UNITED STATES DEPARTMENT OF THE INTERIOR

GEOLOGICAL SURVEY

USER'S MANUAL FOR Z_MP

By Carl A. Carlson

U.S. Geological Survey
Open-File Report 82-621
1982

Table of Contents

List of Illustrations

INTRODUCTION

This is the user's manual for the program z_mp (zeta map plot). Z_mp plots sample location maps (see fig. 1). Z_mp is an extension of the program G239 (map plot) described by Peterson (1977). This section describes the extentions to G239. The user is directed to Peterson (1977) for further information. The next section gives step by step procedures for using z_mp. The following section contains a listing of z_mp. A bibliography completes the manual.

Z_mp impliments these extensions:

- o It now runs on the U.S. Geological Survey Multics computer in Menlo Park, Calif.

- o In place of the two card decks used by G239, it uses two Multics segments. The first is the symbol dictionary. It contains the symbol codes and symbol names for use in interpreting the map data. The map data is contained in the second segment. It comprises the map title, the map scale, the map boundaries, and the sample locations. Both segments posess a format which is identical to the corresponding card decks used by G239. Examples of both formats are shown in figures 2 and 3. These figures also show optional extensions now possible with z_mp.

- o It uses an extended set of map symbols. See figure 4.

- o It will plot optional labels adjacent to each plotted symbol. These symbols can be up to 32 characters long and they can be plotted at any angle.

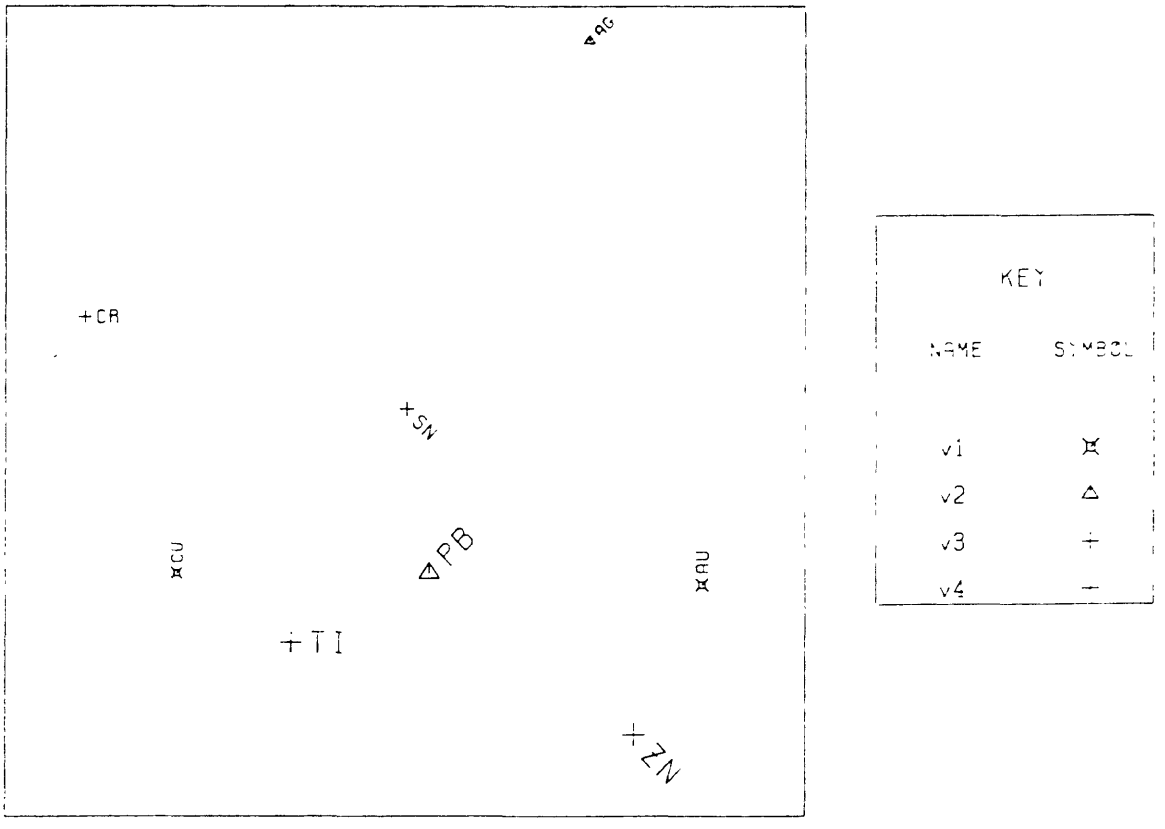
- o It will plot symbols and labels with any of four pens (colors).

- o It will draw a map legend.

- o It will plot composite symbols which can represent multiple data values at a given location. See figure 5; and

- o It will plot the map on a Zeta 6300s Offline Ploting System.

You can create the dictionary and map data segments either by composing them using a Multics text editor such as qedx (Honeywell Information Systems Inc., 1979) or by running the program statmap (Carlson, 1982). Statmap will automatically generate both segments from data contained in a STATPAC file. The STATPAC SYSTEM is described by VanTrump and Miesch (1977).



⊕ DEMONSTRATION OF Z MP

Figure 1. Sample location map. This is a sample of a map produced by z_mp. The following two figures contain the dictionary and the map data files used to plot this map.

```
.....................................................................
v1  210   1  90
v2  202   1  45            This is a four line dictionary
v3  203   1   0
v4  203   1 315
.....................................................................
                                        Column Numbers

         1111111111222222222233333333334444444444555555555566666666667777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890

   ----     ----
---   *    -   *
*     *    *   ****  Angle at which the label is written.  In the case of
*     *    *         composite symbols, this is the angle at which the line,
*     *    *         the symbol, and the label are written. (measured in
*     *    *         degrees counter clock-wise from the three o'clock
*     *    *         direction, right justified)
*     *    *
*     *    ********  Pen number.  Any integer between 1 and 4 is acceptable.
*     *
*     *************  Symbol code. (either standard or composite, right
*                    justified)
*
******************   Symbol name used in the map data file to select the
                     symbol code, pen number, and angle shown on the rest
                     of this line.  (any combination of 1,2, or 3 letters
                     and or numbers that are left justified)
```

Figure 2. Symbol dictionary format. A sample dictionary file is shown between the two dotted lines. Below it are shown the field locations, lengths and definitions.

```
                                    Column Numbers

         1111111111222222222233333333334444444444555555555566666666667777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890

.....................................................................................
DEMONSTRATION OF Z_MP                        <------- 40 columns of map label
     24000<------------------------------------ map scale
        .1          .1<------------------------ map boundaries
        .1       3500.0<---------------------      ditto
    3500.0       3500.0<---------------------      ditto
    3500.0           .1<---------------------      ditto
       0.0           0.0<--------------------- flag to signal end of boundaries
 0.08v1     CU                                     750.0     1050.0
 0.15v2     PB                                    1850.0     1050.0
 0.15v3     TI                                    1250.0      750.0
 0.16v4     ZN                                    2750.0      350.0
 0.09v1     AU                                    3050.0     1000.0
 0.08v2     AG                                    2550.0     3350.0
 0.09v3     CR                                     350.0     2150.0
 0.10v4     SN                                    1750.0     1750.0
.....................................................................................

                                    Column Numbers

         1111111111222222222233333333334444444444555555555566666666667777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
     ---    --------------------------------     --------- ---------
 ----  *                                  *            *         *
    *  *                                  *            *         *
    *  *                                  *            *    UTM Northing
    *  *                                  *            *
    *  *                                  *            **** UTM Easting
    *  *                                  *
    *  *                                  ****  Label (any combination of
    *  *                                               letters, blanks, and
    *  *                                               numbers)
    *  *
    *  ****************************************  Symbol name. (left justified)
    *
    ******************************************  Symbol size. (in inches)
```

Figure 3. Map data format. A sample map data file is shown between the two dotted lines. Each field is defined and its extent shown.

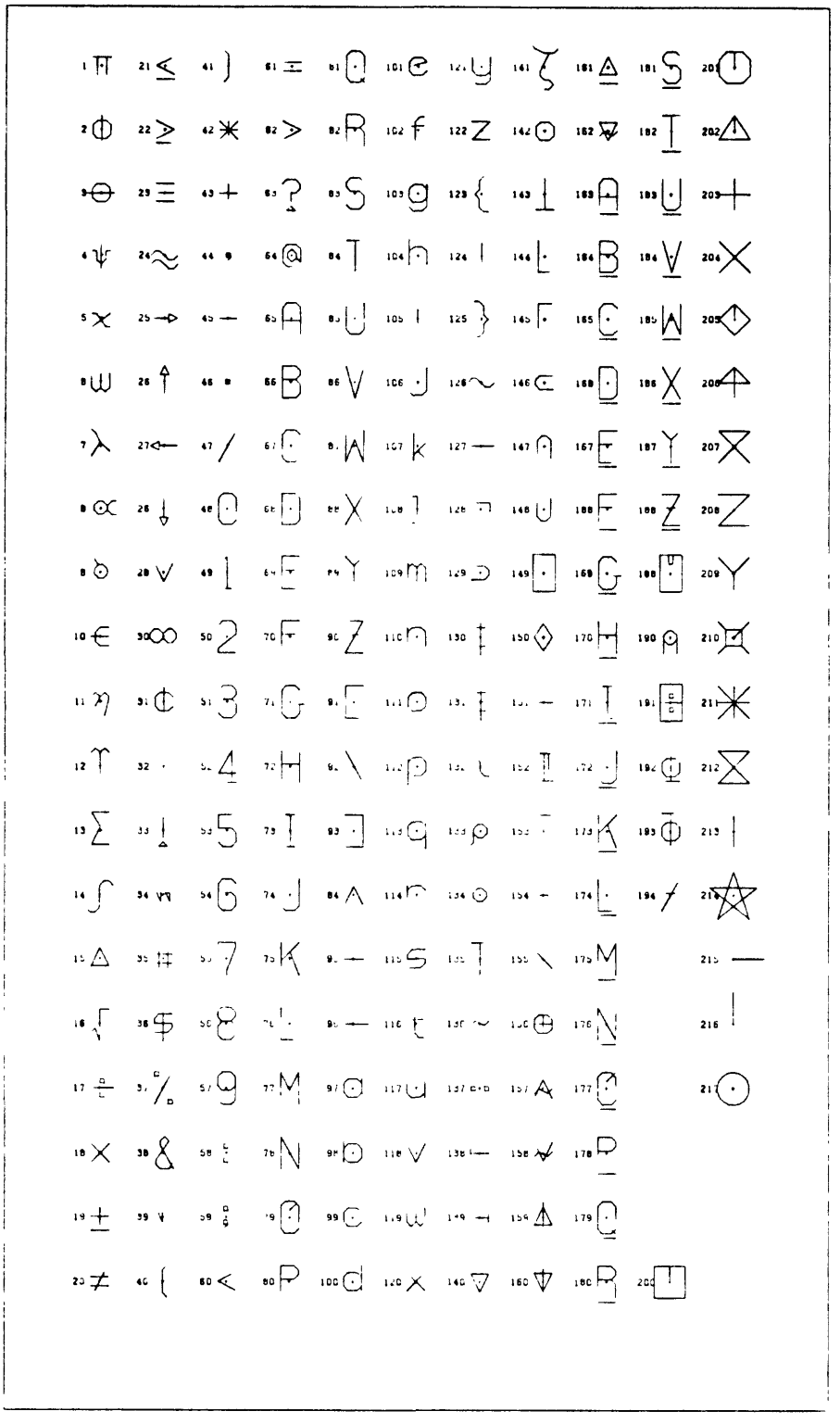


Figure 4. Standard extended character set. The number to the left of each symbol is the symbol code to be used in the symbol dictionary. The small + locates the center of the symbol.

Each composite symbol is composed of one or more plotted line + symbol + label sets. You must have one line of text in your map data file for every set. This text line will have a size, a symbol name, an optional label and UTM's as was shown in Figure 3. The symbol name refers to a line in the dictionary which contains the symbol name, the symbol code, an angle, and a pen number as was shown in Figure 2. The angle refers to the angle measured counter-clock-wise from the three o'clock position along which the line + symbol + label are to be plotted. The meanings of the size and the symbol code are shown below:

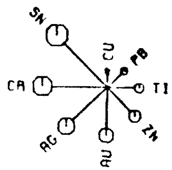
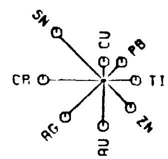
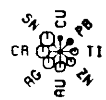
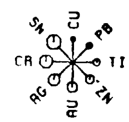
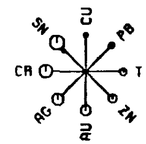
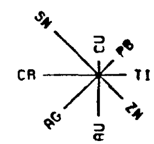
Example:						
Length of the line equals:	The size	The size	.08 in.	.15 in.	.25 in.	The size
Height of the symbol equals:	Size/3	.07 in.	The size	The size	The size	The size
Symbol code equals:	Standard code + 300 C+300	Standard code + 600 C+600	Standard code times -1 -1*C	Standard code + 300 times -1 -1*(C+300)	Standard code + 600 times -1 -1*(C+600)	632
Valid codes:	301 to 517	601 to 817	-1 to 217	-301 to -517	-601 to -817	632

Figure 5. Composite symbol codes. The information needed to compose the composite symbols which are shown at the top of the figure is tabulated in the columns below. Standard code refers to the codes shown in Figure 4.

OPERATING INSTRUCTIONS

Use the following procedures to run z_mp.

Step 1. First you must create the dictionary and the map data files. Use either a Multics editor or statmap. Figures 2 and 3 illustrate the format to be used.

Step 2. Now log on to the Multics computer using the Zeta terminal. The use of the Zeta terminal is described by Zeta Research (1977, 1978). Use the following procedure. Precede all characters intended to be capital letters with a \ (CTRL L).

```
SCA OFF or SCA X.XX Y.YY
ROT OFF or ROT ON
COR ON
COP COM PLO or COP COM TAP
```

Use the standard login procedure except that after the login message you must precede the login command with:

```
MAP
```

Continue with the login procedure as suggested below:

```
\C\CARLSON
PASSWORD
JYTVLHPESBHVG
etc.
```

Step 3. Type:

```
ASR >UDD>\W\MIN\RES>\C\CARLSON
```

Step 4. Type:

```
Z_MP
```

Multics prints:

```
THIS IS Z_MP VERSION 3.0
WHAT IS THE NAME OF THE SEGMENT CONTAINING THE
DICTIONARY?
```

Step 5. Type the chosen name.

Multics prints:

```
WHAT IS THE NAME OF THE SEGMENT CONTAINING THE
MAP DATA?
```

Step 6. Type the chosen name.

Multics prints:

WHAT IS THE NAME OF THE SEGMENT WHERE YOU WANT TO STORE THE PRINTED OUTPUT?

Step 7. Type a name. Multics will create a segment with that name. It will contain error messages, etc.

Multics prints:

DO YOU WANT TO PLOT THE LABELS? (YES OR NO).

Step 8. Type YES or NO depending on whether you want to label each point with the label fields found in the map data segment.

Multics will now plot your map on the plotter (if you used COP COM PLO above), or multics will store the plot on the Zeta tape drive (if you used COP COM TAP).

```
C       ************************************************************
C       *                                                          *
C       *  MINERAL MAPPING PROGRAM (G239)                          *
C       *                                                          *
C       *  THIS PROGRAM PLOTS THE OUTLINE OF A GEOGRAPHICAL REGION *
C       *  AND THEN MARKS THE LOCATIONS OF VARIOUS MINERAL DEPOSITS*
C       *  WITHIN THAT REGION BY APPROPRIATE IDENTIFYING SYMBOLS.  *
C       *  ANY GEOLOGIC DATA REPRESENTABLE BY A MAP POINT IS ALSO  *
C       *  ABLE TO BE PLOTTED                                      *
C       *                                                          *
C       ************************************************************
C       *                                                          *
C       *  MODIFIED TO ACCEPT MAP SCALE INSTEAD OF MAP DIMENSIONS, *
C       *  TO PLOT A QUADRANGLE OR MAP NAME, TO INITIALIZE LIMITS OF*
C       *  PEN MOVEMENTS AND SET ORIGIN, AND TO PROVIDE MORE CHECKS*
C       *  AND ERROR MESSAGES FOR THE USER.                        *
C       *  JOCELYN PETERSON  12/13/76                              *
C       *                                                          *
C       ************************************************************
C       *  REPLACED BENSON-LEHNER CALLS WITH ZETA CALLS            *
C       *  ADDED A SUBROUTINE ADJUST TO REPLACE A CALL TO SPOT     *
C       *  SAM PRIEBE  9/21/79  NEW ADJUST BY CAC REPLACES THIS ONE.*
C       ************************************************************
C       *                                                          *
C       *  MODIFIED TO RUN ON MULTICS.   USER ENTERS SEGMENT NAMES *
C       *  OF:  DICTIONARY; MAP DATA; AND PRINTOUT DESTINATION.    *
C       *  THE PLOT DATA IS SENT DIRECTLY TO THE TERMINAL USING    *
C       *  ZETA'S ERROR CORRECTION.  FEBUARY 15,1980  CARL A CARLSON*
C       *                                                          *
C       ************************************************************

C       *                                                          *
C       *  MODIFIED TO USE DIFFERENT COLORS FOR SYMBOLS, PRINT A   *
C       *  TAG ADJACENT TO EACH SYMBOL, PRINT A KEY TO THE MAP,    *
C       *  AND USE DATA RETRIEVED FROM A STATPAC FILE BY THE PROGRAM*
C       *  STATMAP.  MAY 27, 1980.  CARL A CARLSON                 *
C       *  EXTENSIVE REVISIONS, FEBUARY, 1982                      *
C       *                                                          *
C       ************************************************************
      CHARACTER*30 SEGNAM
      CHARACTER*3 TAGS
      INTEGER MINAM(40), MINCD(40), NAME, ISYM, BLANK, MPSCL,COLOR(40)
      REAL LABLOCX, LABLOCY
      INTEGER AGL(40)
      REAL XBND(25),YBND(25),XMAX,YMAX,XMIN,YMIN
      REAL XSCALE,YSCALE,XLOC,YLOC
      REAL XORG,YORG,XLWR,YLWR,XUPR,YUPR
      REAL QUAD(10),TAG(8)
      DATA BLANK/'   '/
C.....INPUT THE NAMES OF THE SEGMENTS REQUIRED
```

```
2000  FORMAT(V)
      WRITE(0,2000)"this is z_mp:  version 3.0"
      WRITE(0,2000)"what is the name of the segment containing the"
      WRITE(0,2000)"dictionary?"
      READ(0,2000)SEGNAM
      OPEN(10,FORM="FORMATTED",FILE=SEGNAM)
      WRITE(0,2000)"what is the name of the segment containing the"
      WRITE(0,2000)"map data?"
      READ(0,2000)SEGNAM
      OPEN(12,FORM="FORMATTED",FILE=SEGNAM)
      WRITE(0,2000)"what is the name of the segment where you want to"
      WRITE(0,2000)"store the printed output?"
      READ(0,2000)SEGNAM
      OPEN(11,FORM="FORMATTED",FILE=SEGNAM)
      WRITE(0,2000)"do you want to rotate the map 90 deg.? yes or no."
      READ(0,2000)SEGNAM
4000  WRITE(0,2000)"do you want to plot the labels? (yes or no)"
      READ(0,2000)TAGS
      IF ((TAGS.NE."yes").AND.(TAGS.NE."no")) GOTO 4000
C.....CALL PLOTS TO INITIALIZE THE ZETA
      CALL PLOTS(53,0,-1)
C.....CALL SYMDIC TO READ THE NEXT SET OF DICTIONARY LINES
18    CALL SYMDIC(MINAM,MINCD,COLOR,IS,AGL)
C.....READ QUADRANGLE NAME
      READ(12,1000) QUAD
 1000 FORMAT(10A4)
      WRITE(11,1001) QUAD
 1001 FORMAT('1',1X,10A4)
C.....READ MAP SCALE
      READ(12,19) MPSCL
   19 FORMAT(I10)
      WRITE(11,120) MPSCL
  120 FORMAT(' MAP SCALE:   1:',I10)
C.....READ CORNERS OF REGION BOUNDARY
   20 DO 25 IB=1,24
      READ(12,22) XBND(IB),YBND(IB)
   22 FORMAT(2F10.0)
      IF(XBND(IB)+YBND(IB) .EQ. 0.0) GO TO 30
   25 CONTINUE
      WRITE(11,27)
   27 FORMAT(' BOUNDARY DATA NOT TERMINATED BY BLANK CARD WITHIN 24' )
      STOP
   30 IB=IB-1
      WRITE(11,130) (XBND(I),YBND(I), I=1,IB)
  130 FORMAT(' REGION BOUNDARY' / (1X,F10.1,2X,F10.1))
C.....FIND MAXIMUM AND MINIMUM
      XMAX=-1.0E20
      YMAX=XMAX
      XMIN=1.0E20
      YMIN=XMIN
      DO 35 I=1,IB
      IF(XBND(I) .GT. XMAX) XMAX=XBND(I)
      IF(YBND(I) .GT. YMAX) YMAX=YBND(I)
      IF(XBND(I) .LT. XMIN) XMIN=XBND(I)
```

```
      IF(YBND(I) .LT. YMIN) YMIN=YBND(I)
   35 CONTINUE
C.....COMPUTE SCALE FACTORS
      XSCALE=1./(.0254*MPSCL)
      YSCALE=XSCALE
      WRITE(11,140) XMAX,XMIN,YMAX,YMIN,XSCALE,YSCALE
  140 FORMAT(' XMAX=',F10.1,1X,'XMIN=',F10.1,1X,'YMAX=',F10.1,1X,
     *       'YMIN=',F10.1 /1X,'XSCALE=',E12.5,1X,'YSCALE=',E12.5)
C.....SCALE BOUNDARY DATA
      DO 40 I=1,IB
      XBND(I)=(XBND(I)-XMIN)*XSCALE
      YBND(I)=(YBND(I)-YMIN)*YSCALE
   40 CONTINUE
C.....CLOSE BOUNDARY
      IB=IB+1
      XBND(IB)=XBND(1)
      YBND(IB)=YBND(1)
      WRITE(11,150) (XBND(I),YBND(I), I=1,IB)
  150 FORMAT(' SCALED BOUNDARY' / (1X,E12.5,2X,E12.5))
C.....SET PLOT LIMITS
      XORG=0.0
      YORG=0.0
      XLWR=-1.0
      YLWR=-1.0
      XUPR=(XMAX-XMIN)*XSCALE + 3.0
      YUPR=(YMAX-YMIN)*YSCALE
      YMUL=1
C.....CHECK TO INSURE MAP WILL FIT BEFORE PLOTTING
      IF(SEGNAM.EQ."yes") GOTO 5000
      IF(XUPR-XLWR .GT. 200.0) GO TO 300
      IF(YUPR-YLWR .GT. 36.0) GO TO 300
      GOTO 5001
5000  IF(XUPR-XLWR.GT.36.0) GOTO 300
      IF(YUPR-YLWR.GT.200.0) GOTO 300
5001  CONTINUE
C.....INITIALIZE PLOTTER
C      THE FOLLOWING CALL TO FACTOR CONVERTS 2.5 MIL SOFTWARE TO
C      2 MIL HARDWARE
      CALL FACTOR(1.25)
C     CALL PLOT(XLWR,YLWR,-60)          COMMENTED OUT
C     CALL PLOT(XUPR,YUPR,+60)          COMMENTED OUT
C     CALL PLOT(XLWR,YLWR,+3)           COMMENTED OUT
C     CALL PLOT(XORG,YORG,-3)           COMMENTED OUT
C   THE FOLLOWING CALL REPLACES THE PREVIOUS TWO RE-ORIGIN CALLS
      CALL PLOT(1.0,1.0,-3)
      WRITE(11,160) XORG,YORG,XLWR,YLWR,XUPR,YUPR
  160 FORMAT(' PLOT LIMITS SET: XORG=',F6.2,1X,'YORG=',F6.2,1X,
     *       'XLWR=',F6.2,1X,'YLWR=',F6.2,1X,'XUPR=',F6.2,1X,
     *       'YUPR=',F6.2)
C.....PLOT A TIC  MARK FOR CHECKING
C     CALL SPOT(-0.5,-0.5,0.12,66,0.0)          COMMENTED OUT
      CALL SYMBOL(-0.5,-0.5,0.12,3,0.0,-
1)                                        WRITE(11,180)
  180 FORMAT(' FIRST TIC PLOTTED')
```

```
C.....PLOT QUADRANGLE NAME
      CALL SYMBOL(0.0,-0.5,0.12,QUAD,0.0,40)
C.....PLOT BOUNDARY
      X=XBND(1)
      Y=YBND(1)
      CALL PLOT(X,Y,3)
      DO 45 I=2,IB
      X=XBND(I)
      Y=YBND(I)
      CALL PLOT(X,Y,2)
   45 CONTINUE
      WRITE(11,170)
  170 FORMAT(' BOUNDARY PLOTTED')
C.....PLOT SECOND TIC MARK
C     CALL SPOT(-0.5,-0.5,0.06,69,0.0)          COMMENTED OUT
      CALL SYMBOL(-0.5,-0.5,0.06,1,0.0,-1)
      WRITE(11,200)
  200 FORMAT(' SECOND TIC PLOTTED')
C.....READ A MINERAL LOCATION CARD
   50 READ(12,52,END=80) H,NAME,TAG,XLOC,YLOC
   52 FORMAT(1X,F4.2,T6,A3,T13,8A4,T50,F9.1,T60,F9.1)
      IF(NAME .EQ. BLANK) GO TO 70
C.....FOR GROUP DATA WRITE A PARTIAL KEY AND READ A NEW DICTIONARY
      IF(NAME.EQ."NEX")CALL MAPL(XUPR,YUPR,MINAM,COLOR,MINCD,IS,YMUL)
      IF(NAME.EQ."NEX")CALL SYMDIC(MINAM,MINCD,COLOR,IS,AGL)
      IF(NAME.EQ."NEX")GOTO 50
C.....LOOK UP MINERAL SYMBOL CODE
      DO 55 I=1,IS
      IF(NAME .EQ. MINAM(I)) GO TO 65
   55 CONTINUE
      WRITE(11,60) NAME
   60 FORMAT(' NO SYMBOL CODE FOUND FOR: NAME=',A3)
      GO TO 50
   65 ISYM=MINCD(I)
C.....TEST MINERAL LOCATION WITHIN MAP
      IF(XLOC .GT. XMAX) GO TO 302
      IF(XLOC .LT. XMIN) GO TO 302
      IF(YLOC .GT. YMAX) GO TO 302
      IF(YLOC .LT. YMIN) GO TO 302
      GO TO 303
  302 WRITE(11,1003) XLOC,YLOC
 1003 FORMAT(1X,F10.1,2X,F10.1,3X,'LOCATION NOT WITHIN MAP')
      GO TO 50
C.....SCALE LOCATION
  303 XLOC=(XLOC-XMIN)*XSCALE
      YLOC=(YLOC-YMIN)*YSCALE
C.....PLOT LOCATION
C     CALL SPOT(XLOC,YLOC,H,ISYM,0.0)          COMMENTED OUT
C     THE SUBROUTINE SPOT(BENSON-LEHRNER) IS REPLACED BY SYMBOL(ZETA)
C      VIA A CALL TO SUBROUTINE ADJUST TO CORRECTLY POSITION SYMBOLS
      RANGL=AGL(I)
C.....SELECT PEN COLOR
      CALL NEWPEN(COLOR(I))
C.....PLOT THE SYMBOL
```

```
      CALL ADJUST(ISYM,XLOC,YLOC,H,RANGL)
C.....PLOT THE LABEL (TAG)
      IF (TAGS.EQ."no") GOTO 68
      LABLEN=0
      CALL LABELLOC (H, RANGL, TAG, LABLOCX, LABLOCY, LABLEN)
      IF (LABLEN.GT.0) CALL SYMBOL(XLOC+LABLOCX,YLOC+LABLOCY,H,TAG,RANGL
     +   ,LABLEN)
68    CONTINUE
      WRITE(11,190) XLOC,YLOC
  190 FORMAT(' MINERAL LOCATION PLOTTED AT',2E14.5)
      GO TO 50
C.....PLOT THIRD TIC MARK
C  70 CALL SPOT(-0.5,-0.5,0.12,69,0.0)          COMMENTED OUT
   70 CALL SYMBOL(-0.5,-0.5,0.12,1,0.0,-1)
C.....PLOT THE KEY TO THE MAP.
      CALL MAPL(XUPR,YUPR,MINAM,COLOR,MINCD,IS,YMUL)
      WRITE(11,220)
  220 FORMAT(' THIRD TIC PLOTTED')
C.....MOVE ORIGIN DOWN FOR NEXT PLOT
      CALL PLOT(XUPR,0.0,-3)
      GO TO 18
C.....READ AND IGNORE DATA WHEN MAP TOO LARGE TO PLOT
  300 WRITE(11,1002)
 1002 FORMAT(' MAP LARGER THAN PAPER. NOT PLOTTED')
  301 READ(12,52,END=80) NAME
      IF(NAME .EQ. BLANK) GO TO 18
      GO TO 301
C.....PLOT FINAL TIC MARK
C  80 CALL SPOT(-0.5,-0.5,0.12,69,0.0)         COMMENTED OUT
   80 CALL SYMBOL(-0.5,-0.5,0.12,1,0.0,-1)
C.....PLOT THE KEY TO THE MAP
      CALL MAPL(XUPR,YUPR,MINAM,COLOR,MINCD,IS,YMUL)
      WRITE(11,210)
  210 FORMAT(' FINAL TIC PLOTTED')
C.....END OF PLOT
      CALL PLOT(0.0,0.0,999)
      CLOSE(10)
      CLOSE(11)
      CLOSE(12)
      CLOSE(13)
      STOP
      END
C*********************************************************************
C     THIS SUBROUTINE PLOTS THE KEY FOR THE MAP
C     CARL A CARLSON MAR 11, 1980.
C*********************************************************************
      SUBROUTINE MAPL(XUPR,YUPR,MINAM,COLOR,MINCD,IS,YMUL)
      INTEGER MINAM(40),COLOR(40),MINCD(40)
      CALL NEWPEN(1)
      IF(YMUL.NE.1)GOTO 2002
      YMUL=YUPR-3.2
      CALL PLOT(XUPR-2.5,1.5,3)
      CALL PLOT(XUPR-0.5,1.5,2)
      CALL PLOT(XUPR-0.5,YUPR-1.5,2)
```

```
      CALL PLOT(XUPR-2.5,YUPR-1.5,2)
      CALL PLOT(XUPR-2.5,1.5,2)
      CALL PLOT(XUPR-2.5,1.5,3)
      CALL SYMBOL(XUPR-1.83,YUPR-2.0,0.12,5H  KEY,0.0,5)
      CALL SYMBOL(XUPR-2.4,YUPR-2.5,0.1,18H   NAME     SYMBOL,0.0,18)
2002  DO 2001 ICC=1,IS
         ICODE=IABS(MINCD(ICC))
3000     IF (ICODE.GE.301) ICODE=ICODE-300
         IF (ICODE.GE.301) GOTO 3000
      CALL NEWPEN(1)
      X=XUPR-2.05
      Y=YMUL-(ICC-1)*(.33)
      CALL SYMBOL(X,Y,.12,MINAM(ICC),0.0,3)
      CALL NEWPEN(COLOR(ICC))
      X=X+1.1
      Y=Y+.06
      H=.12
      RANGL=0.0
      CALL ADJUST(ICODE,X,Y,H,RANGL)
2001  CONTINUE
      YMUL=Y-.33
      CALL NEWPEN(1)
      END
C*********************************************************************
C          THIS READS A DICTIONARY                                    *
C*********************************************************************
      SUBROUTINE SYMDIC(MINAM,MINCD,COLOR,IS,AGL)
      INTEGER  MINAM(40),MINCD(40),COLOR(40),AGL(40)
10    DO 14 IS=1,40
      READ(10,12,END=16)MINAM(IS),MINCD(IS),COLOR(IS),AGL(IS)
12    FORMAT(A3,I4,1X,I3,1X,I3)
      IF(MINCD(IS).EQ.999)GOTO 16
      IF(COLOR(IS).LT.1)COLOR(IS)=1
      IF(COLOR(IS).GT.4)COLOR(IS)=1
14    CONTINUE
      IS=31
16    IS=IS-1
      WRITE(11,110) (MINAM(I),MINCD(I),I=1,IS)
110   FORMAT(' SYMBOL DICTIONARY' / (1X,A3,2X,I2))
      END
C  *********************************************************
C  *  SUBR:  ADJUST APPROXIMATELY CENTERES SYMBOLS.  PLOTS  *
C  *  SYMBOLS AND SPAR DIAGRAMS.  ADJUSTS LOCATION FOR THE  *
C  *  TAG (LABEL) WHICH IS TO BE PLOTTED IN MAIN PROGRAM.   *
C  *  BY CCARLSON  FEB. 5, 1982                             *
C  *********************************************************
      SUBROUTINE ADJUST( ISYM, XLOC, YLOC, H, RANGL)
      DIMENSION POS(2,196)
      DIMENSION A(196),B(196)
      EQUIVALENCE (POS(1,1),A(1)),(POS(1,99),B(1))
      DATA A/.22,.35,.22,.35,.35,.35,.35,.35,.35,.35,
     + .35,.27,.35,.27,.17,.27,.27,.27,.35,.35,.35,.35,.35,.35,
     + .22,.35,.35,.45,.22,.3,.22,.45,.22,.35,.22,.35,.22,.45,
     + .22,.35,.22,.45,.22,.45,.22,.45,.35,.35,.35,.35,.22,.4,
```

```
     + .35,.35,.22,.45,.22,.35,.35,.27,.22,.35,0.0,0.0,.35,.4,
     + .4,.75,.22,.35,.22,.45,.22,.4,.3,.4,.35,.7,.35,.4,
     + .22,.4,.22,.35,.22,.35,.05,.05,.35,.35,.05,.05,.22,.4,
     + .22,.4,.22,.4,.22,.4,.22,.4,.22,.4,.22,.4,.22,.4,.22,.4,
     + .22,.4,.22,.4,.3,.35,.3,.3,.22,.45,.22,.35,.22,.45,
     + .35,.45,.22,.4,.22,.4,.22,.4,.22,.4,.22,.4,.22,.4,.22,.4,
     + .22,.4,.22,.4,.22,.4,.22,.4,.22,.4,.22,.4,.22,.4,.22,.4,
     + .22,.4,.22,.4,.22,.4,.22,.4,.22,.4,.22,.4,.22,.4,.22,.4,
     + .22,.4,.22,.4,.22,.4,.22,.4,.22,.4,.22,.4,.22,.4,.22,.35,
     + .3,.35,.3,.8,.35,.22,.35,.22/
      DATA B/.35,.22,.35,.22,.35,.22,
     + .3,.35,.35,.22,.35,.22,.35,.22,.35,.1,.22,.35,.35,.35,
     + .35,.22,.35,.22,.35,.22,.35,.22,.35,.22,.35,.22,.35,.22,
     + .27,.35,.35,.22,.35,.22,.35,.22,.35,.22,.35,.22,.35,.22,
     + .17,.35,.22,.35,.4,.35,.35,.35,.35,.0,.22,.17,.35,.4,
     + .22,.45,.22,.45,.35,.35,.4,.4,.4,.4,.22,.4,.22,.27,.22,.75,
     + .35,.35,.35,.35,.22,.37,.35,.22,.22,.35,.22,.4,.17,.4,
     + .17,.4,.35,.4,.4,.3,.4,.3,.27,.35,.35,0.0,.22,-.1,.22,.35,
     + .17,.4,.22,.45,.22,.35,.22,.35,.22,.35,.22,.3,.22,.3,
     + .22,.35,.22,.3,.22,.35,.22,.4,.22,.4,.22,.4,.22,.4,.22,.4,
     + .22,.4,.22,.4,.22,.4,.22,.4,.22,.4,.22,.4,.22,.4,.22,.4,
     + .22,.4,.22,.4,.22,.4,.22,.4,.22,.4,.22,.4,.22,.4,.22,.4,
     + .22,.4,.22,.4,.22,.4,.22,.4,.22,.4,
     + .27,.35,.4,.4,.27,.35,.22,.35,.22,.35,.22,.4,.22,.4,.0,.0/
      ISPOT=ISYM
      XPOS=XLOC
      YPOS=YLOC
      HH=H
C
C          *** SEPERATE THE CASES ***
C
5000  IF ((ISYM.GT.0).AND.(ISYM.LE.195)) GOTO 1000
      IF ((ISYM.GE.200).AND.(ISYM.LE.216)) GOTO 2000
      IF ((ISYM.GE.217).AND.(ISYM.LE.300)) GOTO 3000
      IF ((ISYM.GE.301).AND.(ISYM.LE.900)) GOTO 4000
      IF ((ISYM.LE.-1).AND.(ISYM.GE.-900)) GOTO 4000
      RETURN
C
C          *** CENTER SYMBOLS AND PLOT ***
C
1000  XPOS=XPOS-POS(1,ISYM)*HH*1.25
      YPOS=YPOS-POS(2,ISYM)*HH*1.25
      CALL SYMBOL(XPOS,YPOS,HH,ISPOT,0.0,0)
      RETURN
C
C          *** USE PRE-CENTERED SYMBOLS AND PLOT ***
C
2000  ISPOT=ISYM-200
      CALL SYMBOL(XPOS,YPOS,HH,ISPOT,0.0,-1)
      RETURN
C
C          *** PLOT A CIRCLE ***
C
3000  RAD=HH/2.0
```

```
      CALL CIRCLE(XPOS,YPOS-RAD,-90.0,270.0,RAD,2)
      RETURN
C
C          *** PLOT A SPAR DIAGRAM ***
C
4000  RLENTH=HH
      IF ((ISYM.LE.-1).AND.(ISYM.GE.-300)) RLENTH=.08
      IF ((ISYM.LE.-301).AND.(ISYM.GE.-600)) RLENTH=.15
      IF ((ISYM.LE.-601).AND.(ISYM.GE.-900)) RLENTH=.25
C
C          *** PLOT THE SPAR ***
C
      DANGL=RANGL*(3.1415926/180.0)
      XSPAR=XPOS+RLENTH*COS(DANGL)
      YSPAR=YPOS+RLENTH*SIN(DANGL)
      CALL PLOT(XPOS,YPOS,3)
      CALL PLOT(XSPAR,YSPAR,2)
C
C          *** SET SYMBOL SIZE ***
C
      IF ((ISYM.GE.601).AND.(ISYM.LE.900)) HH=.07
      IF ((ISYM.GE.301).AND.(ISYM.LE.600)) HH=H/3.0
      IF (ISYM.LT.0) HH=H
C
C          *** SET UP LABEL SIZE ***
C
      H=.07
C
C          *** COMPUTE CORRECT SYMBOL CODE ***
C
      IF ((ISYM.GE.301).AND.(ISYM.LE.600)) ISYM=ISYM-300
      IF ((ISYM.GE.601).AND.(ISYM.LE.900)) ISYM=ISYM-600
      IF ((ISYM.LE.-1).AND.(ISYM.GE.-300)) ISYM=-1*ISYM
      IF ((ISYM.LE.-301).AND.(ISYM.GE.-600)) ISYM=-1*(ISYM+300)
      IF ((ISYM.LE.-601).AND.(ISYM.GE.-899)) ISYM=-1*(ISYM+600)
      IF (.NOT.((ISYM.GE.1).AND.(ISYM.LE.300))) RETURN
C
C         *** ADJUST THE LOCATION OF THE PENDANT AND LABEL ***
C
      XPOS=XSPAR
      YPOS=YSPAR
      XLOC=XSPAR
      YLOC=YSPAR
      IF (ISYM.NE.32) XPOS=XSPAR+HH/2.0*COS(DANGL)
      IF (ISYM.NE.32) YPOS=YSPAR+HH/2.0*SIN(DANGL)
      IF (ISYM.NE.32) XLOC=XSPAR+HH*COS(DANGL)
      IF (ISYM.NE.32) YLOC=YSPAR+HH*SIN(DANGL)
C
C          *** GO PLOT PENDANT ***
C
      ISPOT=ISYM
      GOTO 5000
      END
(*     WRITTEN BY D. DELLINGER      *)
```

```
labelloc: proc (h, rangl, tag, labellocx, labellocy, length);
dcl (h, sz, rangl, labellocx, labellocy) float bin;
dcl (i, sgn, length) fixed bin;
dcl  tag char (32);
          do while (substr (tag, 1, 1) = " ");
               tag = substr (tag, 2, 31) ¢¢ " ";
          end;
          length = 32;
          do while (substr (tag, length, 1) = " ");
               length = length - 1;
          end;
          if (rangl > 90) & (rangl <= 270) then do;
               sz = h * (length + .5);
               sgn = -1;
               rangl = rangl + 180;
               if rangl >= 360 then rangl = rangl - 360;
          end;
          else do;
               sz = h;
               sgn = 1;
          end;
          labellocx = (sz * cosd (rangl) * sgn) + ((h/2) * sind (rangl));
          labellocy = (sz * sind (rangl) * sgn) - ((h/2) * cosd (rangl));
          return;
     end labelloc;
```

BIBLIOGRAPHY

Carlson, C. A., 1982, User's manual for statmap. U.S. Geological Survey Open-File Report 82-622, 10 p.

Honeywell Information Systems Inc., 1979. Series 60 (level 68) qedx text editor users' guide, preliminary edition: Order Number CG40-00.

Peterson, J. C., 1977. A modification of mineral mapping program (G239): U.S. Geological Survey Open-File Report 77-713.

VanTrump, Jr., George and Miesch, A. T., 1977. The U.S. Geological Survey RASS-STATPAC system for management and statistical reduction of geochemical data: Computers and Geosciences, v. 3, p. 475-488.

Zeta Research Inc., 1977. Instrumentation manual, 3600 Series Incremental Plotters, Serials F/158: Zeta Research Division of Nicolet Instrument Corporation Publication No. 43-1-005.

_____1978, User's manual, Model 6300 Off-line plotter controller: Zeta Research Division of Nicolet Instrument Corporation Publication No. 43-1-025.