60454730

CONTROL DATA CORPORATION

UNIPLOT
VERSION 2.0
REFERENCE/USER GUIDE

CDC® OPERATING SYSTEMS:
NOS
NOS/BE

LIST OF EFFECTIVE PAGES

New features, as well as changes, deletions, and additions to information in this manual, are indicated by bars in the margins or by a dot near the page number if the entire page is affected. A bar by the page number indicates pagination rather than content has changed.

PAGE	REV	PAGE	REV	PAGE	REV	PAGE	REV	PAGE	REV
Front Cover	-								
Title Page	-								
ii	A								
iii	A								
v	A								
vi	A								
vii	A								
1-1	A								
1-2	A								
2-1 through 2-21	A								
3-1 through 3-15	A								
A-1 through A-6	A								
B-1	A								
B-2	A								
C-1	A								
D-1	A								
D-2	A								
E-1	A								
F-1	A								
G-1	A								
G-2	A								
H-1	A								
H-2	A								
H-3	A								
H-4	A								
I-1	A								
I-2	A								
Index-1 through Index-3	A								
Comment Sheet	A								
Reply Env.	-								
Back Cover	-								

PREFACE

CONTROL DATA CORPORATION® UNIPLOT® (Universal Plotting Software) is a set of basic plotting routines which link a CalComp-compatible application program with supported plotters or graphic devices. UNIPLOT is composed of two software components: (1) a set of CalComp call-compatible neutral plot routines, and (2) a post processor. Called by a user's application program, the neutral plot routines generate a neutral picture file. This file contains a sequence of pictures which may be separately accessed and plotted by the post processor.

The primary function of this publication is to describe the neutral plot routines and post processor and how they are used. UNIPLOT can be used with the following operating systems and supported devices.

- CDC® CYBERNET® users can employ CDC UNIPLOT under the CYBERNET, SCOPE 3.4 and NOS 1.0 operating systems.
- Systems users can employ CDC UNIPLOT under either the NOS or NOS/BE operating systems.
- Refer to appendix D for a list of supported devices.

This manual assumes user familiarity with FORTRAN Extended, standard CalComp routines, and CDC NOS or NOS/BE operating systems.

Control Data manuals which may relate to the operation of CDC UNIPLOT under CYBERNET Service include:

Publication	Publication Number
CYBERNET NOS User's Reference Manual	84000029
CYBERNET SCOPE 3.4 Reference Manual	60307200
CYBERNET FORTRAN Extended 4 Reference Manual	60497800
CYBER Loader Reference Manual	60429800
CYBERNET UPDATE Reference Manual	60342500
777 IGS Reference Manual	16321800
777 IGS User's Guide	17322500
LCGT IGS Reference Manual	76079100
LCGT IGS User's Guide	76077400
Beginning Graphics User's Guide	76077300
UNISTRUC Reference Manual	76076800
GODAS V3.0 User's Guide, Volume 2	76076300

Control Data publications that may relate to the operation of CDC UNIPLOT under NOS/BE include:

Publication	Publication Number
NOS/BE Reference Manual	60493800
FORTRAN Extended 4 Reference Manual	60497800
INTERCOM Reference Manual	60494600
CYBER Loader Reference Manual	60429800
Record Manager Reference Manual	60495700
UPDATE Reference Manual	60449900
777 IGS Reference Manual	17321800
777 IGS User's Guide	17322500
777 IGS Remote Job Entry User's Guide	76077200
LCGT IGS Reference Manual	76079100
LCGT IGS User's Guide	76077400
Beginning Graphics User's Guide	76077300

CONTENTS

APPENDIXES

INDEX

FIGURES

GENERAL DESCRIPTION

UNIPLOT, Universal Plotting Software, is a set of FORTRAN routines which are CalComp call-compatible. These routines write a neutral file which can then be read by a post processor to produce output for a variety of plotters. UNIPLOT is designed to be used by all applications programs which make CalComp calls to produce graphic output in batch mode. Programs which require real-time or interactive graphic displays are not supported. Figure 1-1 illustrates an overview to UNIPLOT processing.

UNIPLOT provides the user with great flexibility in the selection of a device for displaying graphic output. Even after an application program is finished, the user maintains control over the display device, drawing scale, drawing layout on a display surface, mirroring of the drawing, and windowing (displaying only a selected rectangular portion of a drawing).

SYSTEM COMPONENTS

The UNIPLOT system is composed of two software components: a set of CDC neutral plot routines and a post processor. These components communicate through a standard format file known as the neutral picture file.

NEUTRAL PLOT ROUTINES

The neutral plot routines are called by a user's application program in the same manner as standard CalComp routines. However, instead of producing a file that drives only a single CalComp plotter, the neutral plot routines create what is called a neutral picture file. This file, in turn, contains a sequence of pictures, each of which may be accessed and plotted separately on a variety of devices by executing the post processor.

UNIPLOT includes a wide variety of neutral plot routines which perform a number of functions such as generating vectors, character strings, arcs and curves, structuring the plot data into pictures and figures, controlling the placement of pictures on a display surface, and manipulating data in order to mirror, offset, and scale display elements before they are placed in the picture file.

POST PROCESSOR

The post processor is a stand-alone program which reads the neutral picture file. The post processor can be instructed to modify the placement, scale, and rotation of each picture or a portion of each picture. The graphics data is translated into a format which is acceptable by the controller of the selected display device, and the picture may be produced directly on an on-line display device, or written to a tape for plotting on an off-line device.

INITIALIZATION AND TERMINATION

The neutral plot routines are called by an application program as are the CalComp Basic Plotting Software programs. An intermediate device-independent file, called the neutral picture file, is produced. This file contains a sequence of pictures made up of character string control and placement information and other graphic data. Each of these pictures may be accessed and plotted separately by the post processor.

Initially, it is necessary to identify the neutral picture file to the neutral plot routines. This is accomplished in one of two ways:

CALL PLTFIL(ldev)

or via the standard CalComp routine

CALL PLOTS(ibuf,nloc,ldev)

Both of these calls are described later in this section.† It should be noted, however, that CALL PLTFIL and CALL PLOTS are used only once in an application program.

Both routines contain dummy parameter strings for compatibility with UNIPLOT Version 1.0 and CalComp. However UNIPLOT Version 2.0 does not require any buffer space or logical file name from the calling program. The neutral plot routines will always create a neutral picture file called NPFILE.

In addition to PLTFIL and PLOT calls, another call, NEWPLT, allows greater flexibility in defining individual pictures within the entire graphic output.

The standard method of starting a new picture, defining a new origin with a call to PLOT with a negative pen command, is satisfactory for pen plotters. However, in order to display a picture properly on a CRT, the origin or beginning must be known. Some of the application programs which produce CalComp output establish multiple minor origins causing the original origin to be indistinguishable. This problem can be overcome by using the routine NEWPLT. The format is:

CALL NEWPLT(level,xmin,ymin,xmax,ymax,xloc,
yloc,irot,name)

The size of the total picture is defined by the xmin,ymin,xmax and ymax arguments on the NEWPLT call. A rectangle drawn through these coordinates defines a frame or window around the picture. The lower left-hand corner of this rectangle is placed at the xloc,yloc position on the display surface if the post-processor is executed in FILE mode.

Included in the picture defined by a NEWPLT call are one or more figures defined by a negative third argument. All coordinates in the figures must be within the rectangle defined by the minimum and maximum x and y arguments on the last call to NEWPLT.

UNIPLOT is not responsible for the resulting display if any vectors extend across the window boundaries of the picture.

The final call which the application program makes to the neutral plot routines must be a call to the PLOT routine in the following form.

CALL PLOT(x,y,999)

This call empties the buffer of graphics data and ends the neutral picture file (refer to PLOT Routine described later in this section). It should only be called once in the program.

WRITING GRAPHIC DATA

This section briefly describes a number of the routines used to scale and write graphic data. The routines referred to are described fully later in this section under Descriptions of the Neutral Plot Routines.

Via the PLOT routine, the user can employ any of the following four kinds of lines: solid, invisible, dashed, or dotted. Character strings and a variety of centered symbols are written via the SYMBOL routine. The UNIPLOT character set is described in appendix B, and the centered symbols are illustrated in appendix C. Floating point numbers may be plotted as a string of digits, and punctuated, by the NUMBER routine. Circles or arcs may be drawn by either the ARC or ARC2 routine. The difference in these last two calls is in the manner in which the arc is specified.

The SCALE, AXIS and LINE routines are designed to simplify the task of plotting graphs. SCALE inspects an array of data and computes a scale and a starting value which, when applied to the data, assures that the array will fall within a specified number of inches. The user specifies a scale factor in his program using a SCALE=fact* directive, where fact is the factor by which the figure is to be scaled (magnified). Another directive, SCHAR=fact* allows the user program to scale text in a drawing while leaving the figure unchanged.

The starting value and scale are stored in the inspected array in the two locations beyond the last word inspected. Consequently, the DIMENSION statement should be at least two words larger than required by the array. The inspected data need not be contiguous in core, but may be separated by a given number of words. In this case the starting value and scale are placed at the last word address plus increment, and last word address plus two times the increment, respectively.

The AXIS routine is used to generate the axes for a graph. It draws an axis with a linear scale, using the values computed by the SCALE routine or a SCALE and starting value calculated by the user, place tic marks on the axis, and label the axis with numbers and letters.

†Refer to Descriptions of the Neutral Plot Routines in this section.

Instructions for writing a new device control routine are provided in appendix E of this manual. For instructions on loading the corresponding specific device control routine with the post processor, contact your local CDC analyst.

ERROR RETURNS

New applications using the neutral plot routines may choose to use the argument check and error return feature. An additional integer argument may be added to calls to the neutral plot routines for which illegal argument combinations are possible. If the arguments to the routine are legal, this additional argument returns with a value of zero; if not, a value greater than zero is returned. The particular value indicates the type of error. The error codes are explained in appendix F.

If a call is coded without the extra argument, the parameters of that particular call are not checked.

For any given run, the error checking mode must be turned on by a call to routine ERRON. Subsequently, the mode may be turned off by a call to ERROFF. The error checking mode may be turned on and off any number of times during a run.

DESCRIPTIONS OF THE NEUTRAL PLOT ROUTINES

PLOTS ROUTINE

The PLOTS routine initializes the neutral plot routines and starts the first picture. Figures cannot be included within a picture defined by a PLOTS call.

Call format:

```
CALL PLOTS(ibuf,nloc,ldev)
```

ibuf — Unused (Provided for compatibility with CalComp

nloc — Unused (Provided for compatibility with CalComp)

ldev — Unused (provided for compatibility with CalComp. UNIPLOT Version 2.0 always creates a neutral picture file name NPFILE.).

Example:

```
PROGRAM EXAMP (INPUT,OUTPUT)
      .
      .
      .
CALL PLOTS
```

Each time the PLOTS routine is called it opens a file named NPFILE. Subsequent calls to other neutral plot routines use that file as the neutral plot file.

PLOTS also initializes the modal parameters of the neutral plot routines as shown in table 2-2. Thus PLOTS or PLTFIL must be called before any other UNIPLOT routine and should not be called more than once in a program.

TABLE 2-2. INITIALIZING NEUTRAL PLOT ROUTINE PARAMETERS

Parameter	Initial Value	Calling Routine
XLOC	0.0	NEWPLT
YLOC	0.0	NEWPLT
FACT	1.0	FACTOR
SCALEX	1.0	OFFSET
SCALEY	1.0	OFFSET
OFFSETX	0.0	OFFSET
OFFSETY	0.0	OFFSET
ZLINE	0.0	MIRROR
RATIO	1.0	RATIO

```
      PROGRAM EXAMP1(OUTPUT)
C-----
C         INITIALIZE
C-----
      CALL PLTFIL
C-----
C         CALL ROUTINE TO DEFINE AND FRAME A PICTURE
C-----
      CALL FRAME(0.0,0.0,4)
      CALL FRAME(3.0,1.0,2)
      CALL FRAME(6.0,0.0,5)
C-----
C         TERMINATE PLOTTING
C-----
      CALL PLOT(3.0,-1.0,999)
      END
      SUBROUTINE FRAME(XORG,YORG,IC)
C-----
C         ROUTINE TO DEFINE AND FRAME A PICTURE
C-----
      DATA XMIN,YMIN,XMAX,YMAX/-1.0,-1.0,1.0,1.0/
      CALL NEWPLT(1,XMIN,YMIN,XMAX,YMAX,XORG,YORG,0,1H )
      CALL PLOT(1.0,1.0,-3)
      CALL PLOT(XMIN,YMIN,3)
      CALL PLOT(XMAX,YMIN,IC)
      CALL PLOT(XMAX,YMAX,IC)
      CALL PLOT(XMIN,YMAX,IC)
      CALL PLOT(XMIN,YMIN,IC)
      RETURN
      END
```

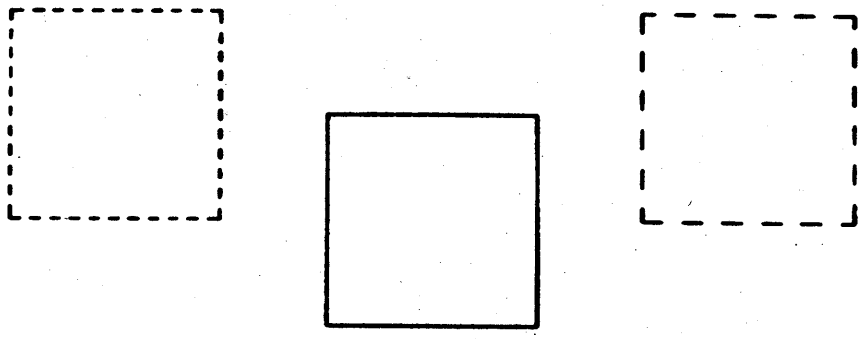


Figure 2-1. NEWPLT Call

```
      PROGRAM EXAMP2(OUTPUT)
C----
C         INITIALIZE AND DEFINE AND FRAME THE PICTURE
C----
      CALL PLTFIL
      CALL NEWPLT(1,-1.5,-1.5,8.5,3.5,0.0,0.0,0,1H )
      CALL PLOT(1.5,1.5,-3)
      CALL FRAME(2)
C----
C         ESTABLISH A FIGURE ORIGIN AND DRAW A 2 INCH SQUARE AROUND IT
C----
      CALL PLOT(0.0,0.0,-3)
      CALL FRAME(4)
      CALL PLOT(3.0,1.0,-3)
      CALL FRAME(2)
      CALL PLOT(3.0,-1.0,-3)
      CALL FRAME(5)
C----
C         TERMINATE PLOTTING
C----
      CALL PLOT(3.0,-.5,999)
      END
      SUBROUTINE FRAME(IC)
C----
C         ROUTINE TO DRAW A FRAME USING DIFFERENT LINE STYLES
C----
      DATA XMIN,YMIN,XMAX,YMAX/-1.5,-1.5,8.5,3.5/
      CALL PLOT(XMIN,YMIN,3)
      CALL PLOT(XMAX,YMIN,IC)
      CALL PLOT(XMAX,YMAX,IC)
      CALL PLOT(XMIN,YMAX,IC)
      CALL PLOT(XMIN,YMIN,IC)
      XMIN=YMIN=-1.0
      XMAX=YMAX=1.0
      RETURN
      END
```

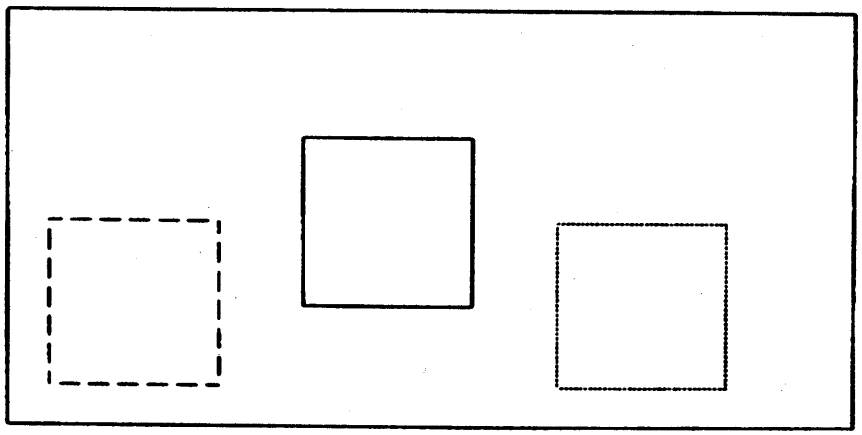


Figure 2-2. PLOT Call

FACTOR ROUTINE

The FACTOR routine sets the value of the modal parameter, fact. The x,y coordinates are multiplied by fact causing the picture to be enlarged or reduced. The lower left corner of the window remains fixed.

Call format:

CALL FACTOR(fact)

fact — The ratio of the desired picture size to the default picture size. If fact is 2.0, all vectors will be twice as long as the default when fact is 1.0.

WHERE ROUTINE

The WHERE routine returns to the calling program the current value of fact and the end-point coordinates of the last vector or character generated by the neutral plot routines and modified by the modal parameters.

Call format:

CALL WHERE(x,y,fact)

x,y — The coordinates of the end-point of the last display element generated, with respect to the figure origin after modal parameters have been applied.

fact — The current scale factor that multiplies all coordinates and sizes that are placed in the neutral picture file (set in a call to FACTOR).

NEWPEN ROUTINE

The NEWPEN routine allows the user to select which pen to use on a multipen plotter.

Call format:

CALL NEWPEN(ipen)

ipen — Pen number.

SCALE ROUTINE

A user's program usually contains virtual x and y coordinates in two arrays, It would be unusual if the range of values in each array corresponded exactly with the number of inches available on a display surface. For some problems the range of data is predictable, and the axis scale values can be predetermined by the programmer. However, in general, these factors are not known in advance.

The SCALE routine examines the data values in an array and determines a starting value, either minimum or maximum, and a scaling factor, positive or negative. The starting value determines that the scale annotation on the axis properly represents the data values in the array and that the data points plotted by the LINE routine fit on the display surface. These two values are computed and stored by SCALE at the end of the array.

The scaling factor (scale) that is computed represents the number of data units per inch of axis, but is adjusted so that it is always an interval of 1, 2, 4, 5, or 8 x 10^n (where n is an exponent consistent with the original unadjusted scaling factor). Thus, an array may have a range of values from 301 to 912, to be plotted over an axis of 10 inches. The unadjusted scaling factor is (912-301)/10 = 61.1 units/inch. The adjusted scale would be 8 x 10^1 = 80.

The starting value (start), which appears as the first annotation on the axis, is computed as some multiple of scale that is equal to or outside the limits of the data in the array. For the example given above, if a minimum is wanted for start, 240 would be chosen as the best value. If a maximum is desired instead, 960 would be selected.

Call format:

CALL SCALE(array,axlen,npts,incr)

array — The first word of the array of data points to be examined. Array must be dimensioned by (npts+2)*incr.

axlen — The length in inches of the axis to which the data is to be scaled. Its value must be greater than 1.0 inch.

npts — The number of data values to be scanned in the array.

incr — An integer whose absolute value is used by SCALE as the increment with which to select the data values to be scanned in the array. Normally incr equals 1; if it is 2, every other value is examined.

If incr is positive, the selected starting value (start) approximates a minimum, and the scale factor (scale) is positive.

If incr is negative, the selected starting value (start) approximates a maximum, and the scaling factor (scale) is negative.

The starting value (start) calculated by SCALE is stored at ARRAY((NPTS+1)*INCR). The scale value (scale) is stored at ARRAY((NPTS+2)*INCR).

SCALE is typically called twice, once for the X axis and once for the Y axis.

An example of the use of the SCALE routine is included with the AXIS example in figure 2-4.

AXIS ROUTINE

The AXIS routine draws an annotated axis. The SCALE routine can be used to compute the values of scale and start which produces an axis appropriate for plotting an array of values or the user can calculate his own scale and starting values. Typically, AXIS is called twice, once for the X axis and once for the Y axis.

Call format:

CALL AXIS(x,y,ititle,n,axlen,angle,start,scale)

x,y — Coordinate starting point of the axis relative to the origin of the last figure defined.

ititle — Array containing BCD title of the axis in A10 format.

n — The absolute value of n is the number of characters in the title. If n is greater than or equal to 0, title and annotation on counterclockwise side of axis. If n is less than 0, title and annotation on clockwise side of axis.

axlen — The angle, in degrees, of the axis with respect to the positive X axis. A positive angle gives a counterclockwise rotation, a negative value gives clockwise rotation.

start — The initial value of the axis, plotted at the first tic mark of the axis; it could be the value computed by the SCALE routine or a value specified by the user.

scale — The increment added to the starting value for each inch along the axis; it could be the value computed by the SCALE routine or a value specified by the user.

LINE ROUTINE

The LINE routine produces a line plot from a set of points stored in a matched pair of X and Y arrays. The points may be represented by centered symbols, and/or connecting lines between points.

There are two methods of scaling the line plot; the array method using the starting and scale values calculated by the SCALE routine or the parameter method using the modal parameter offset, scalex, offsety, and scaley, set by the OFFSET routine.

Call format:

CALL LINE(x,y,n,incr,ilin,isym)

x,y — Arrays containing the virtual coordinates to be plotted and the values calculated by SCALE or their equivalent supplied by the user.

n — The number of points in the arrays to be connected by the line not including the start and scale values.

If n is positive, the array method of scaling is in force and the final three arguments of the call are interpreted as follows:

incr — The increment to be used in accessing the points in x and y (for example, incr equals 2 accesses every other value).

ilin — The absolute value of ilin is the occurrence period of the centered symbols (for example, if the absolute value of ilin equals 3, then every third point in x,y is marked by a centered symbol). If ilin equals 0, no centered symbols are drawn. If ilin greater than or equal to 0, solid lines are drawn between the points. If ilin is less than 0, no lines are drawn between the points.

isym — The number of the centered symbol to be used (refer to appendix C for a list of symbols) or the line type to be used. If ilin equals 0 and:

If isym = 2, points are connected with solid lines.

4, points are connected with dashed lines.

5, points are connected with dotted lines.

12, coordinates are modified by the modal parameters and connected by solid lines.

14, coordinates are modified by the modal parameters and connected by dashed lines.

15, coordinates are modified by the modal parameters and connected by dotted lines.

If n is negative the parameter method is used to scale the data, and the final three arguments in the call are interpreted as follows.

incr — The line type as specified in isym above.

ilin — The absolute value of ilin is the occurrence period of the centered symbols.

isym — The number of the centered symbol to be used (refer to appendix C).

When using the parameter method of scaling, LINE always enters arrays x and y using an increment of one (refer to figure 2-5).

CUBIC ROUTINE

The CUBIC routine draws a cubic spline curve through the set of points specified by corresponding elements of arrays x and y (refer to figure 2-6). There must be at least four pairs of points. CUBIC is not affected by the values calculated by SCALE.

Call format:

CALL CUBIC(x,y,n,ic)

x,y — A matched set of arrays containing the x coordinates and y coordinates, respectively, of the points through which a smooth curve is drawn starting at point (x(1), y(1)), and ending at point (x(n), y(n)).

n — The number of coordinate pairs in the arrays x and y (n must be 4 or larger).

ic — An integer which selects the type of line used to draw the curve and controls the computation of the coordinates using scales and offsets. Default is a solid line (2).

If ic =
- 2, a solid line is selected.
- 4, a dashed line is selected.
- 5, a dotted line is selected.
- 12, a solid line with coordinates computed using scales and offsets is selected.
- 14, a dashed line with coordinates computed using scales and offsets is selected.
- 15, a dotted line with coordinates computed using scales and offsets is selected.

RATIO ROUTINE

The RATIO routine sets the value of the modal parameter, ratio. This parameter controls the ratio of height to width of the alphanumeric characters and centered symbols drawn by routine SYMBOL. The alphanumeric characters are drawn on a seven by seven grid. The three right-most columns in the grid do not contain parts of the character, but serve as the inter-character space in a character string. Thus a ratio of 1 actually gives a character with a height-to-width of 7:4, since the grid has a height-to-width ratio of 1.

Call format:

CALL RATIO(ratio)

ratio — The ratio between heights and widths of character rectangles.

SYMBOL ROUTINE

The SYMBOL routine interprets its arguments in one of two ways, depending on the sign of the final parameter, n. If n is greater than or equal to 0, the character string mode is used, and if n is less than 0, then the centered symbol mode is used. These two modes of operation are described separately.

The Character String Mode (n≥ 0)

In this mode the SYMBOL routine draws a string of alphanumeric characters with the starting-point at the lower left-hand corner of the first character. Figure 2-7a is an example.

Call format:

CALL SYMBOL(x,y,size,iarray,angle,n)

x,y — Coordinates of the starting-point of the string. If (x,y) equals (999.,999.), the current pen position is assumed to be the starting-point of the character string.

size — An argument which performs two functions:

1. Its absolute value is character height in inches.
2. Its sign indicates the format of iarray unless n equals 0. Positive indicates A10 format is used. Negative indicates A1 format is used.

iarray — An array of BCD characters with the format and number selected by the size and n arguments.

angle — The angle, in degrees, with respect to the positive X axis, through which the baseline of the character string is to be rotated. A positive angle implies counterclockwise rotation, and a negative angle a clockwise rotation. The center of rotation is the starting-point (x,y) of the character string.

n — The number of characters in iarray.

- n > 0 — The number of characters in iarray.
- n = 0 — One character is to be taken from the right-most character position of the first word of iarray (for example, IARRAY(1)=1RA).
- n < 0 — Refer to centered symbol mode (below).

The Centered Symbol Mode (n < 0)

In this mode a line of the type specified by the n parameter is drawn to point x,y, where a centered symbol (specified by the isym parameter) is drawn. An example is shown in figure 2-7b.

Call format:

CALL SYMBOL(x,y,size,isym,angle,n)

x,y	The coordinates of the center of the centered symbol to which a line is drawn from the end-point of the previous display element produced by a call to a graphic data routine. If (x,y) = (999.,999.), the current pen position is assumed to be the center of the centered symbol.
size	The height, from bottom to top, of the centered symbol to be plotted.
isym	An integer between 0 and 13 selecting which of the centered symbols to be plotted (refer to appendix C).
angle	The angle in degrees that the centered symbol is rotated about its center. A positive value is a counterclockwise rotation. A negative value is a clockwise rotation.
n	A negative integer which selects the type of line which is generated from the starting-point to the center of the centered symbol. This is the first vector generated by this call.

If n ≥	0,	see character string mode is selected.
=	-1,	draw an invisible line is selected.
	-2,	draw a solid line is selected
	-3,	draw an invisible line is selected.
	-4,	draw a dashed line is selected.
	-5,	draw a dotted line is selected.
	-6 through -11	draw a solid line is selected.
	-12,	draw a solid line with scale and offset applied is selected.
	-13,	draw an invisible line with scale and offset applied is selected.
	-14,	draw a dashed line with scale and offset applied is selected.
	-15,	draw a dotted line with scale and offset applied is selected.
<	-15,	draw a solid line with scale and offset applied is selected.

```
      PROGRAM EXAMP7(OUTPUT)
      DIMENSION X(5),Y(5)
      DATA X/1.5,2.5,3.5,4.5,5.5/
      DATA Y/2.0,2.5,3.0,2.5,2.0/
C-----
C         INITIALIZE, DEFINE AND FRAME A PICTURE
C-----
      CALL PLTFIL
      CALL NEWPLT(1,0.0,0.0,7.0,5.0,0.0,0.0,0,7HSYMBOL2)
      CALL PLOT(0.0,0.0,-3)
      CALL PLOT(7.0,0.0,2)
      CALL PLOT(7.0,5.0,2)
      CALL PLOT(0.0,5.0,2)
      CALL PLOT(0.0,0.0,2)
C-----
C         DRAW A TITLE
C-----
      CALL RATIO(2.0)
      CALL SYMBOL(1.5,4.25,.5,16HCENTERED SYMBOLS,0.0,16)
C-----
C         DRAW SYMBOLS WITH CONNECTING LINES OF TYPES -1 THRU -5
C-----
      CALL RATIO(1.0)
      DO 10 I=1,5
      CALL SYMBOL(X(I),Y(I),.75,1,0.0,-I)
   10 CONTINUE
C-----
C         DRAW EACH SYMBOL AND ITS CORRESPONDING NUMBER
C-----
      XX=-.25
      DO 20 I=1,14
      J=I-1
      XX=XX+0.5
      CALL SYMBOL(XX,1.25,0.2,J,0.0,-1)
      CALL RATIO(1.25)
      CALL NUMBER(XX-.09,0.9,0.14,FLOAT(J),0.0,-1)
      CALL RATIO(1.0)
   20 CONTINUE
C-----
C         TERMINATE PLOTTING
C-----
      CALL PLOT(8.0,0.0,999)
      END
```

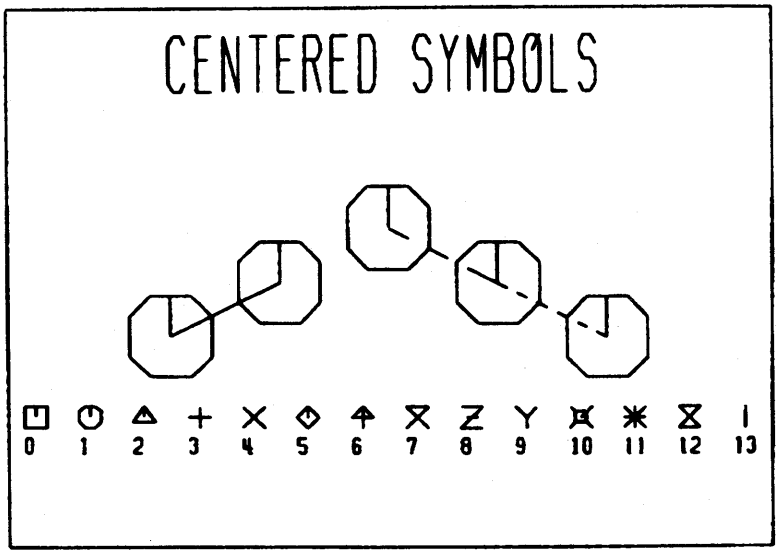


Figure 2-7b. Centered Symbol Mode

ARC ROUTINE

The ARC routine draws a circular arc. The format for the call is presented below, and figure 2-9 illustrates its use.

Call format:

CALL ARC(x,y,angle,xc,yc,ic)

x,y — Coordinates of the starting-point of the arc.

angle — The angle in degrees swept out by the arc. If the angle is positive, the arc moves in a counterclockwise direction away from the starting point. If the angle is negative the movement is clockwise.

xc,yc — Coordinates of the center of the arc.

ic — An integer which selects the type of line used to draw the arc and controls the computation of coordinates using scales and offsets. If an illegal value is specified, default to 2 occurs (solid line).

If ic =
- 2, a solid line is used.
- 4, a dashed line is used.
- 5, a dotted line is used.
- 12, a solid line with coordinates computed using scales and offsets is used.
- 14, a dashed line with coordinates computed using scales and offsets is used.
- 15, a dotted line with coordinates computed using scales and offsets is used.

Alternate call format: (ARC2)

CALL ARC2(x,y,xe,ye,xc,yc,isense,ic)

x,y — Coordinates of the starting-point of the arc.

xe,ye — Coordinates of the end-point of the arc. If they are the same as the start-point coordinates, a full circle is drawn.

xc,yc — Coordinates of the center of the arc.

isense — Direction in which the arc is formed.

If isense =
- 0, the direction is counterclockwise from the start-point.
- 1, the direction is clockwise from the start-point.

ic — An integer which selects the type of line used to draw the arc and controls the computation of coordinates using scales and offsets. If an illegal value is specified, a solid line (2) will be drawn (as default).

If ic =
- 2, a solid line is selected.
- 4, a dashed line is selected.
- 5, a dotted line is selected.
- 12, a solid line with coordinates computed using scales and offsets is selected.
- 14, a dashed line with coordinates computed using scales and offsets is selected.
- 15, a dotted line with coordinates computed using scales and offsets is selected.

TURNING ERROR MESSAGES ON OR OFF

Routines ERRON and ERROFF can be called to turn on error message capability or turn it off. The following message turns the capability on.

CALL ERRON

The following message turns it off.

CALL ERROFF

When on, ERRON issues messages if errors occur (refer to appendix F); conversely, ERROFF disables error checking mode when it is selected.

DEVICE MODE ROUTINES

Three device mode routines are provided for compatibility with ZETA 230 plotting devices. These routines place device mode records into the neutral picture file. Routine SMODE also sets the neutral plot routine ratio parameter to one per aspect. These routines are device dependent and should only be used if the ZETA is the desired plotting device. (Refer to the ZETA User's Guide, available from ZETA Research, Inc..)

Call formats:

CALL SLINE

CALL CLINE

CALL SMODE(ifast,aspect)

ADDITIONAL ROUTINES

The following routines are included for compatibility with the ZETA plotting routines. A call to any of these routines has no effect.

Calls:

CALL PON

CALL POFF

CALL TON

MERGING WITH A USER'S PROGRAM

The UNIPLOT neutral plot routines are CalComp call compatible. The majority of programs making standard calls to the CalComp basic plotting software should be able to use UNIPLOT with no changes. However, there are some exceptions. The following is a list of possible incompatibilities.

- If your program is overlaid, you may have to include two common blocks in the main (0,0) overlay.
- If your program contains routines with names in common with UNIPLOT routines, those names must be changed.
- If your program contains more than one CALL PLOT (X,Y,-3) per plot, you may wish to utilize the UNIPLOT PLTFIL and NEWPLT routines.
- If your program uses more than the standard fourteen centered symbols, the additional symbols must be changed.

AN OVERLAID PROGRAM

UNIPLOT uses two labeled common blocks for storing and transferring plot information. These common blocks are labeled /YCOM/ and /ZCOM/ and are 18 and 129 words in length, respectively.

If the program is overlaid, with the calls to the UNIPLOT routines in a primary or secondary overlay, the following statements should be included in the main (0,0) overlay.

COMMON/YCOM/UNI1(18)

COMMON/ZCOM/UNI2(129)

This addition to the main overlay ensures the plot information is not destroyed when another overlay is loaded.

DUPLICATE ROUTINES

Besides containing the routines with names compatible with the CalComp basic plotting software, UNIPLOT contains an expanded set of user callable routines and some internally called routines.

Table 2-3 contains a list of the UNIPLOT routines. If your program contains routines with a name listed in the table, the names in your program must be changed.

MULTIPLE PLOT ORIGINS

Some programs use CALL PLOT(X,Y,-3) to define multiple origins within one plot. In order for UNIPLOT to provide a selective drawing capability, the post processor must interpret this call as the beginning of a new plot unless the NEWPLT routine is used.

TABLE 2-3. UNIPLOT ROUTINES

ADDF	FASFIG	NEWPLT	SCALE
ADDP	FASGET	PON	SYMBOL
ARC	FASPUT	POFF	SLINE
ARC2	GETE	PUTE	SMODE
AXIS	GETF	PUTH	WHERE
CLINE	GETH	TON	XTRANS
CUBIC	GETP	NUMBER	XYCHECK
DEACTF	ICCHECK	OFFSET	ZDADO
DEACTP	INDEV	PLOT	ZPUT
ENDGRP	FACTOR	PLOTS	ZSLOPE
ERRON	GROUP	PLTFIL	ZTRANS
ERROFF	LINE	RATIO	
FASEBP	MIRROR	REACTF	
BASEBU	NEWPEN	REACTP	

CENTERED SYMBOLS

Some CalComp and Houston Instrument plotting software has the capability of using the fourteen special characters plus any of the standard sixty-four characters as centered symbols. UNIPLOT uses only the fourteen special characters as centered symbols (refer to appendix C). If a program uses one of the 64-character sets as a centered symbol, the calls to routine SYMBOL must be changed to invoke the character string mode instead of centered symbol mode. In addition, an x and y offset must be calculated since the lower left corner of the character is placed at the coordinate specified, not at the center of the character.

OVERALL OPERATION

It is assumed that the user is familiar with the NOS/BE or NOS operating systems, commands, and control cards as necessary, and is able to attach the UNIPOST file which contains the absolutized post processor program (refer to appendix A).

The post processor uses the following four files.

NPFILE	A neutral picture file.
PLOTF	The device dependent plot data output by the post processor.†
OUTPUT	A listing of the parameter input statements and error messages.
INPUT	Post processor directives.

The control card to initiate post-processor execution is as follows.

UNIPOST, D=device, file options.

device	Keyname of desired device (appendix D).	
file options	I=lfn	Local file name containing the post-processor directives.
	I	Default to reading directives from INPUT.
	I=	(Omitted), do not read directives if D is present; otherwise, default to reading directives from INPUT.
	O=lfn	Local file name to which all informative messages are written.
	O=	(Omitted) write informative messages to OUTPUT.
	F=lfn	Local file name of the neutral picture file.
	F=	(Omitted), default to reading neutral picture data from NPFILE.
	P=lfn	Local file name to which the device dependent plot data is written.
	P=	(Omitted), write device dependent plot data to PLOTF.

†The SC4020 device writes to FILMPL.

The default execution command is:

UNIPOST.

This command reads directives from INPUT, reads the neutral picture data from NPFILE, writes any informative messages to OUTPUT and creates a device dependent plot file named PLOTF. A directive record containing the DEVICE= name card must be supplied in the input stream (refer to Selecting a Device later in this section).

This command format cannot be used on an interactive SCOPE 3.4/INTERCOM system unless INPUT and OUTPUT are connected prior to execution.

All of the pictures on the neutral picture file can be obtained with no modification by using the following command.

UNIPOST,D=device name.

Using this command eliminates the need to provide a directive record on INPUT, since the post-processor defaults all directive values. If special directives are needed, an I must be added to the command as follows:

UNIPOST,D=device name,I.

This is the format which should be used for interactive SCOPE/INTERCOM. Add the I only if post-processor directives are to be used.

The logical file names of files NPFILE, PLOTF, OUTPUT, and INPUT may be changed (for example, respectively to PLOTIN, PLOTOUT, LIST, and DIRECT) by executing the post processor via a control card in the following form:

UNIPOST, F=PLOTIN, P=PLOTOUT, O=LIST, I=DIRECT.

DIRECTIVE INPUT STATEMENT SYNTAX

All directive statements take the form:

directive name = values*

The directive name may start in any character position in the record (that is, in any card column), and may continue over record boundaries. Eighty characters are scanned in each record. All blanks, even those within names and numbers, are ignored. The asterisk (*) is the delimiter between directives.

The set of legal directive names, along with the values permitted for each directive, is provided later in this chapter in the section entitled Post Processor Directives. The directive names must be less than or equal to ten characters and may be shortened to their first four characters.

MIRRORING

Mirroring is accomplished by executing a MIRROR directive before the appropriate DRAW or DELETE statement. The values of the MIRROR statement select mirroring in the X or Y dimension (or no mirroring), and if mirroring is selected, optionally define the line in the virtual space about which mirroring takes place.

Example:

```
MIRROR = Y, 2.0*
DRAW = 1*
MIRROR = OFF*
DRAW = 1*
MIRROR = X*
DRAW = 1$
```

This example turns on mirroring, draws picture A mirrored about a line which intersects the Y axis at the point 2.0 (refer to figure 3-1), and then turns mirroring off and draws picture A again without mirroring. Picture A is drawn for a third time, with mirroring in place in the X direction.

WINDOWING

The selection of a rectangular area within the virtual space, "windowing," is done using the WINDOW statement. The values of the window statement are two numeric ranges. Figure 3-2 illustrates the relationship of the axes to placement of window.

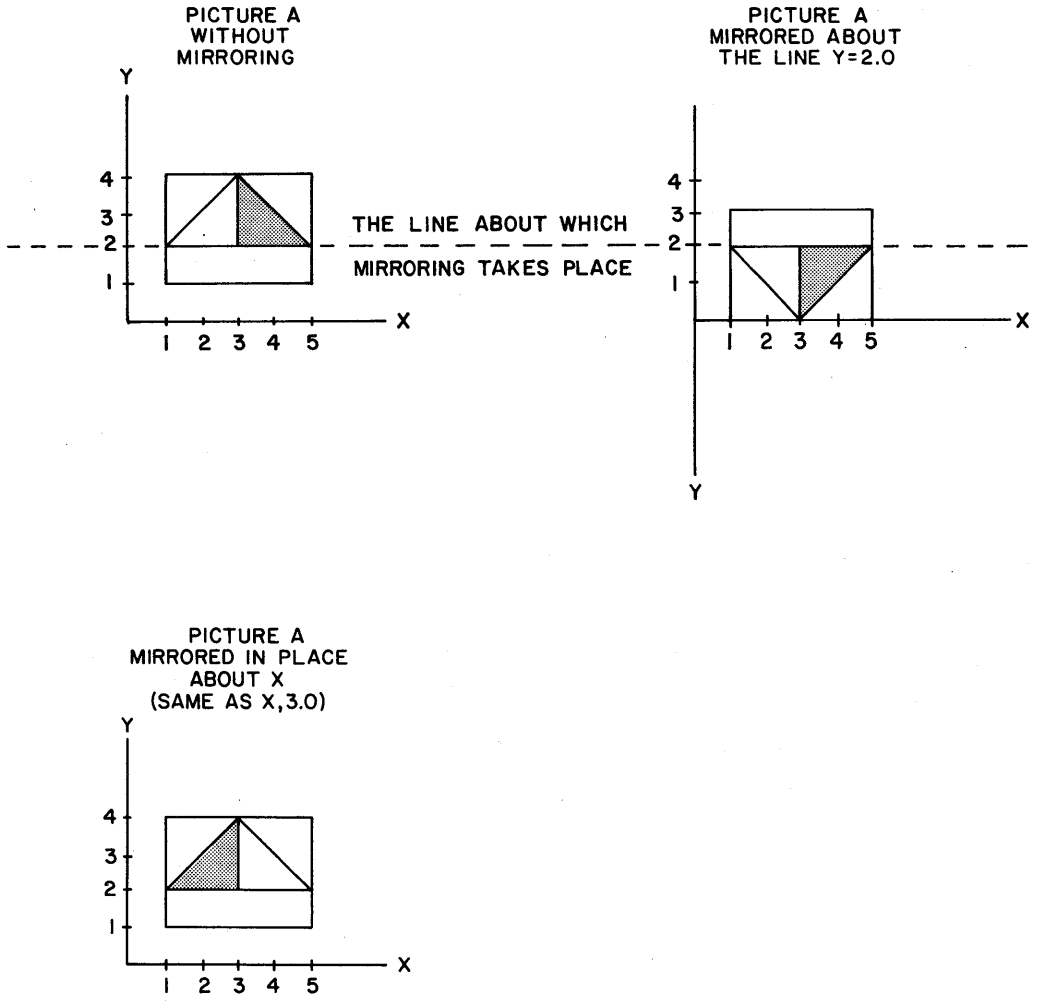


Figure 3-1. Mirroring

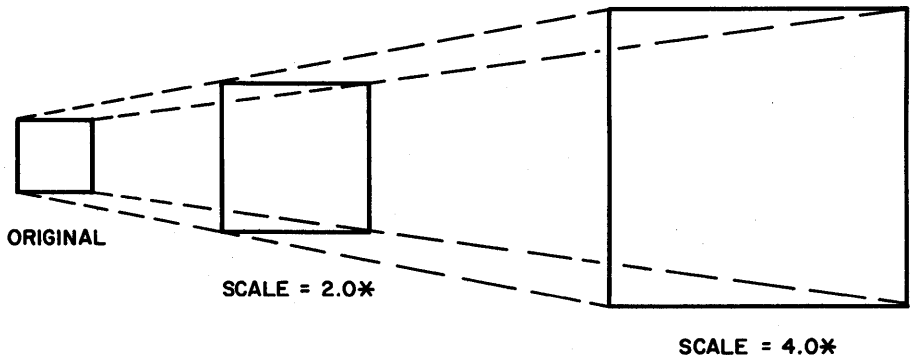


Figure 3-4. Modify with SCALE

PICTURE SIZE

The size of the picture may be varied by the SCALE or SIZE statements. The SCALE statement sets scale factors that multiply all coordinates of the picture. The X and Y coordinates may be multiplied by different values if two values are present in the scale statement. If the X and Y coordinates are multiplied by different values the picture is distorted when it is displayed or plotted. Figure 3-4 illustrates a two-dimensional square scaled upward using SCALE.

The SIZE statement specifies the length of the sides of a rectangle in inches on the display surface into which the selected window is to be fitted. This rectangle is called the display area (refer to figure 3-5). The picture is not distorted, even though some space in the display area is not filled by the windowed graphic data. SIZE is not to be confused with SURFACE which specifies the physical dimensions of the entire display surface available (that is, the total length and width of a paper roll for a drum plotter). Various examples of windowing are shown in figure 3-6.

```
WINDOW = 0-10, 0-20*
SIZE = 5, 5*
DRAW = 1*
```

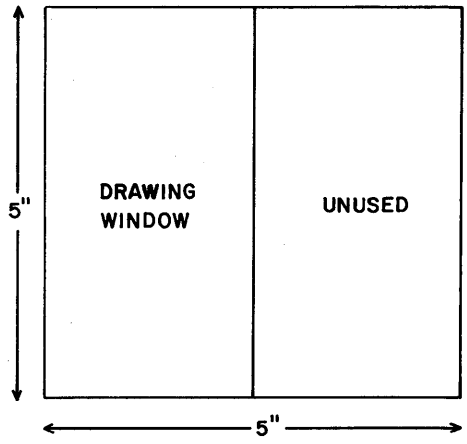


Figure 3-5. Window Size

SELECTIVE MODE

If the display device selected is a CRT, the selective mode is default. This mode places the lower left-hand corner of the picture window at the lower left-hand corner of the display surface unless otherwise specified by the ORIGIN directive.

The selective mode is also valuable for its "cut and paste" capability which could be used to position a string of pictures on a flat-bed plotter to make use of the entire surface. This is done by using the ORIGIN directive whose values are the X and Y display surface coordinates of the lower left-hand corner of the window.

Example:

```
ORIGIN = 17.5, 4*
```

The previous directive places the drawing 17.5 inches to the right and 4 inches above the origin. (The origin is the lower left-hand corner of the display.)

When selective mode is used on a device that is not a CRT, an ORIGIN directive must be used before the DRAW directive for each picture or overplotting occurs.

The available space on the display surface may be such that a rectangle selected by WINDOW needs to be turned sideways (90° in a counterclockwise direction) by the ROTATE statement.

```
ROTATE = ON*
```

The ROTATE directive remains active until it is turned off by the statement:

```
ROTATE = OFF*
```

In the following example segments of pictures 1, 2, and 6 are selected and placed on a 24 inch by 16 inch display surface.

```
WINDOW = 0-10, 0-15*
ORIGIN = 1, .5*
DRAW = 1*
WINDOW = 0-5, 0-3.5*
SCALE = 2*
ORIGIN = 13, .5*
DRAW = 2*
WINDOW = 5-10, 3-10*
SCALE = 1*
ORIGIN = 13, 8.5*
ROTATE = ON*
DRAW = 6$
```

This set of parameter statements results in the layout illustrated in figure 3-7.

A more extensive example of the use of window and the selective mode, to enlarge a selected portion of a picture, is found under WINDOW Statement later in this section.

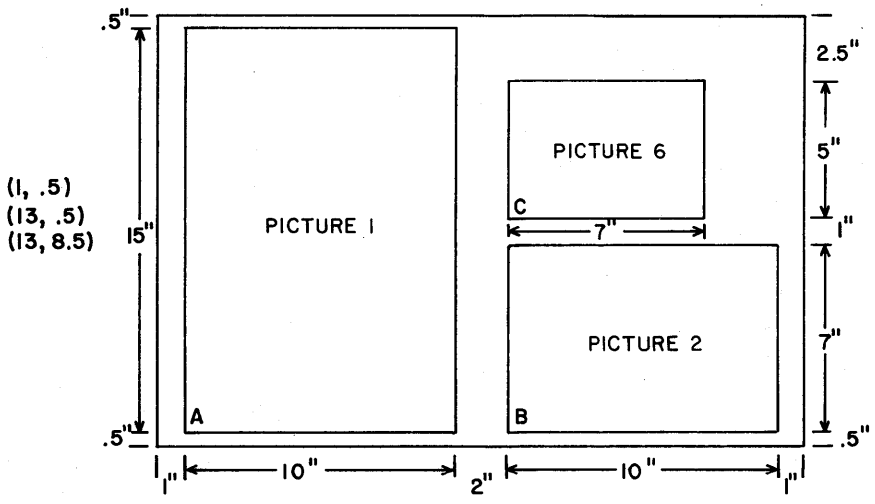


Figure 3-7. Picture Layout

Statement format:

SURFACE = xlength, ylength*

xlength	Horizontal length of the display surface in inches. This is the length of the roll of paper on a drum plotter or the length of the bottom edge of a flat-bed or CRT.
ylength	Vertical length of the display surface in inches. This is the width of the paper on a drum plotter or the number of inches from the bottom to the top of a flat-bed or CRT.

Example:

Assume the device being used is a Houston Instrument 22-inch plotter attached to a 200 user terminal. The default size for DEVICE = HI200 is 11 by 1,440 inches. If the default surface size were used, it would be impossible to obtain plots larger than 11 inches. Therefore, the following statements would be used.

DEVICE = HI200*SURFACE = 22, 1440*

PEN STATEMENT

The PEN statement allows the user to specify the number of pens available on his plotter if it differs from the default as listed in appendix D.

Statement format:

PEN = n*

n	Number of pens available on the plotter.

RESTORE STATEMENT

The RESTORE statement serves two purposes. It restores all directive values back to their default or moves the pen to the bottom edge of a plotter.

Statement formats:

RESTORE*

or

RESTORE = PEN*

If the first format is used, all directive values except the device name, increment, size and number of pens are reset to their defaults. This is particularly useful when using an interactive device such as the Tektronix product. When the second format is used, the post processor generates a command to move the pen to the bottom of a pen plotter. If this feature is desired between every picture the following sequence should be used.

RESTORE = PEN*DRAW=1*
RESTORE = PEN*DRAW=2*
RESTORE = PEN*DRAW=3*

DRAW STATEMENT

The DRAW statement selects a set of pictures from the neutral picture file and displays them on the selected device. A maximum of 30 parameters may be used. All pictures specified are modified by the directive statements preceding the DRAW. Omitting the DRAW statement causes all pictures on the neutral picture file to be displayed.

Statement format:

DRAW = number 1 number n *
numeric range 1 ,..., numeric range n

or

DRAW*

number n	A positive integer in the parameter list which specifies a picture on the neutral picture file to be plotted.
numeric range n	An inclusive specified sequence of picture on the neutral picture file to be plotted. The ends of the ranges must be positive integers or the keyword END which draw through the end of the data.

Example:

DRAW = 1,3,6-8,12-END*

The previous example displays pictures 1, 3, 6, 7, 8, and 12 through the end of data.

Example:

DRAW*
DRAW = 1-END*

The previous examples along with omitting the DRAW display the entire neutral picture file.

NOTE

For a further description of the numeric range, refer to the Directive Input Statement Syntax earlier in this section.

DELETE STATEMENT

The DELETE statement specifies that all pictures from the neutral picture file except the ones specified in the parameter list are to be displayed on the selected device. A maximum of 30 parameters may be used.

type 1 — An area for which the minimum and maximum virtual coordinates for X and Y are specified (for example, WINDOW=0-8,0-11*)

type 2 — An area for which the maximum Y virtual coordinate is omitted (for example, WINDOW=0-8,0-END*)

type 3 — An area for which the maximum X virtual coordinate is omitted (for example, WINDOW=0-END,0-11*)

type 4 — An area for which both of the maximum virtual coordinates are omitted (for example, WINDOW=0-END,0-END*)

It should be noted that a default window is always present as specified by the user program. If END is used in place of a maximum coordinate, the default value is used.

If a user wishes to use the window statement, he or she should use the INFORM statement to determine the virtual coordinates of the default window.

Window maxima and minima refer to the virtual space or user coordinates of the neutral picture file, before any scaling, mirroring or rotating takes place. For an example of windowing, refer to figures 3-6 through 3-9.

INFORM STATEMENT

The INFORM statement allows the user to obtain a list of the default window in the virtual space for each picture displayed. This window can then be used to determine the position in the virtual space of any portion of the picture.

Statement format:

```
                number 1              number n      *
INFORM =
                numeric range 1  ,...,  numeric range n
```

or

```
INFORM*
```

number n — A positive integer which specifies a picture on a neutral picture file for which the window information is printed.

numeric range n — An inclusive specified sequence of pictures on the neutral picture file for which the window information is printed. The ends of the ranges must be positive integers or the keyword END which lists through the end of data.

If the window information is wanted before any pictures are displayed, the DELETE statement must be used in conjunction with INFORM as follows:

```
INFORM*DELETE*
```

This example lists the information for all pictures on the neutral picture file without displaying any of them.

```
INFORM = 1, 5, 8-11 *
```

The above directive lists the information for pictures 1, 5, 8, 9, 10, and 11 and displays all pictures (omitting the DRAW directive defaults to display everything on the neutral picture file).

SCISSOR STATEMENT

The SCISSOR statement allows the user to turn on and off the scissoring feature of the post processor. If a line or character extends across the window boundary or outside the display surface, and scissoring is activated, the line is truncated at the point of intersection. Default is OFF unless the WINDOW directive has been used. This value remains set until another SCISSOR directive is encountered.

Statement format:

```
                ON
SCISSOR =                 *
                OFF
```

If partitioning of a picture is desired in linear placement mode which is default for a drum plotter, use the following:

```
MODE = LINEAR*
SCISSOR = ON
```

ROTATE STATEMENT

The ROTATE statement causes the window to be rotated 90° counterclockwise on the display surface and the new lower left-hand corner of the window placed at the specified display surface origin.

Statement format:

```
                ON
ROTATE =                  *
                OFF
```

It should be noted that for rotation to be possible, the window must have definite maximum coordinates.

MIRROR STATEMENT

The MIRROR statement allows a user to obtain mirror image displays of neutral picture file pictures.

Statement format:

```
                X, number 1
MIRROR =        Y, number 2       *
                OFF
```

number 1 — Optional X coordinate of a vertical line about which the picture is to be mirrored.

number 2 — Optional Y coordinate of a horizontal line about which the picture is to be mirrored.

OFF — No mirroring.

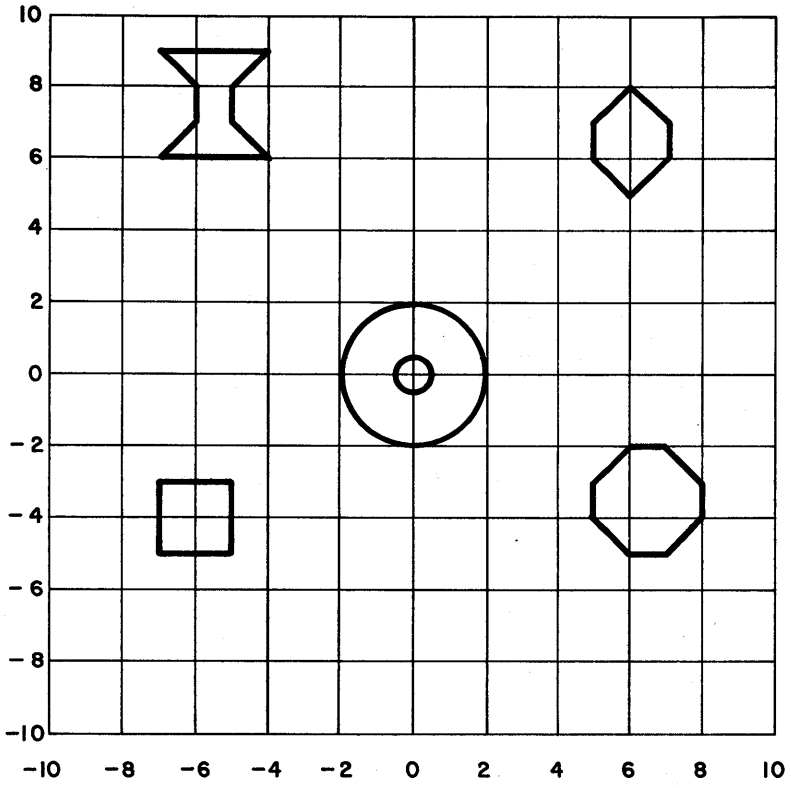


Figure 3-9. Virtual Picture Creation

NOTE

The Tektronix 4014 automatically initiates MODE=SELECTIVE.

DRAW=1* Display the first picture of the neutral picture file placing the lower left-hand corner of the default window (-10-10,-10-10) at the default origin (0,0) of the display surface.

NOTE

The various dotted lines in the drawings are not displayed, but are used in the examples to show the area of the window on the display surface.

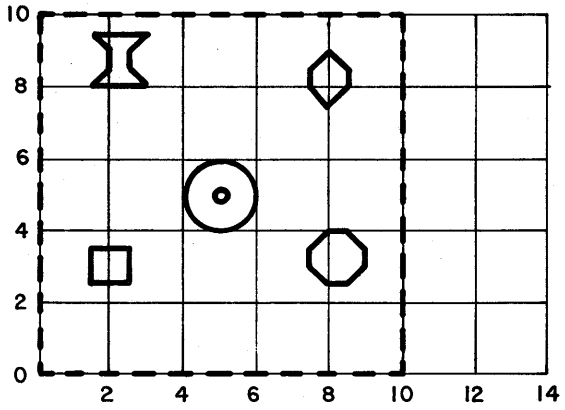


Figure 3-10. Window Area on Display Surface

Figure 5-11 illustrates how the display appears when only the center of the original virtual picture (figure 3-9) is called into the window by the following directives:

TABLE 3-1. POST PROCESSOR DIRECTIVES

Directive	Purpose	Default
CHAR = HARD* = SOFT*	Select hardware or software generated characters	SOFT
DELETE = picture number*	Select the pictures which should not be displayed	All displayed
DEVICE = keyname* = DUMP*	Select the display device or get a debug dump of the neutral picture file	Fatal error
DRAW = picture numbers*	Select the pictures for display	All displayed
INCR = increment*	Select the increment size of the device	Refer to appendix D
INFORM = picture number*	Obtain a listing of the window information for the specified picture	No list
MIRROR = X,const* = Y,const* = OFF*	Obtain a mirror image of the pictures specified on the following draw	OFF
MODE = FILE* = LINEAR* = SELECTIVE*	Select a placement mode	FILE for flat-bed devices; LINEAR for drum devices; SELECTIVE for CRT
ORIGIN = x,y*	Define the location on the display surface of the lower left-hand corner of the window (used with SELECTIVE mode)	0,0
OVERPLOT = ON* = OFF*	Specify if a screen erase will occur between pictures when displayed on a CRT	OFF
PEN = n*	Specify the number of pens on the device	Refer to appendix D
RESTORE*	Reset all directives to the default values (does not affect DEVICE)	No action
RESTORE = PEN*	Drive the pen to the bottom edge of a plotting device	No action
ROTATE = ON* = OFF*	Rotate 90 degrees counterclockwise all pictures specified on the following DRAW	OFF
SCALE = xfact,yfact*	Reduce or enlarge the virtual picture	1.0,1.0
SCHAR = fact*	Scale the characters separately from the rest of the picture	SCALE value
SCISSOR = ON* = OFF*	Specify scissoring of vectors if they cross display surface or window boundaries	ON
SIZE = xlength,ylength*	Define the size of the display area on the display surface	Display surface size
SURFACE = length,width*	Define the size of the display surface	Refer to appendix D
WINDOW = xmin-xmax,ymin-ymax*	Define the portion of the picture to be displayed	Refer to INFORM listing

SCOPE 3.4

The following is an example of compiling, loading and executing the user's UNIPLOT program and post-processing the neutral picture file in one job step on SCOPE 3.4.

Command	Description
MYJOB,MT1.	
USER,...	
FTN.	Compile user's program.
APPLIC(UNIPLT)	Attach UNIPLOT.
REQUEST,NPFILE,*PF.	Request permanent file space.
LGO.	Load and execute program.
CATALOG,NPFILE,NPFILE, ID=MYID.	Catalog neutral picture file.
FILE(PLOTF,RT=S,BT=C)	Define record type for plot tape.†
REQUEST,PLOTF,MT,HI, RING,NL,SV.	Request the plot tape.††
UNIPOST,D=TPLOT.	Execute post processor with defaults.

7-8-9

```
PROGRAM MYPRO(OUTPUT)
CALL PLTFIL
      .
      .                    User's program.
      .
CALL PLOT(0,0.,999)
END
```

6-7-8-9

To execute the post-processor for an on-line device such as the Houston Instrument use the following:

Command	Description
APPLIC(UNIPST)	Attach the post processor.
ATTACH,NPFILE,ID=MYID.	Attach the neutral picture file.
UNIPOST,D=HI200.	Execute post processor.
REWIND,PLOTF.	Rewind the plot file.
COPYBF,PLOTF,OUTPUT.	Copy the plot file to OUTPUT.

6-7-8-9

To CYBERLINK the neutral picture file to NOS for previewing use the following.

Command	Description
ATTACH,NPFILE,ID=MYID.	Attach the neutral picture file.
COPYBR,INPUT,SEND.	Copy the NOS control cards to the CYBERLINK file.
COPYBF,NPFILE,SEND.	Copy the neutral picture file to the CYBERLINK file.
ROUTE,SEND,DC=FR,ST=id.	CYBERLINK the file.†††

7-8-9

JOB.	
ACCOUNT,number,password, family.	
CHARGE,number,project.	
DEFINE,NPFILE.	Define the permanent file.
COPYBF,INPUT,NPFILE.	Copy the neutral picture file to the permanent file.

6-7-8-9

†The FILE card is not needed for most on-line plotting such as Houston Instrument or UNITECH. It should only be used if the device expects a binary formated file. CYBERNET users should list APPINFO for this information.

††Check with the local center for plotting procedure and format of the REQUEST card. No tape is required if plotting is done through a remote terminal.

†††id = KEA; System A
KEB; System B
KEC; System C

Command	Description
RECOVER/CHARGE: CHARGE, number, project	Enter charge number if applicable.
READY.	
NEW,MYPRO	Define a new local file.
READY.	
TEXT	Enter text mode.
ENTER TEXT	
PROGRAM MYPRO (OUTPUT) CALL PLTFIL . . . CALL PLOT (0.,0.,999) END	Enter program.
EXIT TEXT MODE	Type BREAK key to exit text mode.
READY.	
PACK	Pack the new file.
READY.	
NOSORT	Do not sort the file by line numbers.
READY.	
SAVE	Save MYPRO as an indirect access file.
READY.	
DEFINE,NPFILE=TAPE1.	Use this command if the neutral picture file should be saved.
-UNIPROC F=MYPRO	Execute user's program and post-process the neutral picture file on a Tektronix 4014.
ENTER TERMINAL TYPE	
1 = 4010 SYNCHRONOUS 2 = 4010 ASYNCHRONOUS 3 = 4014 SYNCHRONOUS 4 = 4014 ASYNCHRONOUS 5 = 4014 EGM SYNCHRONOUS 6 = 4014 EGM ASYNCHRONOUS	
?2	Enter your terminal type. All NOS communications are asynchronous.
INPUT DIRECTIVES	
?$	Enter a $ to display all pictures with default parameters. If an * is entered directives will again be requested after all pictures are displayed.

The pictures are now displayed. Whenever a bell sounds and a question mark is printed, type a C and a carriage return to continue.

Use the following commands for post-processing to a ZETA plotter.

Command	Description
-UNIPROC, S=1POST (NPFILE = TAPE 1, TEK = ZETA)	
ENTER PLOTTER TYPE 1=230, 2=1240, 3=3640	
?1	Specify a model 230.
ENTER BAUD RATE	
?300	Specify baud rate.
INPUT DIRECTIVES	
?INCR=.005$	Change the increment size.
READY.	
REWIND, PLOTF	Rewind the plot file.
READY.	
BATCH	Enter batch mode.
RFL, 20000.	
/COPY, PLOTF	Copy plot data to output to obtain the plot on the ZETA plotter.

Use the following commands for post-processing to a Houston Instrument unit.

Command	Description
ASC II	Enter ASCII mode.

Command	Description
READY.	
SAVE	Save MYPRO as an indirect access file.
READY.	
BATCH, 60000	Enter batch mode requesting 60,000 octal words of core.
/FTN, I=MYPRO	Compile MYPRO.
/ATTACH, UNIPLOT/UN=number	Attach UNIPLOT library from installation user number.
/LIBRARY, UNIPLOT	Define UNIPLOT as a library.
/DEFINE, NPFILE	Define permanent file space for the neutral picture file.
/LGO	Execute user program.
END MYPRO	
/ATTACH, UNIPOST/UN=number	Attach post-processor from installation user number.
/UNIPOST, D=TEK	Initiate execution for a Tektronix device.
ENTER TERMINAL TYPE	
1 = 4010 SYNCHRONOUS	
2 = 4010 ASYNCHRONOUS	
3 = 4014 SYNCHRONOUS	
4 = 4014 ASYNCHRONOUS	
5 = 4014 EGM SYNCHRONOUS	
6 = 4014 EGM ASYNCHRONOUS	
?2	Enter your terminal type.
DIRECTIVES	
?$	Enter $ to display all pictures. Using an * causes a request for directives after all pictures are displayed.

The pictures are now displayed. Whenever a bell sounds and a question mark is printed, type a C and a carriage return to continue.

NOS/BE

The following is an example of a deck which would be submitted to NOS/BE through a batch terminal to create a tape for off-line plotting.

Command	Description
MYJOB, MT1.	Job card.
USER,	Accounting information if applicable.
FTN.	Compile user's program.
ATTACH, UNIPLOT, ID = UNIPLOT, MR=1.	Attach UNIPLOT neutral plot routine.
LIBRARY, UNIPLOT.	Define UNIPLOT as a library.
REQUEST, NPFILE, *PF.	Request permanent file space.
LGO.	Execute user's program.
CATALOG, NPFILE, ID=MYID.	Catalog neutral picture file.
ATTACH, UNIPOST, ID= UNIPLOT, MR=1.	Attach the post processor.
FILE (PLOTF, RT=S, BT=C)	Define record type for plot tape.†

Command	Description
REQUEST,PLOTF,MT,HI, RING,N,VSN=XXXXX.	Request plot tape.
UNIPOST,D=CAL570.	Execute post processor.

7-8-9

```
PROGRAM MYPRO(OUTPUT)
CALL PLTFIL
      .
      .
      .
CALL PLOT(0.,0.,999)
END
```

User's program.

6-7-8-9

To execute the post processor for a device attached to a remote printer (that is, a Houston Instrument DP-3 on a 200 user terminal), use the following.

Command	Description
ATTACH,UNIPOST,ID= UNIPLOT,MR=1.	Attach the post processor.

†This file card is only needed for those devices which expect binary formatted plot data if not output by a BUFFER OUT.

Figure B-1 illustrates the UNIPLOT software produced character set available for plotting text. The characters are plotted in display code order from the top line to the bottom line in table B-1.

TABLE B-1. UNIPLOT CHARACTER SET

Octal Display Codes	Plotted Characters	Octal Display Codes	Plotted Characters
01	A	41	6
02	B	42	7
03	C	43	8
04	D	44	9
05	E	45	+
06	F	46	-
07	G	47	*
10	H	50	/
11	I	51	(
12	J	52	)
13	K	53	$
14	L	54	=
15	M	55	blank
16	N	56	,
17	O	57	.
20	P	60	≡
21	Q	61	[
22	R	62	]
23	S	63	÷
24	T	64	≠
25	U	65	→
26	V	66	∨
27	W	67	∧
30	X	70	↑
31	Y	71	↓
32	Z	72	<
33	0	73	>
34	1	74	≤
35	2	75	≥
36	3	76	¬
37	4	77	;
40	5	00	:

This plot contains the representation of the fourteen UNIPLOT centered symbols.

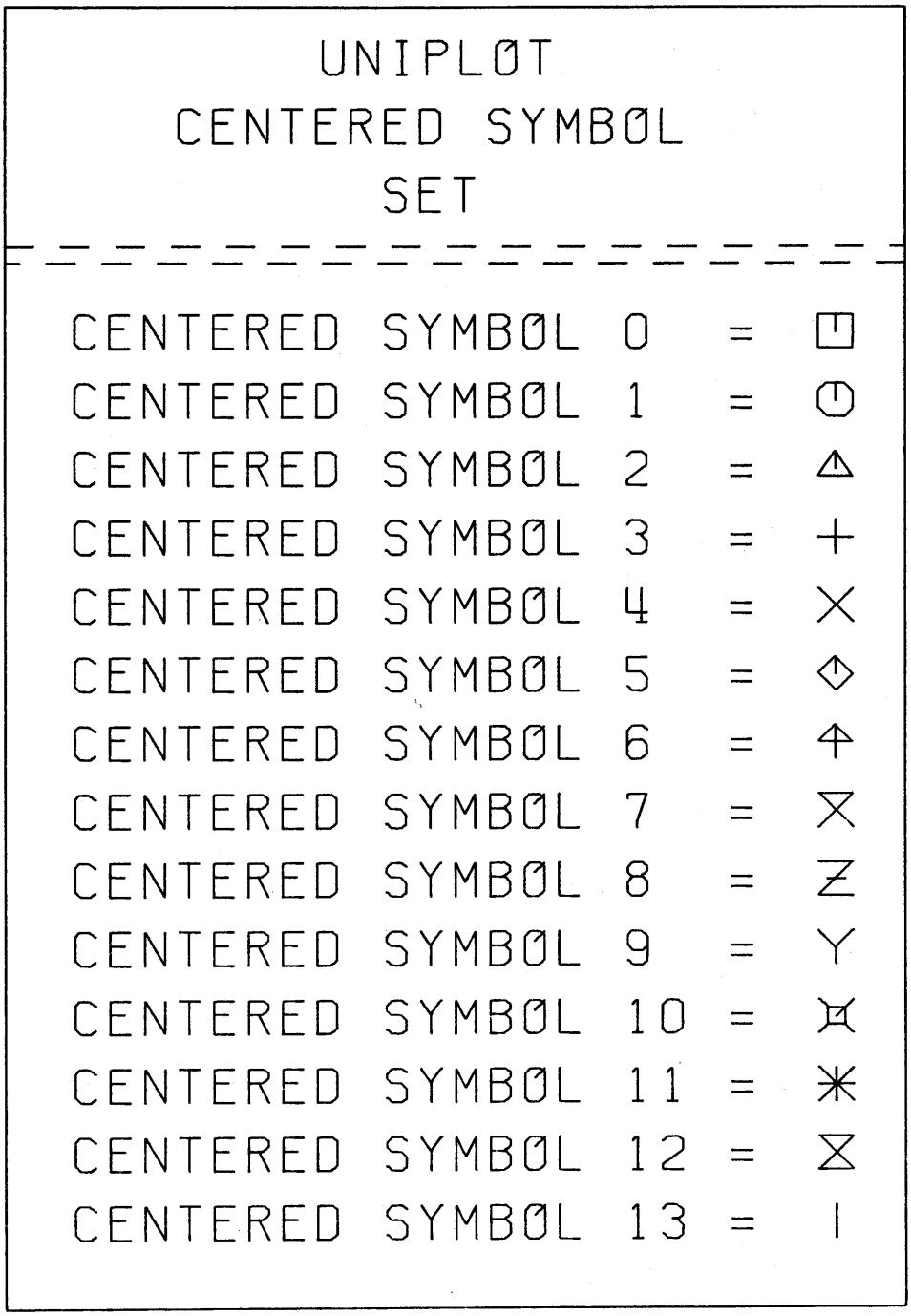


Figure C-1. UNIPLOT Centered Symbol Set

The devices shown in table D-1 are supported by UNIPLOT under SCOPE 3.4 and NOS/BE. The keynames listed for a device are used on the UNIPOST control card or the DEVICE directive statement. Table D-2 lists devices supported by UNIPLOT under NOS. Contact the local CDC office if a needed device is missing.

TABLE D-1. DEVICES SUPPORTED ON NOS/BE (SCOPE 3.4)

Device	Increment	Size	No. of Pens	Keynames
CalComp 770 Controller 763 Plotter	.005	1728x28	1	BOSTON CAL770 MPLSZIP
CalComp 780 Controller 763 Plotter	.005	1728x28	1	CAL780 SUNNYVA
CalComp 7-Track 900 Controller 936 Plotter	.005	1728x30	3	C9007T NEWYORK
CalComp 9-Track 900 Controller 936 Plotter	.002	1728x30	3	C9009T
CalComp 900 Controller † 1136 Plotter	.0025	1440x34	3	ROCKVIL
CDC 160-A/165 Controller CalComp 563 Plotter	.001	1200x30	1	LA MPLS PHILY TPLOT TUSCON
CDC MARC IV Terminal CalComp 570 Controller 565 Plotter	.005	1728x10	1	CAL570 DALLAS
CDC MARC IV Terminal CalComp 770 Controller 763 Plotter	.0025	1728x30	1	MARC770
CDC MARC III Terminal CalComp 915 Controller 1136 Plotter	.001	1440x34	3	MARC915
CDC 200 User Terminal Houston Instrument BTC-7 Controller	.01	1440x10.5	1	HI200
COPE 1200 Terminal † 200 UT Mode CalComp 563 Plotter	.005	1200x30	1	COPE2UT COPEOCC
Houston Instrument MTR4 7 Track Controller DP-7 Plotter	.002	1800x34.5	3	HOUSTON HIMTR47T
Houston Instrument MTR4 9-Track Controller DP-7 Plotter	.0025	1800x34.5	3	HIMTR9T

The following are the functions performed by a plotting device routine. After a listing of the functions, a sample routine is provided.

- A labeled COMMON block, /ZCOM/, is provided with variable, ITYP, and an array, IBUF.
- A plotting device routine converts its numeric arguments into a standard fixed-point decimal format. If the argument is an integer, the conversion is performed by the statement:

  J = I*10000

  If the argument is a floating point, then the conversion statement is:

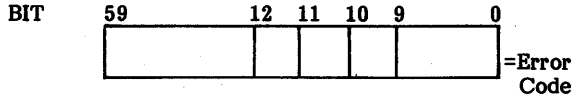
  J = IFIX (X*10000.0+SIGN(.5,X))

  A BCD argument with a limit of 10 characters need not be converted.
- The routine places the converted arguments into array IBUF, starting with IBUF(3), and going on to a maximum number of arguments at IBUF(14).
- The routine places the name of the plotting device, a BCD string, in IBUF(1). This name is the same as the name used in the post processor DEVICE statement.
- The routine also places the name of the present routine, a BCD string, in IBUF(2).
- And lastly, the routine places an 11 in ITYP and calls routine ZPUT, which has no arguments but uses ITYP and IBUF in labeled common/ZCOM/ for communication. ZPUT writes the necessary data to the neutral plot file.

Example:

```
SUBROUTINE CAMRAV (ICAM)
COMMON/ZCOM/ITYP,IBUF(3), I(97)
ITYP = 11
IBUF(1) = 6HSC4020
IBUF(2) = 6HCAMRAV
IBUF(3) = ICAM*10000
CALL ZPUT
END
```

The neutral plot routines optionally return an error code to the calling program as previously described under Error Codes in section 2. If the returned code is zero, then no errors in the calling sequence were detected. If the returned code is nonzero, then the type of error is indicated by which bits are set in the code as shown below.



BITS 0-9: Bits 0 through 9 are set when the parameter of corresponding number is out of range or has illegal value. For example, if the second and fourth parameters in a call to a neutral plot routine are out of range, bits 1 and 3 in the returned error code are set.

BIT 10: Bit 10 is set when the parameter list in an ARC2 subroutine call does not define a circle.

BIT 11: Bit 11 is set by the CUBIC subroutine when the nth and (n+2)th coordinate pairs in the X, Y points arrays are the same.

BIT 12: Bit 12 is set by the LINE subroutine when either the DELTAV or FIRSTV parameters are equal to zero. DELTAV and FIRSTV are usually computed by SCALE prior to calling LINE.

NOTE

The calling application need not initialize the error code prior to using it in a call.

Messages	Explanation
****BAD NUMERIC RANGE	1. Second parameter in numeric range; if not, numeric is not END. 2. If both parameters are numeric then first number is greater than second number.
****DEVICE SPECIFIED ON CONTROL CARD	Device has been given on the UNIPOST control card and overrides this specification.
****FIELD xx INVALID TYPE	Field number in quotes is not the proper type.
****INVALID CONTROL CARD	The UNIPOST control card contains an error. Program is aborted.
****INVALID SYMBOL(S) PRECEDING=	The UNIPOST control card has either more than one character or an illegal character preceding an = sign. Program is aborted.
****LINEAR MODE ILLEGAL FOR ALL BUT DRUM TYPE PLOTTERS	A MODE statement selected LINEAR plot placement but device is not a drum plotter. Processing continues.
****ORIGIN NOT WITHIN PLOTTER LIMITS	An ORIGIN statement specified that a drawing be placed outside the display surface. Processing continues.
****OVERPLOT REQUIRES SELECTIVE DEVICE	OVERPLOT has been turned on but device is not a selective type device. Processing continues.
****PARAMETER COUNT EXCEEDS MAXIMUM LENGTH	Maximum of 30 parameters on a directive has been exceeded.
****PARAMETER(S) MISSING	Directive is missing one or more parameters.
****PICTURE xxxxx NOT IN NEUTRAL PLOT FILE	The drawing selected on a DRAW statement (or implicitly on a DELETE statement) was not found in the neutral plot file. Processing continues.
****TOO MANY PARAMETERS	Directive contains more parameters than permitted.
****WINDOW TRUNCATED	A specified window lies outside the picture or outside the display surface upon transformation. Window limits are truncated to fit and processing continues.
****xxxx CONTAINS ILLEGAL CHARACTER	Parameter in quotes contains an illegal character.
****xxxx CONTAINS NEGATIVE PARAMETER	Directive in quotes contains a negative parameter.
****xxxx EXCEEDS MAXIMUM CHARACTER LENGTH	More than 10 characters in field.
****xxxx ILLEGAL KEYWORD	Parameter occurring to right of an = in a directive is not a keyword recognized by UNIPLOT.
****xxxx IS AN ILLEGAL DEVICE	The plotting device specified by the DEVICE statement or UNIPOST control card is not supported by UNIPLOT. Processing continues with the next input statement.
****xxxx IS AN INVALID DIRECTIVE	Not a UNIPLOT directive.

A dump of the neutral picture file is obtained by the following:

```
UNIPOST,D=DUMP.
7-8-9
DUMP=1$          for single picture
        or
DUMP=1-END$      for an inclusive numeric range.
```

If DUMP is used on an INTERCOM system, INPUT and OUTPUT must be connected prior to executing UNIPOST. A sample dump is included at the end of this appendix (figure H-2). The following discussion describes the contents of the dump.

FILE CREATION DATA

This information uniquely identifies a neutral picture file and is printed at the beginning of each dump.

JOB	Job name used when the file was created.
ACCOUNT	Account name used when the file was created.
DATE:	Date that the file was created.
TIME:	Time of day that the file was created.

PICTURE AND FIGURE DATA

This heading is printed to indicate the beginning of picture and figure information. The heading will be printed once for each picture.

PICTURE DATA

Picture data is printed immediately following the picture and figure data heading. This information will be printed once for each picture.

PICT NO	The picture number.
PICT NAME	The picture name. The name DEFAULT is given to a picture if a name was not included on the call that created the picture. This name appears for all pictures created by UNIPOST.
A	Picture active/inactive status. All pictures created by UNIPOST are active =Y (Picture active) =N (Picture inactive)
R	Picture rotation 0 - Picture is not to be rotated. 1 - Picture is to be rotated.
X - ORIG	Picture X origin.
Y - ORIG	Picture Y origin.
X - MAX	Maximum picture X coordinate.
XMIN	Minimum picture X coordinate.
XMAX	Maximum picture Y coordinate.
XMIN	Minimum picture Y coordinate.

FIGURE DATA

Figure information will be printed once for each figure of a picture.

FIGURE NO.	The figure number.
A	Figure active/inactive status. All figures created by UNIPOST are active. =Y (Figure active) =N (Figure inactive)
X ORIG	Figure X origin.
Y ORIG	Figure Y origin.

ELEMENT DATA

Element information is printed for each figure and immediately follows the figure data.

A type code and variable number of words is printed for each element record contained on the neutral picture file. The number of words that are printed for records that contain a fixed number of entries remain constant. For example, one word will always be printed for the pen select records which have one entry (pen number). The number of words printed for records that contain entries that repeat a variable number of times depends on the number of occurrences of the repeating entries and any fixed entries which are always contained within the record. For example, a vector record always contains three words as a fixed portion of the record and two additional words for each vector contained in the record. Thus, a one-vector record would be printed in five words and a seven-vector record would be printed in 17 words.

Element records are printed as an array. Contents of the individual words are dependent on the record type and the number occurrence of repeated entries. Record contents are printed in figure H-1.

Definition of the record entries are as follows:

Word	Parameter	Contents
1	IGNAM	Group name
2	ISGNAM	Sub-group name
3	IPEN	Pen number
4	INUM	Number of occurrences of a set of repeating parameters. Number of characters for a BCD character string.
5	INTEN	Intensity to be used when displaying the display items. 0 Ignore 1 INTEN 16- New intensity setting.
6	ISTYLE	Line style 0 Symbol plotters with interconnected lines. 1 Symbol plotting only. 2 Solid line. 3 Invisible line. 4 Dashed line. 5 Dotted line.
7	ISUB	Subroutine name.
8	IXCO	X coordinate relative to previous X origin.
9	IYCO	Y coordinate relative to previous Y origin.
10	IANG	Angle. BCD string-angle of base line.
11	ISYM	Symbol to be plotted.
12	IHIGH	Character height.
13	IRAT	Ratio of character width to height.
14	IBCD	Words containing character string.
15	IXSCO	Starting X coordinate of an ARC.
16	ITSCO	Starting Y coordinate of an ARC.
17	IXCCO	X coordinate of center point of an ARC.
18	IYCCO	Y coordinate of center point of an ARC.
19	IVAL	A parameter to be passed to a subroutine.

Annotation	The labels drawn at one inch spacing along the axis.
Application programmer	The programmer who writes graphics programs making calls to UNIPLOT. Although the console operator and application programmer may at times be the same person, this term refers only to the programmer.
Argument	Parameters entered by the graphics program in a call to the UNIPLOT routines.
Centered symbol	A special symbol which can be drawn with a virtual point at its center.
Command mode	The INTERCOM (or TELEX) mode which allows the graphics terminal user to enter and execute time-sharing commands directly from the terminal.
CRT	Cathode-ray tube.
Device independent file	A plot data file which can be displayed on any device through post-processing. It is not in a format compatible with any specific device.
Directive	The input to the post-processor which controls device specification size and placement.
Directive record	The file containing the post-processor directives; usually INPUT.
Display area	A rectangular portion of the display surface in which the picture is displayed. The display area is defined by the SIZE directive.
Display device	A device capable of presenting, on a viewing surface or image area, the display elements that provide a visual representation of data, for example, cathode-ray tube, pen plotter, or microfilm plotter.
Display elements	The basic building symbols used by an application to construct display images, for example, absolute vectors and characters.
Display image	The collection of display elements which are visually represented together in a specified region of the display surface of a display device.
Display surface	The medium upon which a display image is produced, for example, CRT screen, film, or paper.
Factor	A special form of scale in which a single multiplier is applied to both the X and Y axes.
Figure	A collection of display elements which contains its own origin.
Increment	The smallest addressable unit available on a display device.
Inter-character space	The spacing between the characters in a character string.
Mirror	The reflection of display elements with respect to specific straight lines on the display surface.
Mirroring	The process of producing a mirror image of a picture.
Modal parameters	Those values set by user callable UNIPLOT routines that modify the size and placement of the picture.
Neutral picture file	The device independent plot data file generated by the UNIPLOT neutral plot routines containing display elements generated by an application program.
Neutral plot routines	A library of UNIPLOT routines; a subset of which are CalComp call compatible.
Offset	A value subtracted from the x or y coordinate in order to change its position.
Origin	The 0,0 point on the display surface or in virtual space to which all other points are related.
Partitioning	The process of drawing in segments a picture which is too large to fit on a given display surface.
Pen	A drawing instrument mounted in a display device that uses a paper display surface, or those parts of other display devices which perform analogous functions, for example, beam in a CRT.
Pen restore	Moving the pen on a plotter to the bottom edge of the display surface.
Picture	A collection of figures in the neutral file which is preceded by a header record generated by NEWPLT or PLOT. A picture contains one figure if the header record is generated by PLOT.
Plotting device	Refer to display device.
Post processor	The application program, UNIPOST, which reformats the neutral picture file into data acceptable to the display device.

INDEX

COMMENT SHEET

MANUAL TITLE UNIPLOT Version 2.0

Reference/User Guide

PUBLICATION NO. 60454730 REVISION A

FROM: NAME:

BUSINESS ADDRESS:

COMMENTS:

This form is not intended to be used as an order blank. Your evaluation of this manual will be welcomed by Control Data Corporation. Any errors, suggested additions or deletions, or general comments may be made below. Please include page number references and fill in publication revision level as shown by the last entry on the Record of Revision page at the front of the manual. Customer engineers are urged to use the TAR.

NO POSTAGE STAMP NECESSARY IF MAILED IN U. S. A.

FOLD ON DOTTED LINES AND STAPLE