

# SHARE PROGRAM LIBRARY AGENCY



PROGRAM NUMBER

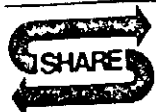
**320001**

---

## University of Miami

1365 MEMORIAL DRIVE - CORAL GABLES, FLORIDA  
(305) - 284-6257

# SHARE PROGRAM LIBRARY SUBMITTAL FORM



SPLA

CONTROL NUMBER: 231

SHARE PROGRAM LIBRARY AGENCY  
Triangle Universities Computation Center  
Post Office Box 12076  
Research Triangle Park, North Carolina USA 27709

This form should be completed and submitted with the program package to the SHARE Program Library Agency at the address shown above. Standards and instructions for submitting programs are in the SHARE Reference Manual, Section 6.

(1) Program Number (to be filled by SPLA) . . . . . 360D-32.0.001

(2) Title of Program . . . . . FORMPLOT-A Forms Design Program

(3) System Type(s) (Machine) . . . . . IBM 360/370

(4) Search Key(s) . . . . . FORMS DESIGN

DATA COLLECTION

PLOTTING

GRAPHICS

PL/I

(5) Programming Systems/Languages . . . . . PL/I

(6) Primary Subject Code . . . . . 32-0

(7) Minimum System Requirements Region of 250K, CALCOMP TYPE PLOTTER

(8) New (N) or Revision (R) (if revision, show prior Program Number in Item 1) N

(9) Date of Submittal . . . . . 12/78

(10) Documentation (number of original pages submitted) . . . . . 33

(11) Author's Name and Address . . . . . DAVID MOFFAT

UNC PROGRAMMING SERVICES

UNC COMPUTATION CENTER

PHILLIPS HALL

CHAPEL HILL NC 27514

(12) Direct Technical Inquiries to Name & Address  
(if different than Author)

(13) Submitter's Installation Membership Code . . . . . UNC

(14) Abstract (should contain sufficient information for a reader to determine the value of the program). Listed on the reverse side of this form are subjects which may serve as a guide for a descriptive abstract.

# SHARE PROGRAM LIBRARY SUBMITTAL FORM

Subject Guide:

- a. Purpose
- b. Programming Language used
- c. Version and modification level or release number
- d. Field of application
- e. Type of routine (main program, subroutine, etc.)
- f. Specific description of machine requirements

SEE ATTACHED SHEETS

(Please attach additional pages if necessary) . . . . . Total pages attached 2

An "Acknowledgement of Assistance" statement must be attached to this Submittal Form.

Permission to Publish

"I hereby give the SHARE Program Library Agency permission to reprint, reproduce, and distribute this program"

(15) Signature of Submitter and Date *[Signature]* 106278

(15) Signature of Installation Addressee *Judith H. Hallman* 12/1/78

Tape Key 360D-32.0.001

Standard label UM3008

File 1: DSN=FORMPLOT.SOURCE,DCB=(RECFM=FB,LRECL=80,BLKSIZE=3520)

File 2: DSN=FORMPLOT.TESTDECK,DCB=(RECFB=FB,LRECL=80,  
BLKSIZE=3440)

## I. ABSTRACT

FORMPLOT is a generalized program for designing keypunch forms, optical character reader forms, and other data collection forms. It can also be used to design notices, ledgers, marksense forms, or to indicate the layout of artwork on forms to be printed. FORMPLOT can produce estimates of the number of forms that can be scanned or keypunched per hour. Forms can be plotted or displayed on the Tektronix 4013 or its equivalent.

## II. SOURCE

FORMPLOT was developed by the UNC Computation Center to be used for designing forms to be scanned by the SCAN-DATA 2250. FORMPLOT greatly reduced the time required to design a new form. It made forms easier to change and it eliminated the necessity of drawing forms by hand.

In the Spring of 1977, UNCCC generalized FORMPLOT. The generalized version allows users to draw keypunch forms and special graphics, and to add labels and text to their forms.

FORMPLOT is written in PL/I. It calls the Calcomp Plotter routines described in LSR-048. In order to implement FORMPLOT at other installations, it would probably be necessary to modify the calls to the plotter subroutines. All calls to the plotter subroutines are isolated in a single section of the FORMPLOT program.

## III. DESCRIPTION

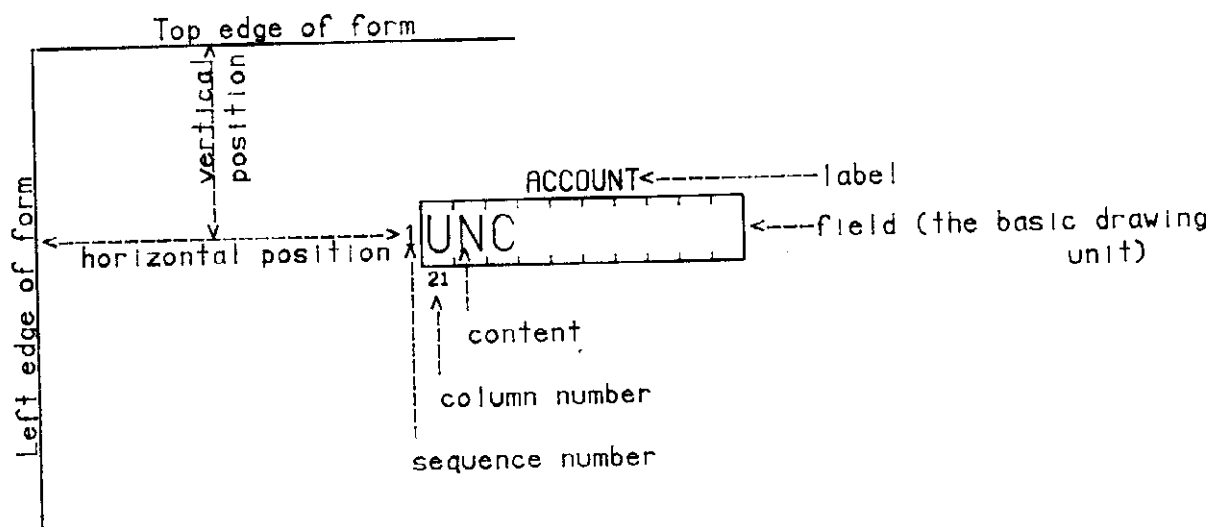
Using FORMPLOT, custom designed forms can be drawn easily and quickly. Several forms may be drawn in a single FORMPLOT job. All forms drawn by FORMPLOT are variations on similar problems. Each form consists of a set of data areas or fields. These data areas are spaces where answers may be filled in on the form. FORMPLOT allows the user to draw standard data areas for keypunch forms or for optical character reader forms, as well as some special graphics. The user can draw sixteen different kinds of fields or data areas using FORMPLOT.

Fields for keypunching and the optical character reader can be labeled and numbered in a variety of ways. Special instructions and text may be included where needed on the form.

Data areas for the optical character reader are designed to be scanned by the SCAN-DATA 2250. FORMPLOT parameters and limitations are designed to make it easy to draw forms that can be scanned with this equipment, although forms may also be drawn to be scanned with other equipment.

Control cards indicate to FORMPLOT the type of field and its location on the form. Additional optional parameters on the control cards indicate labels for fields and whether or not the field should be duplicated in other areas on the form. The user can indicate that a field should be duplicated as many times as necessary from left to right across a form and from top to bottom down a form.

The following form illustrates some of the labeling, numbering, and positioning features available in FORMPLOT.



#### IV. JOB CONTROL

To execute FORMPLOT, use the following Job Control Language:

```
//jobname JOB xxx.yyy.zzz,username,PLOTS=plots
//*PW=password
// EXEC FORMPLOT
//xxxPLOT DD SYSOUT=C
//SYSIN DD *
    FORMPLOT control cards
//
```

where xxx is UNC, DUKE, RTI, NCS, SCOT, or UNCW, depending on where the plotting will be done.

#### V. CONTROL CARDS

There are five types of control cards in FORMPLOT:

- \$FORM statements
- \$DRAW and \$NODRAW statements
- comments
- field descriptions
- STOP, NAME, and END statements

\$FORM statements indicate the beginning of a form description. \$DRAW and \$NODRAW are used to indicate which portions of the form are to be drawn and which are not. Comment cards may appear anywhere in the control cards for documentation. Field descriptions indicate what kind of fields should be drawn and where they are to be drawn on the form. Field descriptions also indicate text and field labels. Field descriptions may be continued on as many cards as necessary. STOP, NAME, and END statements are used to set up user-defined fields. That is, they allow users to construct any kind of field they wish and give it a name. A user-defined field may appear anywhere any other FORMPLOT field is used.

The type of control card is indicated in the first column of the card. \$FORM statements and field descriptions include a number of parameters. These are specified by including keywords, which may be followed by additional options in parentheses. Keywords are separated by one or more blanks or commas. Keywords may appear in any order on the card in either upper or lowercase. Any keyword may be spelled out in full or abbreviated to the first four characters. Some keywords may be abbreviated even further. When shortened abbreviations are acceptable they are indicated in the description. Control cards may contain sequence numbers in columns 73 through 80. If control cards contain sequence numbers, the sequencing must be all numeric. For example, the following control cards might be used to draw two forms:

Column 1

```
$FORM drawing options      (Start a new form)
C      comments            (Comment card)
F001  1st field description (Field parameters)
.
.
.
C                                (Comments and field
                                descriptions)
.
.
.
another field
+nnn  more description      (Continuation card)
.
.
.
$FORM options              (Start another form)
etc.
```

Each type of control card is described below. Variables of the form x.y are real numbers (e.g., 19.13).

#### A. \$FORM Card

The \$FORM card is used to start a new form. If a form is already being described, it also marks the end of that form description. Options on the \$FORM card indicate the size of the form, the parts of the form to be drawn, the data entry calculations, the title of the form, and the mapping information to be printed. The format of the \$FORM card is:

Column 1

```
$FORM    drawing options
or
$F
```

#### Options

The following three options determine which portions of a form will be drawn. By varying the parts of the form that are drawn, several partial drawings of the same form can be produced. This allows separate negatives to be made for different parts of a form to be printed in different colors, which is particularly useful for optical character reader forms. If none of the options below are specified, the field descriptions will be checked for syntax but not drawn. (This is like compiling a program to check for syntax errors but not executing the program.) If both BLUE and BLACK are specified, every field on the form will be drawn.



BLUE - Draw everything but preprinted handprint characters and typed field separators, including labels. A description of preprinted handprint fields is contained on Page 12.

TEXT - Draw labels only. The default is not to draw labels unless BLUE is specified.

BLACK - Draw preprinted handprint fields and typed field separators. The default is not to draw these fields.

The following options determine whether or not various information is printed.

SCANRATE - Calculate and print estimated scanning rates.

KEYRATE - Calculate and print estimated keypunch rates.

MAP - Generate a table showing field type and absolute location of every field drawn on the form.

Sometimes the user may want to magnify or reduce the dimensions of a form or of some of the fields on the form. The following options determine the size and title of the form and the size of some of the fields on the form.

MAG(x.y) - Magnification (scale)  $x.y > 0.0$ . The default magnification is 1.0. The size of the entire form and the size of every field on the form is scaled by the magnification.

SIZE(width,length) - width and length  $> 0.0$ . The default width is 8.5 inches. The default length is 11.0 inches.

TITLE(anything) - up to eight alphabetic or numeric characters. The title will be drawn beneath the form. The default is 'FORM# n' where n is 1 for the first form, 2 for the second, etc.

KEYSIZE(height,width) - width and maximum height of keypunch fields. The default width is 0.167 inch. The default height is 0.333 inch.

#### Examples

\$FORM BLACK BLUE SCAN KEYR  
(Entire 8 1/2 x 11 form drawn actual size, both rates calculated.)

\$F MAG(1.5) SIZE (,14.0) BLUE  
(Length set to 14 inches, only blue parts drawn at 1 1/2 times actual size; thus the form drawn will be 12.75 inches wide and 21 inches long.)

## B. \$DRAW and \$NODRAW Cards

The \$DRAW and \$NODRAW cards are used to start and stop drawing within a form. This is useful when the user is modifying a complicated form and does not wish to plot the entire form until the modification has been perfected. \$DRAW is the default; it results in all fields indicated by the parameters on the \$FORM card being drawn. All fields will continue to be drawn until a \$NODRAW card is encountered. \$NODRAW suppresses drawing fields until the next \$DRAW card. Absolute and relative horizontal and vertical coordinates are determined just as though all fields were to be drawn on the form. The formats for the \$DRAW and \$NODRAW cards are:

Column 1

\$DRAW  
and  
\$NODRAW  
or  
\$NODR

## C. Comment Card

Cards that begin with a 'C' in the first column are comment cards. They may be used to document a form description. The user may include as many comment cards as necessary in a job.

### Examples

```
$FORM BLUE BLAC MAG(2.0) SIZE(8.0,8.0)
C
C      "CALABASH" FORM
C      V1.0 25 JAN 77 D.V.M
C
C ID AREA:
F010 etc.
```

## D. Field Description Card

The field description card contains parameters describing the position, type, labeling, and contents of a "field" or "data area" on the form.

Field description cards begin with an 'F' in Column 1, followed by a field number that may be up to three digits. The format for the field description card is:

Column 1

Fnnn description of field

where  $1 \leq nnn \leq 999$

The field numbers must be in ascending sequence in a form description. Each field description card must have a unique field number.

The field description is specified by the parameters and suboptions described on the following pages.

The user may continue the field description on as many cards as needed. Continuation cards should follow immediately after the field description they continue. A parameter cannot be split across cards. The format for the continuation card is:

Column 1

+nnn more parameters

where "nnn" = the same number as on the "Fnnn"  
card (for your reference)

### Examples

```
F010 LOCATION(1,2) KEY(20) LABEL('NAME')
F20 LOC(2,2) TYPED(20) COLUMN(21)
F30 LOC(1,2) KEYB(20) COL(1)
+30 LABEL('PRINT NAME HERE')
```

The field description is specified by parameters and suboptions of these parameters. There are several types of parameters that may be combined on a field description card to completely describe a field.

- Position of the field on the form.
- Field Type (marksense, keypunch, typed, or user-defined).
- Labeling (labels, column numbers, contents, or sequence numbers).
- Repeats (to draw the same kind of field repeatedly across or down the form).
- Amount of data expected (for rate calculations).

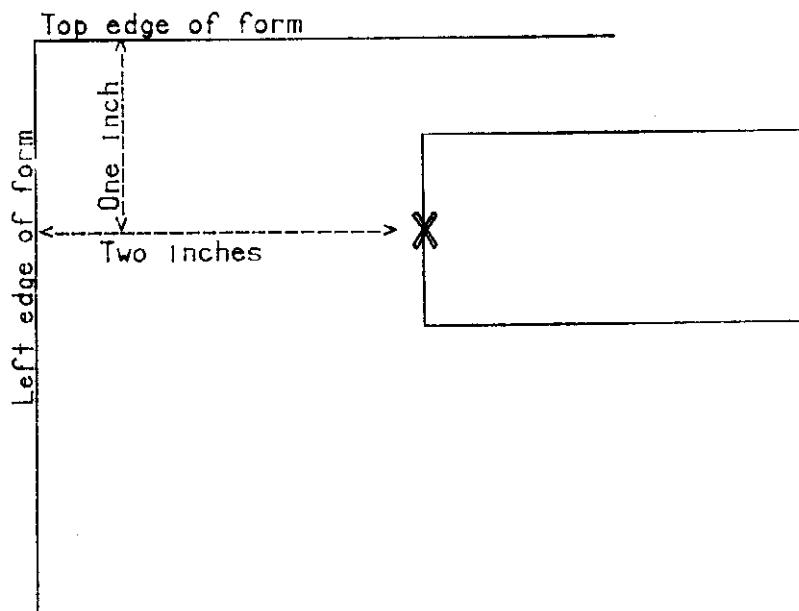
Each of these types of parameters is discussed on the following pages.

Most parameters have suboptions. These are specified in parentheses following the parameter keyword. Suboptions must be specified in the correct order and separated by commas.

#### 1. Position of the Field on the Form

The position of a field is described by indicating where the vertical center of the left edge of the field is located. The user can specify the position of the field by giving its location on an absolute grid, or by giving horizontal and vertical coordinates separately. All measurements are in inches.

If the position is specified by horizontal and vertical coordinates, it may be specified using either relative or absolute coordinates. Absolute coordinates are measured from the upper left corner of the form. If no location is specified, the default location for the field is the last location specified. If no location is specified for the first field described, the location is 0.3 inch from the left edge of the form and 0.467 inch from the top of the form. For example,



is specified as:

```
F01 RECTANGLE(2,1) VERT(1,A) HORI(2,A)
```

RECTANGLE is explained on Page 15.

To specify the position of a field, use one of the following parameters:

- a. LOCATION (horizontal location, vertical location)  
or  
LOC

Absolute location of field in inches.

- b. HORI(x.y,A)  
or  
H

Horizontal location in inches.

where:  $x.y \geq 0.3$  if 'A' is specified.  
'A'=absolute. Distance is from left edge of the form. Default (no, 'A') is relative (i.e., distance is from the value given) unless the last absolute field was a handprint digit. Then distance is calculated as the last absolute value + 0.2 inch. If an absolute location has not been specified earlier in the description, measurements are relative to the left edge of the form + 0.3 inch.

- c. VERT(x.y,A)  
or  
V

Vertical location in inches.

where:  $x.y \geq 0.467$  if 'A' is specified.  
'A'=absolute. Distance is from top edge. If 'A' is not specified, the distance is relative to last vertical given. If an absolute location has not been specified in the form description, measurements are relative to the edge of the form + 0.467 inch.

## 2. Field Types

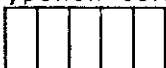
Every field description must include a field type parameter. There are sixteen different field type parameters. It is possible to draw areas for optical character recognition, for keypunched or typed data, and to draw special graphics.

Each field type is described below. Illustrations to the left of the keywords indicate what the area looks like when drawn on a form. In addition to the sixteen different types of fields already available, the user can define his own field type. The user can include a user-defined field type in a field description instead of any of the field types explained below.

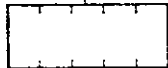
a. Keypunch Areas

There are four different types of keypunch areas. The user can draw keypunch boxes, partitions, or spacers, or have an area for labels on a keypunch form.

keypunch boxes



keypunch partitions



keypunch spacers



The four formats for keypunch areas are:



- (1) KEYB (count, spacing, height)  
or  
KB

This parameter produces keypunch boxes each 0.167 inch wide or as wide as specified in the KEYSIZE option on the \$FORM card.

where: count = number of boxes or data columns.  $1 \leq \text{count} \leq 80$ .  
The default count is 1.  
spacing = unit (box width) spacing between successive boxes in the field.  $1 \leq \text{spacing}$ .  
The default spacing is 1.  
height = S, M, or L - The maximum height is specified in the KEYSIZE option of the \$FORM card. Page 5 contains a description of the KEYSIZE option.

S(small) is 50% of maximum. M(medium) is 75% of maximum. L(large) is 100% of maximum.

If the KEYSIZE is not specified, S is 0.167 inch, M is 0.025 inch, and L is 0.333 inch. If height is not specified, the default height is 100% of the maximum height of the KEYSIZE option.



- (2) KEYP (count, spacing, height)  
or  
KP

This parameter produces keypunch "partitions" each 0.167 inch wide or as wide as specified on the KEYSIZE option. Options are the same as for KEYB.



- (3) KEYS (count, spacing, height)  
or  
KS

This parameter produces keypunch "spacers" each 0.167 inch wide or as wide as specified on the KEYSIZE option. Options are the same as for KEYB.

- (4) KSPACE (count, spacing, height)  
or  
KSP

This parameter is the space definition for extra labels on keypunch forms. Options are the same as for KEYB. This keyword does not result in an area being drawn on the form. It merely delimits an area having the shape and size of a keypunch area. However, it is useful for indicating where labels and text should appear on a keypunch form.

#### b. Typed Areas

Data areas for typed data may appear on a keypunch form or on a form to be read by an optical character reader.

There are three field types for typed data. Typed characters are assumed to be 10 per inch (10 pitch). To draw typed data areas use the following parameters.



- (1) TYPED (count, spacing, height)  
or  
TYP

This provides the outlines for a typed (OCR or punched) field. Unlike keypunch areas, no vertical lines or tick marks separate characters. Unit spacing is 0.1 inch. Count and spacing options are the same as for keypunch areas. However, the height is not affected by the KEYSIZE option of \$FORM.

Height = S - small - 0.167 inch high  
M - medium - 0.250 inch high  
L - large - 0.333 inch high  
The default height is 0.167 inch.

- (2) TSPACE (count, spacing, height)  
or  
TSP

This is the space definition for extra labels on typed forms. Options are the same as for TYPED. This keyword does not result in an area being drawn on the form. However, it is useful for indicating where labels and text should appear on a typed form.

- | (3) SEPARATOR (height)  
or  
SEP

This is the field separator for typed (OCR) forms. The separator is 0.02 inch wide. It occupies a unit space of 0.1 inch. The height option is the same as for TYPED. Field separators are drawn when the user specifies BLACK on the \$FORM card.

#### c. Marksense and Handprint Areas

There are six different types of marksense and handprint fields that can be drawn on forms to be scanned by an optical character reader.



- (1) BOX (count, spacing, outline)

These are the standard OCR handprint boxes for digits. Each box is drawn in an area 0.2 inch wide.



where: count = number of boxes or data columns.  $1 \leq \text{count} \leq 40$ . The default is 1.

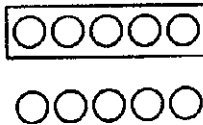


spacing = unit (box width) spacing  
 between successive boxes in  
 the field.  $1 \leq \text{spacing} \leq 5$ .  
 The default is 1. This  
 results in contiguous boxes.  
 The horizontal distance  
 between boxes will be  $0.20$   
 spacing (inches).  
 outline = Y-yes, draw an outline  
 around the group of boxes.  
 N-no, do not draw an  
 outline.



(2) MARK (count, spacing, outline)

This is the standard OCR marksense field for marks. Options are the same as for BOX, except that the default for outline is N. Note that  $1 \leq \text{spacing} \leq 5$  if the marksense field is to be scanned as one unit. Unit spacing is 0.2 inch.



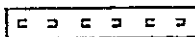
(3) CIRCLE (count, spacing, outline)  
 or  
 CIR

This is standard OCR marksense areas for X's. Options are the same as for MARK. Unit spacing is 0.2 inch.

(4) HSPACE (count, spacing)  
 or  
 HSP

This is the space definition for extra labels on handprint (OCR) forms (with BOX, MARK, and CIRC). Units are 0.2 inch wide.

== (5) HMARK (count, spacing, outline)



These are horizontal marksense areas for the SCANTRON mark reader. This area can also be used for drawing standard marksense areas if the form is to be scanned from side to side instead of from top to bottom. Options are the same as for BOX. Unit spacing is 0.333 inch. Use RECTANGLE instead of PP(1) to draw timing marks for these areas.

1 2345  
67890  
ACTX

(6) PP (what)

This is the standard preprinted (black) OCR handprint character.

where: what = the desired digit.  
0<=digit<=9.  
or  
A, C, T, or X  
No default.

Ordinarily, for marksense areas to be scannable, every line of one or more handprint (BOX, CIRC, MARK) fields begins with a preprinted one (0.025 inch wide by 0.188 inch high ).

Every handprint field is positioned horizontally relative to a preprinted character. A location of HORI(0.0) is the point exactly 0.2 inch to the right of the left-most edge of the preprinted character.

#### d. Special Graphics

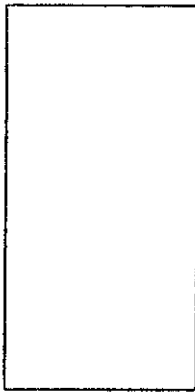
In addition to keypunch, typed, and marksense handprint areas, some special graphics are also available. To draw lines or rectangles, use the following keywords:

(1) LINE (hori, vert, A)

This draws a line anywhere on the form.

where: hori and vert are the ending coordinates of the line. The line begins at the coordinates specified as the position of the field on the form. Field positions are described on Page 8.

If A is given, the end coordinates are absolute (to a given point on the form). Otherwise the coordinates are relative to the starting point. Thus either coordinate may be zero (defaults).  
NOTE: 'LINE' cannot have an associated 'LABEL.'



(2) RECTANGLE (width, height)  
or  
REC

This draws a rectangle of any size anywhere on the form.

where: width and height are the width and height of the rectangle. Note that the figure is located with reference to the midpoint of its left edge. Either dimension may be 0. (Both default to 0, resulting in a single point.)

### 3. Labels Associated with a Field

In addition to drawing the fields on the form, the user may wish to add labels, titles, special instructions, and other text. FORMPLOT includes five different parameters to add text of this sort. Any of these parameters may be included in a field description for keypunch, typed, or marksense and handprint areas. These parameters should not be used with LINE.

Labels are positioned relative to a field. A label can appear in any of five vertical locations relative to the field it labels. Some labels may be aligned three different ways horizontally. For notation, see the location chart below. The default for empty data areas (KSPACE, TSPACE, and HSPACE) is middle, centered. The default for all other areas is top (above), centered.

#### POSITIONS FOR LABEL OF A FIELD

Vert. Location	Hori Location			Vert. Only*
	Left Justified	Centered	Right Justified	
TOP (above)	TL	TC	TR	T
UPPER (inside)	UL	UC	UR	U
MIDDLE (inside)	ML	MC	MR	M
LOWER (inside)	LL	LC	LR	L
BOTTOM (under)	BL	BC	BR	B

\*For SEQ, CON, COL

Locations Relative to a Field

TL	TC	TR
UL	UC	UR
ML	MC	MR
LL	LC	LR
BL	BC	BR

To add text or labeling to a data area use one of the following keywords.



- a. LABEL ('string,' position, size, angle)  
or  
LAB

This draws a label over, in, or under the field.

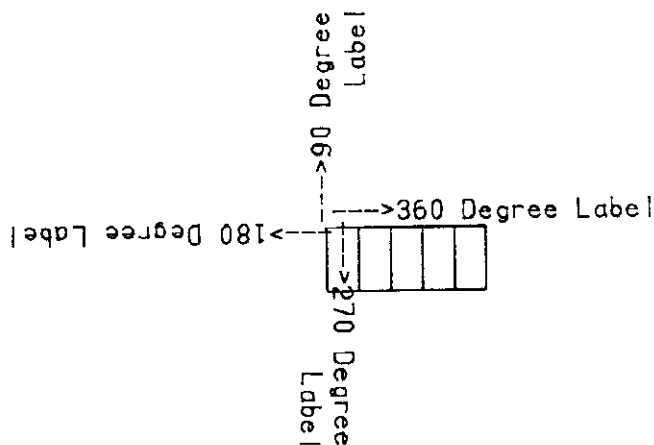
where: 'string' = quoted string of up to 70 characters. (A quote inside the string is represented by two quotes.) No default.

position = the vertical and horizontal position of the label relative to the field. The default position is determined by the type of field being drawn. The default position is TC for KSPACE, TSPACE, and HSPACE. The default is MC for all other data areas.

size = S (Small) - 0.10 inch high letters (Default).  
M (Medium) - 0.183 inch high letters.  
L (Large) - 0.267 inch high letters.  
x.y>010 (desired height).

angle = degrees from the horizontal as shown here.

45 Degree Label



Two lines  
of label



The user should be careful about where he locates a label. Two-part labels can be specified by separating the parts with an underscore (\_). If the first part is drawn at a vertical location of T (top), the second part is drawn at B (bottom). If the first is drawn at U

(upper), the second is drawn at L (lower). If the first is drawn M, L, or B, the second is not drawn. Labels drawn at an angle should be located top-left (TL). Labels on LINES are not well defined. Labels positioned inside some data fields may not be legible because of tick marks or crossbars.

12345

A B C D E

- b. CONTENT (data, vertloc, size, how)  
or  
CON

This specifies letters or numbers drawn in a field, one per unit space, to predefine the contents of the field. If CONTENT is specified with MARK, the two vertical lines that ordinarily are drawn inside the data area are not drawn so that the contents can be read.

where: data = n - numeric sequence starting with n.  $0 \leq n < 99$ .  
= a - alphabetic sequence starting with a.  $A \leq a \leq Z$ .  
= 'string' - any string of characters (up to 32).  
= PP(n) - characters standard handprint digit(s).  $0 \leq n \leq 9$ . Beginning with n. The default is numeric sequencing starting with one.

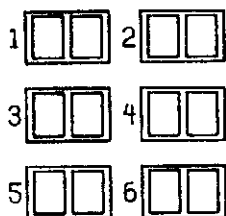
vertloc = vertical location, as shown in the table on Page 15.

size = S - Small. (Default for MARK, CIRC, HMAR and all TYPED.) Actual size same as for LABEL.  
= M - Medium. (Default for BOX, HSPACE, and all keypunch areas.)  
= L - Large.  
= x.y > 0.0 - actual height in inches.

how = 0 - Once. One character was given, or one occurrence of 'string.'

= S - Sequence to end of field (same as 0 for 'string').

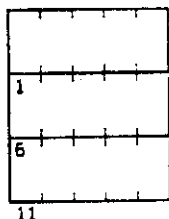
= R - Repeat. Repeat 'string,' or start over with 0 after 99, or start over with 'A' after 'Z,' or start over with PP(0) after PP(9). The default is S for numeric and alphabetic constants. The default is 0 for 'string' and PP(n).



- c. SEQUENCE (data, vertloc, size, how)  
or  
SEQ

This specifies the letters or numbers (up to three digits) drawn to left (outside) of a field, one for each occurrence of the field (for numbering fields drawn repetitively in a loop).

where: data, vertloc, size, how, are defined as in CONTENT. The defaults are the same as for CONTENT.



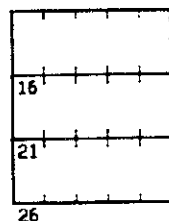
- d. COLUMN (start#, vertloc, spacing)  
or  
COL

This specifies up to three digits to be drawn in the first character position of a field to indicate the punch column of the data. The digits are 0.067 inch high.

where: start# = punch column for the first data character in the field.  $0 \leq \text{start\#} \leq 999$ . Default=0. If positive, the number given is drawn. If 0, the next available column number is drawn (beginning with 1 for the first field on the form).

vertloc = the vertical position of the column number relative to the field. (See table on Page 15.) The default is B (under).

spacing = number of columns to space over to insert another column number within the same (long) field. 0 = first column only.  $0 \leq \text{spacing} \leq 10$ . Default is 10.



Many lines  
of text  
can be  
written at  
one time

- e. TEXT ('lines,' position, size, spacing)

This writes several lines of text. It may be used instead of LABEL.

where: lines = one or more lines of text separated from each other with an underscore(\_). The first line is drawn at the given vertical position within the field; succeeding lines are spaced at regular intervals down the page. 'lines' cannot be continued on another card.

position = the starting position of the first line of text (see table on Page 15.) The horizontal position of succeeding lines will be the same as the first.

size = S - small letters, 0.1 inch high (default).  
M - medium, 0.183 inch high (default).  
L - large, 0.250 inch high (default).  
x.y >0 - Actual height in inches.

spacing = the distance between successive lines in inches. The default is 0.167 inch.

#### 4. Repeats

Repeats are used to replicate a field at regular intervals across a page or to replicate a group of fields at regular intervals down a page.

- a. DOFIELD (count, spacing)  
or  
DOF

This draws a field at regular intervals across the form. DOFIELD will not change the horizontal reference.

where: count = the number of times the field is to be drawn, including the first time.  $1 \leq \text{count}$ . (Use only for  $\text{count} > 1$ .)

spacing = the distance between the horizontal locations of the fields (from the start of one to the start of the next).

- b. REPEAT (field, count, spacing)  
or  
REP

This draws a group of fields repeatedly at regular intervals down a form.

where: field = the field number of the beginning of the group (always prior to, or equal to, this field). If no field is supplied, only the current field will be duplicated.

count = as in DOFIELD

spacing = the vertical spacing between the position of the current field and the next occurrence of the group.

REPEAT alters the vertical location. For example:

F10 CIRC(5) REP(10,3,.333)

moves the vertical location 1 inch down the page.

## 5. Expected Data

CHARACTERS (x.y)  
or  
C

This parameter should be included if the user uses either the KEYRATE or the SCANRATE options on the \$FORM card. It is used to calculate estimated keypunch rates and scanning rates. It has no effect when used with LINE or RECTANGLE.

where: x.y = average number of characters that will be entered into the field.  $0 \leq x.y$ . The default is 0. The keyrate indicates the number of forms that can be keyed per hour assuming 10,000 keystrokes per hour. The scanning rate indicates the number of forms that can be scanned in an hour assuming good quality handprint and typed areas. The scanning rate is calculated for the Scan Data 2250/1 with the 425 reader.

The formula for determining the scanning rate is based on a number of variables:

- V - the length of the form
- L - the number of lines of data + 2
- F - the number of fields on the page
- C - the average number of characters
- H - the total line length for all marksense and handprint areas, excluding preprinted characters
- O - the total line length for all other data areas not included in H

If R is the scanning rate, then

$$R = 3600 / (.04 + ((V+1)/12) + .008L + .0092F + .0004C + .00613 * (H * 5.4 + O * 1.215))$$



## 6. User-Defined Fields

Using the different kinds of fields already described, the user may construct an arbitrarily complex field called a user-defined field. This is similar to writing a subroutine in a program language. The statements necessary to set up a user-defined field are discussed later in this section.

By using the EXEC parameter in a field description, the user can include user-defined fields in a form just as he would include any other field. Thus, the EXEC statement is used like a subroutine CALL in a programming language.

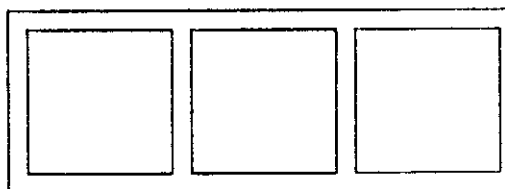
The format for the EXEC parameter is:

EXEC(name,width,height)

where:        name = name of the user-defined field.  
                              It can be one to eight  
                              characters.

width and height = the width and height of the user-  
                              defined area. These are  
                              optional. They are used to  
                              determine where to place labels,  
                              text, column numbers, and  
                              sequence numbers.

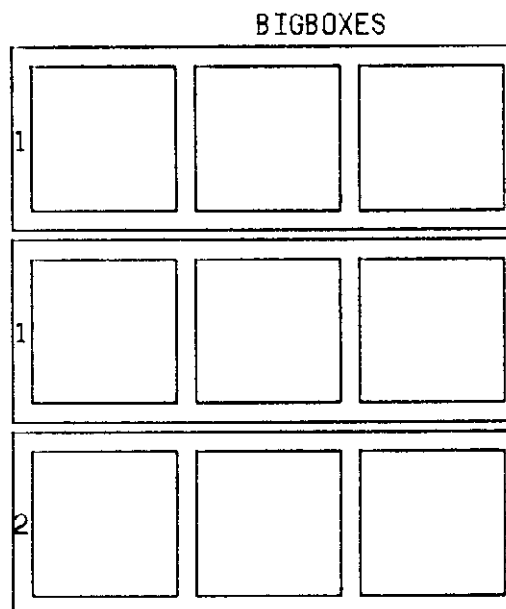
For example, suppose 'BIGBOXES' is the name of a user-defined field that draws an area that looks like



Then

```
F10 EXEC(BIGBOXES,3,1) LAB('BIGBOXES') SEQ  
F20 V(1) EXEC(BIGBOXES,3,1) SEQ REP(20,2,1)
```

produces:



The STOP, NAME, and END cards are used to construct user-defined fields. The NAME statement indicates the beginning of a new definition. The END statement indicates the end of the definition. Statements defining user fields are separated from the form description by a STOP statement. There must be a STOP statement just before the user fields are defined. STOP statements may not appear anywhere else.

The user may include definitions for any number of user fields. Any field type, including an EXEC field, is allowed in a user-defined field. However, the definition of a user field may not occur inside the definition of another user field.

The user will probably want to specify relative locations for all fields within the definition. The reference point for the first location is the position given in the EXEC statement using the definition. For example:

```
F10 LOC(1,1) EXEC(BOXCOL,1.0,.333) LAB('COL.1')
F20 LOC(3,1) EXEC(BOXCOL,1.0.333) LAB('COL.2')
.
.
.
F100 STOP
F900 NAME(BOXCOL)
F910 BOX(5) REPEAT(910,5,.333)
F920 END(BOXCOL)
```

produces:

col. 1	col. 2
<div></div>	<div></div>
<div></div>	<div></div>
<div></div>	<div></div>
<div></div>	<div></div>
<div></div>	<div></div>

The formats for the STOP, END, and NAME statements are given below:

a. STOP

Column 1

Fnn STOP

No other parameters may be included in the STOP statement.

b. NAME

Column 1

Fnn NAME(name)

where: 'name' is one to eight alphabetic characters. It indicates the name of the field being defined. There is no default name. No other parameters are allowed.

c. END

Column 1

Fnnn END(name)

where: 'name' is the name of the field that has just been defined. The name must be included. No other parameters are allowed.

## VI. DEFAULTS AND MINIMAL NOTATION

DEFAULT	MINIMUM NOTATION
LOCATION(*,*)	LOC(*,*)
HORIZONTAL(0.0)	H (or nothing)
VERTICAL(0)	V (or nothing)
KEYB(1,1,L)	KB
KEYP(1,1,L)	KP
KEYS(1,1,L)	KS
KSPACE(1,1,L)	KSP
TYPED(1,1,S)	TYPE
TSPACE(1,1,S)	TSP
SEPARATOR(S)	SEP
BOX(1,1,Y)	BOX
MARK(1,1,N)	MARK
CIRCLE(1,1,Y)	CIRC
HSPACE(1,1)	HSP
HMARK(1,1,N)	HMAR
PP(*)	PP(*)
LINE(0,0)	LINE(draws nothing)
RECTANGLE(0,0)	REC (draws nothing)
LABEL(*,TC,S,0.0) with KB,KS,KP,TYPE, MARK,BOX,CIRC,PP,SEP	LAB(*)
LABEL(*,MC,S,0.0) with KSP,TSP,HSP	LAB(*)
SEQUENCE(1,M,S,S)	SEQ
CONTENT(1,M,S,S) with MARK,CIRC,TYPE, TSP	CON
CONTENT(1,M,M,S) with KB,KS,KP,BOX, KSP,HSP	CON
COLUMN(0,B,10)	COL
TEXT(*,ML,S,.167)	TEXT(*)
DOFIELD(1,0)	DOF
REPEAT(0,1,.333)	REP
CHARACTERS(0.0)	C
EXEC(*,0,0)	EXEC(*)
STOP	STOP
NAME(*)	NAME(*)
END(*)	END(*)

\*These options must always be given, along with both parentheses and any preceding commas.

## VII. SAMPLE FORMS

All of the illustrations in the preceding text were generated by FORMPLOT. The user can run a job to produce each illustration with a label identifying the field by executing the following JCL:

```
//jobname JOB xxx.yyy.zzz,username,PLOTS=10000  
// EXEC FORMPLOT  
//xxxPLOT DD SYSOUT=C  
//SYSIN DD DSN=UNCCC.DATA(FORMPLOT),DISP=SHR
```

Several additional examples of FORMPLOT illustrations are shown on following pages.

# Example 1.

```

$POPM BLACK BLUE SIZE(8.5,14.0) MAG(1.117) TITLE('GP75')

PLD DESCRIPTION-----
C
C          "ANSHEET" FORM
C          30 AUGUST 1977
C
C    PRINT THE TITLE
C
1 F010  V(0.5,A) H(3.7,A) HSPACE LABEL('10 CHOICE',ML,.15)
2 F020  V(0.25) H(3.55,A) HSPACE LABEL('ANSWER SHEET',ML,.15)
C
C    PRINT THE IDENTIFICATION AREA WHICH CONSISTS OF A COLUMN
C    OF (1) A BOX IN WHICH TO WRITE THE NUMBER (NOT SCANNED),
C    (2) A PREPRINTED ONE, AND (3) MARK SENSE SELECTION AREA FOR
C    THE DIGIT PRINTED IN THE BOX.
C
3 F024  V(0.5) H(1.1,A) HSPACE LABEL('IDENTIFICATION',TL,S)
4 F025  V(0.1) H(0.4,A) KSPACE LABEL('PRINT',UL,.07)
5 F026  H(2.2,A) KSPACE LABEL('MARK',UL,.07)
6 F030  V(0.1) H(0.5,A) BOX(1) SEQ(1,UL,.07,S)
7 F040  H(1.3,A) PP(1)
8 F050  MARK(10) CONTENT(0,MC,S,S)
C
9 F060  V(0.333) H(0.5,A) BOX(1) SEQ(2,UL,.07,S)
10 F070  H(1.3,A) PP(1)
11 F080  MARK(10) REPEAT(60,8,0.333) CON(0,MC,S,S)
C
C    ALLOW FOR MULTI-PAGE QUESTIONNAIRES OR EXAMS BY SCANNING
C    A PAGE NUMBERING FIELD
C
12 F090  V(0.667) H(0.5,A) PP(1)
13 F100  MARK(10) LABEL('PAGE #',TC,S) H(0.2) CONTENT(0,MC,S,S)
C
C    PRINT THE 100 QUESTION SELECTION AREAS, THE BODY OF THE FORM
C
14 F110  V(0.333) H(2.9,A) HSPACE LABEL('SELECT ANSWERS BELOW',LL,.15)
15 F120  V(0.333) H(0.5,A) PP(1)
16 F130  H(0.2) MARK(10) SEQ(1,UL,S,S) CONTENT(0,MC,S,S)
+130    DOFIELD(3,2.4) REPEAT(120,25,0.333)
C
C    BACK TO THE TOP TO WRITE THE INSTRUCTIONS
C
17 F140  V(1.25,A) H(5.2,A) HSPACE LABEL('INSTRUCTIONS',TL,S)
18 F150  H(4.27,A) HSPACE TEXT('1. Use a #2 pencil.',UL)
19 F160  H(4.27,A) V(0.15) HSPACE
+160    TEXT('2. Select digits or answers with a single',UL)
20 F170  H(4.55,A) V(0.15) HSPACE
+170    TEXT('vertical stroke.',UL)
21 F180  H(4.27,A) V(0.333) HSPACE TEXT('Example:',UL)
22 F190  H(5.5,A) MARK(10) CONTENT(0,MC,S,S)
23 F200  H(4.27,A) V(0.333) HSPACE
+200    TEXT('3. Mark only in the specified areas.',UL)
24 F220  H(4.27,A) V(0.15) HSPACE
+220    TEXT('4. Erase any errors completely.',UL)
25 F230  H(4.27,A) V(0.15) HSPACE
+230    TEXT('5. Do not make any stray marks or fold or',UL)
26 F240  H(4.55,A) V(0.15) HSPACE TEXT('damage the sheet.',UL)
C
C    PRINT THE UNSCANNED, POSSIBLY KEYPUNCHED INFORMATION AREA
C
27 F250  V(0.4) H(4.0,A) BOX(5) LABEL('SEQUENCE NUMBER',TC,.07)
28 F260  H(4.0,A) HSPACE LABEL('10',BL,.07)
29 F270  V(0.5) H(4.0,A) BOX(20) LABEL('16',BL,.07)
30 F280  V(0.5) H(4.0,A) BOX(20) LABEL('36',BL,.07)
31 F290  V(0.5) H(4.0,A) BOX(20) LABEL('56',BL,.07)

```

THERE ARE 0 ERRORS IN THIS FORM DESCRIPTION.

GP 75  
27

Example 2.

SPORN BLUE BLACK SIZE (8,4)

PLD DESCRIPTION-

c

C

### EXAMPLES OF KEYPUNCH AREAS

c

```

1 P010 LOCA (1.2,1.3)  KEYB (10)
2 P020 LOCA (3.2,1.3)  KEYB (10,,M)
3 P030 LOCA (5.2,1.3)  KEYB (10,,S)
4 P040 LOCA (1.2,2.3)  KEYP (10)
5 P050 LOCA (3.2,2.3)  KEYP (10,,M)
6 P060 LOCA (5.2,2.3)  KEYP (10,,S)
7 P070 LOCA (1.2,3.3)  KEYS (10)
8 P080 LOCA (3.2,3.3)  KEYS (10,,M)
9 P090 LOCA (5.2,3.3)  KEYS (10,,S)

```

THERE ARE 0 ERRORS IN THIS FORM DESCRIPTION.

FORM# 1



Example 3.

FORM BLUE BLACK SIZE (2, 3.5)

FLD DESCRIPTION-

5

C      EXAMPLES OF TYPE AND SEP

C

```

1 F100 VERT(1,A) HOPI(.1) TYPE(10)
2 F110 HOPI(1,A) SEP
3 F120 HOPI(1.8) TYPE(5,,M)
4 F130 HOPI(2.7) SEP(M)
5 F140 HORT(3.1) TYPE(20,,L)
6 F150 HORT(5.5) SEP(L)
7 F160 HOPI(5.9) TYPE(3)
8 F170 TYPE(10) VERT(.5) HOPI(1,A)
9 F180 HOPI(2) TYPE(10,2)
10 F190 VERT(.333) TYPE(10,,M)
11 F200 VERT(.333) TYPE(10,,L)
12 F210 VERT(.667) TYPE(10,,S) LABEL('NAME')

```

THERE ARE 0 ERRORS IN THIS FORM DESCRIPTION.

[illegible]

FORM# 2

Example 4.

SPORN BLUE BLACK SIZE (8, 3)

```

PLD  DESCRIPTION-----
C
C      EXAMPLES OF ALL HANDPRINT AREAS.
C
1  P220 VERT(.5,A)  HORI(.5,A)  PP(1)
C
C      NOTE THAT THE HORIZONTAL LOCATIONS OF THE FOLLOWING HANDPRINT
C      FIELDS ARE ALL RELATIVE TO THE LOCATION OF THE PP(1)
C      PLUS 0.2 INCHES.
C
2  P230 BOX(5)
3  P240 HORI(2)  MARK(5)
C
C      MARK SENSE AREAS FOR SCANTRON EQUIPMENT
C
4  P250 HORI(4,A)  HMARK(5)  VERT(2,A)  REPR(250,2,,2)
C
C      A TYPICAL MARKSENSE VARIATION:
C
5  P260 VERT(2,A)  HORI(.5,A)  MARK(5,2)  CON(A)

THERE ARE      0 ERRORS IN THIS FORM DESCRIPTION.

```

[illegible]

FORM# 3

# Example 5.

\$FORM BLUE BLACK SIZE(8,4)

```

PLD DESCRIPTION-----
C
C   EXAMPLES OF LABELS: LABEL, CONTENT, AND COLUMN
C
1 P280 LOCA(1.1,1.2) KEYB(6) LAB('WHAT?')
2 P290 LOCA(3.1,1.2) KEYB(9) LAB('SOC. SEC. #',TL)
3 P300 LOCA(5.1,1.2) KEYS(11) LAB('SOC. SEC. #',BC)
4300 CON(' - - ',L)
4 P320 LOCA(1.1,2.2) KEYP(11) LAB('SOC. SEC. #') COL(20)
4320 CON(' - - ',M)
5 P330 VERT(1) HCRI(3.5,A) BOX CON(X) LAB('ALRIGHT?',TL,,45)
C
C   EXAMPLES OF TWO-PART LABELS
C
6 P340 LOCA(5.2,2.2) TYPE(20,,L) LAB('TOP PART_BOTTOM',UC)
7 P350 LOCA(5.2,2.8) TYPE(20,,L) LAB('TOP PART_BOTTOM')
8 P360 LOCA(5.2,3.4) TYPE(20,,L) LAB('TOP PART_BOTTOM',MC)

```

THERE ARE 0 ERRORS IN THIS FORM DESCRIPTION.

The diagram illustrates the layout of a form with the following elements:

- WHAT?**: A label above a 6-column box.
- SOC. SEC. #**: A label above a 10-column box.
- SOC. SEC. #**: A label above a 10-column box, with a 20-30 column range indicated below it.
- ALRIGHT?**: A label above a box containing a checked checkbox.
- TOP PART** and **BOTTOM**: Labels above a 10-column box, with the text "TOP PART" and "BOTTOM" appearing in separate boxes.

FORM# 4

Example 6.

SPORM BLUE BLACK SIZE(8,4)

FLD DESCRIPTION-----

C

C     EXAMPLES OF REPEATS, SEQUENCE, AND COLUMN.

C

- 1 F370 VERT(.667) HOPI(4,A) KSPACE TAB('CODING SECTION',,L)
- 2 F380 VERT(.667) HORI(1,A) KEYP(10) DOPI(3,2) SEQ REPE(380,3,.333)  
+380                    COL(0,U)
- 3 F390 VERT(1) HOPI(1,A) MAPK(5) CON SEQ(A) DOPI(4,1.4)

THERE ARE    0 ERRORS IN THIS FORM DESCRIPTION.

CODING SECTION																			
1					2					3									
4					5					6									
7					8					9									
A 1 2 3 4 5					B 1 2 3 4 5					C 1 2 3 4 5					D 1 2 3 4 5				

FORM#    5

Example 7.

\$FORM BLUE BLACK TITLE(BOXES) SIZE(8,5)

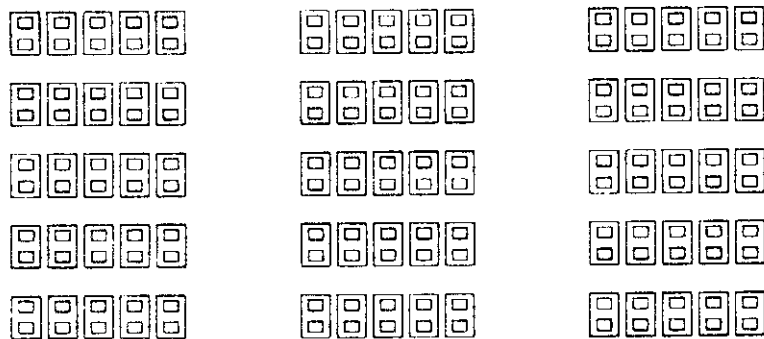
```

PLD DESCRIPTION-----
C
C USER DEFINED FIELDS
C
1 F10 LOC(4.5,1.0) MSP LAB('COMPLEX BOXES')
C COMBOX IS A USER DEFINED FIELD
2 F20 LOC(1.0,2.0) EXEC(COMBOX) DCF(3,2.0) REP(20,5,0.5)
3 F30 STOP
C 'STOP' INDICATES THE END OF THE FORM
C WHAT FOLLOWS ARE THE DEFINITIONS OF THE USER FIELDS
4 F980 NAME(COMBOX)
C YOU CAN REFERENCE USER DEFINED FIELDS IN THE DEFINITION OF
C A NEW USER DEFINED FIELD
5 F981 EXEC(BOXES) DOP(5,0.25)
6 F982 EXEC(UPPER) DOP(5,0.25)
7 F983 EXEC(LOWER) DOP(5,0.25)
8 F984 END(COMBOX)
C YOU CAN DEFINE AS MANY USER DEFINED FIELDS AS YOU WISH
9 F985 NAME(BOXES)
10 F986 REC(0.2,0.3)
11 F987 END(BOXES)
12 F988 NAME(UPPER)
13 F989 HORI(.05) VERT(-.075) REC(0.1,.075)
14 F990 END(UPPER)
15 F991 NAME(LOWER)
16 F992 HORI(.05) VERT(.075) REC(0.1,.075)
17 F993 END(LOWER)

```

THERE ARE 0 ERRORS IN THIS FORM DESCRIPTION.

COMPLEX BOXES



BOXES