# LABELBAR, A PACKAGE OF ROUTINES TO DRAW LABELLED BARS

**David J. Kennison**
**NCAR, P.O. Box 3000, Boulder, Colorado 80307-3000**
**email: kennison@ncar.ucar.edu (128.117.64.4)**

1.0 INTRODUCTION ....................1
2.0 ROUTINES ....................1
2.1 LBLBAR (IHOV, XLEB, XREB, YBEB,YTEB, NBOX, . . . ) ....................1
2.1.1 Usage ....................2
2.1.2 Arguments ....................2
2.2 LBGETI (PNAM,IVAL) and LBGETR (PNAM,RVAL) ....................4
2.2.1 Usage ....................4
2.2.2 Arguments ....................4
2.3 LBSETI (PNAM,IVAL) and LBSETR (PNAM,RVAL) ....................4
2.3.1 Usage ....................4
2.3.2 Arguments ....................4
2.4 LBFILL (IFTP,XCRA,YCRA,NCRA,INDX) ....................5
2.4.1 Usage ....................5
2.4.2 Arguments ....................5
3.0 INTERNAL PARAMETERS ....................6
4.0 ERROR HANDLING ....................6
5.0 EXAMPLES ....................8
6.0 MISCELLANY ....................9

# LABELBAR, A PACKAGE OF ROUTINES TO DRAW LABELLED BARS

**David J. Kennison**
**NCAR, P.O. Box 3000, Boulder, Colorado 80307-3000**
**email: kennison@ncar.ucar.edu (128.117.64.4)**

## 1.0 INTRODUCTION

This document describes a small package of NCAR Graphics routines that allow a user to create a labelled, filled, rectangular bar to serve as a key for a filled plot. The bar may be positioned horizontally or vertically in any desired portion of the plotter frame. Rectangular boxes along the bar are filled with different colors or patterns. Labels are written next to the bar in such a way as to be associated with the boxes or with the lines dividing the boxes from each other.

A label bar is drawn by a single call to a routine called LBLBAR. Exactly what a given call does is determined by the values of the arguments in the call and by the values of six internal parameters, each of which specifies either a color or a line width to be used while drawing some portion of the label bar.

The routines LBGETI and LBGETR allow a user to retrieve the values of these internal parameters and the routines LBSETI and LBSETR allow a user to give new values to the internal parameters.

The routine LBFILL, which, under certain circumstances, is called to fill the boxes, is really just a dummy; the default version will do color fill, but it essentially serves as a placeholder and is intended to be replaced by the user with a version to do some other kind of fill.

All of the routines mentioned above are discussed in detail below, in the section named “ROUTINES”.

## 2.0 ROUTINES

### 2.1 LBLBAR (IHOV, XLEB, XREB, YBEB,YTEB, NBOX, . . . )

(The remaining arguments are WSFB, HSFB, LFIN, IFTP, LLBS, NLBS, and LBAB.)

This is the routine that actually draws a label bar.

### 2.1.1 Usage

The FORTRAN statement

```
 CALL LBLBAR (IHOV,XLEB,XREB,YBEB,YTEB,NBOX,WSFB,HSFB,LFIN,IFTP,
+             LLBS,NLBS,LBAB)
```

causes a complete label bar to be drawn in a manner determined by the values of the arguments and by the values of internal parameters of the package LABELBAR.

### 2.1.2 Arguments

**IHOV** (an input expression of type INTEGER) is zero if the bar is to be oriented horizontally and non-zero if the bar is to be oriented vertically.

**XLEB**, **XREB**, **YBEB**, and **YTEB** (input expressions of type REAL) specify a rectangular area in the plotter frame in which the entire bar, including labels, is to fit. Each is a real number between 0 and 1, inclusive. XLEB specifies the position of the left edge of the area; XREB, the position of the right edge of the area; YBEB, the position of the bottom edge of the area; and YTEB, the position of the top edge of the area. A horizontal bar should probably be made wider than it is high; similarly, a vertical bar should probably be made higher than it is wide.

**NBOX** (an input expression of type INTEGER) may be positive or negative; its absolute value specifies the number of rectangular pieces (let's call them boxes) into which the bar should be divided. All the boxes will be the same size. A horizontal bar will be divided into boxes using vertical lines and a vertical bar will be divided into boxes using horizontal lines. The boxes in a horizontal bar are considered to be ordered from left to right and the boxes in a vertical bar from bottom to top. This ordering determines the order in which fill indices and labels are given. If NBOX is positive, the boxes will be outlined after being drawn; if NBOX is negative, the boxes will not be outlined.

**WSFB** and **HSFB** (input expressions of type REAL) determine what part of each box is to be color-filled or pattern-filled. Each is a number between 0 and 1; they specify the horizontal and vertical dimensions, respectively, of the filled sub-box, as fractions of the horizontal and vertical dimensions of the whole box. The exact position of each sub-box within its box will be determined for you, depending on how you choose to have the bar labelled.

**LFIN** (an input array of type INTEGER) contains a list of indices, one for each box in the bar; each index determines how the filling of the associated box is to be done. These may be color indices or some other kind of indices, depending on the value of the next argument, IFTP.

**IFTP** (an input expression of type INTEGER) specifies how the sub-boxes should be filled, as follows:

- If the value of IFTP is zero, LBLBAR fills each sub-box by calling the routine SFSGFA, in the package SOFTFILL. In this case, the value of the SOFTFILL internal parameter ‘TY’ determines whether filling is done by calls to the GKS routine GFA or by drawing closely-spaced colored lines, which simulates solid fill, or by drawing lines which form patterns of different apparent densities.
- If the value of IFTP is non-zero, the routine LBFILL will be called to do the filling. You may supply your own version of the routine LBFILL.

Prior to the filling of the boxes, the current polyline color and width will be set as specified by the values of the internal parameters 'CFL' and 'WFL', respectively. Therefore, if lines are used to do the fill and the method chosen does not otherwise set polyline color and width, the values of 'CFL' and 'WFL' will take effect.

After filling is complete, the filled sub-boxes may be outlined using the color and line width specified by the LABELBAR internal parameters ‘CBL’ and ‘WBL’. (As mentioned above, this is done only if the argument NBOX is positive; it is suppressed if NBOX is negative.)

**LLBS** (an input array of type CHARACTER) provides a list of labels for the bar; the number of such labels may be equal to

- the number of boxes less one, in which case the labels are associated with the divisions between the boxes, or to
- the number of boxes, in which case the labels are associated with the boxes themselves, or to
- the number of boxes plus one, in which case the first label is associated with the beginning of the bar, the last label with the end of the bar, and the ones in between with the divisions between the boxes.

The labels are drawn by calls to the PLOTCHAR routine PLCHHQ; they are drawn in the color specified by the LABELBAR internal parameter 'CLB' and using the polyline width specified by the LABELBAR internal parameter 'WLB'.

**NLBS** (an input expression of type INTEGER) specifies the number of labels in the array LLBS.

**LBAB** (an input expression of type INTEGER) specifies on which side or sides of the bar the labels are to be written. The value 0 specifies that the bar is to be unlabelled, the value 1 that the labels are to be below a horizontal bar or to the right of a vertical bar, the value 2 that the labels are to be above a horizontal bar or to the left of a vertical bar, and the value 3 that the labels are to be repeated on both sides of the bar. The labels will be scaled in such a way as not to overlap one another or to run outside the area in which the label bar is to lie (except perhaps for labels at the ends of the bar).

## 2.2 LBGETI (PNAM,IVAL) and LBGETR (PNAM,RVAL)

These two subroutines are used to retrieve the current values of internal parameters.

### 2.2.1 Usage

Use the FORTRAN statement

```
CALL LBGETx (PNAM,xVAL)
```

where “x” is an “I” or an “R” (depending on the type of the value involved), to retrieve, in the variable xVAL, the value of the parameter specified by PNAM.

### 2.2.2 Arguments

**PNAM** (an input constant or variable of type CHARACTER) is a string three or more characters in length, beginning with one of the six character strings 'CBL', 'CFL', 'CLB', 'WBL', 'WFL', or 'WLB'.

**xVAL** (an output variable of type INTEGER or REAL, depending on whether “x” is an “I” or an “R”) is the name of a variable into which the value of the parameter specified by PNAM is to be retrieved.

See the section “INTERNAL PARAMETERS” for more information.

## 2.3 LBSETI (PNAM,IVAL) and LBSETR (PNAM,RVAL)

These two subroutines are used to give new values to internal parameters.

### 2.3.1 Usage

Use the FORTRAN statement

```
CALL LBSETx (PNAM,xVAL)
```

where “x” is an “I” or an “R” (depending on the type of the value involved), to set the value of the parameter specified by PNAM to the value specified by the argument xVAL.

### 2.3.2 Arguments

**PNAM** (an input constant or variable of type CHARACTER) is a string three or more characters in length, beginning with one of the six character strings 'CBL', 'CFL', 'CLB', 'WBL', 'WFL', or 'WLB'.

**xVAL** (an input expression of type INTEGER or REAL, depending on whether “x” is an “I” or an “R”) specifies the desired value of the parameter.

See the section "INTERNAL PARAMETERS" for more information.

## 2.4 LBFILL (IFTP,XCRA,YCRA,NCRA,INDX)

### 2.4.1 Usage

This routine is called by LBLBAR when LBLBAR's argument IFTP is non-zero. LBFILL is called once for each sub-box in the label bar, using a statement of the form

```
CALL LBFILL (IFTP,XCRA,YCRA,NCRA,INDX)
```

and is expected to fill the sub-box defined by its arguments. The default version of the routine looks like this:

```
SUBROUTINE LBFILL (IFTP,XCRA,YCRA,NCRA,INDX)
  DIMENSION XCRA(*),YCRA(*)
  CALL GSFACI (INDX)
  CALL GFA (NCRA-1,XCRA,YCRA)
  RETURN
END
```

This version does color fill of the box, using the last argument as a color index. The user may replace this routine with a version that does any desired sort of fill; usually, this can be done by simply compiling the desired version, so that the default one from the package will not be loaded.

### 2.4.2 Arguments

**IFTP** (an input expression of type INTEGER) is just the value of the argument IFTP from the user's call to LBLBAR.

**XCRA** and **YCRA** (input arrays of type REAL) are real arrays containing the X and Y coordinates of five points defining a rectangular box which is to be filled in some manner. The fifth point defined by these arrays will always be a duplicate of the first.

**NCRA** (an input expression of type INTEGER) is the number of points defining the rectangular box. Its value will always be five.

**INDX** (an input expression of type INTEGER) is one of the indices from the array LFIN, in the user's call to LBLBAR. Its value may be used as a color index or as a pattern selector.

# 3.0 INTERNAL PARAMETERS

LABELBAR has six internal parameters, each of which affects the behavior of LBLBAR in a particular way. The current value of one of these parameters may be retrieved by calling either LBGETI or LBGETR and the value of one of them may be reset by calling either LBSETI or LBSETR. The following table lists the internal parameters of LBLBAR:

| Name | Type | Description |
|---|---|---|
| 'CBL' | I | Color of box lines. The default value is -1. |
| 'CFL' | I | Color of fill lines. The default value is -1. |
| 'CLB' | I | Color of labels. The default value is -1. |
| 'WBL' | R | Width of box lines. The default value is 0. |
| 'WFL' | R | Width of fill lines. The default value is 0. |
| 'WLB' | R | Width of label lines. The default value is 0. |

Each of the “color” parameters may be given a negative value to indicate that color is unspecified; in this case, the color of an object drawn will be that specified by the current polyline color index (in the case of box lines and fill lines) or the current text color index (in the case of labels). A value which is zero or greater specifies the color index to be used.

Each of the “width” parameters may be given a value of zero or less to indicate that line width is unspecified; in this case, the line width used to draw an object will be that specified by the current polyline width scale factor. Note that setting polyline width before drawing the labels does affect their appearance; this is because labels are drawn by calls to the routine PLCHHQ, which strokes out characters using lines.

# 4.0 ERROR HANDLING

When a LABELBAR routine detects an error condition, it calls the routine SETER, which is the principal routine in the error-handling package for NCAR Graphics. (There is a programmer document describing SETER and associated routines; see that document for complete information about error handling in NCAR Graphics.)

By default, SETER prints a line and STOPs. The line printed will look something like this:

```
ERROR    3 IN LBGETR - PARAMETER NAME NOT KNOWN - X
```

The error number (“3”, in the example) may be of use to a consultant (to determine exactly where the error occurred), but is not otherwise meaningful. The actual error message consists of the name of the routine in which the error occurred (“LBGETR”, in the example), a blank, a minus sign, another blank, and, lastly, a short description of the error.

All errors are “recoverable” in the sense that, if the user program puts SETER into “recovery mode”, control will be returned to the caller of the LABELBAR routine in which the error occurred. In some cases, it is then possible to take remedial action to get around whatever problem has occurred; in any case, the error flag can be cleared and execution of the user's program can continue.

When SETER is in recovery mode (and, occasionally, even when it is not), error messages may have a somewhat more complicated form, like this:

```
LBGETI/LBGETR - PARAMETER NAME NOT KNOWN - X
```

What this particular error message says is that LBGETI called LBGETR, which detected an error condition (a bad parameter name) and called SETER. Upon getting control back from LBGETR, LBGETI detected the fact that LBGETR had logged an error. It augmented the error message by prepending its own name, followed by a slash, and then passed control back to the user. Of course, there can be more than two such levels of routine calls indicated in the error message: in a few cases, seven or eight routine names may be listed, each separated from the next by a slash.

The various error conditions in LABELBAR are described in the list below. Each bulleted item includes an error message and a thumb-nail description of the error. The items in the list are arranged in alphabetical order. If you get an error message with one or more prefixed subroutine names, as described above, omit them and look up the result in this list. Note that, since LABELBAR routines sometimes call other routines, elsewhere in NCAR Graphics, that can detect error conditions and call SETER, the error message you get by calling a LABELBAR routine may not be listed here, but in the programmer document for some other package.

- **LBGETI - UNCLEARED PRIOR ERROR**

  This error message indicates that, at the time LBGETI was called, there was an unrecovered outstanding error. In this case, LBGETI cannot continue; it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **LBGETR - PARAMETER NAME NOT KNOWN - X**

  The first argument in a call to LBGETR is not the name of one of the internal parameters of LABELBAR. "X" is the offending name.

- **LBGETR - PARAMETER NAME TOO SHORT - X**

  The first argument in a call to LBGETR is too short to be the name of one of the internal parameters of LABELBAR. "X" is the offending name.

- **LBGETR - UNCLEARED PRIOR ERROR**

  This error message indicates that, at the time LBGETR was called, there was an unrecovered outstanding error. In this case, LBGETR cannot continue; it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **LBLBAR - ERROR EXIT FROM GQFACI**

  The GKS routine GQFACI, which is called to get the current value of the fill area color index, has returned a non-zero error code.

- **LBLBAR - ERROR EXIT FROM GQLWSC**

The GKS routine GQLWSC, which is called to get the current value of the line width scale factor, has returned a non-zero error code.

- **LBLBAR - ERROR EXIT FROM GQPLCI**

  The GKS routine GQPLCI, which is called to get the current value of the polyline color index, has returned a non-zero error code.

- **LBLBAR - ERROR EXIT FROM GQTXCI**

  The GKS routine GQTXCI, which is called to get the current value of the text color index, has returned a non-zero error code.

- **LBLBAR - UNCLEARED PRIOR ERROR**

  This error message indicates that, at the time LBLBAR was called, there was an unrecovered outstanding error. In this case, LBLBAR cannot continue; it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **LBSETI - UNCLEARED PRIOR ERROR**

  This error message indicates that, at the time LBSETI was called, there was an unrecovered outstanding error. In this case, LBSETI cannot continue; it forces the error message for the outstanding error to be printed and then substitutes this one for it.

- **LBSETR - PARAMETER NAME NOT KNOWN - X**

  The first argument in a call to LBSETR is not the name of one of the internal parameters of LABELBAR. “X” is the offending name.

- **LBSETR - PARAMETER NAME TOO SHORT - X**

  The first argument in a call to LBSETR is too short to be the name of one of the internal parameters of LABELBAR. “X” is the offending name.

- **LBSETR - UNCLEARED PRIOR ERROR**

  This error message indicates that, at the time LBSETR was called, there was an unrecovered outstanding error. In this case, LBSETR cannot continue; it forces the error message for the outstanding error to be printed and then substitutes this one for it.

# 5.0 EXAMPLES

On a Unix system on which NCAR Graphics has been installed, the command “ncargex” may be used to acquire the code for and run an example of LABELBAR. The name of the example is “elblba”.

# 6.0 MISCELLANY

LABELBAR was written early in 1989 as part of the examples for the package CONPACK and was made into a separate utility in June, 1989. The entire package is written in FORTRAN 77. Entry-point names in the package are LBLBAR, LBFILL, LBGETI, LBGETR, LBSETI, and LBSETR. There is one internal common block, named LBCOMN. The only I/O is the graphic output generated. The SPPS routines GETSET and SET are called. The routine PLCHHQ, in the package PLOTCHAR, is used to plot labels. The routine SFSGFA, in the package SOFTFILL, is used to fill boxes.