GX Progress Windows

Although the GXBAR$ and GXBRU$ routines allow a single progress bar to be displayed within a special Progress Bar Window they do not allow a progress bar to be displayed within a standard application window. This note describes a technique that allows one, **or more**, progress bars to be displayed on a standard application window.

Furthermore, the format of the Progress Bar Window displayed by the combination of the GXBAR$ and GXBRU$ sub-routines is very rigid (e.g. only a small, fixed number of text lines can be displayed above and below the actual bar). The technique described in this document allows complete control over the fields and labels displayed around the "move-able" progress bar.

A Progress Bar is added to a standard application window by coding a special label of the form:

"~P*ndddxxxxx*"

where:

P	Progress Bar control specifier
n	Index value in the range 1 to 9
ddd	Depth (in pixels of the control)
xxxxx	Filler text to set the width of the control

Important Note: The depth specified in pixels thus the progress bar control can be displayed over multiple lines in the application window. Care must be taken to ensure there is sufficient space to allow for the progress bar control.

The range of values (minimum and maximum) that the progress control displays is specified using the GXPRG$ routine (see GXPRG$.DOC). If the GXPRG$ routine is not used the control automatically defaults the minimum and maximum values to 0 and 100, respectively.

The position of the progress bar is updated by updating a PIC 9(9) COMP data field that is defined **at the same position in the window** as the label containing the "~P*ndddxxxxx*" text. For example:

```
03 03 "Units"                       * Description of 1st progress bar
03 15 "~P10257890"                  * Label Specifying 1st progress bar
03 15 Z-D1      9(9) C  NSC         * Data field to update 1st progress bar
*
05 03 "Tens"                        * Description of 2nd progress bar
05 15 "~P20257890"                  * Label Specifying 2nd progress bar
05 15 Z-D2      9(9) C  NSC         * Data field to update 2nd progress bar
*
07 03 "Hundreds"                    * Description of 3rd progress bar
07 15 "~P30257890"                  * Label Specifying 3rd progress bar
07 15 Z-D3      9(9) C  NSC         * Data field to update 3rd progress bar
```

If the value in the associated data field (e.g. Z-D1) when it is displayed (typically by using the SHOW verb) is below the minimum value specified using the GXPRG$ routine, the minimum value will be used. Similarly, if the value in the associated data field (e.g. Z-D1) when it is displayed (typically by using the SHOW verb) is above the maximum value specified using the GXPRG$ routine, the maximum value will be used.

Version Requirements

GX V3.4j
GSM SP-16

See Also

GXPRG$

Example

```
FRAME PROGRS
DATA DIVISION
*
77 Z-C1     PIC 9(9) COMP
77 Z-C2     PIC 9(9) COMP
77 Z-C3     PIC 9(9) COMP
77 Z-COUNT  PIC 9(9) COMP
*
77 Z-WHDL   PIC 9(4) COMP
77 Z-INDX   PIC 9(4) COMP
77 Z-PRM1   PIC 9(9) COMP
77 Z-PRM2   PIC 9(9) COMP
*
77 Z-D1     PIC 9(9) COMP
77 Z-D2     PIC 9(9) COMP
77 Z-D3     PIC 9(9) COMP
*
01    P1
  02  P1VER PIC 9(4) COMP
      VALUE 1
  02  P1ID  PIC 9(2) COMP
  02  P1MIN PIC 9(9) COMP
  02  P1MAX PIC 9(9) COMP
*
WINDOW W1
*
BASE AT 5 5
*
01 02 "Progress control display window" A12
*
03 03 "Units"
03 15 Z-D1  9(9) C      NSC
03 15 "~P10257890"
05 03 "Tens"
05 15 Z-D2  9(9) C      NSC
05 15 "~P20257890"
07 03 "Hundreds"
07 15 Z-D3  9(9) C      NSC
07 15 "~P30257890"
*02 02 W1NULL     X(0)        NUL
*
09 20 "  Break    "           BTN U-7
*
ENDWINDOW
*
WINDOW W2
*
BASE AT 10 10
*
02 02 "Progress control window"     A12
*
04 02 W2TEMP      X(4)              NSC NUL
*
06 06 "  Start    " BTN U50
*
ROUTINES SECTION
*
R-FUNC.
```

```
    IF $FUNC < 50 EXIT
    IF $FUNC > 99 EXIT

    IF $FUNC = 50
       PERFORM CE-START-TEST
       ON EXCEPTION
        CLEAR WINDOW W1
       END
    END

    EXIT

PROCEDURE DIVISION
SECTION AA-MAIN.
*
    ENTER WINDOW W2
    IGNORE EXCEPTION
    EXIT
*
SECTION CA-SET-RANGE
*
      MOVE Z-INDX TO P1ID
      MOVE Z-PRM1 TO P1MIN
      MOVE Z-PRM2 TO P1MAX
      CALL GXPRG$ USING P1          * ASSUME CURRENT WINDOW
      EXIT
*
SECTION CE-START-TEST
*
    DISPLAY WINDOW W1
    IGNORE EXCEPTION

    CALL B$GX-7

    MOVE 0 TO Z-WHDL
    MOVE 1 TO Z-INDX
    MOVE 0 TO Z-PRM1
    MOVE 10 TO Z-PRM2
    PERFORM CA-SET-RANGE
    MOVE 2 TO Z-INDX
    PERFORM CA-SET-RANGE
    MOVE 3 TO Z-INDX
    PERFORM CA-SET-RANGE

    MOVE 0 TO Z-C1
    MOVE Z-C1 TO Z-D1

    MOVE 0 TO Z-C2
    MOVE Z-C2 TO Z-D2

    MOVE 0 TO Z-C3
    MOVE Z-C3 TO Z-D3

    DISPLAY WINDOW W1

    DO FOR Z-COUNT = 1 TO 999
       CALL TEST$
       ON EXCEPTION EXIT WITH 1
       ADD 1 TO Z-C1
       IF Z-C1 = 10
        MOVE 0 TO Z-C1
        ADD 1 TO Z-C2
        IF Z-C2 = 10
           MOVE 0 TO Z-C2
           ADD 1 TO Z-C3
           MOVE Z-C3 TO Z-D3
           SHOW WINDOW W1 FIELD Z-D3
        END
        MOVE Z-C2 TO Z-D2
        SHOW WINDOW W1 FIELD Z-D2
       END
```

```
       MOVE Z-C1 TO Z-D1
       SHOW WINDOW W1 FIELD Z-D1
    ENDDO
    CLEAR WINDOW W1
    EXIT
*
ENDFRAME
ENDSOURCE
```