GXBRU$ - Update Progress Bar on GX

The GXBRU$ routine is used to update a Progress Bar on GX windows. Furthermore, GXBRU$ will return an exception if the optional Cancel button has been pressed.

The GXBAR$ routine must be used to display an initial Progress Bar before GXBRU$ can be used.

1. Invocation

To update a Progress Bar code:

```
CALL GXBRU$ USING up
```

where *up* is a control block of the following format:

```
01    UP
  02  UPVERS      PIC 9(2) COMP     * Block version number
      VALUE 1                       * Must be set to 1
  02  UPPRG PIC 9(9) COMP           * Progress value which
                                    * must be between the
                                    * minimum and maximum
                                    * range specified in the
                                    * the call of GXBAR$
                                    * top open the window
  02  UPTITL      PIC 9(4) COMP     * -1 = do not wish to update title string
                                    * 0 = zero-terminated updated title string
                                    * N =length of updated title string
  02  UPTITP      PIC PTR           * Pointer to updated title string
  02  UPTXT OCCURS 3                * Updated text strings
   03 UPTXTL      PIC 9(4) COMP     * -1 = do not wish to update text
                                    * 0 = zero-terminated text string
                                    * N =length of text string
   03 UPTXTP      PIC PTR           * Pointer to text sting
```

2. STOP Codes and Exception Conditions

The following STOP codes may be generated by GXBRU$:

STOP code	**Description**
14503	GXBRU$ has been called by an application that is not running on GX.
14504	Unknown version of UP control block (i.e. UPVERS does not contain 1)
14505	Incompatible version of GX
14506	The total length of the GX command block has exceeded 8192.

The following exception conditions may be returned by GXBRU$:

EXIT code	$$COND	Description
14501	1	It was not possible to allocate a temporary 32-bit page to hold the GX command block.
14502	2	Cancel button clicked

3. Programming Notes

GXBRU$ is only available when running on GX. Any attempt to use GXBRU$ on a non-GX terminal will result in a STOP code.

Before GXBRU$ is used to *update* a Progress Bar, GXBAR$ **must** be used to display an *initial* Progress Bar.

Important Note 1: In order for the Progress Bar to maintain the focus there must be no accepts in any other window between the initial call of GXBAR$ (to display the Progress Bar) and the final call of GXBAR$ (to remove the Progress Bar).

Important Note 2: GXBRU$ is typically called repeatedly within a processing loop. Although a single GXBRU$ operation is relatively fast the time required to complete multiple GXBRU$ calls can be significant if this routine is called during **every** iteration of a processing loop. You are advised to include logic to avoid calling GXBRU$ every time within a loop. For example:

```
MOVE 0 TO Z-DIVISOR                     * INITIALISE DIVISOR
DO WHILE condition TRUE                 * TOP OF LOOP
      ADD 1 TO Z-DIVISOR                * BUMP DIVISOR
      IF Z-DIVISOR = 100                * TIME FOR ANOTHER UPDATE
            MOVE 0 TO Z-DIVISOR         * RESET COUNTER
            CALL GXBRU$ etc.            * DILUTED GXBRU$ CALLS
      END
      Main processing in loop
ENDDO
```

4. Examples

[EXAMPLE REQUIRED]

5. Copy-Books

See copy-book "u$" in copy-library S.SYS32. Note that this copy-book **MUST** be expanded using a SUBSTITUTING clause. For example:

```
COPY "u$" USING "UP"
```

6. See Also

GXBAR$ - Display initial Progress Bar