B$GTWK - Get Window Control Block

The B$GTWK routine returns a “sanitised” version of the Window Control Block structure.

1. Invocation

To obtain a "sanitized" Window Control Block code:

CALL B$GTWK USING *window wk*

where *window* is the Window Definition of the target window; and *wk* is the "sanitized" Window Control Block defined by copy-book W$ in S.SYS32.

For GSM SP-18, or later, a 3 parameter call is supported:

CALL B$GTWK USING *window wk flag*

where *window* and *wk* are as above; and *flag* is an optional PIC 9(4) COMP field, or literal, that defines the operating mode as described in section 3, below.

2. STOP Codes and Exception Conditions

No STOP codes are generated by B$GTWK.

No exception conditions are returned by B$GTWK.

3. Programming Notes

The B$GTWK has been implemented to allow Window Control information to be obtained in a standard and future-proof manner. Any code that uses questionable or dubious techniques to obtain the Window Control Block information (e.g. using a redefinition of SVC-64) must be changed to use B$GTWK.

Important Note: By default for most versions of this sub-routine, the record number at the top line in the scrolled area (first RDA), WKADD1, is always the record number for the **current** window being processed (i.e. when returning this result field B$GTWK does not use the window, *window*, that is **passed** to the sub-routine). An address of –1 is returned in WKADD1 if the record number of the record (in the current window) is unavailable. All the other fields returned by B$GTWK are for the window, *window*, that is **passed** to the sub-routine.

For GSM SP-18, and later, it is possible to modify the manner in which the B$GTWK routine returns the WKADD1 field by passing the optional *flag* parameter. The following table summarizes the options:

CALL B$GTWK USING *window wk*	WKADD1 is the record number for the **current** window
CALL B$GTWK USING *window wk* 0	WKADD1 is the record number for the **current** window
CALL B$GTWK USING *window wk* 1	WKADD1 is the record number for the **passed** window

The exact handling of the WKADD1 field has gone through a number of revisions:

Revision	Date of BA$MID DLM	Comments

GSM SP-16, and earlier	08/09/2004, and earlier	Always returns WKADD1 as the record number of the first record of the **current** window. May cause ILLEGAL OPERATION CODE AT BA$MID/#0ADE under some circumstances.
GSM SP-17, pre-release	02/12/2004	Always returns WKADD1 as the record number of the first record of the **passed** window. This can result in illegal values returned resulting in GLOBAL.EXE crashes.
GSM SP-17, official release	14/02/2005	Always returns WKADD1 as the record number of the first record of the **current** window but includes additional code to validate the pointers involved. Despite the extra validation, may cause ILLEGAL OPERATION CODE IN BA$MID under some circumstances.
GSM SP-18, and later	16/03/2005, and later	Introduction of optional third parameter to determine whether the record number of the first record refers to the **current** or the **passed** window.

4. Examples

```
FRAME ORDER  "Order Entry and Maintenance" ******************************
**                                                                      **
**************************************************************************
*
ACCESS CU TR OR OL ST
*
DATA DIVISION
*
77    Z-REC       PIC 9(6) COMP
01    WK
COPY W$ SUBSTITUTING "WK"
*
WINDOW W1 USING OR
*
SEL                           * ENABLE ADD,ENQ/SEL & MNT MODES
AUTOPGE                       * FWD EXIT ONLY WITH LOCKED RECORD
BASE AT  3  3
SCROLL 10
02 02 "Order       Number"
02 22 "Order Total"
02 42 "Customer  Number"
02 62 "Required date"
            03 02 ORORDN
            03 22 ORTOTL
            03 42 ORCSNO
            03 62 ORRQDT
ROUTINES SECTION
R-SELECT.
      ENTER WINDOW W2         * ENTER LINE-ITEM WINDOW
      IGNORE EXCEPTION        * IGNORE <BCK>
*
      CLEAR WINDOW W2
      EXIT
ENDWINDOW
*
WINDOW W2 USING OL$SQ DEPENDANT ON OR ************************************
**                                                        **
**    O R D E R   L I N E   I T E M W I N D O W . . .         **
**                                                        **
**************************************************************************
```

```
*
AUTOPGE                         * DISPLAY 1ST PAGE ON ENTRY
SEL                             * LOOP TILL <NXT> KEYED.
REPEAT
BASE AT 8   8
SCROLL      8 BY  1 SPLIT  1 OFFSET  1
02 02 "Stock"
03 10 "Description"
            05 02 OLSTNO
03 02 "Number"
            05 10 OLDESC
02 33 "Date"
03 33 "Reqd"
            05 31 OLDTRQ
02 45 "Unit"
03 44 "Price"
            05 41 OLPRCE
02 51 "Order"
03 52 "Qty"
            05 51 OLORQT
02 62 "Line"
03 61 "Amount"
            05 58 OLLAMT
*
ROUTINES SECTION
R-FUNC.
      IF $FUNC = 11                 * ON EXIT
          CALL B$GTWK USING W1 WK 0
          MOVE WKADD1 TO Z-REC
          GET OL KEY      Z-REC     * GET THE FIRST LINE IN W2
          MOVE WKADDR TO Z-REC      * GET CURRENT RECORD IN W1
          GET OR KEY      Z-REC
      END
      EXIT
ENDWINDOW
PROCEDURE DIVISION ************************************************
**                                                  **
**    ALL THE WINDOWS ARE CONTROLLED HERE!                    **
**                                                  **
**********************************************************************
*
      CALL B$OPN USING "DEMON" "FLS" 0
AA-000. ENTER WINDOW W1             * ENTER ORDER HEADER WINDOW
      ON EXCEPTION EXIT WITH 1      * <BCK> EXIT..
*
      GOTO AA-000                   * THE ORDER TOTAL & RE-ENTER.
*
ENDFRAME
ENDSOURCE
```

5. Copy-Books

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

```
      COPY "W$" USING "WK"
```

6. See Also

None.