B$BAD - Create new Global format Speedbase Database

The B$BAD routine is used to create a new Global format Speedbase database.

This routine provides the same functionality as the Create New Database functionality of $BADGN.

1. Invocation

To create a new Global format Speedbase database code:

CALL B$BAD USING *ba* [*bs*]

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

```
01    BA
  02  BADICT      PIC X(5)                * Data dictionary name
  02  BADIUN      PIC X(3)                * Data dictionary unit
  02  BATITL      PIC X(30)               * Database title
  02  BARECS OCCURS 36 PIC 9(9) COMP      * Number of records
                                          * (SPACES or invalid default to 50)
  02  BAFILN OCCURS 36 PIC 9 COMP         * Database file number
                                          * (i.e. 1, 2 or 3)
                                          * (0 or invalid default to 1)
  02  BAINDX OCCURS 36 PIC X(6)           * Index names returned by B$BAD
  02  BANAM0      PIC X(5)                * Index file name
                                          * (SPACES default BADICT)
  02  BASIZ0      PIC 9(2) COMP           * %Size of index file
                                          * 0 = use BASIZ1
                                          * N = %age increase on default size
                                          * -1 = BASIZ1 multiplication factor
                                          * on default size
  02  BASIZ1      PIC 9(9) COMP           * Size of index file
                                          * (0 = use default)
  02  BAUNI0      PIC X(3)                * Index file unit
                                          * (SPACES default BADIUN)
  02  BAUNIS OCCURS 3 PIC X(3)            * Units for database files 1,2 or 3
                                          * (SPACES = same unit as index file)
```

and *bs* is an **optional** control block. For GSM SP-17 and GSM SP-18 only the following (version 1) format BS block is supported:

```
01    BS
  02  BSVERS      PIC 9(4) COMP           * Block version number
                                          * must be set to 1 for version 1 blk
  02  BSFREE OCCURS 36 PIC 9(2) COMP      * %age minimum free space required
                                          *(0 = don't check this record set)
```

For GSM SP-19, and later, both the version 1 BS block (above) and the version 2 BS block are supported:

```
01    BS
  02  BSVERS      PIC 9(4) COMP           * Block version number
                                          * must be set to 2 for version 2 blk
  02  BSFREE OCCURS 36 PIC 9(2) COMP      * %age minimum free space required
                                          *(0 = don't check this record set)
  02  BSFFLG      PIC 9 COMP              * 1 = BSFREE table is valid
                                          * 0 = Ignore BSFREE table
  02  BSFPRG      PIC 9 COMP              * 1 = Progress Message Rtn ptr set
                                          * 0 = No progress Message Rtn ptr
  02  BSPRTN      PIC PTR                 * Pointer to the Progress Message
                                          * Routine
```

The version 2 BS block has been designed to allow all combinations of a BSFREE table and a Progress Message Display Routine as explained in this table:

BSFFLG value	BSFPRG value	Comments
0	0	BSFREE table is ignored; no Progress Message Display Routine. This combination of BSFFLG & BSFPRG is pointless as the optional *bs* block may as well be omitted.
1	0	BSFREE table is recognised; no Progress Message Display Routine. This combination of BSFFLG & BSFPRG is slightly pointless as a version 1 *bs* block may as well be passed.
0	1	BSFREE table is ignored; Progress Message Display Routine pointer **must** be valid.
1	1	BSFREE table is recognised; Progress Message Display Routine pointer **must** be valid.

2. STOP Codes and Exception Conditions

The following STOP codes may be generated by B$BAD:

STOP code	Description
25262	The version number in the BS block is invalid (i.e. not 1 or 2)

The following EXIT codes may be returned by B$BAD:

EXIT code	$$COND	Description
25201	1	Error opening dictionary file
25202	2	Error opening index file
25203	3	Error opening data files
25204	4	I/O error on dictionary file
25205	5	I/O error on index file
25206	6	I/O error on data files
25207	7	Invalid file type

25217	17	Cannot create DBxxxxx9 file

3. Programming Notes

The database file created by B$BAD is merely a template file. **A call to B$BAD must be immediately followed by a call to B$RBG to complete the database creation process**.

The optional *bs* parameter is only recognised by GSM SP-17 or later. For GSM SP-17 and GSM SP-18 only the version 1 BS block is supported. For GSM SP-19, and later, both version 1 and version 2 of this control block are supported.

If a version 1 *bs* parameter block is passed then it will be assumed that the free space table, BSFREE, is valid and a DB*xxxxx*9 file will be created from the BSFREE table. B$ST2N routine can be used to check the free space available in a record set.

For the version 2 *bs* parameter block you **must specifically set the BSFFLG to1** if you want to set a free space table i.e. the contents of BSFREE will be ignored unless BSFFLG is set to 1.

If the BSFPRG flag is set then BFPRTN **MUST** point to a "Progress Message Display Routine" in the application. This Progress Message Display Routine will be called by B$BAD to display any progress messages. The Progress Message Display Routine entry-point should be coded as follows:

ENTRY *routine* USING *pr*

where *pr* is defined as follows:

```
01     PR
  02   PRMODE       PIC 9(4) COMP          * Calling mode (should always be
                                           * returned with a value of 1)
  02   PRFILE       PIC 9(2) COMP          * Data file number of the data file
                                           * being initialised
  02   PRCOUNT      PIC 9(2) COMP          * Progress count (0-10)
```

The progress count provides an indication of the progress of the potentially time consuming data file initialised phase executed by B$BAD.

At the very least the Progress Message Display Routine will be called once with a count number of 0 (start) and again with account number of 10 (finish). Whether the Progress Message Display Routine is called for count numbers between 1 and 9 will depend on the size of the file being created. For small data files some count numbers may be skipped.

4. Examples

[EXAMPLE REQUIRED]

5. Copy-Books

None.

6. See Also

B$RBG	Database Rebuild routine
B$RBL	Database Re-Index routine

B$BCN	Convert Global format Speedbase database
B$ST2N	Check free space in Global format Speedbase database record set