

B\$RBL - Database Re-Index Routine

The B\$RBL allows the indexes of a single or all record types to be rebuilt. The rebuilding process is functionally identical to a partial database rebuild as performed by the database rebuild utility \$BARBL. This system routine may be run from time to time to re-organise indexes following heavy processing such as mass deletions or insertions in order to optimise performance. The routine has the advantage that only a selected record type need be re-built, thus potentially saving unnecessary re-building time of other record types.

This routine provides the same functionality as the \$BARBL utility.

1. Invocation

To re-index a Global format Speedbase database code:

```
CALL B$RBL USING dbid rcid opr [col] [row]
```

where:

- dbid* is the PIC X(5) Database ID of the database to be rebuilt, which must be open with exclusive access prior to the call;
- rcid* is the PIC X(2) ID of the record type to be rebuilt. If this parameter contains spaces then the indexes of all record types will be rebuilt;
- opr* is the PIC X operator response flag, which if set to "Y" will request a pause before continuing. Otherwise this flag should be set to "N";
- col* is an optional PIC 9(2) COMP variable containing the column number at which the status window is to be displayed. The column number must be in the range 1 to 34 inclusive. If this parameter is not provided, the default is column 17. Note that this parameter, if coded, **MUST** be coded as a variable. A numeric literal must not be coded;
- row* is an optional PIC 9(2) COMP variable containing the row number at which the status window is to be displayed. The row number must be in the range 1 to 18 inclusive. If this parameter is not provided, the default is row 18. Note that this parameter, if coded, **MUST** be coded as a variable. A numeric literal must not be coded.

Following the call, the status window will be displayed and the indexes of the designated record type(s) are rebuilt. On successful completion control is returned without exception.

2. STOP Codes and Exception Conditions

No STOP codes are generated by B\$RBL although the routine may terminate the application if a fatal error is detected during the rebuild.

The following EXIT codes may be returned by B\$RBL:

EXIT code	\$\$COND	Description

1	1	The specified database is not open
2	2	Specified database is opened non-exclusively
3	3	Database Dictionary not found or in use
4	4	Database Dictionary file of invalid file type
5	5	Database Dictionary file is corrupt
6	6	Database and Dictionary generations are different
7	7	Database is corrupt and cannot be rebuilt
8	8	The supplied Record -id is invalid
9	9	The supplied Record-id has no indexes
11	11	Insufficient memory to start re-indexation
12	12	Key Errors during re-index; database is corrupt
13	13	Unix C-ISAM channel cannot be opened
14	14	PSQL (Btrieve) or MSQL database cannot be rebuilt (use B\$BNR)

3. Programming Notes

The rebuilding process is memory intensive, and requires quite substantial amounts of memory to perform quickly.

Any serious error detected during the re-building process will cause the routine to terminate with a stop code. Note that this will leave the database in a partially rebuilt state, and a full re-build should be performed before further processing using the database rebuild utility \$BARBL.

Exception 14 is only supported by GSM SP-30, or later.

4. Examples

[EXAMPLE REQUIRED]

5. Copy-Books

None.

6. See Also

B\$BAD	Create new Global format Speedbase database
B\$RBG	Database Rebuild routine
B\$RBX	Database Re-Index routine
B\$BCN	Convert Global format Speedbase database