

# B\$BNR - Rebuild non-DBX SQL Format Speedbase Database

The B\$BNR routine can be used to rebuild a non-DBX Pervasive SQL or Microsoft SQL format Speedbase database. This routine provides the same functionality as the "Rebuild database" option of \$BN32, \$BS32, \$BADN and \$BADS.

## 1. Invocation

To rebuild a non-DBX Pervasive SQL or Microsoft SQL format Speedbase database code:

```
CALL B$BNR USING rb
```

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

```
01  RB
02  RBVERS      PIC 9(4) COMP      * Rebuild block version number
                                VALUE 1      * Must be set to 1
02  RBSCHE      PIC X(5)          * Schema file
02  RBSUID      PIC X(3)          * and unit-id
02  RBPROG      PIC 9 COMP        * 0 = No Progress Message Display required
                                * 1 = Progress Message Display required
02  RBPRTN      PIC PTR           * Pointer to the Progress Message Display
                                * routine
                                * This pointer must be initialised
                                * if RBPROG = 1
```

## 2. STOP Codes and Exception Conditions

The following STOP codes may be generated by B\$BNR:

STOP code	Description
25258	Invalid RB block version number
25259	Incompatible GSM configuration

The following exceptions may be returned by B\$BNR:

EXIT code	\$\$COND	Description
25279	79	Database already open
25280	80	Database in use
25281	81	I/O error on database open

25282	82	Unexpected error on database on open
25283	83	Could not open dictionary
25284	84	Dictionary of wrong type
25285	85	Corrupt dictionary
25286	86	Insufficient memory to allocate work data
25287	87	Server interface error (error code returned in \$\$CRES)
25288	88	Error opening/creating database (error code returned in \$\$CRES)
25289	89	Unable to retrieve master record
25290	90	GVA overflow
25291	91	Duplicate primary key
25292	92	Error on close (error code returned in \$\$CRES)

### 3. Programming Notes

B\$BNR can only be used to rebuild a non-DBX database. Use B\$DBXR to rebuild a DBX database.

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

```
ENTRY routine USING ms dt
```

where *ms* is defined as follows:

```
77 MSNO PIC 9(4) COMP * Message Number
```

and *dt* is defined as follows:

```

01      DT
02      DTRCNO      PIC 9(4) COMP      * Record number
02      DTRCID      PIC X(2)          * Record id
02      DTRNAM      PIC X(6)          * Record name
02      DTBLOC      PIC 9(6) COMP      * Block number
02      DTUSED      PIC 9(6) COMP      * Total blocks
    
```

**Important Note:** The *dt* parameter is only valid for **some** Message Numbers (see below).

The following Message Numbers are defined:

Message Number	Meaning
1	Looking for server (the <i>dt</i> parameter is <b>not</b> valid for this Message Number)
2	Creating new database and/or tables (the <i>dt</i> parameter is <b>not</b> valid for this Message Number)
3	Processing record number <i>dtrcno</i> record id/name <i>dtrcid/dtrnam</i> block number <i>dtbloc</i> of <i>dtused</i> (the <i>dt</i> parameter <b>is</b> valid for this Message Number)
4	Closing database (the <i>dt</i> parameter is <b>not</b> valid for this Message Number)

For GSM SP-17, and later, the code that handles the RBPRTN pointer is more "defensive". This pointer can be set to HIGH-VALUES to indicate that a Progress Message Display Routine is **not** required.

RBPROG	RBPRTN	Comments
0	Not HIGH-VALUES	No Progress Message Display Routine required
0	HIGH_VALUES	No Progress Message Display Routine required
1	Not HIGH-VALUES	Progress Message Display Routine required
1	HIGH-VALUES	No Progress Message Display Routine required (prior to GSM SP-17 this combination will produce a NO BASE exception)

## 4. Examples

[EXAMPLES REQUIRED]

## 5. Copy-Books

See copy-books "\$4" and "\$5" in copy-library S.SYS32. Note that these copy-books **MUST** be expanded using a SUBSTITUTING clause. For example:

```

COPY "$4" SUBSTITUTING "RB"
COPY "$5" SUBSTITUTING "DT"
    
```

## 6. See Also

```

B$DBXR      Rebuild DBX database
B$BNC       Create new non-DBX database
B$BNN       Convert non-DBX database
B$BND       Delete non-DBX database
    
```

B\$BNP      Change path of non-DBX database