B$DBXN - Convert DBX Database

The B$DBXN routine can be used to convert an existing Pervasive SQL or Microsoft SQL format DBX Speedbase database to a new Generation Number. This routine provides the same functionality as the "Convert to new database" option of $DXU.

1.   Invocation

To convert an existing Pervasive SQL or Microsoft SQL format DBX Speedbase database to a new Generation Number code:

```
CALL B$DBXN USING cn
```

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

```
01    CN
  02  CNVERS     PIC 9(4) COMP    * Block version number
                 VALUE 0          * This must be set to 0 or 1
  02  CNSCHE     PIC X(5)         * Source schema file
  02  CNSUID     PIC X(3)         * and unit
  02  CNDSCH     PIC X(5)         * Destination schema file
  02  CNDUID     PIC X(3)         * and unit
  02  CNPATH     PIC X(60)        * Destination directory path
  02  CNDICT     PIC X(5)         * Conversion dictionary
  02  CNCUID     PIC X(3)         * and unit-id
  02  CNSIZE     PIC 9(6) COMP    * If this is a Microsoft SQL
                                  * database which is being
                                  * created, you may specify
                                  * an initial file size in
                                  * Mbytes; otherwise this field
                                  * must be set to 0
  02  CNNAME     PIC X(23)        * If this is a Microsoft SQL
                                  * database then must set the
                                  * database name
  02  CNPROG     PIC 9 COMP       * 0 = No Progress Message Display required
                                  * 1 = Progress Message Display required
  02  CNPRTN     PIC PTR          * Message Display routine
                                  * which must be set if CNPROG is set
```

2.   STOP Codes and Exception Conditions

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

STOP code	Description
25417	Invalid CN block version number
25418	Incompatible GSM configuration
25419	No database name

The following exception conditions may be returned by B$DBXN:

EXIT code	$$COND	Description
25444	44	Unable to access server ($$CRES contains condition)
25448	48	Unable to open source dictionary exclusively
25449	49	Source dictionary of invalid type
25450	50	I/O error on source dictionary (the dictionary may be corrupt)
25451	51	Unable to open source schema file exclusively
25452	52	I/O error on source schema file (the schema file may be corrupt)
25453	53	Source schema file is of invalid type
25454	54	Cannot access server (error code returned in $$CRES)
25455	55	Interface error on open (error code returned in $$CRES)
25456	56	Error on open (error code returned in $$CRES)
25457	57	Destination schema file already exists
25458	58	Error deleting old temporary file
25459	59	Unable to rename old schema file
25460	60	Database already exists
25461	61	Cannot open conversion dictionary exclusively

25462	62	The conversion dictionary was of an invalid type
25463	63	I/O error on conversion dictionary
25464	64	Unable to open destination dictionary exclusively
25465	65	Unable to create destination dictionary
25466	66	Unable to copy dictionaries
25467	67	Unable to open new destination schema file
25468	68	I/O error on destination schema file
25469	69	Cannot create BDCF file (error code returned in $$CRES)
25470	70	I/O error on BDCF file (error code returned in $$CRES)
25471	71	Gateway error in conversion (error code returned in $$CRES)
25472	72	Error writing record (error code returned in $$CRES)
25473	73	Interface error on close (error code returned in $$CRES)
25474	74	Error on close (error code returned in $$CRES)
25475	75	No index on record set

3.   Programming Notes

If the source and destination schema files are the same then the schema file will be overwritten. Otherwise if the destination schema file already exists then an "Already Exists" error will be returned.

If CNPATH is spaces (i.e. no output directory is supplied) the current directory from the existing Schema file will be used.

If the CNPROG flag is set then CNPRTN **MUST** point to a "Message Display Routine" in the application. This Message Display Routine will be called by B$DBXN to display any progress messages. The Message Display Routine entry-point must 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
```

The *dt* bock is only valid for message number 3, 10, 11, 12, 13, 14.

Important Note-1: The *dt* parameter is only valid for **some** Message Numbers (see below).

Important Note-2: The Message Display Routine will be called for **every** message with message numbers of 10, 11, 12, 13 or 14. Such message numbers generally indicate that an exception condition has occurred during the conversion. For some database conversions these messages may be called very frequently (e.g. for every record undergoing conversion). Consequently, you are recommended to avoid executing a DISPLAY statement for every progress message as the numerous DISPLAY statements may significantly increase the time of the conversion process.

Progress messages with a message number of 3 are **always** expected to be called very frequently. For these messages B$DBXN only calls the Message Display Routine when the *dtbloc* value changes.

The following Message Numbers are defined:

Message Number	Meaning
1	Creating new database (the *dt* parameter is **not** valid for this Message Number)
2	Database successfully created - will now load it (the *dt* parameter is **not** valid for this Message Number)
3	Processing file number *dtrcno* record id/name *dtrcid/dtrnam* block number *dtbloc* of *dtused* (the *dt* parameter **is** valid for this Message Number)
4	Closing Pervasive SQL input files (the *dt* parameter is **not** valid for this Message Number)
5	Closing old Microsoft SQL database (the *dt* parameter is **not** valid for this Message Number)
6	Closing Pervasive SQL output files (the *dt* parameter is **not** valid for this Message Number)
7	Closing new Microsoft SQL database (the *dt* parameter is **not** valid for this Message Number)

8	Generating schema file (the *dt* parameter is **not** valid for this Message Number)
9	Generating BDCF file (the *dt* parameter is **not** valid for this Message Number)
10	Invalid conversion for record type *dtrcid* (the *dt* parameter **is** valid for this Message Number)
11	Numeric overflow on record type *dtrcid* (the *dt* parameter **is** valid for this Message Number)
12	Master record not found on record type *dtrcid* (the *dt* parameter **is** valid for this Message Number)
13	GVA overflow on record type *dtrcid* (the *dt* parameter **is** valid for this Message Number)
14	Duplicate primary key on record type *dtrcid* (the *dt* parameter **is** valid for this Message Number)

IMPORTANT NOTE: The Message Numbers for B$DBXN, as listed above, have different meanings from the similar Message Numbers generated by the related B$BNN routine.

Important Note: This routine was revised for GSM SP-17. The following changes ensure the DT block for the B$DBXN routine is compatible with the DT block for the B$BNN routine:

- The allow CNVERS field can be 1 as well as 0. With this change the CN block for the B$DBXN routine is compatible with the CN block for the B$BNN routine.

- The format of the DT-block has been **extensively revised** as follows:

  DTRCNO has been changed from PIC 9(4) to PIC 9(4) COMP
  DTBLOC has been changed from PIC 9(6) to PIC 9(6) COMP
  DTUSED has been added

4.   Examples
None.

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

```
COPY "n$" SUBSTITUTING "CN"
```

6.   See Also
B$DBXC   Create new DBX database
B$DBXR   Rebuild DBX database
B$DBXD   Delete DBX database
B$DBXP   Change path to DBX database
B$DBXL   Load pre DBX database to new DBX database

Note that the $DXU "Import from C-ISAM dumpset" function is **not** available as a sub-routine.