

DBXBEGIN**INTRINSIC NUMBER 420**

Designates the beginning of a sequence of TurboIMAGE/XL procedure calls that are to be regarded as a dynamic transaction of a single database or dynamic transaction spanning multiple databases (DMDBX) for the purposes of logging and dynamic roll-back recovery. The *text* parameter can be used to log user information to the log file. DBXBEGIN is used in conjunction with DBXEND to begin and end a dynamic transaction.

OPENTURBO vs TurboIMAGE Difference

100%

OPENTURBO Performance Enhancements

N/A.

OPENTURBO Additional Features

2PC Support A.02.00

Syntax

```
DBXBEGIN, { base
           baseidlist } , te0xt,mode,status, textlen
```

Parameters

base is the name of the array used as the base parameter when opening the database. The first element of the array must contain the base ID returned by DBOPEN. (Refer to DBOPEN for more information about the base ID.)

baseidlist is the name of the integer array containing the base IDs of the databases which are involved in a DMDBX. Use **baseidlist** when calling DBXBEGIN mode 3 (DMDBX). The layout of this array is shown here (each element is a half word or two bytes):

Element	Contents
1-2	Application program must set these two half words to binary 0s before calling DBXBEGIN. After returning to the calling program, these two halfwords contain the transaction ID. Use this same baseidlist with the corresponding DBXEND or DBXUNDO intrinsics.
3	Number of base IDs involved in the DMDBX. This must be a number between 1 and 15 inclusive.
4- n	Base IDs of the databases involved in the DMDBX. Each base ID occupies one half-word or 2 bytes and it is the first halfword of the base parameter used to call TurboIMAGE/XL intrinsics.

`text` is the name of an array up to 256 halfwords long that contains user ASCII or binary data to be written to the log file as part of the DBXBEGIN log record. The `text` argument is used to assign each particular transaction a distinct name. (Refer to “Discussion” below for more information.)

`mode` `s` is an integer indicating the transaction type:

Mode 1: Indicates a dynamic transaction which spans only one database.

Mode 3: Indicates a dynamic transaction spanning multiple databases (DMDBX). If user logging is enabled for the databases, mode 3 generates *multiple* entries in the log file, one for each database.

`status` is the name of an array of 10 halfwords in which TurboIMAGE/XL returns status information. If the procedure executes successfully, the status array contents are:

Element	Contents
1	If the procedure succeeds, the return status is 0. Table 5-24. describes the contents of element 1 when the procedure does not succeed.
2-4	Unchanged from previous procedure call using this array.
5-10	Procedure call information. Refer to “Library Procedure sError Messages” in appendix A for a description of this information.

`textlen` is an integer equal to the number of halfwords to be logged from the `text` parameter, or is a negative integer equal to the number of bytes. Length can be zero.

Discussion

DBXBEGIN is called to designate the beginning of a sequence of TurboIMAGE/XL procedure calls that are jointly considered a single dynamic transaction. The end of such a sequence is designated by a matching call to DBXEND. DBXBEGIN cannot be called if another transaction started by DBXBEGIN or DBBEGIN is active. The intrinsic does not begin a dynamic transaction if AUTODEFER is enabled for the database.

Before including a database in a DMDBX, DBXBEGIN mode 3, DBCONTROL mode 7 needs to have been done *once* for each of the databases in the DMDBX. DBCONTROL mode 7 remains active until the database is closed or the application terminates. DBCONTROL mode 7 also enables the database for deadlock detection. If deadlock is encountered, it returns an error 26, instead of triggering a process hang.

NOTE DBXBEGIN is not allowed with DBOPEN mode 2 nor with AUTODEFER enabled. Logging and DBRECOV are not needed with dynamic transactions, because the database can be recovered dynamically. However, if the calling process is logging, DBXBEGIN causes a record to be written to the log file to identify the beginning of a dynamic transaction.

Table 5-24. DBXBEGIN Return Status Values

File System, Memory Management, and Transaction Management Failures:	-4	FREADLABEL error.
	-228	DBXBEGIN encountered XM error <i>nn</i> when starting dynamic transaction.
Calling Errors:	-11	Bad database reference.
	-31	Bad (unrecognized) DBXBEGIN mode: <i>n</i> .
	-139	Invalid number of base IDs.
	-140	Bad base ID list.
	-151	Text length greater than 512 bytes.
	-152	DBXBEGIN called while a transaction is in progress.
	-217	DBOPEN mode <i>n</i> incompatible with Dynamic Rollback.
	-218	Output deferred not compatible with DBX.
	-219	Remote database access incompatible with Dynamic Rollback.
	-221	Cannot begin transaction when a dynamic transaction is active.
	-222	Only DBXUNDO allowed when a dynamic transaction encounters an error.
-242	Error in TurboGTX file operation.	
Logging System Failures:	-111	WRITELOG intrinsic failure.
Exceptional Conditions:	67	DBU disabled; potential damage; only DBCLOSE allowed.
	-332	Error in QLOCK table operation.

Consult appendix A for more information about these conditions.

