DBXEND

INTRINSIC NUMBER 421

Designates the end of a sequence of TurboIMAGE/XL procedure calls regarded as a dynamic transaction for the purposes of logging and dynamic roll-back recovery. The *text* parameter can be used to log user information to the log file. `DBXEND` is used in conjunction with `DBXBEGIN` 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

DBXEND ,{ base

baseidlist } text,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 the `DMDBX`. Use the same `baseidlist` used with `DBXBEGIN` when calling `DBXEND` mode 3 to end the `DMDBX`. The layout of this array is same as in `DBXBEGIN`, except that the transaction ID is already set in the first two halfwords by `DBXBEGIN`.

`text` is 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 `DBXEND` log record.

`mode` must be a halfword equal to 1, 2 or 3.

Mode 1: End of dynamic transaction spanning one database.

Mode 2: End of dynamic transaction spanning one database, started with `DBXBEGIN` mode 1, and write contents of the transaction management (XM) logging buffer in memory to disk. If logging is enabled, the contents of the logging buffer in memory will also be written to disk.

Mode 3: Indicates the end of a `DMDBX` started with `DBXBEGIN` mode 3. If user logging is enabled for the databases, mode 3 generates *multiple* entries in the log file, one for each database in the `DMDBX`, in order to mark the end of a dynamic transaction.

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

Element **Contents**

1 If the procedure succeeds, the return status is 0. Table 5-25. 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 Error 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

`DBXEND` is called to designate the end of a sequence of TurboIMAGE/XL procedure calls that are collectively considered a dynamic transaction. If an intrinsic fails within a dynamic transaction, you may use `DBXEND` to end the dynamic transaction at that point. The modifications completed until that point will remain in the database(s). The beginning of such a sequence is designated by a previous call to `DBXBEGIN. DBXEND` cannot be called to end a transaction started by `DBBEGIN`.

Logging and `DBRECOV` are not needed with dynamic transactions, because the database can be recovered dynamically. However, if the calling process is logging, `DBXEND` causes a record to be written to the log file to identify the end of a dynamic transaction.

If you call `DBXEND` with mode 2, `DBXEND` forces the XM log buffer (and user log buffer if the process is logging) to be written from memory to disk before returning to the calling process. Use this option only for critical transactions; too many mode 2 `DBXEND` calls can degrade performance by causing a disk access each time a dynamic transaction ends. `DBXEND` returns an error condition if it is called without a prior matching call to `DBXBEGIN.DBXEND` is not necessary after a `DBXUNDO`.

Table 5-25. DBXEND Return Status Values

File System, Memory Management, and Transaction Management Failures:	`-213`	DBXEND encountered XM error *nn* when ending dynamic transaction.
Calling Errors:	`-11`	Bad database reference.
	`-31`	Bad (unrecognized) DBXEND mode: *n*.
	`-140`	Bad baseid list.
	`-151`	Text length greater than 512 bytes.
	`-219`	Remote database access incompatible with Dynamic Rollback.
	`-222`	Only `DBXUNDO` allowed when a dynamic transaction encounters an error.
	`-223`	Cannot `DBXEND` or `DBXUNDO` a transaction which was not active.
	`-237`	Cannot `DBXEND` or `DBXUNDO` a `DBBEGIN` transaction.
	`-238`	`MDBX DBXBEGIN, DBXEND` mode mismatch.
Logging System Failures:	`-111`	`WRITELOG` failure.
Exceptional Conditions:	`67`	DBU disabled; potential damage; only `DBCLOSE` allowed.
	`-213`	XM error on `DBXEND`.
	`-242`	Error in TurboGTX file operation.
	`-332`	Error in `QLOCK` table operation.

Consult appendix A for more information about these conditions.