DBEND

INTRINSIC NUMBER 413

Designates the end of a sequence of TurboIMAGE/XL procedure calls regarded as a static or multiple database transaction (based on the mode) for the purposes of logging and recovery. The *text* parameter can be used to log user information to the log file. `DBEND` is used in conjunction with `DBBEGIN` to begin and end a static or multiple database transaction.

OPENTURBO vs TurboIMAGE Differece
Pseudo END Transaction – Commit Work.

OPENTURBO Performance Enhancements
Commit Work, the OPENTURBO transaction initiated by DBBEGIN.

OPENTURBO Additional Features
Commit Work, the OPENTURBO transaction initiated by DBBEGIN

Syntax

DBEND,{ base

baseidlist

transid } ,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.) Use with `DBEND` mode 1 or 2.

`baseidlist` is the name of an integer array containing the list of base IDs which are involved in a multiple database transaction. Use with `DBEND` mode 3 or 4, and set the first two halfwords to binary zeroes. The layout of this array is shown here (each element is a halfword, or two bytes):

Element	Contents
1-2	Must be set to binary zeroes.
3	The number of base IDs involved in the multiple database transaction. Must be a number between 1 and 15, inclusive.
4- `n`	Base IDs of the databases involved in the transaction. Base ID is the first halfword of the `base` parameter used to call TurboIMAGE intrinsics.

`transid` is the name of the integer array containing the two-halfword transaction ID. The transaction ID was returned by `DBBEGIN` mode 3 or 4. Use with `DBEND` mode 3 or 4, and do not set the first two halfwords to binary zeroes.

`text` is an array up to 256 halfwords long which contains user ASCII or binary data to be written to the log file as part of the `DBEND` log record.

`mode` must be an integer equal to 1, 2, 3, or 4.

Mode 1: End of static transaction.

Mode 2: Write contents of the logging buffer in memory to disk, and end the static transaction.

Mode 3: End of multiple database transaction. If user logging is enabled for the databases, mode 3 generates *multiple* entries in the log file in order to mark multiple database transactions. For example, assume that base IDs 11, 12, and 13 are involved in a multiple database transaction. `DBEND` mode 3 (with base IDs 11, 12, and 13 specified in the `baseidlist` parameter) generates the following log record sequence:

DBBEGIN (11, 1/3)

DBBEGIN (12, 2/3)

DBBEGIN (13, 3/3)

.

.

.

database updates

.

.

.

DBEND (11, 1/3)

DBEND (12, 2/3)

DBEND (13, 3/3)

where the notations 1/3, 2/3, 3/3 in the log records indicate "first of three," "second of three," and "third of three." Refer to chapter 7 for more information about user logging.

Mode 4: Write contents of the logging buffer in memory to disk, and end the multiple database transaction. If user logging is enabled for the databases, mode 4 generates *one* entry in the log file in order to mark multiple database transactions. For example, assume that base IDs 11, 12, and 13 are involved in a multiple database transaction. `DBEND` mode 4 (with base IDs 11, 12, and 13 specified in the `baseidlist` parameter) generates the following log record sequence:

MDBXEND (11, 12, 13)

.

.

.

database updates

.

.

.

MDBXEND (11, 12, 13)

Refer to chapter 7 for more information about user logging.

`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 5-8. 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 to be logged. Length can be zero.

Discussion

`DBEND` is called to designate the end of a sequence of TurboIMAGE/XL procedure calls which are collectively considered a static or multiple database transaction. The beginning of such a sequence is designated by a previous call to `DBBEGIN`.

NOTE `DBEND` is not valid if a transaction was begun with `DBXBEGIN`. `DBEND` checks for any active dynamic transactions before executing.

If the process is logging, DBEND causes a log record to be written to the log file which includes such information as the time, date, and user text buffer. DBEND log records are used by the database recovery program DBRECOV to identify the end of static and multiple database transactions. However, if a program aborts, a static or multiple database transaction that has not been completed by a call to DBEND will be recovered by default. For additional information, refer to the discussion of the ABORTS and NOABORTS options under the description of the DBRECOV >CONTROL command in chapter 8.

If you call DBEND with mode 2 or 4 and logging is enabled, DBEND forces the log buffer to be written from memory to disk before returning to the calling process. This flush of the log buffer occurs after the intrinsic has logged the end of the logical transaction. Use this option only for critical transactions; too many mode 2 or mode 4 DBEND calls can degrade performance by causing a disk access each time a static or multiple database transaction ends.

NOTE When you call DBEND with mode 2 or 4 to force writing a static or multiple data base transaction to disk, logging must have been enabled prior to executing the transaction.

DBEND returns an error condition if it is called without a prior matching call to DBBEGIN, whether the process is actually logging or not.

Table 5-9. DBEND Return Status Values

Calling Errors:	-11	Bad database reference.
	-31	Bad mode.
	-146	Invalid transaction ID.
	-147	Mode doesn't match DBBEGIN mode.
	-148	Base ID list doesn't match DBBEGIN base ID list.
	-151	Text length greater than 512 bytes.
	-153	No transaction in progress to end.
	-216	Cannot end a dynamic transaction with a DBEND.
	-222	Only DBXUNDO allowed when a dynamic transaction encounters an error.
Communications Errors:	-102	DSWRITE failure.
	-106	Remote 3000 data inconsistent.
	-107	NS 3000 or DS 3000 system error.
Logging System Failures:	-111	WRITELOG failure.
	-113	FLUSHLOG returned error number to DBEND.
Exceptional Conditions:	-193	DBU control block is full.
	63	DBG disabled; potential damage; only DBCLOSE allowed.

Consult appendix A for more information about these conditions.