DBBEGIN

INTRINSIC NUMBER 412

Designates the beginning 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. `DBBEGIN` is used in conjunction with `DBEND` to begin and end a static or multiple database transaction.

OPENTURBO Specific

Pseudo BEGIN transaction—ORACLE Mode.

OPENTURBO Performance Enhancement

Commit Work and initiate an OPENTURBO transaction.

OPENTURBO Additional Feature

Commit Work and initiate an OPENTURBO transaction.

Syntax

```
    DBBEGIN,{ 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.) Use `base` when calling `DBBEGIN` mode 1 (static transaction).

`baseidlist` is the name of the integer array containing the base IDs of the databases which are involved in a multiple database transaction. Use `baseidlist` when calling `DBBEGIN` mode 3 or 4 (multiple database transaction). The layout of this array is shown here (each element is a halfword or two bytes):

Element **Contents**

1-2 The application program must set these two halfwords to binary 0s before calling `DBBEGIN`. After returning to the calling program, these two halfwords contain the transaction ID. This ID can be used in order to end the transaction by calling `DBEND`.

3 The number of base IDs involved in the multiple database transaction. This 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/XL intrinsics.

`text` is the name of 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 `DBBEGIN` log record. The *text* argument is used to assign each particular transaction a distinct name. (Refer to “Discussion” below for more information.)

`mode` is an integer indicating the type of transaction desired as follows:

Mode 1: Indicates a static transaction.

Mode 3: Indicates a multiple database transaction with one log record per database. 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. `DBBEGIN` 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 details about user logging.

Mode 4: Indicates a 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.

DBBEGIN mode 4 (with base IDs 11, 12, and 13 specified in the `baseidlist` parameter) generates the following log record sequence:

MDBXBEGIN (11, 12, 13)

.

.

```
          .
     database updates
          .
          .
          .
     MDBXEND (11, 12, 13)
```

Refer to chapter 7 for more details about user logging.

`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 5-3. 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.

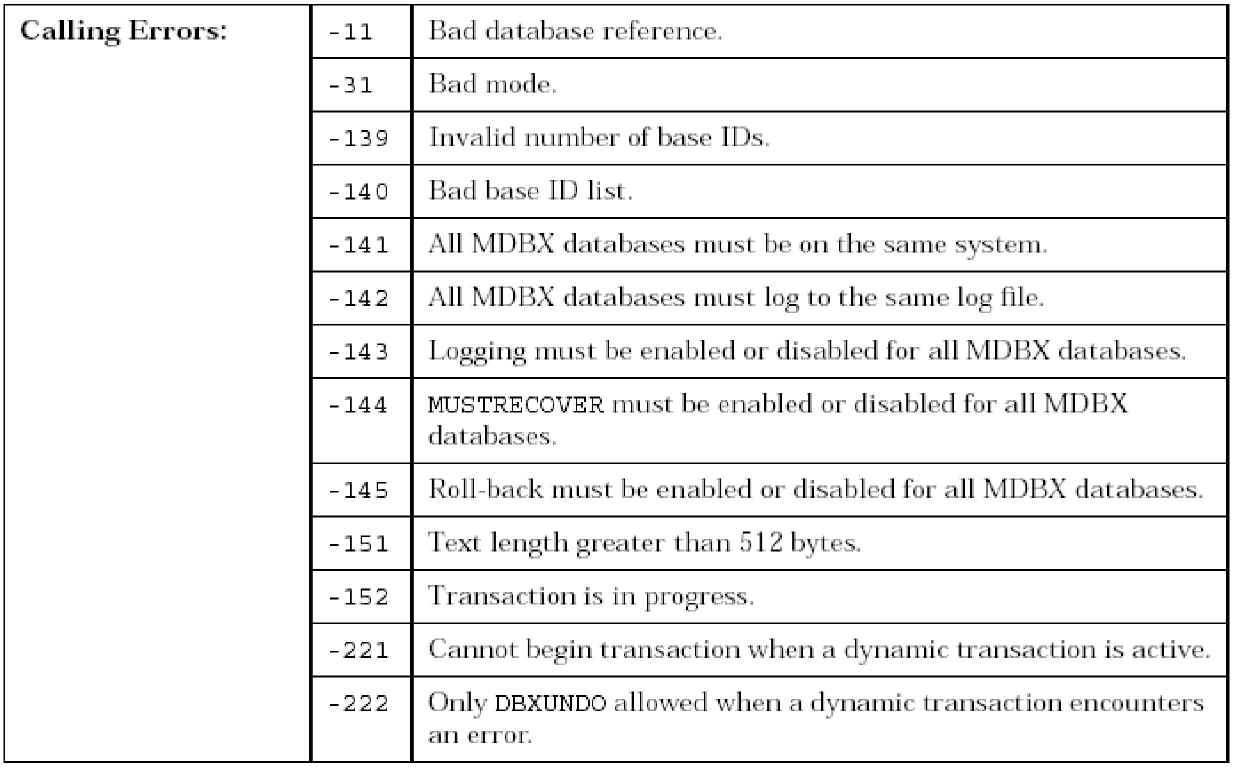
textlen is an integer equal to the number of halfwords to be logged from the *text* parameter, or is a negative integer equal to the number of bytes to be logged. Length can be zero.

Discussion

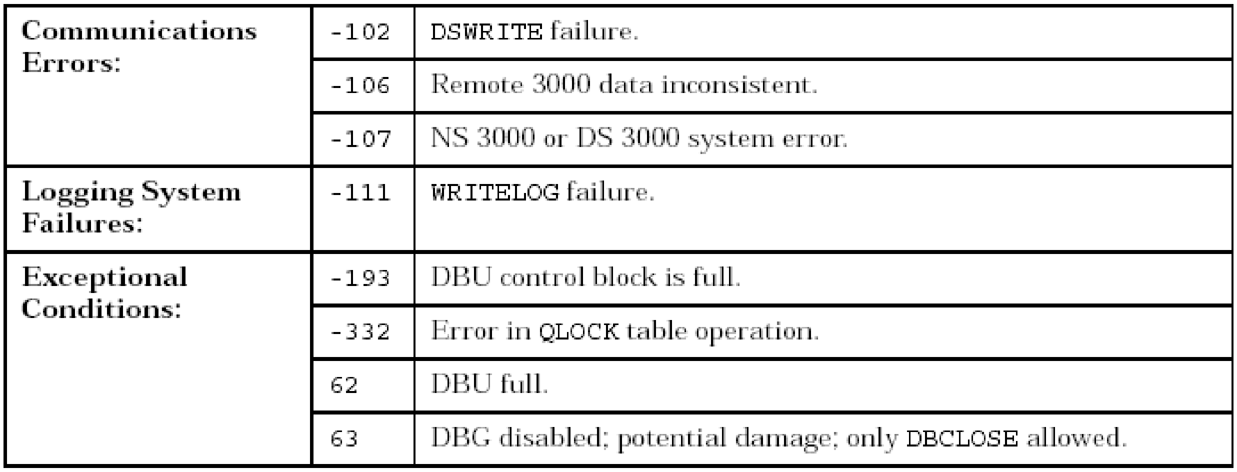
`DBBEGIN` is called to designate the beginning of a sequence of TurboIMAGE/XL procedure calls which are jointly considered a single logical transaction. The transaction is either a static or multiple database transaction, based on the mode. The end of such a sequence is designated by a matching call to `DBEND`. If the calling process is logging, `DBBEGIN` causes a log record to be written to the log file which includes such information as the time, date, and user text buffer. `DBBEGIN` log records are used by the database recovery program `DBRECOV` to identify the beginning of all logical transactions.

`DBBEGIN` returns an error condition if it is called twice without an intervening call to `DBEND` or if it is called while a dynamic transaction is still active, whether the process is actually logging or not.

Table 5-4. DBBEGIN Return Status Values



Calling Errors:	`-11`	Bad database reference.
	`-31`	Bad mode.
	`-139`	Invalid number of base IDs.
	`-140`	Bad base ID list.
	`-141`	All MDBX databases must be on the same system.
	`-142`	All MDBX databases must log to the same log file.
	`-143`	Logging must be enabled or disabled for all MDBX databases.
	`-144`	`MUSTRECOVER` must be enabled or disabled for all MDBX databases.
	`-145`	Roll-back must be enabled or disabled for all MDBX databases.
	`-151`	Text length greater than 512 bytes.
	`-152`	Transaction is in progress.
	`-221`	Cannot begin transaction when a dynamic transaction is active.
	`-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.
Exceptional Conditions:	`-193`	DBU control block is full.
	`-332`	Error in `QLOCK` table operation.
	`62`	DBU full.
	`63`	DBG disabled; potential damage; only `DBCLOSE` allowed.

Consult appendix A for more information about these conditions.