

DBUPDATE

INTRINSIC NUMBER 406

Modifies values of data items in the entry residing at the current record address of a specified data set. To call DBUPDATE, you must open the database in access mode 1, 2, 3, or 4. The update is always carried out correctly against the latest version of the data, regardless of modifications made by other users.

In database access mode 1, 3, or 4, you can use DBUPDATE to modify the values of detail data set search and sort items if permitted by the critical item update (CIUPDATE) option settings for the database and the current process. Master data set key item values *cannot* be modified even if CIUPDATE is permitted.

OPENTURBO vs TurboIMAGE Difference

100%-- CIUPDATE is supported.

OPENTURBO Performance Enhancements

Elimination of AUTOMATIC Dataset

No performance degradation for CIUPDATE, same as NON-CIUPDATE

OPENTURBO Additional Features

Update all fields (columns) always

Syntax

DBUPDATE, base, dset, mode, status, list, buffer

OPENTURBO Specific

CIUPDATE is supported. No performance degradation for CIUPDATE, same for NON-CIUPDATE. Elimination of AUTOMATIC Dataset.

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 base ID.)

dset is the name of an array containing the left-justified name of the data set to be read, or is an integer referencing the data set by number. The data set name can be up to 16 characters long. If shorter, it must be terminated by a semicolon or a blank.

mode must be an integer equal to 1.

If your database is enabled for third-party indexing (TPI), refer to your vendor documentation for additional DBUPDATE mode information. The section on DBUTIL in chapter 8 of this book has a brief description of the TPI option.

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

Element Contents

- | | |
|------|--|
| 1 | If the procedure succeeds, the return status is 0. Table 5-23. describes the contents of element 1 when the procedure does not succeed. |
| 2 | Length of the values in buffer (in halfwords). |
| 3-10 | Same word values set by preceding procedure call which positioned the data set at the current entry. If critical item update is permitted, the value contained in element 3 determines the message returned. |

`list` is the name of an array containing an ordered set of data item identifiers, either names or numbers. Values supplied in the buffer array replace the values of data items occupying the same relative position in the `list` array. The user class established when the database is opened must allow at least read access to all the items included in the `list` array.

If the corresponding buffer array values are the same as the current data item values, the `list` array can include data items to which the user has read access only, such as, key, search and sort items. This feature permits reading and updating with the same `list` array contents. Those items to be updated must allow write access and *cannot* be key, search, or sort items.

The list array can contain a left-justified set of data item names, separated by commas and terminated by a semicolon or a blank. No embedded blanks are allowed and no name can appear more than once.

When referencing by number, the first element of the `list` array is an integer `n` followed by `n` unique data item numbers (one-halfword positive integers).

The `list` not only specifies the data items to be updated immediately but is saved internally by TurboIMAGE/XL as the current list for this data set. The current list is unchanged until a different list is specified in a subsequent call to `DBGET`, `DBPUT`, or `DBUPDATE` for the same access path and data set.

Some special list constructs are allowed. These are described in Table 5-20. with the `DBPUT` procedure. List processing is a relatively high overhead operation that can be shortened substantially in subsequent calls by using the asterisk construct to specify that the current list is to be used.

`buffer` is the name of an array containing concatenated values to replace the values of data items occupying the same relative position in the list array. The number of half words for each value must correspond to the number of half words required by its type multiplied by the sub-item count. Search and sort item values can be included in this update list if their values will not change.

Discussion

Before performing an update for a database opened in access mode 1, TurboIMAGE/XL verifies that locks are in effect to cover the data entry both before and after it is modified. The current record number, forward and backward pointers are unchanged. (Refer to the description of status words 3 through 10.)

If the process is logging, a call to `DBUPDATE` causes a log record to be written with such information as the time, date, user identification number, and a copy of both the old and new data item values.

When `DBUPDATE` is called within a dynamic transaction, a log record is written after the successful completion of the physical transaction. If the intrinsic cannot be completed, an error is returned. This error condition must be checked, and you must decide to use `DBXUNDO`, `DBXEND`, or continue with the remainder of the dynamic transaction. `DBXUNDO` will abort the entire transaction. `DBXEND` will terminate the dynamic transaction; the modifications completed thus far within the transaction will remain in the database.

Table 5-23. DBUPDATE Return Status Values

File System, Memory Management, and Transaction Management Failures:	- 1	FOPEN failure.
	- 3	FREADDIR failure.
	- 4	FREADLABEL failure.
	- 5	FWRITEDIR failure.
	- 167	Cannot begin MPE XL XM transaction: XM error.
	- 168	Cannot attach <i>n</i> to MPE XL XM: file system error <i>nn</i> .
	- 169	Invalid mode for XM attach options.
	- 175	Cannot attach <i>n</i> to MPE XL XM: XM error <i>nn</i> .
	- 176	Cannot detach <i>n</i> from MPE XL XM: XM error <i>nn</i> .
	- 178	Cannot detach <i>n</i> from MPE XL XM: file system error <i>nn</i> .
	- 199	Cannot end MPE XL XM transaction: XM error <i>nn</i> .
	- 209	Invalid mode for XM detach options.
Calling Errors:	- 11	Bad database reference.
	- 12	No locks cover the data entry to be updated. (Occurs only if database is open in access mode 1.)
	- 14	Illegal intrinsic in current access mode.
	- 21	Bad data set reference.
	- 31	Bad mode.
	- 51	Bad <i>list</i> length.
	- 52	Bad list or bad item.
	- 82	CIUPDATE is set to DISALLOWED; cannot use critical item update.
	- 222	Only DBXUNDO allowed when a dynamic transaction encounters an error.
	- 1025	The XM transaction size (28 MB) limit has been reached, only DBXEND or DBXUNDO are allowe.
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.
	-264	Error while writing to TPI files.
	-3nn	Internal error.
	-312	Error while reading database file.
	-314	Error while getting path information for set.
	-332	Error in QLOCK table operation.
	17	No entry.
	41	DBUPDATE attempted to modify value of critical item--key, search or sort.
	42	Read only item.
	49	Illegal buffer address.
	50	Buffer too small.
	62	DBG full.
	63	DBG disabled; potential damage; only DBCLOSE allowed.
	68	DBB disabled.

Appendix A contains more information about these conditions.