DBLOCK

INTRINSIC NUMBER 409
Applies a logical lock to a database, one or more data sets, or one or more data entries.

OPENTURBO vs TurboIMAGE Difference
100%

OPENTURBO Performance Enhancements
Thick Client LOCK Manager
Thin Server LOCK Manager

OPENTURBO Additional Features
Entity LOCK support for ORACLE applications (ORACLE LOCK Object).
OPENTURBO and ORALCE applications co-existence support.

```
/*********************************************************************/
/* Send LOCK Request to Server:                                      */
/*                                                                   */
/*      Table1: PID, memo, timstamp                                  */
/*      Table2: PID, level, db, set, item, op, vallen, val           */
/*                                                                   */
/*      PID  : -1 (int)                                              */
/*      memo : max of 35 characters (char[36])                       */
/*                                                                   */
/*      # of Descriptor: (int)                                       */
/*            Descriptor Format =>                                   */
/*                LOCK Level    (int)                                */
/*                                 TI_NO_LOCK        = 0             */
/*                                 TI_DBLEVEL_LOCK   = 1             */
/*                                 TI_SETLEVEL_LOCK  = 3             */
/*                                 TI_ITEMLEVEL_LOCK = 5             */
/*                Database ID  (int)                                 */
/*                Dataset  ID  (int)                                 */
/*                Dataitem ID  (int)                                 */
/*                Operator     (int)                                 */
/*                                 TI_DBLOCK_NOOP = 0                */
/*                                 TI_DBLOCK_EQOP = 1   (==)         */
/*                                 TI_DBLOCK_GEOP = 2   (>=)         */
/*                                 TI_DBLOCK_LEOP = 3   (<=)         */
/*                Value Type    E,R,I,J,P,Z,K,U,X                    */
/*                Value Length -> in number of (int)                 */
/*                Value        -> array of (int)                     */
/*                                pad with trailing 0                */
/*                                MAX = 256 Bytes (64 int)           */
/*                                                                   */
/*                *Note: each item can have the followings:          */
/*                    1) item =  value                               */
/*                    2) item >= value                               */
/*                    3) item <= value                               */
/*                    4) item >= value1 && item <= value2            */
/*                                                                   */
/*********************************************************************/
```

Syntax
DBLOCK, base, qualifier, mode, status

Parameters
`base`   is the name of the array used for 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.)

`qualifier`   **Modes 1 and 2:** Ignored.

                             Modes 3 and 4: An integer variable referencing the data set number or the name
                             of an array containing a data set name. Could also be "@", when applying a
                             database lock.

                             Modes 5 and 6: The name of the array containing the lock descriptors. The
                             format for lock descriptors is given in Figure 5-2. Use care when changing
                             modes. The qualifier parameter can also change.

`mode`   contains an integer indicating the type of locking desired (refer to Table 5-15.).

         NOTE If the database is open in user access mode 1, a lock must be in effect on either the data
         set or the whole database when adding to or deleting from master data sets. If a data entry level
         lock is specified, any subsequent DBPUTs or DBDELETEs will fail with error number −12 and
         the following message is returned:

                  intrinsic name CALLED WITHOUT COVERING LOCK IN EFFECT

         Note, however, that a lock on either the entire database or data set can be achieved with a data
         entry lock when an @ sign is used to specify either all data sets or all data items.

`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:

Element	Contents

         1   If the procedure succeeds, the return status is 0. Table 5-17. describes the
         contents of element 1 when the procedure does not succeed.

         2   The number of lock descriptors that were successfully applied in the DBLOCK
         request. For successful locks in modes 1 through 4 this will be 1.

         3   If the return status is 20, this element contains 0 if the database is locked, 1 if the
         data set or entries are locked.

         4   Reserved: Contents undefined.

5-10   Information about the procedure call and its results. Refer to "Library Procedure Error Messages" in appendix A for a complete description of this information.

NOTE Concurrent processes running in a process-handling environment must have MR capability if they call DBLOCK.

Discussion

The format of the array containing a list of lock descriptors is illustrated in Figure 5-2. and applies only for locking modes 5 or 6. The number of lock descriptors (n) is a one-halfword binary integer. Only the first n lock descriptors are processed. If n is zero, DBLOCK returns without taking any action. The format of a lock descriptor is illustrated in Figure 5-3., and the lock descriptor fields are described in Table 5-16.

The shortest possible descriptor is 9 halfwords long consisting of the length field and a *dset* field containing @. Although the *dset* field only contains an at-sign, it must still be 8 halfwords long. The length of the entire descriptor array cannot exceed 4094 bytes. Lock descriptors are sorted by data set number, then by value provided for the lock item.

TurboIMAGE/XL does not sort by item within the set, because more than one item per

data set constitutes a conflicting lock descriptor (TurboIMAGE/XL error −134).

Table 5-15. Locking Mode Options

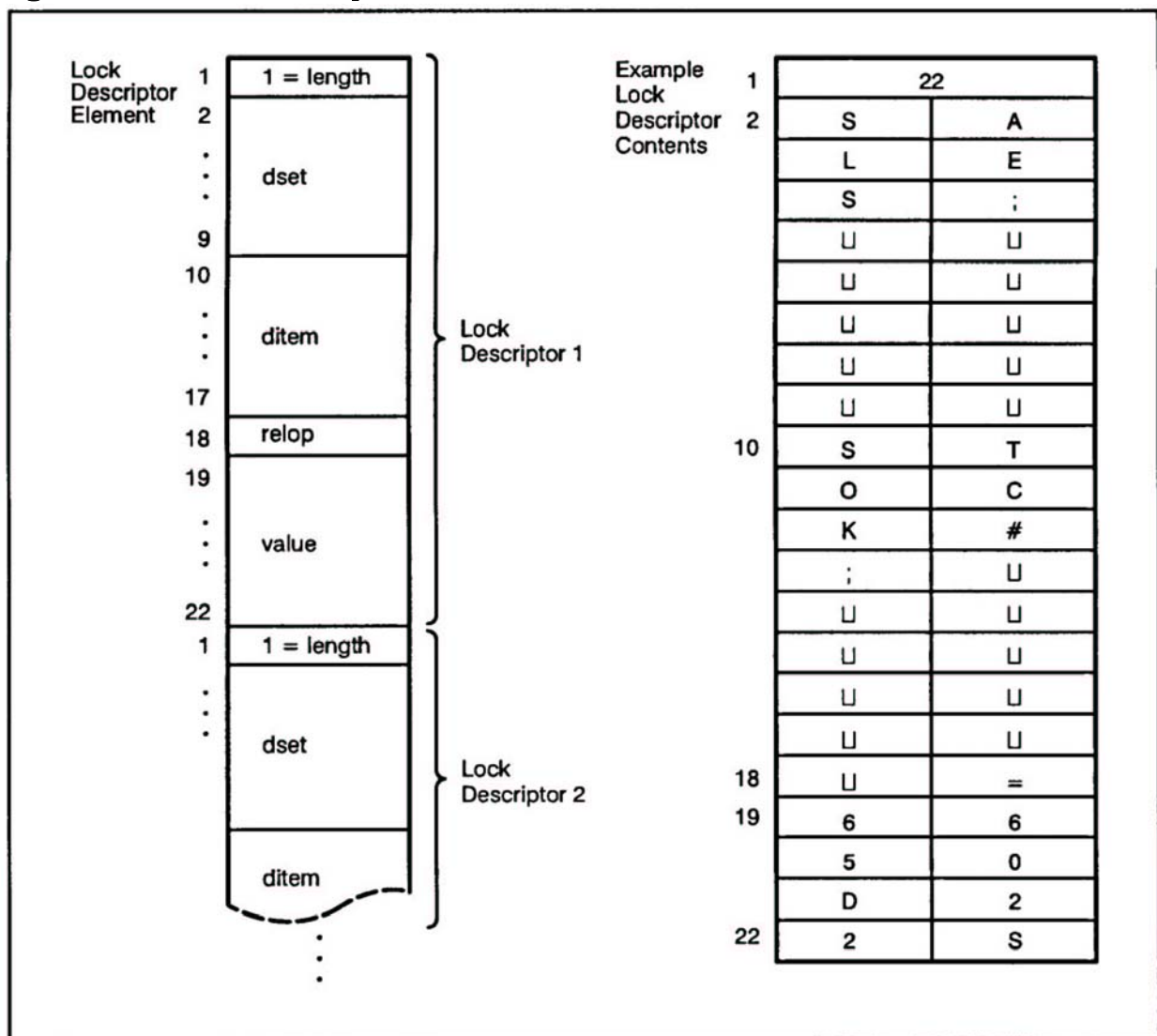| Lock Mode	Lock Level	Locking Type	Description
1	Base	Unconditional	DBLOCK applies an unconditional lock to the whole database, returning to the calling program only after the lock is successful (or if an error occurs). The *qualifier* parameter is ignored.
2	Base	Conditional	DBLOCK applies a conditional lock to the database and returns immediately. A return status of zero indicates success. A non-zero return status indicates the reason for failure. (Refer to Table 5-17.)
3	Set	Unconditional	DBLOCK applies an unconditional lock to a data set. The *qualifier* parameter must specify the name of an array containing the left-justified name of the data set or the name of an integer referencing the data set number. The data set name can be 16 characters long or, if shorter, terminated by a semicolon or blank.
			The data set need not be accessible for read or write access to the user requesting the lock.
4	Set	Conditional	DBLOCK applies a conditional lock of the same type as mode 3. It always returns to the calling program immediately. A return status of zero indicates success and a non-zero return status indicates a reason for failure. (Refer to Table 5-17.)
5	Entry	Unconditional	DBLOCK applies unconditional locks to the data entries specified by lock descriptors. The *qualifier* parameter must specify the name of an array containing the lock descriptors. The format of the array is shown in Figure 5-2. It returns only when all the locks have been acquired.
6	Entry	Conditional	DBLOCK applies conditional locks of the same type as mode 5. If multiple lock descriptors are specified and DBLOCK encounters a lock descriptor that it cannot apply, it returns. All locks that have been applied until that point are retained.
			Because the locks are not executed in the order supplied by the user, it is not predictable which locks are held and which are not after an unsuccessful mode 6 DBLOCK. Status element 2 indicates how many lock descriptors were actually successful. It is recommended that a DBUNLOCK be issued after any unsuccessful mode 6 DBLOCK.

NOTE Be careful when changing modes. The *qualifier* parameter can change.

Figure 5-2. Qualifier Array Format for Locking Modes 5 and 6
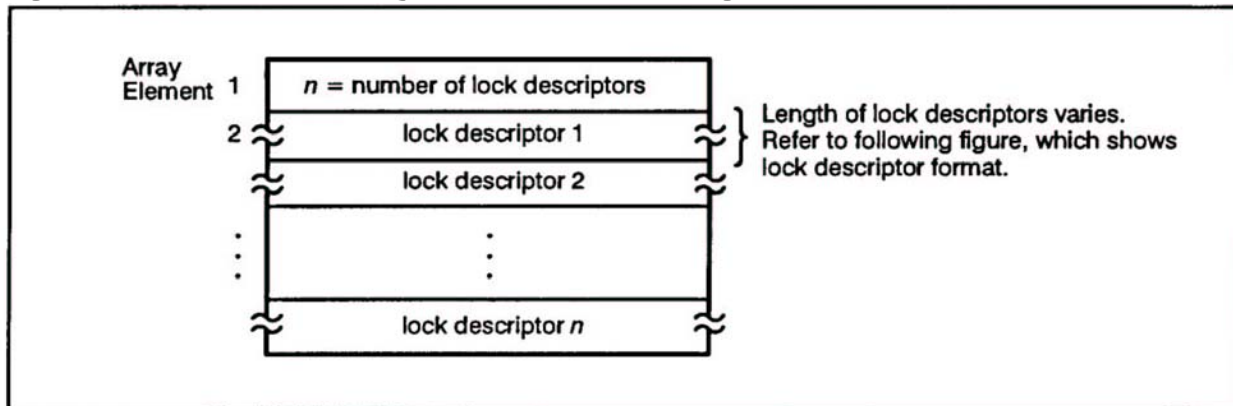
LG200137_014a

Figure 5-3. Lock Descriptor Format

LG200137_015a

Table 5-16. Lock Descriptor Fields

Field Name	Description
length	A halfword integer specifying the physical length in halfwords of the lock descriptor, including the *length* field itself.
dset	Describes the data set in which locks are placed. It is always 8 halfwords long and can be one of the following: • A data set name, left-justified, 16 characters long or, if shorter, terminated with a blank or semicolon (for example, SALES;). • A data set number, an integer in the range of 1 to 199 stored in the first element. • An at-sign (@) stored in the first byte of the *dset* and a lock descriptor length of 2 indicating that the whole database is to be locked. All unused bytes are ignored. In this case, the *ditem, relop,* and *value* fields are ignored and can be omitted if desired. • A blank or semicolon (first byte) or binary zero (first halfword) indicating that the whole lock descriptor is to be ignored. (It is counted as one of the *n* descriptors.) The data set, if specified, need not be accessible for read or write access to the user requesting the lock.
ditem	Always 8 halfwords long unless an @ is stored in the first byte. It can be one of the following: • A data item name, left-justified, 16 characters long or, if shorter, terminated with a blank or semicolon. • A data item number stored as an integer in the first halfword. It can be in the range of 1 to 1023. • An at-sign (@) stored in the first byte of the *ditem* indicating that the whole data set specified in *dset* is to be locked. All unused bytes are ignored and can be omitted if desired. The data item need not be a search item, nor does it have to be accessible to the user requesting the lock. However, it cannot be a compound item or a P-type item longer than P28.

relop	One halfword long, it contains one of the three relational operators represented as two ASCII characters:
	
	    <= less than or equal
	
	    >= greater than or equal
	
	    =⎵ or ⎵= equal (⎵ indicates space character)

value	The value of the data item to be locked. It must be stored in exactly the same way as it is stored in the database. TurboIMAGE/XL extracts as many halfwords as required by the corresponding data item definition (in the schema). The rest (if any) are ignored.

If you specify a data item of type P, U, or Z in a lock descriptor, TurboIMAGE/XL checks that the value is valid for that data item type. The following checks are made:

- If the data item is type P, the right half of the right most byte must contain a sign and all preceding nibbles must contain decimal digits represented in Binary Coded Decimal (BCD) format. For example, if a data item is defined as type P with a length of 20, the format must be as shown here:

Figure 5-4. Lock Descriptor Format

This would be declared in COBOL II as 19 digits plus a sign or 20 nibbles (P20 in the schema):

S9(19) COMP-3

Type P data item used in a lock descriptor cannot exceed 28 nibbles (7 halfwords) in length. The locking system treats all sign digits other than 1101_2 as identical. 1101_2 is assumed to be a negative sign.

- If the data item is type U, the value cannot contain any lowercase alphabetic characters in the range of a through z (for non-native language use only).

- If the data item is type U or X, and a lock specifies an inequality, the language of the database is used.

- If the data item is type Z, each byte preceding the last one must contain an 8-bit digit represented in ASCII format and the last byte must contain a value representing a digit and a sign.

- If the data item is type R, it is sorted based on the HP 3000 floating point number format.

Table 5-17. DBLOCK Return Status Values

Calling Errors:	-11	Bad database reference.	
	-31	Bad mode value.	
	-121	Descriptor count error.	
	-123	Illegal relop in a descriptor.	
	-124	Descriptor too short. Must be greater than or equal to 9.	
	-125	Bad set name/number.	
	-126	Bad item name/number.	
	-127	Attempt to lock using a compound item.	
	-128	Value field too short in a descriptor.	
	-129	P-type item longer than P28 specified.	
	-130	Illegal digit in a P-type value.	
	-131	Lowercase character in type U value.	
	-132	Illegal digit in type Z value.	
	-133	Illegal sign in type Z value.	
	-134	Two descriptors conflict.	
	-135	DBLOCK called when locks already in effect.	
	-136	Descriptor list exceeds 4094 bytes.	
	-222	Only DBXUNDO allowed when a dynamic transaction encounters an error.	
Communications Errors:	-102	DSWRITE failure.	
	-103	Remote 3000 stack too small.	
	-106	Remote 3000 data inconsistent.	
	-107	NS 3000 or DS 3000 system error.	

Exceptional Conditions:			Applicable Modes
	20	Database locked or contains locks.	(2,4,6)
		(Status element 3: 0 = database locked 1 = data set or entries locked)	
	22	Data set locked by another process.	(3,4,5,6)
	23	Entries locked within set.	(4)
	24	Item conflicts with current locks.	(6)
	25	Entry or entries already locked.	(6)
	26	Lock not performed since deadlock would occur.	(1,2,3,4,5,6)
	62	DBG full. (If this error occurs when multiple lock descriptors are specified, some of the descriptors may have been successfully completed. If so, they are not unlocked by TurboIMAGE/XL before returning the error. Therefore, issue a DBUNLOCK after any positive-numbered error, unless you have reason to do otherwise.)	(5,6)
	63	DBG disabled; potential damage; only DBCLOSE allowed.	
	-192	Invalid DBU.	
	-241	Bad tag for TurboLKT table.	

Appendix A contains more information about these conditions.