ELOCK$ - Special Exclusive Lock

The ELOCK$ routine provides an alternative to the standard LOCK verb as described in Global Development File Management Manual. Whereas the standard LOCK verb allows multiple locks from a user on a particular Lock Region, the ELOCK$ routine will return an exception if a user attempts to lock a Lock Region that is already locked by that user.

1. Invocation

To execute a special exclusive lock code:

```
CALL ELOCK$ USING fd region
```

Where filename identifies the file definition (FD) of a file which must have been opened previously, otherwise the program will be terminated in error. If the file was not OPEN SHARED the operation will be ignored; and region, is the name of a 4-byte area used to define the part of the file to which exclusive access is required. Typically, the region code might be:

- The PIC 9(9) COMP record key of the file record that you require to update. The effect of the LOCK statement will then be a record-level lock;

- A PIC X(4) subset of a longer indexed sequential key field. The LOCK statement will then grant exclusive access to records whose keys contain the subset value you have specified.

2. STOP Codes and Exception Conditions

No STOP codes are generated by ELOCK$.

The following EXIT codes may be returned by ELOCK$:

EXIT code	**$$COND**	**Description**
2501	1	All available locks are in use.
2502	2	The region is locked by another user, or by the same user (on a different FD).

3. Programming Notes

Unlike the standard LOCK verb, the ELOCK$ routine does not support a WAIT clause.

4. Examples

[EXAMPLE REQUIRED]

5. Copy-Books

None.

6. See Also

None.