Subject: Re: Accessing an ASCII database?
Posted by thompson on Fri, 20 Dec 2002 16:15:11 GMT
View Forum Message <> Reply to Message

Craig Markwardt <craigmnet@cow.physics.wisc.edu> writes:

> Jonathan Greenberg <greenberg@ucdavis.edu> writes:
>> I'm trying to develop an IDL routine that allows multiple machines running
>> IDL to query an ASCII database, and update it-- are there any prebuilt
>> routines that could help me out with this? I'm running into problems using
>> openu, because if two machines open the text file at the same time, strange
>> things ensue. Anyone have experience with this or can point me to some
>> scripts that could help me out?

> By any chance, is the IDL Astronomy Library database system suitable
> to you? It's extensive, but I've never used it myself.

We do use it, but even there we have to be careful not to have two processes write to the database at the same time. The way we ended up doing it was to use lock files to manage the process. We would create a file called mydatabase.LOCK in the same directory as mydatabase before opening the database for writing, and then deleting the lock file when the database was closed. Any processes wanting to open the file for update would first check to make sure the lock file wasn't there.

We've never bothered to check for locked databases on read (openr), but the writes almost always append to the end of the files, leaving the bulk of the file untouched. Also, we always open the database and close it again for each individual database transaction, so you're always reading from the current database.

I've appended a simple procedure which you could probably modify for your purpose. There's also some alternative, more sophisticated, software which you may find useful at

ftp://sohoftp.nascom.nasa.gov/solarsoft/gen/idl/system/

The files there are apply_lock.pro, check_lock.pro, rm_lock.pro.

Bill Thompson

```
 PRO LOCK_DATABASE, DATABASE, LOCKFILE
;+
; Project     : SOHO - CDS
;
```

```
; Name        : LOCK_DATABASE
;
; Purpose     : Lock a CDS database for write.
;
; Category    : Class4, Operations, Database
;
; Explanation : Locks a catalog database for write access.  If another process
;  has the catalog locked, then wait until it is unlocked before
;  locking it.
;
;  An empty file called <database>.LOCK (e.g. experiment.LOCK) is
;  created in the same directory as the database.  This signals to
;  other processes that the database is locked.
;
; Syntax      : LOCK_DATABASE, DATABASE, LOCKFILE
;
; Examples    : LOCK_DATABASE, 'experiment', LOCKFILE
;  ... write to database, e.g. using DBBUILD ...
;  UNLOCK_DATABASE, LOCKFILE
;
; Inputs      : DATABASE = The name of the database.  The program looks for a
;       file with the given name, and the extension .dbf in
;       either the current directory, or the path given by
;       the environment variable ZDBASE.
;
; Opt. Inputs : None.
;
; Outputs     : LOCKFILE = The complete name of the file <database>.LOCK,
;       including the path.
;
; Opt. Outputs: None.
;
; Keywords    : None.
;
; Calls       : FIND_WITH_DEF, BREAK_FILE, FILE_EXIST, CDS_MESSAGE
;
; Common      : None.
;
; Restrictions: Must have write access in the directory containing the database
;  files.
;
; Side effects: There is no timeout to this procedure.  It will wait forever
;  for the database to be unlocked.  If a process dies leaving the
;  lock file in place, then it must be deleted by hand.
;
; Prev. Hist. : None.
;
; History     : Version 1, 09-Apr-1996, William Thompson, GSFC
```

```
;  Version 2, 12-Apr-1996, William Thompson, GSFC
;   Improved status message
;
; Contact     : WTHOMPSON
;-
;
 ON_ERROR, 2
;
;  Check the input parameters.
;
 IF N_PARAMS() NE 2 THEN MESSAGE, $
  'Syntax:  LOCK_DATABASE, DATABASE, LOCKFILE'
;
;  See if the lock file exists.  If it does, then wait until it's unlocked.
;  Every 30 seconds, print out a message.
;
 WAITED = 0L
 TEMPNAME = FIND_WITH_DEF(DATABASE+'.dbf','$ZDBASE')
 BREAK_FILE, TEMPNAME, DISK, DIR, LOCKNAME, EXT
 LOCKFILE = DISK + DIR + LOCKNAME + '.LOCK'
 WHILE FILE_EXIST(LOCKFILE) DO BEGIN
  IF (WAITED MOD 10) EQ 0 THEN CDS_MESSAGE, /CONTINUE, $
   'Waiting for database ' + DATABASE + $
   ' to be unlocked ...'
  WAIT, 1
  WAITED = WAITED + 1
 ENDWHILE
;
;  Lock the database.
;
 OPENW, UNIT, LOCKFILE, /GET_LUN
 PRINTF, UNIT, ''
 FREE_LUN, UNIT
;
 RETURN
 END
```