Subject: Accessing an ASCII database?
Posted by Jonathan Greenberg on Thu, 19 Dec 2002 17:42:40 GMT
View Forum Message <> Reply to Message

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?

--j

Subject: sec : U Re: Accessing an ASCII database? Posted by Andrew Cool on Thu, 19 Dec 2002 23:14:58 GMT View Forum Message <> Reply to Message

## Jonathan Greenberg wrote:

\_

- > 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?

>

> --j

Jonathon,

Have a look at the FSTAT function.

The fields OPEN, READ and WRITE should allow you to block access by a second party until the file is released by the guy who's currently got the file open.

•

**Andrew** 

Fields of the FSTAT Structure

The following descriptions are of fields in the structure returned by the FSTAT function. They are not keywords to FSTAT.

UNIT � The IDL logical unit number (LUN). NAME � The name of the file.

OPEN � Nonzero if the file unit is open. If OPEN is zero,

the remaining fields in FSTAT will not contain useful information.

READ i¿½ Nonzero if the file is open for read access. WRITE i¿½ Nonzero if the file is open for write access.

ISATTY � Nonzero if the file is actually a terminal instead of a normal file. For example, if you are using an xterm window on a UNIX system and you invoke FSTAT on logical unit 0 (standard input), ISATTY will be set to 1.

ISAGUI i¿½ Nonzero if the file is actually a Graphical User Interface (for example, a logical unit associated with the IDL Development Environment). Thus, if you are using the IDLDE and you invoke FSTAT on logical unit 0 (standard input), ISAGUI will be set to 1.

INTERACTIVE i; 1/2 Nonzero if either ISATTY or ISAGUI is nonzero.

XDR � Nonzero if the file was opened with the XDR keyword, and is therefore considered to contain data in the XDR format.

COMPRESS i¿½ Nonzero if the file was opened with the COMPRESS keyword, and is therefore considered to contain compressed data in the GZIP format.

\_\_\_\_\_\_

Andrew D. Cool
Electromagnetics & Propagation Group
Intelligence, Surveillance & Reconnaissance Division
Defence Science & Technology Organisation
PO Box 1500, Edinburgh
South Australia 5111

Phone: 061 8 8259 5740 Fax: 061 8 8259 6673

Email: andrew.cool@dsto.defence.gov.au

\_\_\_\_\_\_

Subject: Re: Accessing an ASCII database?
Posted by Craig Markwardt on Fri, 20 Dec 2002 01:13:10 GMT
View Forum Message <> Reply to Message

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.

Craig

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 guery 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
```

Subject: Re: Accessing an ASCII database? Posted by condor on Sat, 21 Dec 2002 00:05:19 GMT

View Forum Message <> Reply to Message

Jonathan Greenberg <a href="mailto:greenberg@ucdavis.edu">greenberg@ucdavis.edu</a> wrote in message news:<BA274510.11DB6%greenberg@ucdavis.edu</a>...

- > 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.

Maybe this is a stupid comment, but things like SQL exist exactly to take care of things like this and to keep the programmer/user from having to worry about the underlying bit'n'byte logistics.

Is there a reason why that db HAS to be in ascii? Would it be sufficient if it were in (adabas/oracle/whathaveyou) format and there was a little routine for showing records in ascii when needed?

Subject: Re: Accessing an ASCII database?
Posted by Jonathan Greenberg on Sat, 21 Dec 2002 03:29:09 GMT
View Forum Message <> Reply to Message

I understand that, but can I put together a database using only IDL in an hour or less? I'm using IDL for scientific research, and I don't want to learn an entirely new language/program if I can create a "hack" that will perform more or less just as well as a more elegant solution... If there is an EASY way to do this in IDL, I'd love to know!

--j

On 12/20/02 4:05 PM, in article df160b8f.0212201605.142abb79@posting.google.com, "Big Bird" <condor@biosys.net> wrote:

- > Jonathan Greenberg < greenberg@ucdavis.edu> wrote in message
- > news:<BA274510.11DB6%greenberg@ucdavis.edu>...
- >> I'm trying to develop an IDL routine that allows multiple machines running
- >> IDL to guery 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.

> >

> Maybe this is a stupid comment, but things like SQL exist exactly to

- > take care of things like this and to keep the programmer/user from
- having to worry about the underlying bit'n'byte logistics.

- > Is there a reason why that db HAS to be in ascii? Would it be
- > sufficient if it were in (adabas/oracle/whathaveyou) format and there
- > was a little routine for showing records in ascii when needed?

Subject: Re: Accessing an ASCII database? Posted by Jonathan Greenberg on Sat, 21 Dec 2002 03:30:33 GMT View Forum Message <> Reply to Message

I forgot to answer your question -- no, it does not have to be ascii -- as a matter of fact I've already changed it to a byte file...

--i

On 12/20/02 4:05 PM, in article df160b8f.0212201605.142abb79@posting.google.com, "Big Bird" <condor@biosys.net> wrote:

- > Jonathan Greenberg < greenberg @ ucdavis.edu > wrote in message
- > news:<BA274510.11DB6%greenberg@ucdavis.edu>...
- >> I'm trying to develop an IDL routine that allows multiple machines running
- >> IDL to guery 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.

>

- Maybe this is a stupid comment, but things like SQL exist exactly to
- > take care of things like this and to keep the programmer/user from
- > having to worry about the underlying bit'n'byte logistics.

>

- > Is there a reason why that db HAS to be in ascii? Would it be
- > sufficient if it were in (adabas/oracle/whathaveyou) format and there
- > was a little routine for showing records in ascii when needed?

Subject: Re: Accessing an ASCII database? Posted by Pavel A. Romashkin on Mon, 23 Dec 2002 19:51:19 GMT View Forum Message <> Reply to Message

## Jonathan Greenberg wrote:

- > I understand that, but can I put together a database using only IDL in an
- > hour or less?

I don't think so. Not if you want it to be a real \*databse\*. If all you need is a \*journal\*, then yes. A database is a whole lot more than a searchable text file, and I would not take upon the task of writing a full blown IDL-driven database. But a searchable text file may be sufficient in your case, and that is up to you to determine. And if yes, then of course you can do it in an hour or less:-) Cheers, Pavel