Subject: SQlite file locking - problems to handle
Posted by papascalientes on Mon, 04 Apr 2011 12:18:31 GMT
View Forum Message <> Reply to Message

Hello,

I try to write an application with read/write operations on already existing sqlite databases that should run in a network. Therefore it is necessary to handle file locking situations. The way I tried this is illustrated in the following code example for a function that writes to a database:

***** begin code example ***************************************
+++ calling the function from another function +++

```
try{writezsummary() ;}
catch(String &e)
{
PromptOK(e);
throw Exc("exit");
}
```

+++ function void writezsummary() - only the database writing part -

```
Sqlite3Session z_sql3;
if(!z_sql3.Open(z.db3))
   throw Exc("Can't create or open database file: Z.db3");


Sql sqlz(z_sql3);
z_sql3.Begin();

cInsertstring="INSERT INTO ZSUMMARY VALUES("firtst nametest, last nametest")";      // using
testvariables
int iINT_DEFAULT_NUMBER_OF_TRIES = 5;

for(int p=0;p<10;++p)    //i.e. 10 persons -> 10 times inserting into the database
{
for(int av=0;av<(iINT_DEFAULT_NUMBER_OF_TRIES+1);++av)    // if locked, try again to write
for 5 times
 {
  sqlz.ClearError();
  sqlz.Execute(cInsertstring);
  if(sqlz.WasError())
  {
   if(av<iINT_DEFAULT_NUMBER_OF_TRIES)         // inform user, wait a bit
   {
    status=sqlz.GetLastError()+", "+AsString(av+1)+".try" ;
    status.Sync();
    Sleep(4000+av*1000);
```

```
   }
  }
  else
  {
   sqlz.ClearError();
   if(av>0)
   {
    status="Database update successful";
    status.Sync();
   }
   break;
  }
 }
 if(sqlz.WasError())
 {
  database_locked=true;         // bool variable used somewhere else in the program
  return;
 }
}
z_sql3.Commit();
z_sql3.Close();
```

********** end of example **********
That worked not the way I wanted. So I read more about file locking and concurrency in SQlite and looked for information to handle file locking in U++. I had to realize, that I didn't understand very well how SQL (SQlite) is used in U++. So I hope, someone can help me with the following questions:

1. in U++ I have a session object (z_sql3 in the example) and an Sql object (sqlz(z_sql3) in the example). In SQlite documentation they are always referring to transactions. What corresponds to the transaction in U++, the session object or the sql object?

And a related question:
1a)I can use Commit(), Begin(), Clear(), ... with the session object or with the SQL object. Is there any difference between using the methods with the session object and using the the methods with SQL object or is it the same?

2. (more SQlite specific question/s): If I start a SQlite transaction with Begin() in U++, does the transaction persist
- only when a Commit() failed because of a shared lock
- always if a Commit() failed

And related questions:
2a) If I Close() the session, what happens to the transaction that was started with Begin() and is closed after a successful Commit() but may still be open if the Commit() failed

2b) What happens to the Journal, if it is not possible to write to the database. When and how is it closed/destroyed?

And of course I am also grateful for general tipps how to handle file locking and concurrency database operations in U++.

Thanks in advance,

Petra
