Subject: Schema File
Posted by kevind on Wed, 25 Oct 2006 16:25:19 GMT
View Forum Message <> Reply to Message

First of all, I am very happy with U++. I found it last week and am very impressed with the increase in development speed.

Although, I've hit a road block and don't fully understand a few things.

1. I've created a sqlite3 database with a number of tables. It seems that running the standard code included:

```
#ifdef _DEBUG
SqlSchema sch(SQLITE3);
StdStatementExecutor se(session);
session.SetTrace();
All_Tables(sch);
if(sch.ScriptChanged(SqlSchema::UPGRADE))
 Sqlite3PerformScript(sch.Upgrade(),se);
if(sch.ScriptChanged(SqlSchema::ATTRIBUTES)) {
 Sqlite3PerformScript(sch.Attributes(),se);
}
if(sch.ScriptChanged(SqlSchema::CONFIG)) {
 Sqlite3PerformScript(sch.ConfigDrop(),se);
 Sqlite3PerformScript(sch.Config(),se);
}
sch.SaveNormal();
session.SetTrace();
#endif
```

should create the sch file. But if the file (.sch) doesn't exist at compile time (as specified by MODEL) my program doesn't compile. And if it already existed it doesn't change. I saw the explanation of this code in another thread but it doesn't seem to be behaving as per my expectations.

2. Are there any suggestions on the way to manage a sql session and best practices? If I create a session in one class how do I access it in another?

More concretely: If have my MainClass and I initiate Sqlite3Session in it, I can't access it in any obvious (to me) way from SomeOtherClass. (Could I possibly see a small code example as well?) All the examples I've seen have all the queries in a single class. I am trying to make an app with a different class for each tab and don't want to create a new connection for each tab.

3. Finally, I am trying to create a SqlArray and running into the following problem:

```
MainClass:
 SQL;
 Sqlite3Session session;

 session.LogErrors(true);
 if(!session.Open(ConfigFile("C:\\MyApps\\testapp\\gsm.db"))) {
  LOG("Can't create or open database file\n");
  return;
 }
 SQL = session;

AnotherClass
 CtrlLayout(*this);
 SizePos();
 CtrlLayout(auth);

 SqlId username("username");
 SqlId user_list("user_list");

 ldap_array.SetTable("user_list");
 ldap_array.AddColumn("username","username",3);
 ldap_array.SetOrderBy("username");
 ldap_array.Query();
```

At the last line I get a seg fault. This is a reproducible event if I try to do a session.Open(ConfigFile("")). I wish I could give you more on this but I don't fully understand my error.

I am running 610-dev3 on win32, here is the gdb output:

```
Program received signal SIGSEGV, Segmentation fault.
0x0074468b in Sql::GetSession() const (this=0x879c90)
    at C:/upp/uppsrc/Sql/Sqls.h:174
174             SqlSession& GetSession() const                  { return cn->
GetSession(); }
Current language:  auto; currently c++
(gdb)
```

Problem 3 is probably due to my lack of understanding of 1 & 2.

Thanks for your help.

-Kevin

Subject: Re: Schema File
Posted by mirek on Wed, 25 Oct 2006 23:02:58 GMT
View Forum Message <> Reply to Message

(1) No, that is misunderstanding. .sch file is your description of database shema (what tables you want to have in database). Those lines of code take this schema, create .sql script (containing DDL statements like 'create table...') from them and execute them to have you schema realized in the table.

More specifically, the script executed is so called "upgrade script" - it creates the schema by adding single columns to the table - that way you can add columns to tables without loosing your database data.

(2) SQL is "global default session", so you should be able to get to your database using it. Almost every U++ SQL related interface defaults to this session, so as long as you are accessing only this one, you should be OK.

The likely cause of segfault is (1) - your tables do not exist in database.

BTW

```
 SqlId username("username");
 SqlId user_list("user_list");

 ldap_array.SetTable("user_list");
 ldap_array.AddColumn("username","username",3);
```

you should rather use SqlId constants in SetTable and AddColumn:

```
 ldap_array.SetTable(user_list);
 ldap_array.AddColumn(username,"username",3);
```

Those constants can also get defined by .sch file too. In U++ we tend to uppercase all database names too. E.g.

mydb.sch:

```
TABLE_(MYTABLE)
   INT_(COLUMN)
END_TABLE
```

.cpp:

```
ldap_array.SetTable(MYTABLE);
```

ldap_array.AddColumn(COLUMN, "column");

See examples/SQLApp for fairly complete SQL app skeleton.

Mirek
