Subject: SQLite crashes
Posted by Mindtraveller on Fri, 13 Jun 2008 23:04:06 GMT
View Forum Message <> Reply to Message

Recently I tried to use SQLite U++ plugin and met strange yet unavoidable error: after calling Open(), in the end of scope, sqlite library throws an exception. Let`s consider minimal U++/sqlite console application:

```
#include <Core/Core.h>
#include <plugin/sqlite3/Sqlite3.h>
using namespace Upp;

CONSOLE_APP_MAIN
{
 Sqlite3Session session;
 session.LogErrors(true);
 if (!session.Open("data"))
  Cout() << "Error opening SQL";
}//crash here!
```

Actual crash seems to happen in file /uppsrc/Sql/Sqls.h, line 219

Conditions: WinXP, latest SVN (rev. 297).

Unfortunately I`m not good at databases, so I can`t propose adequate solution on that problem. But I think any plugins should work "from the box" without resulting any crashes.

Any ideas on solving this?

Subject: Re: SQLite crashes
Posted by Mindtraveller on Sat, 14 Jun 2008 07:42:33 GMT
View Forum Message <> Reply to Message

My investigation continues.
Somehow it seems that this code works without exception:

```
#include <Core/Core.h>
#include <plugin/sqlite3/Sqlite3.h>
using namespace Upp;

CONSOLE_APP_MAIN
{
 Sqlite3Session session;
 session.LogErrors(true);
 if (!session.Open("data"))
  Cout() << "Error opening SQL";

 SQL = session;
}
```

Subject: Re: SQLite crashes
Posted by mirek on Tue, 17 Jun 2008 06:10:12 GMT
View Forum Message <> Reply to Message

A bug. QuickFix:

```
void Sqlite3Session::Close() {
 if (NULL != db) {
  int retval;
#ifndef flagNOAPPSQL
  if(SQL.IsOpen() && &SQL.GetSession() == this)
   SQL.Cancel();
#endif
```

Note that it is typical to assign SQL or use NOAPPSQL, that is why this bug was left unnoticed until now (thanks!).

Mirek

Subject: Re: SQLite crashes
Posted by Mindtraveller on Thu, 19 Jun 2008 20:04:19 GMT
View Forum Message <> Reply to Message

Thanx for the fix!

Also I`ve discovered something that could be a problem. It may be SQLite specific - I don`t know for shure at the moment.
When I try to Fetch() after Execute() of some SQL statement which contained error, an exception is thrown. And it seems like it doesn`t caught even with catch(...). Maybe it is C-style exception or something like that. This makes application potentially risky and unstable.
I had no free time to analyze this feature excessively but that certainly could be a problem.

Subject: Re: SQLite crashes
Posted by unodgs on Thu, 19 Jun 2008 21:01:27 GMT
View Forum Message <> Reply to Message

The best would be if you post here a testcase.

Subject: Re: SQLite crashes
Posted by Mindtraveller on Wed, 23 Jul 2008 16:31:59 GMT
View Forum Message <> Reply to Message

OK, I switched to MySQL from SQLite and have the same error.

Usually one checks Execute() before doing Fetch:

```
MySqlSession session;
String q = ".....";
if (!session.Connect(...))
 return;
Sql sql(session);
if (sql.Execute(q))
 while(sql.Fetch())
 {
  for (int i=0; i<sql.GetColumns(); ++i)
   ShowDBMessage(sql.GetColumnInfo(i).name + " : " + (String)sql[i]);
 }
```

But if you forget to check Execute() and start fetching, the Fetch() call will throw an exception:

```
sql.Execute(q);
while(sql.Fetch()) //<-- throws an exception if Execute() failed
{
 for (int i=0; i<sql.GetColumns(); ++i)
  ShowDBMessage(sql.GetColumnInfo(i).name + " : " + (String)sql[i]);
}
```

I think Fetch() should not throw an exception, instead it should return no results if latest Execute() failed.

---

Subject: Re: SQLite crashes
Posted by mirek on Wed, 23 Jul 2008 17:04:40 GMT
View Forum Message <> Reply to Message

Mindtraveller wrote on Wed, 23 July 2008 12:31OK, I switched to MySQL from SQLite and have the same error.

Usually one checks Execute() before doing Fetch:

```
MySqlSession session;
String q = ".....";
if (!session.Connect(...))
 return;
Sql sql(session);
if (sql.Execute(q))
 while(sql.Fetch())
 {
  for (int i=0; i<sql.GetColumns(); ++i)
   ShowDBMessage(sql.GetColumnInfo(i).name + " : " + (String)sql[i]);
 }
```

But if you forget to check Execute() and start fetching, the Fetch() call will throw an exception:

```
sql.Execute(q);
while(sql.Fetch()) //<-- throws an exception if Execute() failed
```

```
{
 for (int i=0; i<sql.GetColumns(); ++i)
  ShowDBMessage(sql.GetColumnInfo(i).name + " : " + (String)sql[i]);
}
```

I think Fetch() should not throw an exception, instead it should return no results if latest Execute() failed.

Ah, this one. It is there since beginning and U++ being started as database interface library, it is perhaps the right thing to do.

Imagine impact if Fetch would actually return false. In that case, a lot of code would do wrong things and pretend everything is OK....

Mirek

---