Subject: MySqlSession Close and Reconnect crashes Application
Posted by hoelblin on Wed, 25 Nov 2020 16:17:33 GMT
View Forum Message <> Reply to Message

Hi,

I have an Application with a Connection to MySql. When the Application is Idle for
a long Period of Time the Connection is Closed with session.Close.

After that session.IsOpen returns false.

When I try to reconnect with session.Reconnect() the App Crashes with a Null Pointer dereference.
It seems like in Sql.Cancel cn->cancel is called while Member cn == NULL.

This behavior is on TheIDE Version 15512 on Windows with CLANG, on TheIDE 13664 with minGW this works as
expected without crashing.

To reproduce the Crash i have taken the SQL_MySql reference Example and make some minor changes.

```
#include <MySql/MySql.h>

// debian: sudo apt-get install libmysqlclient-dev

using namespace Upp;

#define SCHEMADIALECT <MySql/MySqlSchema.h>
#define MODEL <mysqlCrash/schema.sch>
#include <Sql/sch_header.h>
#include <Sql/sch_source.h>
#include <Sql/sch_schema.h>

CONSOLE_APP_MAIN
{
 MySqlSession session;
 // edit the connection parameters if necessary
 if(session.Connect("test_user", "test", "test")) {
  Cout() << "Connected\n";
  SQL = session;

  SqlSchema sch(MY_SQL);
  All_Tables(sch);
  // create the table if necessary
  SqlPerformScript(sch.Upgrade());
  SqlPerformScript(sch.Attributes());
```

```
SQL.ClearError();


try {
 // insert some random data
 SQL & Insert(TEST_TABLE)(VALUE, Uuid::Create().ToString());
 // fetch some data
 Sql sql;
 sql * Select(ID, VALUE).From(TEST_TABLE)
                         .OrderBy(Descending(ID))
                         .Limit(5);
 while(sql.Fetch())
  Cout() << AsString(sql[0]) << ": " << AsString(sql[VALUE]) << "\n";
}
catch(SqlExc &ex) {
 Cerr() << "ERROR: " << ex << "\n";
 SetExitCode(1);
}

session.Close();

Sleep(2000);

if(session.IsOpen()) {
 Cout() << "Connection closed FAILED!\n";
} else {
 Cout() << "Connection closed!\n";
}

Sleep(2000);

Cout() << "Try open Connection again\n";

if(session.Reconnect()) { // CRASH !!!!!!!!!
 try {
  Sql sql;
  sql * Select(ID, VALUE).From(TEST_TABLE)
                          .OrderBy(Descending(ID))
                          .Limit(3);
  while(sql.Fetch())
   Cout() << AsString(sql[0]) << ": " << AsString(sql[VALUE]) << "\n";
 }
 catch(SqlExc &ex) {
  Cerr() << "ERROR: " << ex << "\n";
  SetExitCode(1);
 }
} else {
 Cout() << "ERROR: " << session.GetLastError() << "\n";
```

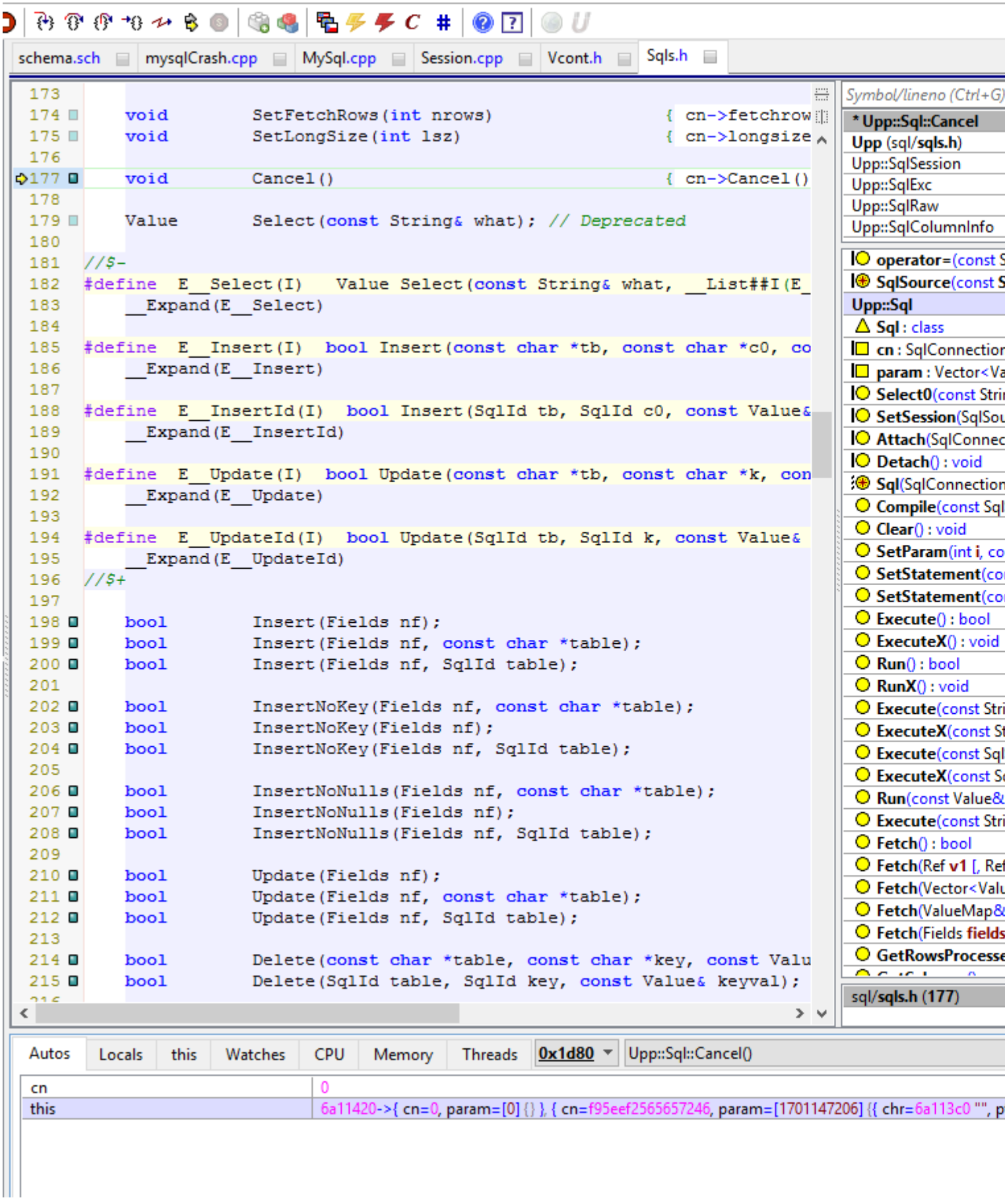
```
  }
 }
 else {
  Cerr() <<"ERROR: Unable to connect to database\n";
  SetExitCode(1);
 }
}
```

// Moderator: PLESE USE CODE TAG FOR THE NEXT TIME!

My Question is what is the right way to do a Reconnect after Close ?

File Attachments

1) mysqlCrash1.PNG, downloaded 1113 times


schema.sch
mysqlCrash.cpp
MySql.cpp
Session.cpp
Vcont.h
Sqls.h
void SetFetchRows(int nrows) { cn->fetchrow
void SetLongSize(int lsz) { cn->longsize
void Cancel() { cn->Cancel()
Value Select(const String& what); // Deprecated
//$-
#define E__Select(I) Value Select(const String& what, __List##I(E_
__Expand(E__Select)
#define E__Insert(I) bool Insert(const char *tb, const char *c0, co
__Expand(E__Insert)
#define E__InsertId(I) bool Insert(SqlId tb, SqlId c0, const Value&
__Expand(E__InsertId)
#define E__Update(I) bool Update(const char *tb, const char *k, con
__Expand(E__Update)
#define E__UpdateId(I) bool Update(SqlId tb, SqlId k, const Value&
__Expand(E__UpdateId)
//$+
bool Insert(Fields nf);
bool Insert(Fields nf, const char *table);
bool Insert(Fields nf, SqlId table);
bool InsertNoKey(Fields nf, const char *table);
bool InsertNoKey(Fields nf);
bool InsertNoKey(Fields nf, SqlId table);
bool InsertNoNulls(Fields nf, const char *table);
bool InsertNoNulls(Fields nf);
bool InsertNoNulls(Fields nf, SqlId table);
bool Update(Fields nf);
bool Update(Fields nf, const char *table);
bool Update(Fields nf, SqlId table);
bool Delete(const char *table, const char *key, const Valu
bool Delete(SqlId table, SqlId key, const Value& keyval);
Symbol/lineno (Ctrl+G)
* Upp::Sql::Cancel
Upp (sql/sqls.h)
Upp::SqlSession
Upp::SqlExc
Upp::SqlRaw
Upp::SqlColumnInfo
operator=(const S
SqlSource(const S
Upp::Sql
Sql : class
cn : SqlConnection
param : Vector<Va
Select0(const Stri
SetSession(SqlSou
Attach(SqlConnec
Detach() : void
Sql(SqlConnection
Compile(const Sql
Clear() : void
SetParam(int i, co
SetStatement(co
SetStatement(co
Execute() : bool
ExecuteX() : void
Run() : bool
RunX() : void
Execute(const Stri
ExecuteX(const St
Execute(const Sql
ExecuteX(const S
Run(const Value&
Execute(const Stri
Fetch() : bool
Fetch(Ref v1 [, Ref
Fetch(Vector<Valu
Fetch(ValueMap&
Fetch(Fields fields
GetRowsProcesse
sql/sqls.h (177)
Autos
Locals
this
Watches
CPU
Memory
Threads
0x1d80
Upp::Sql::Cancel()
cn
0
this
6a11420->{ cn=0, param=[0] {} }, { cn=f95eef2565657246, param=[1701147206] {{ chr=6a113c0 "", p

2) mysqlCrash2.PNG, downloaded 1101 times

```
Upp::Sql::Cancel() at C:\uppClang\upp\uppsrc\Sql\Sqls.h 177
Upp::SqlSession::SessionClose() at C:\uppClang\upp\uppsrc\Sql\Session.cpp 92
Upp::MySqlSession::Close() at C:\uppClang\upp\uppsrc\MySql\MySql.cpp 142
Upp::MySqlSession::Reconnect() at C:\uppClang\upp\uppsrc\MySql\MySql.cpp 79
ConsoleMainFn_() at C:\uppClang\upp\MyApps\mysqlCrash\mysqlCrash.cpp 51
Upp::AppExecute__(app=140002370) at C:\uppClang\upp\uppsrc\Core\App.cpp 442
main(argc=1, argv=18995d0->189f430 "C:\\uppClan") at C:\uppClang\upp\MyApps\mysqlCrash\mysqlCr
WinMainCRTStartup()
mainCRTStartup()
BaseThreadInitThunk()
RtlUserThreadStart()
```