
Subject: MySqlConnection 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
```

```
{  
    MySqlConnection 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: PLEASE 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 744 times

schema.sch | mysqlCrash.cpp | MySQL.cpp | Session.cpp | Vcont.h | Sqs.h

```

173
174 void SetFetchRows(int nrows) { cn->fetchrow
175 void SetLongSize(int lsz) { cn->longsize
176
177 void Cancel() { cn->Cancel()
178
179 Value Select(const String& what); // Deprecated
180
181 // $-
182 #define E_Select(I) Value Select(const String& what, __List##I(E
183     __Expand(E_Select)
184
185 #define E_Insert(I) bool Insert(const char *tb, const char *c0, co
186     __Expand(E_Insert)
187
188 #define E_InsertId(I) bool Insert(SqlId tb, SqlId c0, const Value&
189     __Expand(E_InsertId)
190
191 #define E_Update(I) bool Update(const char *tb, const char *k, con
192     __Expand(E_Update)
193
194 #define E_UpdateId(I) bool Update(SqlId tb, SqlId k, const Value&
195     __Expand(E_UpdateId)
196 // $+
197
198 bool Insert(Fields nf);
199 bool Insert(Fields nf, const char *table);
200 bool Insert(Fields nf, SqlId table);
201
202 bool InsertNoKey(Fields nf, const char *table);
203 bool InsertNoKey(Fields nf);
204 bool InsertNoKey(Fields nf, SqlId table);
205
206 bool InsertNoNulls(Fields nf, const char *table);
207 bool InsertNoNulls(Fields nf);
208 bool InsertNoNulls(Fields nf, SqlId table);
209
210 bool Update(Fields nf);
211 bool Update(Fields nf, const char *table);
212 bool Update(Fields nf, SqlId table);
213
214 bool Delete(const char *table, const char *key, const Valu
215 bool Delete(SqlId table, SqlId key, const Value& keyval);
216

```

Symbol/lineno (Ctrl+G)

- * Upp::Sql::Cancel
- Upp (sql/sqs.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/sqs.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 714 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()
```