
Subject: Re: Sql(SqlSession& session) problem (SOLVED)
Posted by [Sender Ghost](#) on Thu, 18 Jul 2013 16:37:45 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello, Danny.

pfsdanny wrote on Thu, 18 July 2013 15:24After rewrite the code, it works. So the following code that is ok in 4193 does not work in 5485.

Toggle source code

```
class c_db : public MySqlConnection{
public:
typedef c_db CLASSNAME;
String username;
String password;
String database;
String host;
String collate;
String lockId;
int port;
bool connect(bool prompt=true);
void disconnect();

void copyFm(c_db db); <<--- this line cause error !
void copyFm(String xhost, String xdatabase, String xusername, String xpassword, String xcollate,
int sport); <<-- change to this one and no problems !

bool getlock(String name);
bool releaselock();
c_db();
~c_db();
};
```

You are right, that after 4296 revision your previous code doesn't work, because of "Sql, SqlSession: operator= and copy constructors now private".

I think, you could rewrite your code as follows:

Toggle source code

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

using namespace Upp;

struct DBData {
String username;
String password;
String database;
String host;
```

```

String collate;
String lockId;
int port;
DBData() : port(3306) { }
DBData(const String& username, const String& password, const String& database, const String&
host,
    const String& collate, const String& lockId, int port = 3306) : username(username),
    password(password), database(database), host(host), collate(collate), lockId(lockId), port(port) {
}
};

```

```

class DBConnection {
protected:
    MySqlSession session;
    DBData data;
public:
    // DBConnection() { }
    DBConnection(const DBData& data) : data(data) { }
    ~DBConnection() { Disconnect(); ReleaseLock(); }
    MySqlSession& GetSession() { return session; }
    const DBData& GetData() { return data; }
    void SetData(const DBData& d) { data = d; }
    bool Connect(bool prompt = true);
    void Disconnect();
    bool GetLock(const String& name);
    bool ReleaseLock();
};
// TODO: implement
bool DBConnection::Connect(bool prompt)
{
    return true;
}
// TODO: implement
void DBConnection::Disconnect()
{
}
// TODO: implement
bool DBConnection::GetLock(const String& name)
{
    return true;
}
// TODO: implement
bool DBConnection::ReleaseLock()
{
    return true;
}

```

CONSOLE_APP_MAIN

```
{
  DBData data("username", "password", "database", "host", "collate", "lockId");
  DBConnection ctn(data);
  if (!ctn.Connect())
    return;
  SQL = ctn.GetSession();
  /* // Doing some query:
  Sql sql;
  sql * Select(SqlAll()).From(TABLE);
  while(sql.Fetch()) { }
  */
}
```

where you will have DBData structure to avoid the need to copy each variable to c_db::copyFm method and no need to assign MySqlConnection contents.
