Introduction to Mysql++ (c++ API for Mysql) on FreeBSD

(English Version)

Introduction: March 17, 2001

We will try to make a brief introduction to the `complex` C++ API for Mysql on FreeBSD. We are going to explain the installation procedure and introduce the API in light of three samples...

General Notes

Recent version of this document can be found on ;
http://www.enderunix.org/documents/mysql++.html .

-- Murat Balaban
murat@enderunix.org
17 Mart, 2001

Installation for Mysql++-1.7

Let me state in the very begining that, mysql++ assumes you have mysql client libraries installed. If not, go and install them first. If you have a standart mysql-version-client.tgz/ports installation, you don't need to worry about this, they are all in place...

We are going to install our API from FreeBSD ports collection:

```
[root@pathfinder examples]# cd /usr/ports/databases/mysql++
[root@pathfinder mysql++]# make install
```

FreeBSD will download the source for the API from one of the predefined sites, will compile and install it all for us.

Installation will copy the header files to /usr/local/include , libsqlplus.{a|so|so.1} libraries to /usr/local/lib directory.
However, the port in FreeBSD seems somehow broken. Two header files, which are quite necessary for clear compilation are missing. In fact, they are not copied to their usual places, so we need to copy them manually:

```
[root@pathfinder mysql++]# cp
/usr/ports/databases/mysql++/work/mysql++-1.7/sqlplusint/define_short
/usr/local/include/
[root@pathfinder mysql++]# cp
/usr/ports/databases/mysql++/work/mysql++-1.7/sqlplusint/defs /usr/local/include/
```

Sample 1: create table.cpp

Now that we have completed the installation, we can start accessing and playing with mysql via our c++ codes. Now let's have a look at out sample code:
(You can grab this example source code along with others from http://www.enderunix.org/documents/mysql++-samples.tgz .

Figure #1 (create_table.cpp)

```
//-----------------------------------starts here---------------------------------

#include <iostream>
#include <sqlplus.hh> // We need to supply this header file.
#define HOST "localhost" // so, where's your mysql server?
#define DB "enderunix" // and database name?
#define USERNAME "root" // a user granted access to the above database?
#define PASSWORD "" // enter the password for the above user. If there's no password, leave it as it
is...

int main () { // Here we go...

Connection connection (use_exceptions); // create an object `connection` from the Connection class.

// Our API can handle exceptions, so let's utulize this

try { // All the way main() is nothing but a try block

connection.connect("", HOST, USERNAME, PASSWORD); // connecting....

try { // we try to select database, if some exception is returned, we'll catch it and create the
database.

connection.select_db(DB); // select database

} catch (BadQuery er ) { // if returned an exception, catch it

connection.create_db(DB); // so, no database? create it first then.

connection.select_db(DB);

}

Query query = connection.query(); // an object from Query class which is in fact bound to connection
object.

// That is the query. see the overloaded << operator. We can do many things with it.

query << "CREATE TABLE fihrist (id INT not null auto_increment, name TEXT not null , surname TEXT
not null , phone TEXT not null , email TEXT not null , web TEXT not null , PRIMARY KEY (id), INDEX
(id), UNIQUE (id))";

try {

query.execute(); // execute it!

} catch (BadQuery er) { // catch the exception

cerr << "Error: " << er.error << endl; // Print the error on the screen

return -1;

}

} catch (BadQuery er) { // Print the error on the screen

cerr << "Error: " << er.error << endl;

return -1;

}
```

```
return 0;

}

//-----------------------------------------------ends
here---------------------------------------------------------------------
```

Ok, but how to compile?:

```
[root@pathfinder examples]# c++ -D_FIX_FOR_BSD_ -I/usr/local/include/mysql -L/usr/local/lib
-lsqlplus create_table.cpp -o create_table
/usr/local/lib/mysql/libmysqlclient.so.6: warning: tempnam() possibly used unsafely; consider using
mkstemp()
[root@pathfinder examples]#
```

Here, via -I flag, we expand our PATH to /usr/local/include/mysql where mysql header files are located, and via -L flag, we specify the location of the library, link -lsqlplus. There comes a binary called create_table. And, when we run:

```
[root@pathfinder examples]# ./create_table
[root@pathfinder examples]#
```

We have created our database and table without any error, but let's check:

```
[root@pathfinder examples]# mysql

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 33 to server version: 3.22.32

Type 'help' for help.

mysql> show databases;

+-----------+

| Database |

+-----------+

| enderunix |

| murat |

| mysql |

| test |

+-----------+

4 rows in set (0.01 sec)

mysql> use enderunix;

Reading table information for completion of table and column names

You can turn off this feature to get a quicker startup with -A

Database changed

mysql> describe fihrist;

+---------+---------+------+-----+---------+----------------+

| Field | Type | Null | Key | Default | Extra |

+---------+---------+------+-----+---------+----------------+

| id | int(11) | | PRI | 0 | auto_increment |
```

```
| id | int(11) | | PRI | 0 | auto_increment |
| name | text | | | NULL | |
| surname | text | | | NULL | |
| phone | text | | | NULL | |
| email | text | | | NULL | |
| web | text | | | NULL | |
+---------+---------+------+-----+---------+----------------+
6 rows in set (0.01 sec)
mysql>
```

As you can see, our database and table have been created successfully.

Lets write some code to insert data to the table.

Sample #2: insert_data.cpp

Figure #2 (insert_data.cpp)

```
//-------------------------------------------------------starts
here----------------------------------------------------------------
#include <iostream>
#include <sqlplus.hh>
#include <string>
#define HOST "localhost" // so, where's your mysql server?
#define DB "enderunix" // and database name?
#define USERNAME "root" // a user granted access to the above database?
#define PASSWORD "" // enter the password for the above user. If there's no password, leave it as it
is...

int main () {

struct Person { // perhaps useless, but just for future development.

int id;

string name;

string surname;

string phone;

string email;

string web;

};

Person person;



cout << "Please enter name\n"; // we get the information from the user.

cin >> person.name;

cout << "Please enter surname\n";
```

```
cin >> person.surname;

cout << "Please enter phone number\n";

cin >> person.phone;

cout << "Please enter email address\n";

cin >> person.email;

cout << "Please enter web address\n";

cin >> person.web;




Connection connection (use_exceptions);

try {

connection.connect("", HOST, USERNAME, PASSWORD);

connection.select_db(DB);

Query query = connection.query();

//Difference from the first sample: none but the query itself.



query << "INSERT INTO fihrist " << "(id, name, surname, phone, email, web) VALUES (\"\",\"" <<
person.name << "\", \""

<< person.surname << "\", \"" << person.phone << "\", \"" << person.email << "\", \"" <<
person.web << "\" )";

try {

query.execute();

} catch (BadQuery er) {

cerr << "Error: " << er.error << endl;

return -1;

}

} catch (BadQuery er) {

cerr << "Error: " << er.error << endl;

return -1;

}



return 0;

}

//---------------------------------------------------------ends
here---------------------------------------------------------
```

compile:

```
[root@pathfinder examples]# c++ -D_FIX_FOR_BSD_ -I/usr/local/include/mysql -L/usr/local/lib
-lsglplus insert_data.cpp -o insert_data
```

```
-lsqlplus insert_data.cpp -o insert_data
/usr/local/lib/mysql/libmysqlclient.so.6: warning: tempnam() possibly used unsafely; consider using
mkstemp()
[root@pathfinder examples]#
```

run:

```
[root@pathfinder examples]# ./insert_data
Please enter name
Murat
Please enter surname
Balaban
Please enter phone number
+905325566557
Please enter email address
murat@enderunix.org
Please enter web address
http://www.enderunix.org
[root@pathfinder examples]# ./insert_data
Please enter name
Ismail
Please enter surname
Yenigul
Please enter phone number
+905552545511
Please enter email address
ismail@enderunix.org
Please enter web address
http://www.enderunix.org/~yenigul/
[root@pathfinder examples]#
```

All information is successfully inserted to mysql. Lets verify:

```
[root@pathfinder examples]# mysql

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 36 to server version: 3.22.32

Type 'help' for help.

mysql> use enderunix

Reading table information for completion of table and column names

You can turn off this feature to get a quicker startup with -A

Database changed

mysql> select * from fihrist;

+----+--------+---------+--------------+---------------------+----------------------------------+

| id | name | surname | phone | email | web |

+----+--------+---------+--------------+---------------------+----------------------------------+

| 1 | Murat | Balaban | +905325566557 | murat@enderunix.org | http://www.enderunix.org |

| 2 | Ismail | Yenigul | +905552545511 | ismail@enderunix.org | http://www.enderunix.org/~yenigul/
|

+----+--------+---------+--------------+---------------------+----------------------------------+

2 rows in set (0.00 sec)

mysql>
```

As we can see, the data is in the table.

Now, lets make some more fun, and write code to query some given information:

Sample #3: select_data.cpp

Figure #3 (select_data.cpp)

```
//-----------------------------------------------starts
here-----------------------------------------------------------------------------
#include <iostream>
#include <sqlplus.hh>
#include <iomanip>
#include <string>
#define HOST "localhost" // so, where's your mysql server?
#define DB "enderunix" // and database name?
#define USERNAME "root" // a user granted access to the above database?
#define PASSWORD "" // enter the password for the above user. If there's no password, leave it as it
is...

 int main () {

 struct Person {

int id;

string name;

string surname;

string phone;

string email;

string web;

};

Person person;


cout << "Please enter name\n"; // get the name to bu queried.

cin >> person.name;




Connection connection (use_exceptions);

try {

connection.connect("", HOST, USERNAME, PASSWORD);

connection.select_db(DB);

Query query = connection.query();

// Querymiz yapiliyor:


query << "SELECT * FROM fihrist WHERE name = \"" << person.name << "\"";
```

```
try {

Result result = query.store(); // query.store() executes query and stores it.

// and we create a result object of Result class which is bound to query object.

Row row; // this is for row[""]

Result::iterator i;

int count = 0;

for ( i = result.begin(); i != result.end() ; i++ ) { // loop till the end of result.

row = *i;

cout << "\nRecord #" << ++count << "\tID: " << row["id"] << endl;

cout.setf(ios::left);

cout << setw(10) << "Name" << row["name"] << "\n"

<< setw(10) << "Surname" << row["surname"] << "\n"

<< setw(10) << "E-Mail" << row["email"] << "\n"

<< setw(10) << "Phone" << row["phone"] << "\n"

<< setw(10) << "Web" << row["web"] << "\n";

}

cout << "\nTotally, " << result.size() << " records listed.\n\n";

} catch (BadQuery er) {

cerr << "Error: " << er.error << endl;

return -1;

}

} catch (BadQuery er) {

cerr << "Error: " << er.error << endl;

return -1;

}



catch (BadConversion er) {

cerr << "Error: Tried to convert \"" << er.data << "\" to a \""

<< er.type_name << "\"." << endl;

return -1;

}

return 0;

}

//---------------------------------------------------------------ends
here-------------------------------------------------------
```

```
...
```

Compile and run:

```
[root@pathfinder examples]# c++ -D_FIX_FOR_BSD_ -I/usr/local/include/mysql -L/usr/local/lib
-lsqlplus select_data.cpp -o select_data
/usr/local/lib/mysql/libmysqlclient.so.6: warning: tempnam() possibly used unsafely; consider using
mkstemp()
[root@pathfinder examples]#
[root@pathfinder examples]# ./select_data
Please enter name
murat

Record #1 ID: 1
Name Murat
Surname Balaban
E-Mail murat@enderunix.org
Phone +905325566557
Web http://www.enderunix.org

Totally, 1 records listed.

[root@pathfinder examples]#
```

That's it. Good Luck:)

References:

Mysql Manual:
http://www.mysql.com/documentation/index.html

Mysql++ Manual:
http://www.mysql.com/documentation/mysql++/index.html