



(online) = ISSN 2285 – 3642

ISSN-L = 2285 – 3642

*Journal of Economic Development, Environment and People*

Volume 3, Issue 2, 2014

URL: <http://jedep.spiruharet.ro>

e-mail: [office\\_jedep@spiruharet.ro](mailto:office_jedep@spiruharet.ro)

## Open source MySQL Browser for Open Innovation

Lecturer Radu Bucea-Manea-Tonis, PhD <sup>1</sup>

<sup>1</sup> [mysqlbrowser.codeplex.com](http://mysqlbrowser.codeplex.com)

**Abstract.** *Our purpose is to cross-compile MySQL driver source code for Linux on Windows architecture using a tool chain in order to build a neutral valid graphic interface on 32 bits. Once achieving this goal we could say that every possible Open source application can be built and run on Windows with maximum efficiency concerning costs and resource. This browser is an example of open innovation because its source code is free for anybody willing to develop new software apps for business and uses only Open source tools.*

**Keywords:** MySQL, GCC, MinGW, GTK+, Open innovation

**JEL Codes:** M15

Open source methodology is gaining popularity in the business processes owned by modern companies. The benefits derive from its free availability, low-cost implementation and rapid business value growth. Open source business model inspires small companies that have not enough funds to invest in technology. Even big companies like IBM base their development strategies on Open source in order to obtain feedback from every specialist around the World and to facilitate the process of open innovation. The growing success of this paradigm explains the author's decision to develop MySQL Browser.

### 1. MySQL API

The MySQL API[2] consists of wrapping classes to Native API where the most important data structure is `st_mysql_res` which points the current row and some metadata, and contains the following data fields:

Table 1. ST\_MYSQL\_RES Struct

|                              |                            |
|------------------------------|----------------------------|
| <code>my_ulonglong</code>    | <code>row_count</code>     |
| <code>MYSQL_FIELD *</code>   | <code>fields</code>        |
| <code>MYSQL_DATA *</code>    | <code>data</code>          |
| <code>MYSQL_ROWS *</code>    | <code>data_cursor</code>   |
| <code>unsigned long *</code> | <code>lengths</code>       |
| <code>MYSQL *</code>         | <code>handle</code>        |
| <code>MEM_ROOT</code>        | <code>field_alloc</code>   |
| <code>unsigned int</code>    | <code>field_count</code>   |
| <code>unsigned int</code>    | <code>current_field</code> |
| <code>MYSQL_ROW</code>       | <code>row</code>           |



(online) = ISSN 2285 – 3642

ISSN-L = 2285 – 3642

*Journal of Economic Development, Environment and People*

Volume 3, Issue 2, 2014

URL: <http://jedep.spiruharet.ro>

e-mail: [office\\_jedep@spiruharet.ro](mailto:office_jedep@spiruharet.ro)

|                                 |                            |
|---------------------------------|----------------------------|
| MYSQL_ROW                       | current_row                |
| my_bool                         | eof                        |
| my_bool                         | unbuffered_fetch_cancelled |
| const struct st_mysql_methods * | methods                    |

Accessing rows is done iteratively, every row being linked with the next one through the following **st\_mysql\_rows** structure:

Table 2. ST\_MYSQL\_ROWS Struct

|                 |        |
|-----------------|--------|
| st_mysql_rows * | next   |
| MYSQL_ROW       | data   |
| unsigned long   | length |

The standard method to get records from database is calling the **next()** method of the RecordSet object, like in the following example:

```
res = stmt->executeQuery(str);
if(res->next())
{
if(strncmp("char",res->getString(1).c_str(),4)==0)
    TypeArr[i]=G_TYPE_STRING;
...
}
```

This method translates itself into the following instruction: **row = result->fetch\_row()**, where **result** is a **NativeResultSetWrapper** instance and row has the **MYSQL\_ROW** type, structure implemented as an array of counted byte strings.

After calling the **executeQuery()** method upon a **NativeStatementWrapper** object, the **get\_resultset()** method is called and assigns the **use\_result()/store\_result()** return value - called upon a **NativeConnectionWrapper** proxy object - to a **result** instance. **store\_result()** translates into a lower level method to obtain a **Recordset** by executing the following line: **::st\_mysql\_res \* raw= api->store\_result(mysql)**, where the **st\_mysql\_res** object is initialized.

## 2. Building libmysqlcppconn library from scratch

After install, mingw toolchain[3] will contain GCC compiler, linker and make tools. CMake[4] is launched to generate a valid Makefile, after extracting MySQL driver archive into a folder. It will ask for



(online) = ISSN 2285 – 3642

ISSN-L = 2285 – 3642

*Journal of Economic Development, Environment and People*

Volume 3, Issue 2, 2014

URL: <http://jedep.spiruharet.ro>

e-mail: [office\\_jedep@spiruharet.ro](mailto:office_jedep@spiruharet.ro)

valid paths to gcc and g++ compilers, linker and MySQL Server include and lib folders. For higher versions of mysql driver, it will ask for Boost library path but unfortunately this is poorly compatible with our shipped version of GCC compiler(4.6.2).

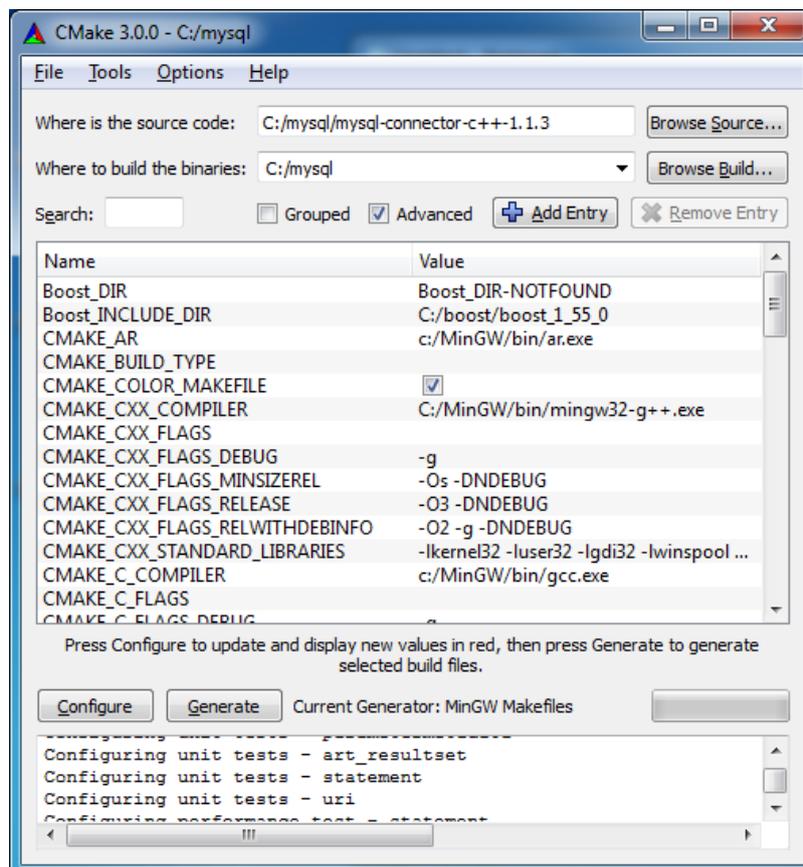


Fig. 1: CMake configure window

Make command is launched to build the project and raised errors will be treated appropriately. There are two types of errors encountered, one type concerning data definitions redundancy in header files (macro's, functions, and managed data types), and another, dealing with templates occurrences in extern C context. If the build succeeds, both static (libmysqlcppconn.dll.a) and dynamic (libmysqlcppconn.dll) libraries are created.



(online) = ISSN 2285 – 3642

ISSN-L = 2285 – 3642

*Journal of Economic Development, Environment and People*

Volume 3, Issue 2, 2014

URL: <http://jedep.spiruharet.ro>

e-mail: [office\\_jedep@spiruharet.ro](mailto:office_jedep@spiruharet.ro)

### 3. Creating a GUI for MySQL driver

GTK+[5] is a multi-platform toolkit for creating graphical user interfaces. Offering a complete set of widgets in combination with the Glade GUI builder, it provides an effective method of rapid application development. We create a `My_GUI` class with the methods described in the class diagram below:

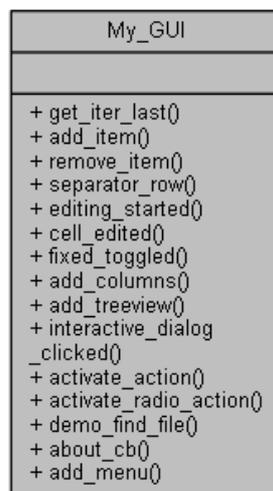


Fig. 2: My\_GUI Class diagram

After calling the `gtk_init()` method that initializes the library for use, sets up default signal handlers, and checks the arguments passed to the application on the command line, we create the main window, following the steps:

```
window = gtk_window_new (GTK_WINDOW_TOPLEVEL);
gtk_window_set_title (GTK_WINDOW (window), "MySQLBrowser");
gtk_container_set_border_width (GTK_CONTAINER (window), 5);
```

then add the window to the container and set its size:

```
gtk_container_add (GTK_CONTAINER (window), vbox);
gtk_window_set_default_size (GTK_WINDOW (window), 320, 200);
```

We call then the `gtk_main()` that runs the main loop. It will not return until `gtk_main_quit()` is called. For showing the results in a grid form we use a tree view `GtkWidget` that connects to the real data through a `GtkTreeModel` object initialized before by calling `create_items_model` method. After adding the menu bar and a couple of modal dialog windows, we get the next picture:

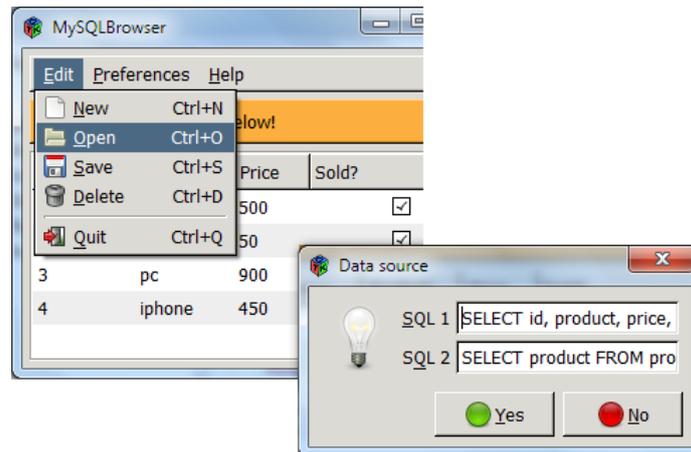


Fig. 3: MySQL Browser for Windows

The overall dependencies graph of the main classes (MySQL and MyGUI) and the environment types is shown below:

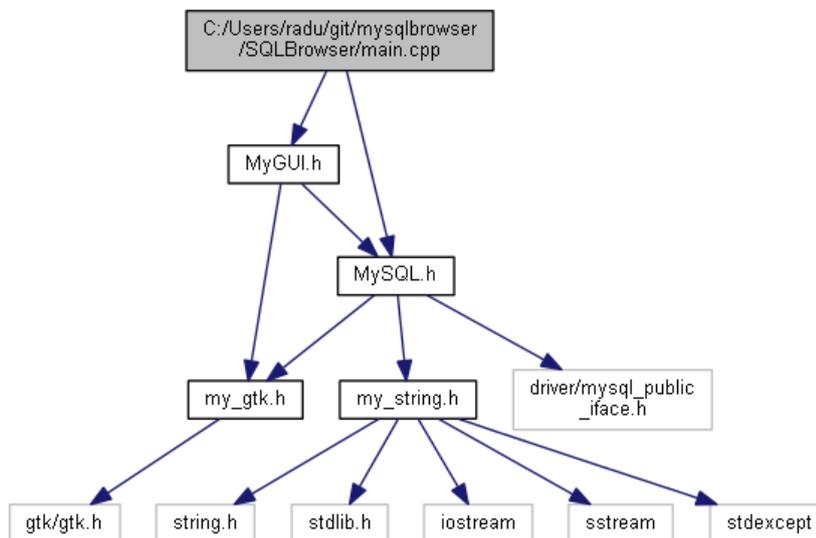


Fig. 4: Dependencies graph generated with Doxygen[7]

It is worth mentioning that GTK libs use UTF-8 international encoding standard, resulting back and forth conversions between back-end (MySQL Server) and front-end (GTK interface).

The system requirements for development are as follows:

- MySQL driver source code (ex. mysql-connector-c++-1.1.3.tar.gz)
- GTK+ project (ex. gtk+-bundle\_2.24.10-20120208\_win32.zip)
- mingw toolchain (ex. mingw-get-inst-20120426.exe)



(online) = ISSN 2285 – 3642

ISSN-L = 2285 – 3642

*Journal of Economic Development, Environment and People*

Volume 3, Issue 2, 2014

URL: <http://jedep.spiruharet.ro>

e-mail: [office\\_jedep@spiruharet.ro](mailto:office_jedep@spiruharet.ro)

- Boost C++libraries[6] (ex. boost\_1\_55\_0.zip)
- Makefile CMake builder(ex. cmake-2.8.11.2-win32-x86.exe)

Most of the .dll's are easy to find in MinGW\bin and GTK+\bin folders. Others are free to download (libmysql.dll) in case some have limited privileges on the computer (and have XAMPP installed).

#### 4. Conclusions

This paper proves the effectiveness of complementary Open source tools in developing useful apps to improve business value and competitiveness. The project have been implemented and shared on the Microsoft Codeplex Open source website [1] where is browsed and downloaded by numerous developers.

#### 5. References

- [1] Open source MySQL Browser for Windows, <http://mysqlbrowser.codeplex.com/>
- [2] C API Data Structures, <http://dev.mysql.com/doc/refman/5.0/en/c-api-data-structures.html>
- [3] Minimalist GNU for Windows(MinGW), <http://www.mingw.org/>
- [4] CMake cross-platform open-source build system, <http://www.cmake.org/>
- [5] GTK+ the GIMP Toolkit, <http://www.gtk.org/>
- [6] Boost C++libraries, <http://www.boost.org/>
- [7] Doxygen Generate documentation from source code, <http://www.stack.nl/~dimitri/doxygen/>