Subject: What do I need to download / configure to use MySql with U++?
Posted by lectus on Wed, 12 Dec 2012 01:01:08 GMT
View Forum Message <> Reply to Message

Do I need the official C++ connector too? Or just the MySql Server?

Thanks

Subject: Re: What do I need to download / configure to use MySql with U++?
Posted by zsolt on Wed, 12 Dec 2012 16:01:42 GMT
View Forum Message <> Reply to Message

I link my exe against MySQL Connector C.

Subject: Re: What do I need to download / configure to use MySql with U++?
Posted by BioBytes on Wed, 16 Jan 2013 20:12:29 GMT
View Forum Message <> Reply to Message

Hi Lectus and Zsolt,

Sorry for late reply.

I connect directly to MySql using server include, bin and lib files.

See native connection example:

Using native connection to MySql
1. Prologue

There are two ways to connect to MySql database using U++ - either through the ODBC layer or directly using MySql libray. Using ODBC is the easier way. The direct connection is more tricky but is more reliable when database is very large and has a lot of records.

2. Database Schema

```
TABLE_(test_table)
 INT_(id)PRIMARY_KEY AUTO_INCREMENT
 STRING_(value,50)
END_TABLE
```

3. Example code

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

using namespace Upp;
```

```
#define SCHEMADIALECT <MySql/MySqlSchema.h>
#define MODEL <NativeMySql/NativeMySql.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("root",NULL,"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 & Select(id,value).From(test_table)
                         .OrderBy(Descending(id))
                         .Limit(5);
   while(SQL.Fetch())
    Cout() << AsString(SQL[0]) << ": " << AsString(SQL[1]) << "\n";
  }
  catch(SqlExc &ex) {
   Cerr() << "ERROR: " << ex << "\n";
   SetExitCode(1);
  }
 } else {
  Cerr() <<"ERROR: Unable to connect to database\n";
  SetExitCode(1);
 }
 SetExitCode(0);
}
```

4. Build Method setting

You have to set correct paths in you build method. In Build methods dialog (Setup > Build methods...), choose your build method (e.g. MSC9) from the list on the left and then add the MySQL server installation paths into the three tabs on the right side of the dialog.

On PATH tab add C:\Program Files\MySQL\MySQL Server 5.5\bin

-

On INCLUDE tab add C:\Program Files\MySQL\MySQL Server 5.5\include

-

And on LIB tab add C:\Program Files\MySQL\MySQL Server 5.5\lib

-

Note that the paths might be slightly different, depending on you system setup and the version of MySQL server installed.

5. Main package configuration

This is a crucial step. To link correctly with the MySql library, you must set the ".MYSQLDLL" build flag.

-

Note: Multithreading is not required, even though mysqlclient.lib is a multithreaded library (according to information shown in MySql website).

Screenshots are available in Theide help topics (see MySql: Native connection).

Hope this could be helpful

Regards

Biobytes
