
Subject: MySql and upp 2017r1

Posted by [Melek](#) on Thu, 26 Jan 2017 01:24:11 GMT

[View Forum Message](#) <> [Reply to Message](#)

```
C:/Ultimatepp/bin/mingw64/32/bin/./lib/gcc/i686-w64-mingw32
/6.2.0/../../../../i686-w64-mingw32/bin/ld.exe: cannot find -lmysql
collect2.exe: error: ld returned 1 exit status
:( :( :(
```

Subject: Re: MySql and upp 2017r1

Posted by [mr_ped](#) on Tue, 31 Jan 2017 02:42:36 GMT

[View Forum Message](#) <> [Reply to Message](#)

Did you follow this part of documentation?

[http://www.ultimatepp.org/srcdoc\\$MySql\\$NativeMySql\\$en-us.htm](http://www.ultimatepp.org/srcdoc$MySql$NativeMySql$en-us.htm) I

Can you verify you have your include/library paths set to your installation of mysql development library?

(it's not part of upp package IIRC)

Subject: Re: MySql and upp 2017r1

Posted by [germax](#) on Sun, 30 Apr 2017 18:07:26 GMT

[View Forum Message](#) <> [Reply to Message](#)

Since this is the latest rendition of the all popular mingw vs mysql error...
I'll reuse this thread.

Exact same error as OP

here are my settings (as closely taken from the SQL_MySql reference - example I am trying to get running)

executable path is set to the mysql-bin directory of course (just not room to patch another snippet into that screenshot ;))

Well...
Anyone?

File Attachments

1) [mingw-mysql-error.png](#), downloaded 628 times

Subject: Re: MySql and upp 2017r1
Posted by [germax](#) on Mon, 01 May 2017 14:02:47 GMT
[View Forum Message](#) <> [Reply to Message](#)

So, just as I expected ;)

Anyways;

for those of you that have the exact same Problem..

the mysql - lib folder is missing a file libmysql.a
that you must create yourself I'm afraid
(dl'ed files off the internet won't work well, since they must exactly match the mysql version installed)

Fear not it's not difficult at all:
five simple steps:

Copy your libmysql.dll to your MinGW bin directory Open a console at said directory Run
pexports libmysql.dll > libmysql.def * Run dlltool -k --input-def libmysql.def --dllname libmysql.dll
--output-lib libmysql.a * Copy libmysql.a back to your mysql server's lib folder *

You are done..

except for when you're encountered issues on the way ;))
first: upp's mingw (at least mine) is not coming with pexports,
you can grab a MinGW standalone off sourceforge and use that instead,
or you could use dlltool itself to create the def file (hopefully to your likings ;))
like so:
dlltool --export-all-symbols --dllname libmysql.dll --output-def libmysql.def
or if you have an older minGW laying around that came with MinGW utils reimp or you're able to
trace that down
just use that on the libmysql.lib file (not the dll ;))
reimp -d libmysql.lib

Now that you got this, the next thing you might trip is making a non-zero sized libmysql.a file
I know I did.

Check your windows environment variables (path to be exact)

if you're like me it's not too short at all..

inspect it thoroughly!

Make sure your mingw-bin path is in there,

AND make sure no interfering path is listed before that (in my case it was an old symbian tool
chain (CSL ARM)))

if in doubt check all listed paths for mingw files.. and remove them for now (or permanently.. it's up
to you)

when your path variable is cleaned try again creating the libmysql.a file.

once the libmysql.a is of non zero size you can go on and use that

The last step (in case needed) [I didn't]
is rather trivial again... once your compiler throws errors referring to mysql..
like
mysql_fetch_row@4 or
mysql_set_character_set@8
or alike

open your def file and append the respective @# part to it
(change mysql_fetch_row to my_sql_fetch_row@4 and mysql_set_character_set to
mysql_set_character_set@8 etc.pp)
redo step 4 (make a new libmysql.a file with dlltool)
copy it over to your mysql library folder and you're finally through.

I hope this info helps a handfull of you.