

转 C/C++连接MySql数据库

2013年10月15日 18:45:50 阅读数：7934

一、VS2008工程设置工作

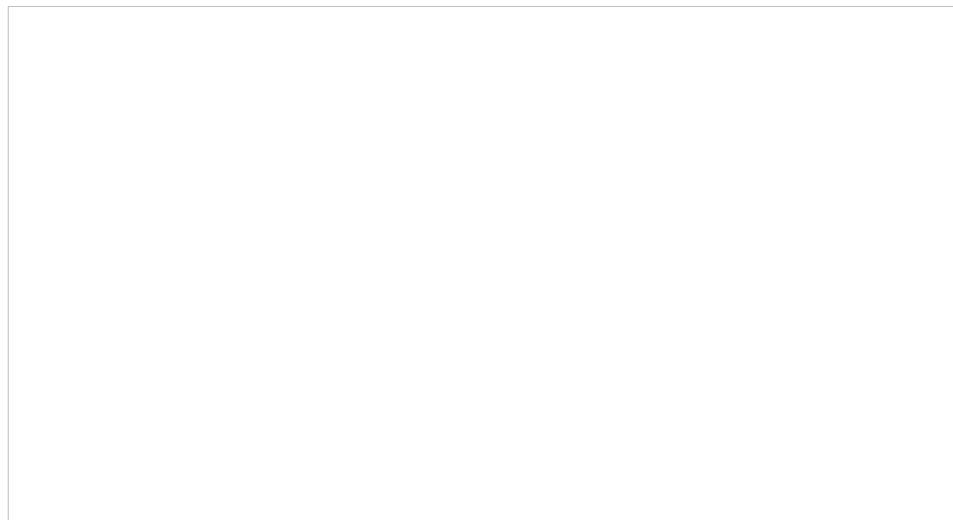
由于我们要使用Mysql的API，并且我们机子上肯定安装了Mysql数据库，所以我们要将工程的头文件路径指向Mysql安装目录的同文件mysql.h所在的位置，将连接库路径指向libmysql.lib所在的路径，在我的机子上，Mysql 的安装路径为：C:\Program Files\MySQL\MySQL Server 5.1

□

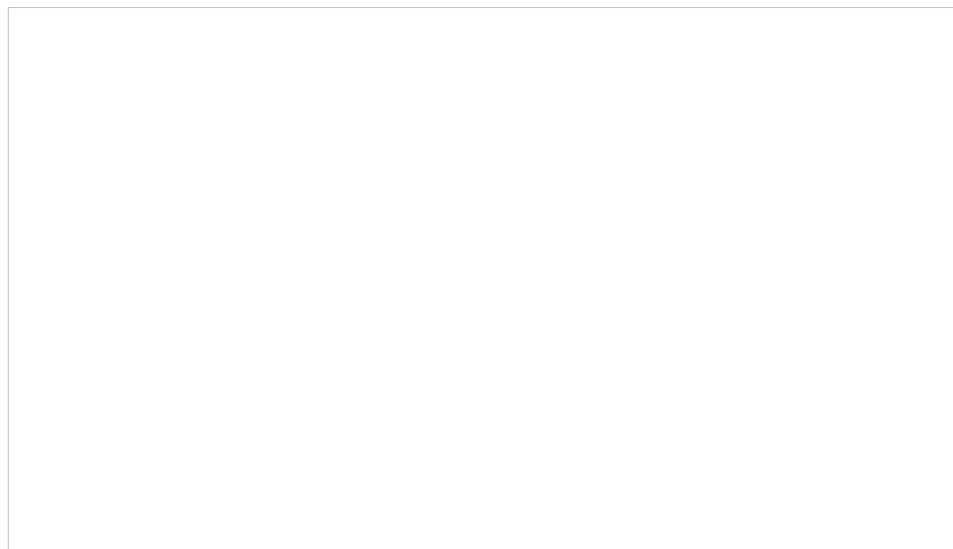
□

我们需要把VS2008的工程中的头文件路径和连接库路径指向上面的两个地方：

将项目属性页的C/C++->常规->附加包含目录指向：C:\Program Files\MySQL\MySQL Server 5.1\include



将项目属性页的链接器->常规->附加库目录指向:C:\Program Files\MySQL\MySQL Server 5.1\lib\opt



将链接器->输入->附加依赖项中添加libmysql.lib

这样就OK了。

二、连接Mysql和从MySQL中取出数据的API介绍

2.1 mysql_real_connect()

2.1.1 函数原型：

MYSQL *mysql_real_connect(MYSQL *mysql, const char *host, const char *user, const char *passwd, const char *db, unsigned int port, const char *unix_socket, unsigned int client_flag)

2.1.2 参数说明：

- 第一个参数应该是一个现存MYSQL结构的地址。在调用mysql_real_connect()之前,你必须调用mysql_init()初始化MYSQL结构。见下面的例子。
- host值可以是一个主机名或一个IP地址。如果host是NULL或字符串"localhost",假定是到本地主机的一个连接。如果OS支持套接字(Unix)或命名管道(Win32),使用他们而不是TCP/IP与服务器连接。
- user参数包含用户的MySQL登录ID。如果user是NULL,假定是当前用户。在Unix下,它是当前登录名。在Windows ODBC下,必须明确地指定当前用户名字。见16.4 怎样填写ODBC管理程序中各种域。
- passwd参数为用户包含口令。如果passwd是NULL,只有在user表中对于有一个空白口令字段的用户的条目将被检查一个匹配。这允许数据库主管设置MySQL权限,使用户获得不同的口令,取决于他们是否已经指定一个口令。注意:不要试图在调用mysql_real_connect()前加密口令;口令加密自动被客户API处理。
- db是数据库名。如果db不是NULL,连接将缺省数据库设置为这个值。
- 如果port不是0,值对于TCP/IP连接将用作端口号。注意host参数决定连接的类型。
- 如果unix_socket不是NULL,字符串指定套接字或应该被使用的命名管道。注意host参数决定连接的类型。
- client_flag值通常是0,但是在很特殊的情况下可以被设置为下列标志的组合：

标志名字 意味着的标志

CLIENT_FOUND_ROWS 返回找到的(匹配的)行数,不是受到影响的行数。

CLIENT_NO_SCHEMA 不允许db_name.tbl_name.col_name语法。这是为了ODBC;如果你使用该语法,导致语法分析器产生一个错误,它是在一些ODBC程序捕捉错误是有用的。

CLIENT_COMPRESS 使用压缩协议。

CLIENT_ODBC 客户是一个ODBC客户。这使mysqld变得对ODBC更友好。

2.1.3 返回值

如果连接成功，一个 MYSQL*连接句柄。如果连接失败，NULL。对一个成功的连接，返回值与第一个参数值相同，除非你传递NULL给该参数。

2.1.4 错误

CR_CONN_HOST_ERROR

不能连接MySQL服务器。

CR_CONNECTION_ERROR

不能连接本地MySQL服务器。

CR_IPSOCK_ERROR

不能创建一个IP套接字。

CR_OUT_OF_MEMORY

内存溢出。

CR_SOCKET_CREATE_ERROR

不能创建一个Unix套接字。

CR_UNKNOWN_HOST

不能找到主机名的IP地址。

CR_VERSION_ERROR

由于试图使用一个不同协议版本的一个客户库与一个服务器连接导致的一个协议失配。如果你使用一个非常老的客户库连接一个没有使用--old-protocol选项启动的新服务器，这就能发生。

CR_NAMEDPIPEOPEN_ERROR;

不能在 Win32 上创建一个命名管道。

CR_NAMEDPIPEWAIT_ERROR;

不能在 Win32 上等待一个命名管道。

CR_NAMEDPIPESETSTATE_ERROR;

不能在 Win32 上得到一个管道处理器。

2.2 mysql_select_db()

2.2.1 函数原型

int mysql_select_db(MYSQL *mysql, const char *db)

2.2.2 参数说明

使得由db指定的数据库成为 在由mysql指定的连接上的缺省(当前)数据库。在随后的查询中,这个数据库对于不包括一个显式的数据库指定符的表的引用是缺省数据库。

除非连接的用户能被认证允许使用数据库，否则mysql_select_db()失败。

2.2.3 返回值

成功，零。如果发生一个错误，非零。

2.2.4 错误

CR_COMMANDS_OUT_OF_SYNC

命令以一个不适当的次序被执行。

CR_SERVER_GONE_ERROR

MySQL服务器关闭了。

CR_SERVER_LOST

对服务器的连接在查询期间失去。

CR_UNKNOWN_ERROR

发生一个未知的错误。

2.3 mysql_real_query

2.3.1 函数原型

int mysql_real_query(MYSQL *mysql, const char *query, unsigned int length)

2.3.2 参数说明

执行由query指向的SQL查询，它应该是一个length个字节的字符串。查询必须由一个单个的SQL语句组成。你不应该在语句后增加一个终止的分号(“;”)或\g。

对于包含二进制数据的查询，你必须使用mysql_real_query()而不是mysql_query()，因为二进制代码数据可能包含“\0”字符，而且，mysql_real_query()比mysql_query()更快，因为它对查询字符串调用strlen()。

2.3.3 返回值

如果查询成功，零。如果发生一个错误，非零。

2.3.4 错误

CR_COMMANDS_OUT_OF_SYNC

命令以一个不适当的次序被执行。

CR_SERVER_GONE_ERROR

MySQL服务器关闭了。

CR_SERVER_LOST

对服务器的连接在查询期间失去。

CR_UNKNOWN_ERROR

发生一个未知的错误。

2.4 mysql_store_result

2.4.1 函数原型

MYSQL_RES *mysql_store_result(MYSQL *mysql)

2.4.2 返回值

A **MYSQL_RES** result structure with the results. **NULL** (0) if an error occurred.

2.5 mysql_fetch_row()

Description

Retrieves the next row of a result set. When used after **mysql_store_result()**, **mysql_fetch_row()** returns **NULL** when there are no more rows to retrieve. When used after **mysql_use_result()**, **mysql_fetch_row()** returns **NULL** when there are no more rows to retrieve or if an error occurred.

The number of values in the row is given by **mysql_num_fields(result)**. If **row** holds the return value from a call to **mysql_fetch_row()**, pointers to the values are accessed as **row[0]** to **row[mysql_num_fields(result)-1]**. **NULL** values in the row are indicated by **NULL** pointers.

The lengths of the field values in the row may be obtained by calling **mysql_fetch_lengths()**. Empty fields and fields containing **NULL** both have length 0; you can distinguish these by checking the pointer for the field value. If the pointer is **NULL**, the field is **NULL**; otherwise, the field is empty.

Return Values

A **MYSQL_ROW** structure for the next row. **NULL** if there are no more rows to retrieve or if an error occurred.

Errors

Note that error is not reset between calls to **mysql_fetch_row()**

- **CR_SERVER_LOST**

The connection to the server was lost during the query.

- CR_UNKNOWN_ERROR

An unknown error occurred.

参考资料：<http://dev.mysql.com/doc/refman/5.6/en/index.html>

三、利用Mysql库提供的API编写连接Mysql和从Mysql中取出数据的代码

```
[cpp]
1. #include <windows.h>
2. #include "stdio.h"
3. #include "winsock.h"
4. #include "mysql.h"
5.
6.
7. int main()
8. {
9.
10.  MYSQL * con; // = mysql_init((MYSQL*) 0);
11.  MYSQL_RES *res;
12.  MYSQL_ROW row;
13.
14.
15.  char tmp[400];
16.
17.  //database configuration
18.  char dbuser[30]="root";
19.  char dbpasswd[30]="apple";
20.  char dbip[30]="localhost";
21.  char dbname[50]="excel";
22.  char tablename[50]="test";
23.  char *query=NULL;
24.
25.
26.  int x;
27.  int y;
28.  int rt;//return value
29.  unsigned int t;
30.
31.  int count = 0;
32.
33.
34.  printf("input x,y\n");
35.  scanf("%d,%d",&x,&y);
36.  fflush(stdin);
37.  printf("input over\n");
38.  con = mysql_init((MYSQL*) 0);
39.
40.
41.  if ( con !=NULL && mysql_real_connect(con,dbip,dbuser,dbpasswd,dbname,3306/*TCP IP端口*/,NULL/*Unix Socket 连接类型*/,0/*运行成ODBC数据库标志*/) )
42.  {
43.      if (!mysql_select_db(con,dbname))
44.      {
45.          printf("Select successfully the database!\n");
46.
47.          con ->reconnect = 1;
48.
49.          query = "set names 'GBK'";
50.          //mysql_query(con,"set names 'GBK'");
51.
52.          rt=mysql_real_query(con,query,strlen(query));
53.          if (rt)
54.          {
55.              printf("Error making query: %s !!!\n",mysql_error(con));
56.          }
57.          else
58.          {
59.              printf("query %s succeed!\n",query);
60.          }
61.
62.      }
63.  }
64.
65.  else
66.  {
67.      MessageBoxA(NULL,"Unable to connect the database,check your configuration!", "",NULL);
68.
69.  }
70.
71.  //sprintf(tmp,"update %s set 商品='%s',卖出=%d,成交=%d,涨跌=%d,买进=%d,总量=%d,涨幅=%f,时间
72.  = '%s' where %s",tablename,goods,sold,deal,fluctuate,buy,total,delta,time,UpdateCon);
73.  sprintf(tmp,"insert into %s values(%s,%d,%d)",tablename,"null",x,y); //注意如何向具有自增字段的数据库中插入记录
74.  //MessageBoxA(NULL,tmp,tmp,MB_OK);
75.  //mysql_query(con,tmp);
76. }
```

```

76.     rt=mysql_real_query(con,tmp,strlen(tmp));
77.     if (rt)
78.     {
79.         printf("Error making query: %s !!!\n",mysql_error(con));
80.     }
81.     else
82.     {
83.         printf("%s executed!!!\n",tmp);
84.     }
85.
86.     sprintf(tmp,"select * from %s",tablename);
87.
88.     rt=mysql_real_query(con,tmp,strlen(tmp));
89.     if (rt)
90.     {
91.         printf("Error making query: %s !!!\n",mysql_error(con));
92.     }
93.     else
94.     {
95.         printf("%s executed!!!\n",tmp);
96.     }
97.
98.     res = mysql_store_result(con); //将结果保存在res结构体中
99.
100.    while(row = mysql_fetch_row(res))
101.    {
102.        /**
103.         * MYSQL_ROW STDCALL mysql_fetch_row(MYSQL_RES *result);
104.         * 检索行
105.         */
106.
107.        for(t=0;t<mysql_num_fields(res);t++)
108.        {
109.            printf("%s ",row[t]);
110.        }
111.        printf(".....\n");
112.        count ++;
113.    }
114.    printf("number of rows %d\n",count);
115.    printf("mysql_free_result...\n");
116.    mysql_free_result(res);
117.
118.    mysql_close(con);
119.    return 0;
120.
121. }

```

四、运行结果

□

五、数据库脚本

```

1. SET FOREIGN_KEY_CHECKS=0;
2.
3. -- Table structure for `test`
4.
5. DROP TABLE IF EXISTS `test`;
6. CREATE TABLE `test` (
7.   `x` bigint(4) NOT NULL AUTO_INCREMENT,
8.   `y` int(4) DEFAULT NULL,
9.   `z` int(4) DEFAULT NULL,
10.  PRIMARY KEY (`x`)
11. ) ENGINE=InnoDB AUTO_INCREMENT=118 DEFAULT CHARSET=latin1;
12.
13.
14. -- Records of test
15.
16. INSERT INTO `test` VALUES ('95', '12432', '4334');
17. INSERT INTO `test` VALUES ('96', '213', '321');
18. INSERT INTO `test` VALUES ('97', '213', '213');
19. INSERT INTO `test` VALUES ('98', '123', '231');
20. INSERT INTO `test` VALUES ('99', '321', '231');
21. INSERT INTO `test` VALUES ('100', '123', '32132');
22. INSERT INTO `test` VALUES ('101', '777', '32213');
23. INSERT INTO `test` VALUES ('102', '123', '213');
24. INSERT INTO `test` VALUES ('103', '21', '321');
25. INSERT INTO `test` VALUES ('104', '324', '432');
26. INSERT INTO `test` VALUES ('105', '132', '231');
27. INSERT INTO `test` VALUES ('106', '324', '342');
28. INSERT INTO `test` VALUES ('107', '23', '23');
29. INSERT INTO `test` VALUES ('108', '12', '21');
30. INSERT INTO `test` VALUES ('109', '231', '321');
31. INSERT INTO `test` VALUES ('110', '123', '231');
32. INSERT INTO `test` VALUES ('111', '123', '231');
33. INSERT INTO `test` VALUES ('112', '123', '123');
34. INSERT INTO `test` VALUES ('113', '312', '231');
35. INSERT INTO `test` VALUES ('114', '312', '321');
36. INSERT INTO `test` VALUES ('115', '23', '3');
37. INSERT INTO `test` VALUES ('116', '213', '312');
38. INSERT INTO `test` VALUES ('117', '2', '3');
39.
40.
41. -- Table structure for `xqdata`
42.
43. DROP TABLE IF EXISTS `xqdata`;
44. CREATE TABLE `xqdata` (
45.   `代码` varchar(20) NOT NULL DEFAULT '',
46.   `商品` varchar(20) CHARACTER SET utf8 DEFAULT NULL,
47.   `卖出` bigint(20) DEFAULT NULL,
48.   `成交` bigint(20) DEFAULT NULL,
49.   `涨跌` bigint(20) DEFAULT NULL,
50.   `买进` bigint(20) DEFAULT NULL,
51.   `总量` bigint(20) DEFAULT NULL,
52.   `涨幅` double DEFAULT NULL,
53.   `时间` time DEFAULT NULL,
54.   PRIMARY KEY (`代码`)
55. ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
56.
57.
58. -- Records of xqdata
59.
60. INSERT INTO `xqdata` VALUES ('FITX*1', '商品', '34', '43', '23', '34', '0', '1.4', '13:23:08');

```

原文地址：<http://www.cnblogs.com/justinzhang/archive/2011/09/23/2185963.html>

文章标签：[C++](#) [MySQL](#) [数据库](#) [连接](#)

个人分类：[纯编程](#)

此PDF由spygg生成,请尊重原作者版权!!!

我的邮箱:liushidc@163.com