[otl通过myodbc连接mysql](http://blog.csdn.net/genganpeng/article/details/7402229)

参考网址：

<http://blog.csdn.net/genganpeng/article/details/7402229>

---shuishanga

1. **Otl简介**

OTL 是 Oracle,Odbc andDB2-CLI Template Library 的缩写，是一个C++编译中操控关系数据库的模板库，它目前几乎支持所有的当前各种主流数据库，例如Oracle,MS SQL Server,Sybase,Informix,MySQL,DB2,Interbase/Firebird,PostgreSQL,SQLite,SAP/DB,TimesTen,MS ACCESS等等。OTL中直接操作Oracle主要是通过Oracle提供的OCI接口进行，进行操作DB2数据库则是通过CLI接口来进行，**至于MS的数据库和其它一些数据库，则OTL只提供了ODBC来操作的方式。**当然Oracle和DB2也可以由OTL间接使用ODBC的方式来进行操纵。

1. **Myodbc**

因为otl只能通过odbc连接mysql，所以首先需要安装myodbc。

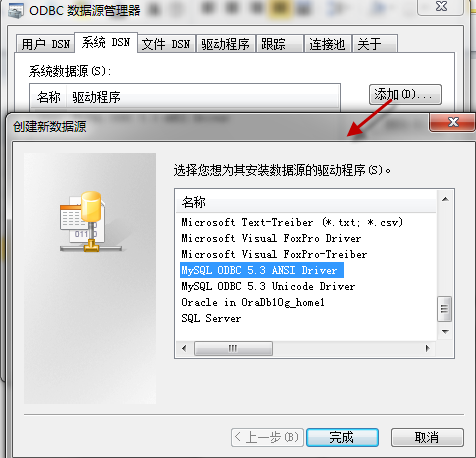
myodbc官网下载地址：<http://dev.mysql.com/downloads/connector/odbc/>



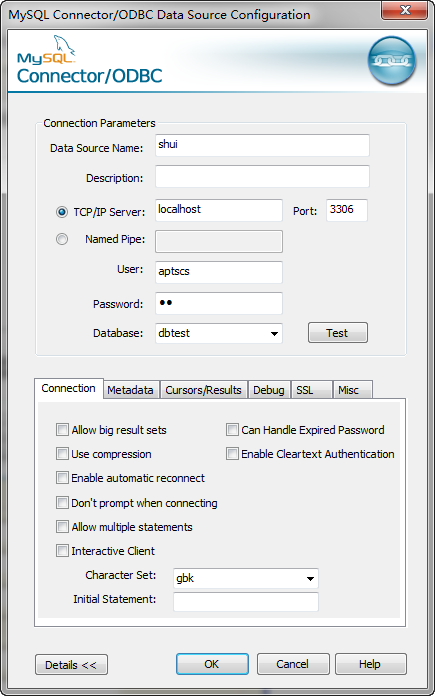
1. **添加数据源**

打开控制面板-》点击管理工具-》点击数据源，然后添加数据源

* 1. **创建数据源**



* 1. **进行配置**



1. **64位系统的配置**

在64位Windows系统中，默认“数据源(ODBC)”是64位的，包括“控制面板-》管理工具-》数据源 ”或在“运行”中直接运行“ODBCAD32”程序。

如果客户端是32位应用程序，仍然需要配置32位ODBC数据源，这时需要运行“C:\Windows\SysWOW64\odbcad32.exe”来启动“ODBC数据源管理器”，添加32位的ODBC数据源。

1. **测试**
   1. **测试代码**

注意引入头文件”otlv4.h”

#define OTL\_ODBC // Compile MySQL

#include "otlv4.h" // include the OTL 4 header file

#include <iostream>

using namespace std;

otl\_connect db; // connect object

void insert() // insert rows into table

{

otl\_stream wr

(50, // buffer size

"insert into test\_tab values(:f1<int>, :f2<char[7]>)", // SQL statement

db // connect object

);

char tmp[7];

for (int i=1; i<=20; ++i)

{

sprintf(tmp, "Name%d", i);

wr << i << tmp;

}

}

void select()

{

otl\_stream rd

(50, // buffer size

"select \* from test\_tab where f1>=:n1<int> and f1<=:n2<int>", // SELECT statement

db // connect object

);

rd << 8 << 16;

int f1;

char f2[7];

while (!rd.eof()) // while not end-of-data

{

rd >> f1 >> f2;

cout << "f1 = " << f1 << ", f2 = " << f2 << endl;

}

}

int main()

{

otl\_connect::otl\_initialize(); // initialize OTL environment

try

{

db.rlogon("aptscs/cs@shui"); // connect to Oracle

otl\_cursor::direct\_exec

(db,

"drop table test\_tab",

otl\_exception::disabled // disable OTL exceptions

); // drop table

otl\_cursor::direct\_exec

(

db,

"create table test\_tab(f1 int, f2 varchar(6))"

); // create table

insert(); // insert records into table

select(); // select records from table

}

catch (otl\_exception& e) // intercept OTL exceptions

{

cerr << "ERROR: code[" <<e.code<<"]\n";

cerr << e.msg << endl; // print out error message

cerr << e.stm\_text << endl; // print out SQL that caused the error

cerr << e.var\_info << endl; // print out the variable that caused the error

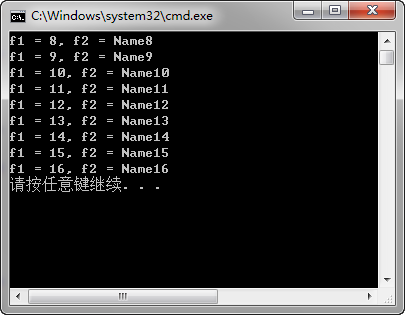
}

db.logoff(); // disconnect from Oracle

return 0;

}

* 1. **运行结果**：



* 1. **数据库中**

