



C# ODBC Database Programming

Author: Rajinder Yadav

Date: Aug 21, 2007

Web: <http://devmentor.org>

Email: rajinder@devmentor.org

Perquisites

I am going to assume you already have install MySQL on your system along with the ODBC driver from www.mysql.org

You should also know basic SQL and be able to create and populate your own database tables. A quick primer on SQL can be found at: www.w3schools.com/sql/default.asp

Creating a Database

Bring up MySQL Query Browser and type and execute the following commands.

a) First we will need to create a database named "store"

```
create database store
```

b) Next we need to set the database we will be working with

```
use store
```

c) We will create a database table with the following schema: **fruits(name, cost)**

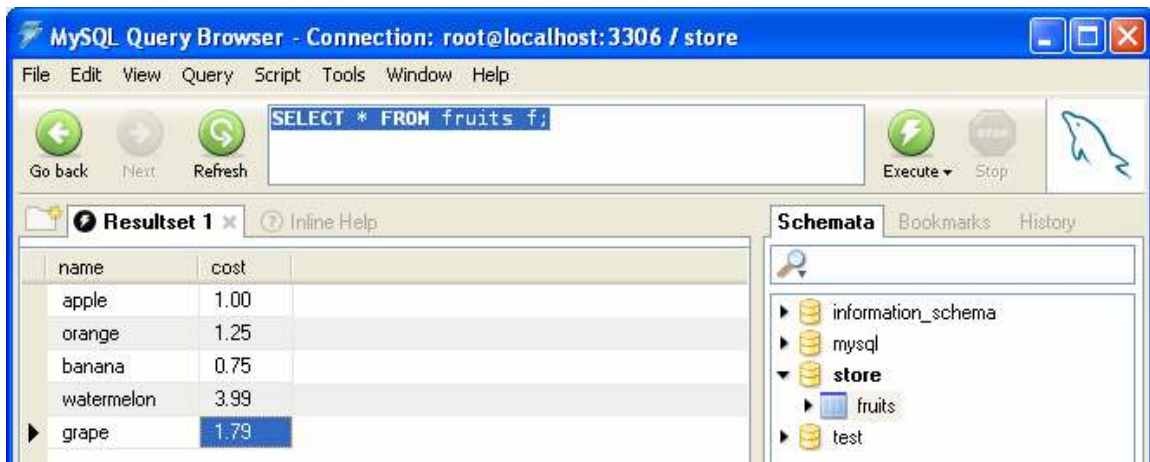
```
create table fruits (  
name varchar(20),  
cost decimal(5,2)  
);
```

d) Once a table has been created, it needs to be populated with data.

```
insert into fruits value  
("apple", "1.00"),  
("orange", "1.25"),  
("banana", "0.75"),  
("watermelon", "3.99"),  
("grape", "1.79");
```

We can perform a simple SQL **select** to make sure our table was properly populated.

```
SELECT * FROM fruits f;
```



1. Before you can start using the ODBC class definitions, you will need to include the right module.

```
using System.Data.Odbc; // ODBC definitions
```

2. A Connection is made using the **OdbcConnection** class and passing a connection string to the object being created.

```
string strConnect = "DSN=MySQL;UID=root;PWD=admin;DATABASE=store";  
OdbcConnection dbMySQL = new OdbcConnection(strConnect);
```

Understanding the ODBC connection string

Each ODBC name/value pair is separated with a ';' semicolon. There should be no spaces between the '=' equal sign and the name and value.

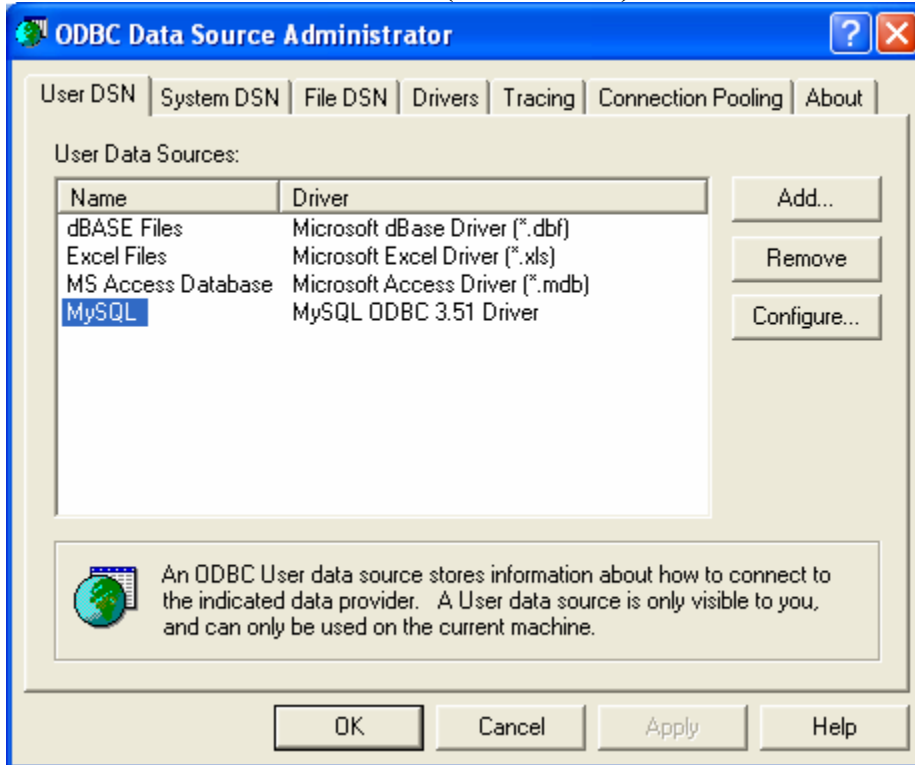
The "**DSN**" setting defines the name of the **ODBC Data Source** to connect to.

The "**UID**" ODBC connection string defines the "User ID"

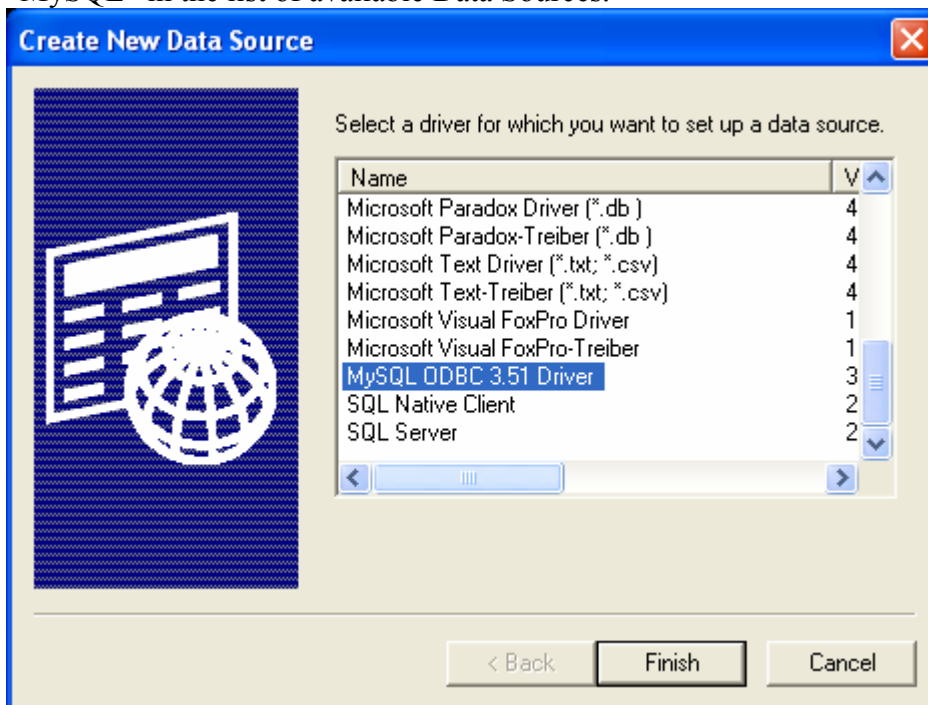
The "**PWD**" ODBC connection string define the "Password"

On the following page we will setup the ODBC Data Source, this is what our C# application will use when connecting to our database.

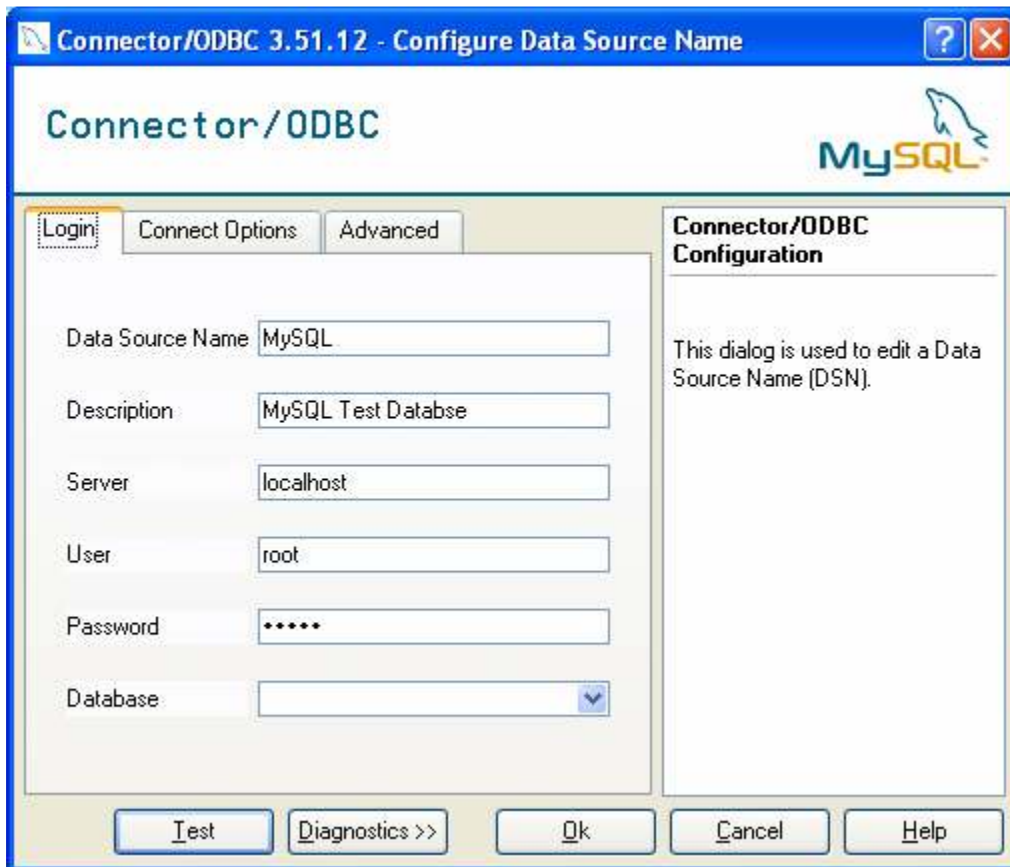
To bring up the ODBC Data Source dialog box, go to the **Control Panel** and select **Administrative Tools → ODBC (Data Source)**.



To create a new Data Source, you will need to click on the “Add” button and then find “MySQL” in the list of available Data Sources.



Click on the Finish button and then populate the fields are shown below, the "User" and "Password" to connect to a running instance of a MySQL database will be whatever you defined during the installation process of the MySQL database.



Before clicking on the "OK" button, you should click the "Test" button to verify that you are able to connect to the database. Make sure that the MySQL database service is running first. If everything is correct then you will see the following dialog box after clicking on the "Test" button.



3. Once we have a valid ODBC connection, we need to open it.

```
dbMySQL.Open();
```

If an error occur because we fail to make a connection to the database, an **OdbcException** exception will get thrown. So we will need to place our code inside a **try/catch** block.

```
string strConnect = "DSN=MySQL;UID=root;PWD=admin;DATABASE=test";
OdbcConnection dbMySQL = new OdbcConnection(strConnect);
try
{
    dbMySQL.Open();

    // do some database stuff

    dbMySQL.Close();
}
catch(OdbcException e)
{
    Console.WriteLine("Database Error\n\n{0}", e.ToString());
}
finally
{
    if(dbMySQL != null) dbMySQL.Close();
}
```

We need to make sure to close the connection once we are done, otherwise the connection will remain open even after the object goes out of scope.

4. After we have attained a open database connection, we need to get the command object in order to execute our SQL.

```
OdbcCommand sqlCommand = dbMySQL.CreateCommand();
```

5. Next we prepare the SQL we want to execute

```
sqlCommand.CommandText = "select * from fruits order by cost";
```

6. Since this SQL statement will be returning a result table we need to call the **ExecuteReader()** method on the **OdbcCommand** object "sqlCommand".

```
OdbcDataReader sqlReader = sql.ExecuteReader();
```

7. Once we have a data reader object, we can begin to read in the row value one at a time using the **Read()** method. This method will return **'true'** if there are more rows to be fetched, and **'false'** when there are no more rows existing in the result table.

8. The **OdbcCommand** object has several methods to retrieve the value from a column on the current row. We will use the **GetString(iCol)** method to fetch the value from column 'iCol' which is a zero based integer index.

9. Once we are done we need to make sure to close the **OdbcCommand** and **OdbcDataReader** objects. Since **OdbcCommand** does not have a close method, you will need to call it's **Dispose()** method.

Here is the complete code listing

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Data.Odbc; // ODBC definitions

namespace csMySQL
{
    class Program
    {
        static void Main(string[] args)
        {
            // prepare ODBC database connection
            string strConnect = "DSN=MySQL;UID=root;PWD=admin;DATABASE=store";

            OdbcConnection dbMySQL = new OdbcConnection(strConnect);
            try
            {
                // open database connection
                dbMySQL.Open();

                // get sql command object from odbc connection
                OdbcCommand sqlCommand = dbMySQL.CreateCommand();

                // execute sql
                sqlCommand.CommandText =
                    "select * from fruits order by cost";

                OdbcDataReader sqlReader = sqlCommand.ExecuteReader();

                // display result set column names
                Console.WriteLine("{0,-12} {1}", sqlReader.GetName(0),
                    sqlReader.GetName(1));

                Console.WriteLine("{0,-12} {1}", sqlReader.GetName(0),
                    sqlReader.GetName(1));

                Console.WriteLine("{0,-12} {1}", sqlReader.GetName(0),
                    sqlReader.GetName(1));

                // display row values
                while(sqlReader.Read())
                {
                    Console.WriteLine("{0,-12} {1}", sqlReader.GetString(0),
                        sqlReader.GetString(1));
                }
                sqlReader.Close();
                sqlCommand.Dispose();
                dbMySQL.Close();
            }
            catch(OdbcException e) {
                Console.WriteLine("Database Error\n\n{0}", e.ToString());
            }
            Finally {
                if(dbMySQL != null) dbMySQL.Close();
            }
        }
    }
}
```