MySQL and ODBC

By W.J. Gilmore

Table of Contents

Introduction

Although creating custom Web–based GUIs (Graphical User Interfaces) for your MySQL backend is a fairly common method for administrating database information, it is not without its problems. Slow Internet connections and cross–browser coding issues are just a few issues that could inhibit your client from updating their database information in the most timely and efficient way possible. Of course, problems such as these are largely out of your control, but don't try telling that to the client!

Well, if not a Web–based GUI, what then? The alternative is actually a quite interesting one. You may not be aware that MySQL is compatible with the ODBC (Open DataBase Connectivity) standard, and even offers its own ODBC drivers (known as MyODBC) free for download from the MySQL site (http://www.mysql.com). This is great, because MyODBC makes it possible to incorporate MySQL into quite a few interesting applications, including Microsoft Access. In this article, I'll demonstrate just how easy it is to install and configure MyODBC and subsequently use Microsoft Access to manage MySQL data.

Developer Shed

MyODBC Installation

The first step in the MyODBC installation process is to go to the MySQL Website and download the latest version of the MyODBC drivers. They are located at: http://www.mysql.com/downloads/api–myodbc.html. Incidentally, the drivers are available for the Windows 95/98/NT, UNIX, and Solaris platforms. However, since the demonstration will be devoted specifically to integration with MS Access, it is assumed that either the 95/98 or NT version of the drivers will be downloaded. To summarize the installation process:

1. Download the drivers from the URL given above.
2. Unzip it using your favorite ZIP package. WinZip (http://www.winzip.com) is particularly popular solution.
3. Click on *Setup.exe*. This will begin the installation process.
4. A Welcome prompt will appear. Click Continue.
5. A *Install Drivers* prompt will appear. Highlight the MySQL driver in the *Available ODBC Drivers* box and press OK.
6. Installation of the various files will begin. If you get an error referring to the MFC30.dll file, please read the Note found below. Otherwise, proceed to step 7.
7. A 'Data Sources' prompt will appear. Just press Close for now.
8. A *Setup Succeeded!* prompt appears. Congratulations, MyODBC is now installed.

Note: Installation problems relating to the MFC30.dll file are often encountered when installing MyODBC. If error messages appear while installing MyODBC, there are two options that I am aware of:
1. Restart Windows in Safe Mode (Press F8 during reboot). Once rebooted, perform the installation process as outlined above. Then reboot once more, this time normally.
2. Edit the odbc.inf file, deleting the line (only do this if using Windows 98/NT): "MFC" = 1, mfc30.dll,,,, 1995–07–11, ,,, ,,,,,, 322832,,,, 3.2.0.0,

The boring installation part is complete. In the next section, a sample project will be introduced, which will provide a practical basis for learning more about how MyODBC is used as the interface between MySQL and MS Access.

Developer Shed

The Sample Project

Suppose our client was the Wichita Widget Company. They are interested in making client information available over the Internet for the perusal of sales representatives located across the nation. This information will be maintained by the sales department at the central Kansas office. Rather than use a custom Web interface, they would like MS Access to be used to create a friendly, Windows–like application which can be used to maintain this data.

The first step in this process is the creation of the MySQL database and table(s) which will be used to store the data. The mysqladmin interface was used to create a new database named 'widget' (Privileged access is required to use mysqladmin):

```
%>mysqladmin create widget
```

Next, a new user and appropriate privileges were assigned to that user for the database 'widget':

```
%>mysql -u root -p mysql
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 17 to server version:
3.22.34-shareware-debug

Type 'help' for help.

mysql>GRANT select, insert, update, delete on widget.*
->to admin@123.456.789.0 IDENTIFIED BY ´secret´;
```

This creates not only the new user ´admin´ capable of connecting from IP '123.456.789.0' with the password ´secret', but it also grants that user selection, insertion, update and deletion privileges on the database ´widget´.

Note: For more information about MySQL and the privilege tables, check out the article MySQL Administration, published here on Devshed. Although direct table manipulation has since been deprecated by the GRANT/REVOKE syntax, this article will provide you with valuable information regarding the purpose of these tables.

Finally, log in as the user ´admin´ from the host IP ´123.456.789.0´, and create the table used to store client the information:

```
mysql> create table clients (
-> clientId mediumint not null auto_increment,
-> name char(30) not null,
-> state char(2) not null,
-> telephone char(10) not null,
-> email char(55) not null,
-> primary key(clientID) );
```

All necessary preparatory tasks involving MySQL are now complete. The next step is to update the available ODBC Data Sources with the new database information. Information regarding how this is accomplished is the subject of the next section.

Developer Shed

Updating the ODBC Data Sources

The client machine must be able to communicate with the MySQL database before it can be incorporated into an Access project. This is accomplished by adding it to the ODBC Data Source Administrator (ODBC DSA). The ODBC DSA can be accessed by clicking on the *ODBC Data Sources (32bit)* icon located in the "Control Panel" folder. This icon is shown in Figure 1–1.

Figure 1–1: The ODBC Data Sources (32bit) icon located in the Control Panel folder.

ODBC Data
Sources (32bit)

Clicking on the icon will result in the appearance of a window very similar to the one shown in Figure 1–2. Click on the *System DSN* tab found at the top of the window, and then click on the Add... button to add a new data source.

Figure 1–2: The ODBC Data Sources window
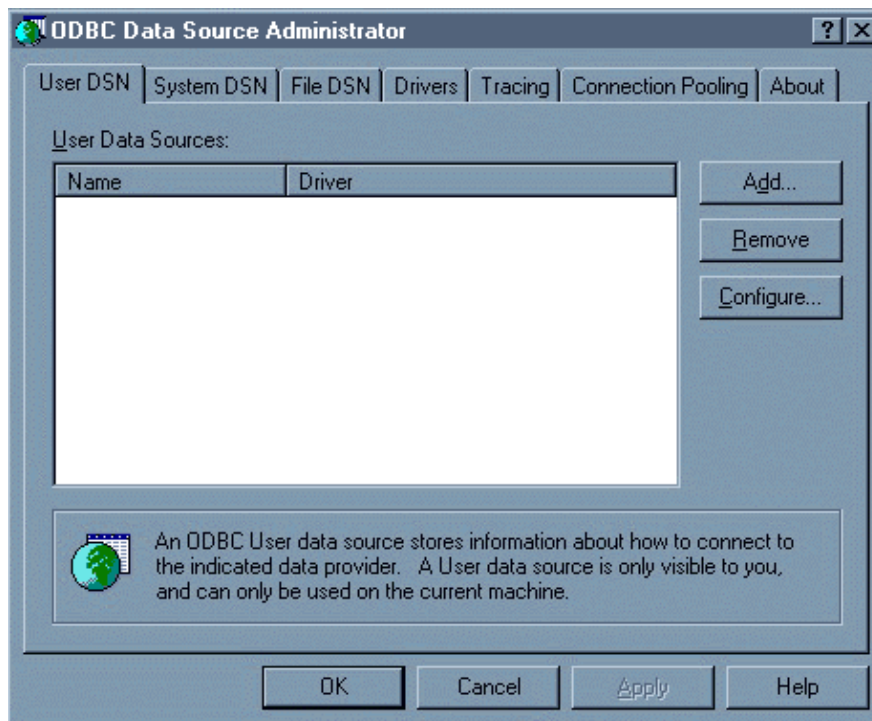
Clicking on the Add... button will produce a *Create New Data Source* prompt. The user is requested to select a driver for which a new data source should be set up. Scroll down and highlight the one that says *MySQL*. Click *Finish*. This in turn will produce a new window, which contains a series of textfields relevant to this new datasource. Although all textfields are in some way relevant to the datasource, there are a few which are particularly important:

Windows DSN Name

The Data Source Name (DSN) is the name used to represent the data source that is to be made available. I typically assign this the same name as the database, or choose a name that unmistakably refers to that database.

MySQL Host (name or IP)
This is the hostname assigned to connect to the MySQL database specified in the field "MySQL database name" (see below). Of course, this hostname should be the same as the host from which this client will connect, or the connection will fail.

MySQL database name This is the name of the database made available via ODBC. In the case of the database used in this tutorial, the name would be "widget".

User
This is the user name assigned to connect to the MySQL database specified in the field "MySQL database name".

Password
This is the password assigned to connect to the MySQL database specified in the field MySQL database name.

Fill each of these in which the correct information as it applies to the parameters you specified when setting up the widgets database. Click the OK button, and you will see that the Widget datasource has been added to the datasource list. It is now ready to be used by Access, or any other MyODBC–capable application.

In the next section, instruction regarding how to import the newly ODBC–accessible MySQL database into Access. Also, preliminary information regarding making concurrent updates to the MySQL database through an Access GUI is provided.

The MS Access GUI

Synchronizing the MySQL database to an Access GUI may seem like a rather difficult feat; On the contrary, however, it is actually a fairly simple procedure, carried out in just a few steps. The first steps is simply to fire up MS Access. Upon doing so, the user will be greeted with a screen very similar to the one shown in Figure 1–3.

Figure 1–3: Initial MS Access Window

Choose the radio button at the top of the window, labeled Blank Access Database, and press OK. A new window will appear, prompting for a location in which to save the new database. Name the new database widgets.mdb, and navigate to the location in where the database should be saved. Press "Create" to save the database.

The widgets database will be saved, and a new window will subsequently appear. This window will be very similar to the one shown in Figure 1–4.

Figure 1–4: New database window
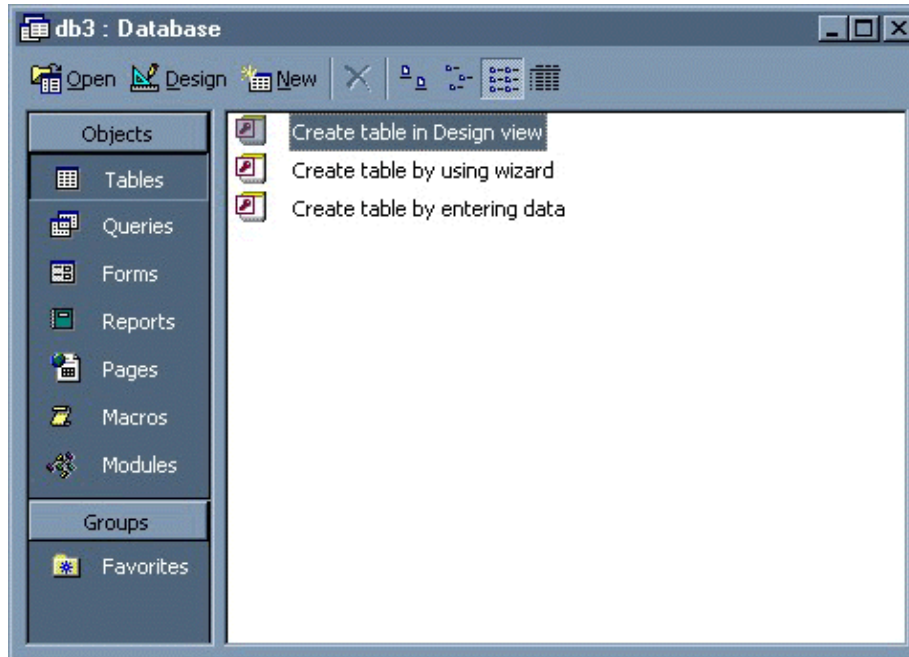
At this point, the MySQL database can be connected to via ODBC, and its structure and contents updated to the widgets.mdb Access database. Since this particular step is often the cause of confusion for those users attempting their first connection between MS Access and MySQL (or any other database), a separate section is devoted to it.

Developer Shed

Importing and Linking the Data Source

Given the initial confusion caused by connecting a MS Access database to a ODBC data source, a few clarifications are in order. There are essentially two functions provided by MS Access which are used to associate the MS Access database with the data source, each accessible via the File−−>Get External Data submenu:

Import
The Import option should be used if any data found in the data source will be subsequently used exclusively in MS Access. That is to say that no other applications will have the data source data available to them.

Link Tables
The Link Tables option should be used if the data found in the data source will be updated using applications other than the MS Access. This is likely the choice that the user will want to employ, since it is the intention that MS Access will update the data stored within the MySQL database server in addition to its own internal tables.

Therefore, choose Link Tables from the Get External Data submenu. Doing so will create a "link" between the data source tables and MS Access. Any information updated using the Access Interface (which we'll create in the next section), will automatically update the MySQL database, making the information immediately available from any application using that data. Choosing Link Tables will result in the familiar looking navigation window to appear. However, nothing will actually be selected in the window. Instead, go to the Files of type drop−down menu located at the bottom of the window, and choose ODBC Databases(). This is illustrated in Figure 1−5.

Figure 1−5: Choosing ODBC Databases()
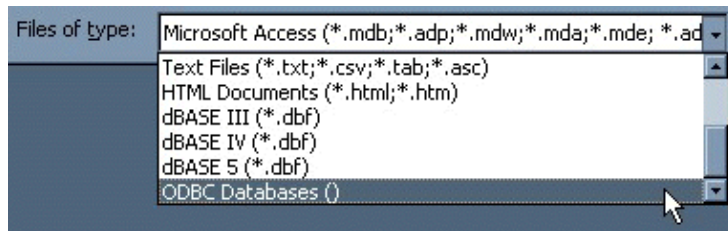
Doing so will cause yet another window to appear. Choosing the tab entitled Machine Data Source will produce a window very similar to that shown in Figure 1−6. Notice that the widget DSN is included in the data source name listing. Click on the widget DSN and press OK. This will produce one final window, which lists all of the tables found in the widget database, in this case only one, clients. Click on the clients table and press OK. The window will close and the original database window (Shown in Figure 1−4) will reappear, this time including the names of those tables chosen in the window. Next to each table name will be a globe with an arrow pointing to it. This signifies that these tables have been linked to those found in the data source.

Note: The client machine must be connected to the Internet, otherwise it will be impossible for MS Access to update the new database with the information found in the data source tables!

Figure 1−6: Machine Data Source window
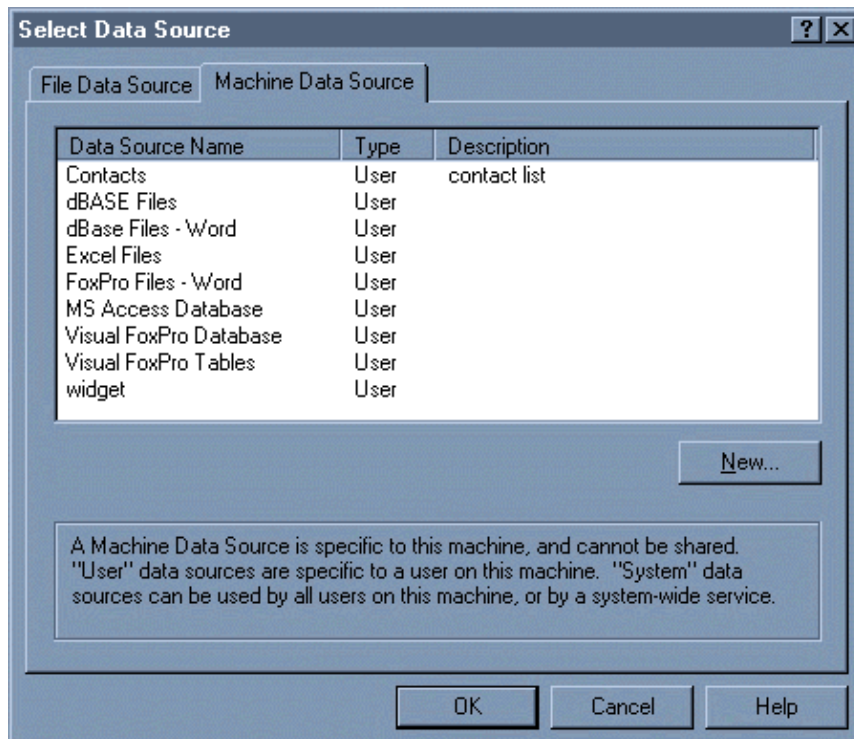
Double–click the clients table name, and window will appear displaying the table structure. Notice that the columns are the same as those found within the MySQL clients table created earlier in this tutorial.

Congratulations. An ODBC connection to a MySQL database has been successfully carried out. In the next section, a demonstration of how inserting information into the linked tables via the Access database will cause the MySQL database to be simultaneously updated.

Updating The Database

Figure 1–7 shows a simple GUI created using the MS Access Form Wizard. After creating this GUI, I used it to insert a sample record into the database.

Figure 1–7: Clients GUI
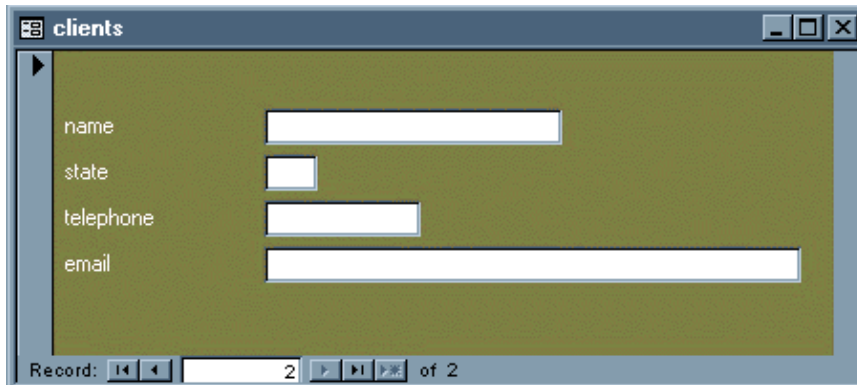
Create a GUI similar to the one shown above, and experiment with inserting information into it. Immediately upon doing so, connect to the MySQL database and execute a select * query on the clients table. The table will show that it has been updated with the information entered using the MS Access GUI! Figure 1–8 shows the clients table as seen from MS Access after entering one record. Following Figure 1–8 is a view of the MySQL clients table after executing a SELECT * query.

Figure 1–8: View of the clients table from MS Access


```
mysql> select * from clients;
+----------+-----------------+-------+------------+---------------------
| clientId | name | state | telephone | email |
+----------+-----------------+-------+------------+---------------------
| 1 | Widget Fanatics | OH | 4125555678 |
widgets@widgetfanatics.com |
+----------+-----------------+-------+------------+---------------------
```

Developer Shed

To convince yourself that the synchronous updates occur regardless of whether the update takes place from MS Access or directly from MySQL, go ahead and delete the newly added row from within the MySQL monitor. Then return to MS Access and reopen the clients table. You will see that the data has disappeared!

For those readers new to MS Access, take some time to experiment with the various Wizards made available from within the program. Using these Wizards will better acquaint you with how Access works. However, to tweak the various components of the Access interface to your specific needs, you will need to learn how to modify the component properties manually. To learn more about doing so, I would suggest searching the Web and consider purchasing one of the many books available on the subject.

Conclusion

This tutorial was intended to instruct the reader on how a MySQL database can be easily connected to MS Access using the MyODBC drivers. A number of topics were covered, including:

- How to install the MyODBC drivers
- Creating a new ODBC Data Source through the ODBC DSA.
- Linking the ODBC Data Source to an MS Access database
- Updating the MySQL database through an MS Access GUI

Hopefully, the information provided has introduced you to a whole new methodology for providing your clients with an easy solution for maintaining MySQL data. If you create any particularly interesting applications using the information found in this tutorial, I would love to hear about it!

Developer Shed