Access a Password Protected Microsoft Access Database from within Visual Basic 6

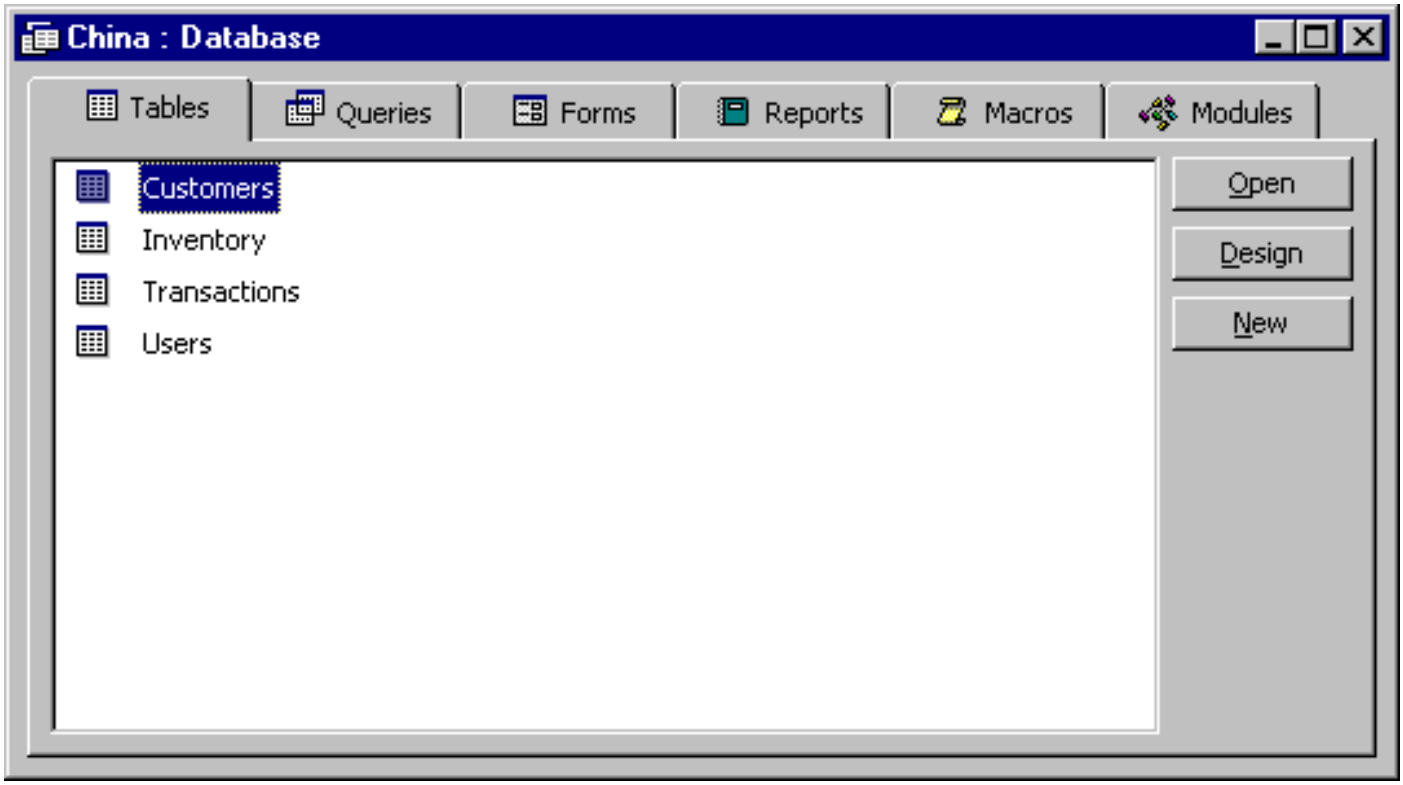
Have you ever wanted to password protect an Access Database that is a Data Store (a repository of Data) used in one of your Visual Basic programs?

Password protecting an Access Database is pretty easy, and I'll be showing you how to do that in a minute. Getting to its data from within Visual Basic has been known to be a real pain for beginner programmers---and that's what I want to show you how to do in this article.

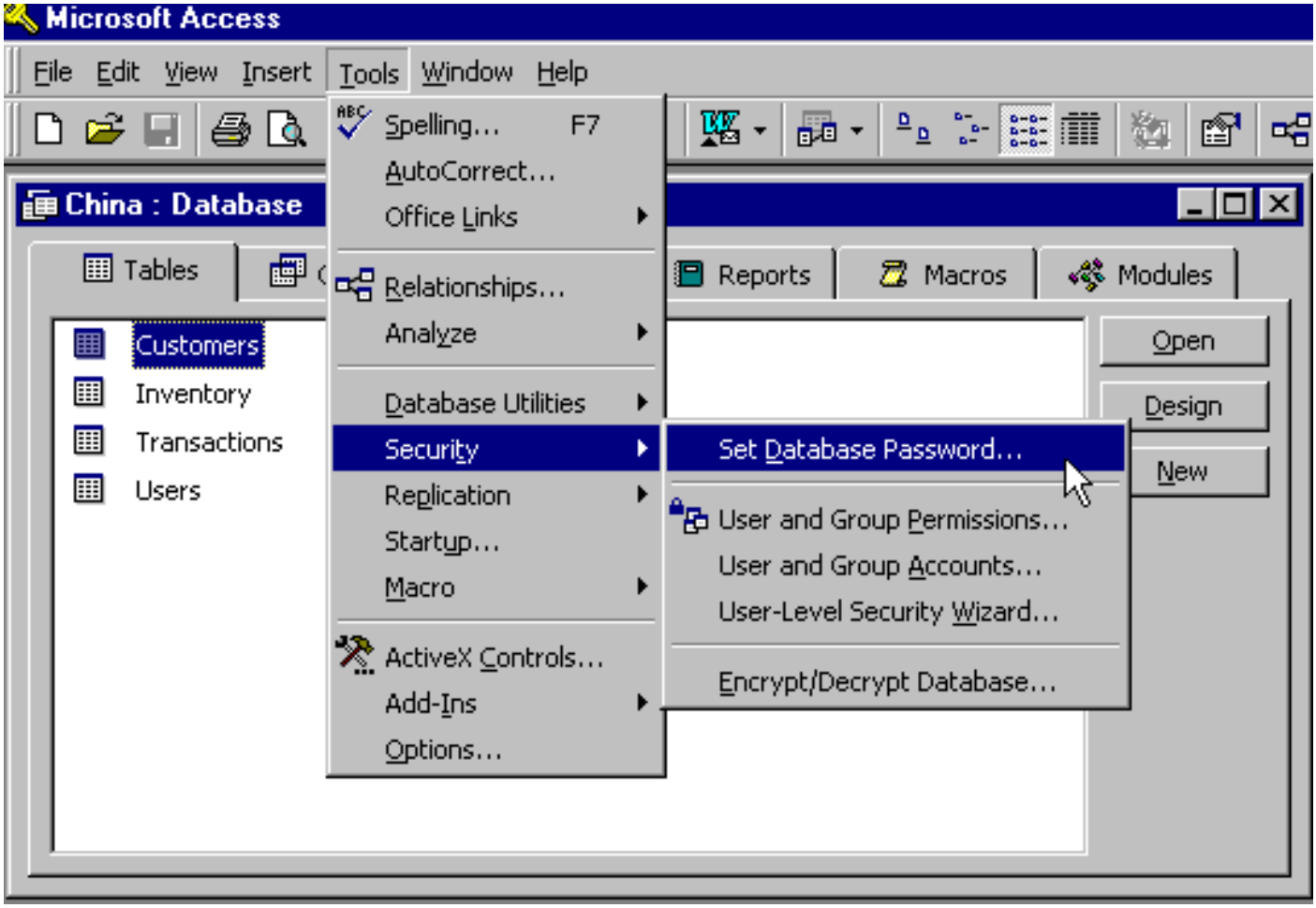
First though, let's password protect one of my favorite Databases---the China Shop Database.

Password Protect an Access Database

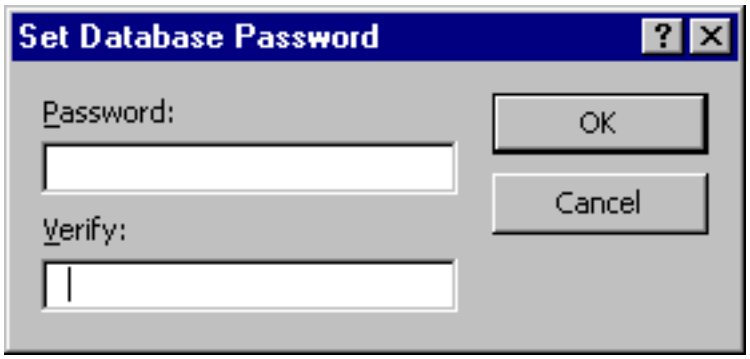
The first thing we need to do is fire up Access (my screenshots are from Access 97), and select the China Database. With the Database Window in view …



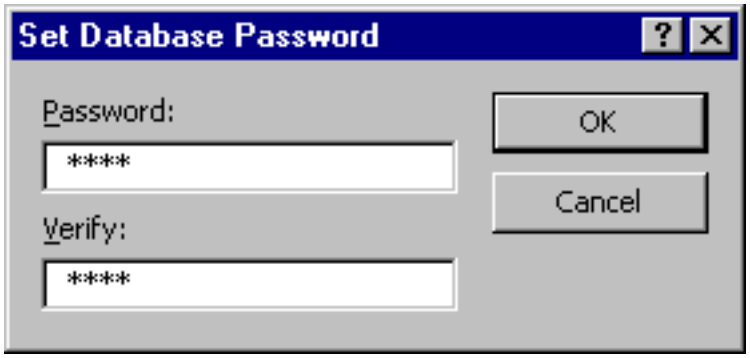
Select Tools-Security-Set Database Password from the Access Menu Bar…



When you do, the following window will appear…



Enter a password of your choice (I entered 'dish'---by the way, Access passwords are case sensitive) and click on the OK button (there is no confirmation of any kind).



By the way, to set a password, you must have the Database opened for Exclusive access, otherwise you'll receive an error message when you try to set the password. Exclusive Mode is the default open mode for a database---when you select the database to be opened from the Access menu bar, the Exclusive checkbox should be selected.

That's great---our database is now password protected. If we now close the Database, and then attempt to open it again, we'll be prompted for a password. Enter it (don't forget, Access passwords are case sensitive)

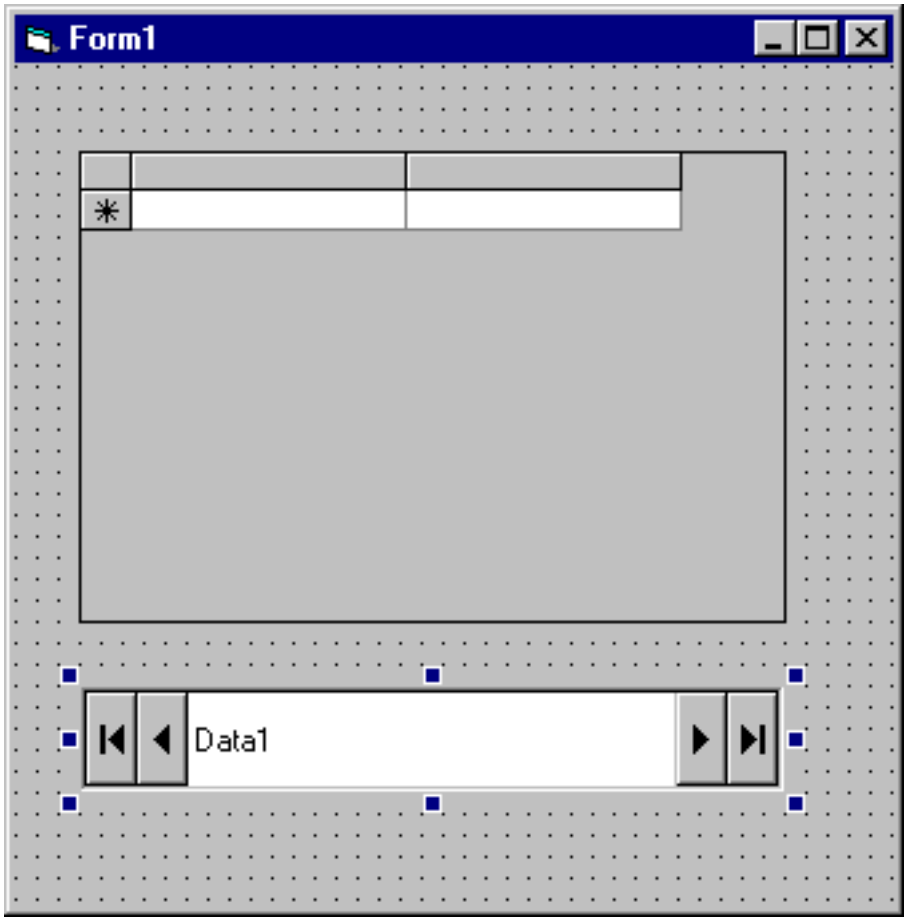
…and your database will be opened.

Open the Database from within Visual Basic Using Data Controls

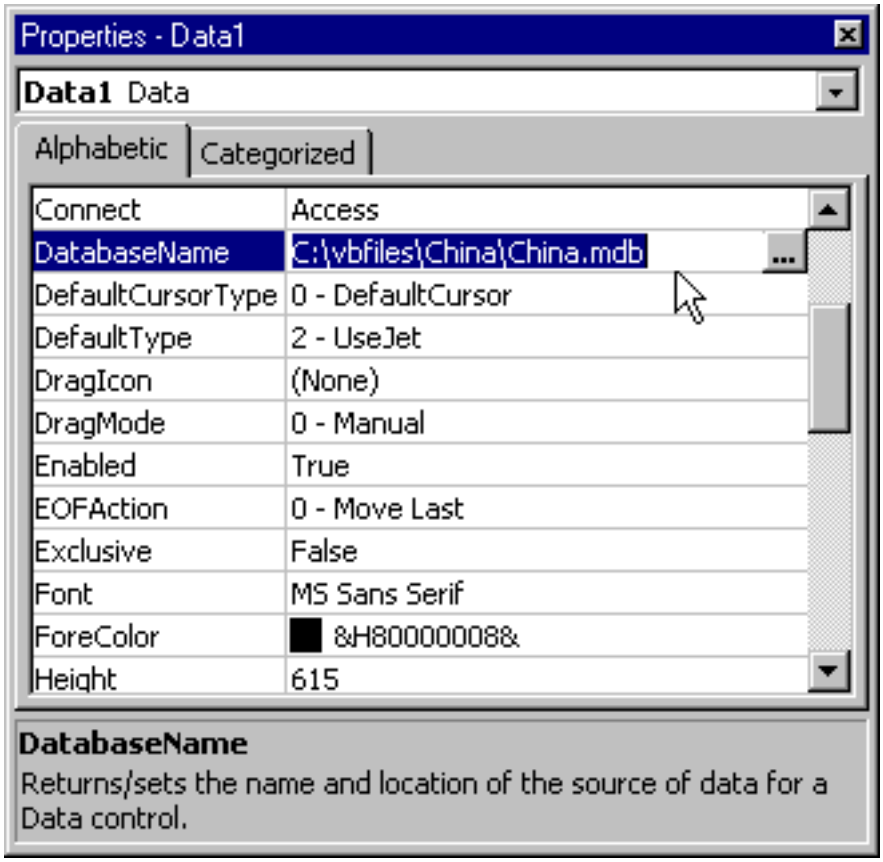
Setting the password in Access is pretty easy. Now let's see how we can get to the Database from within Visual Basic. I want to show you how to open the Database using the Data Control first (both the DAO and ADO Data Control) and then I'll show you how to use Data Objects to open the Database.

The DAO Data Control

Let's work with a simple table---the Users table has just two fields, so I'll create a form with a VB5 Data Grid and a DAO Data Control.



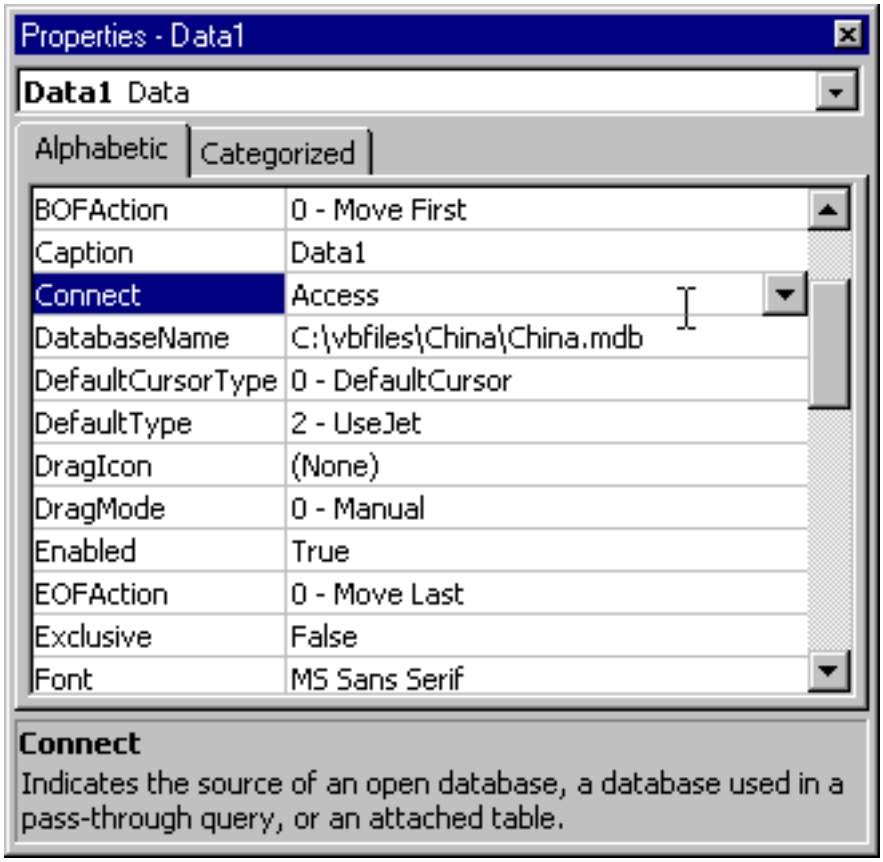
Ordinarily, I would set the DatabaseName Property of the Data Control to point to the China Database…



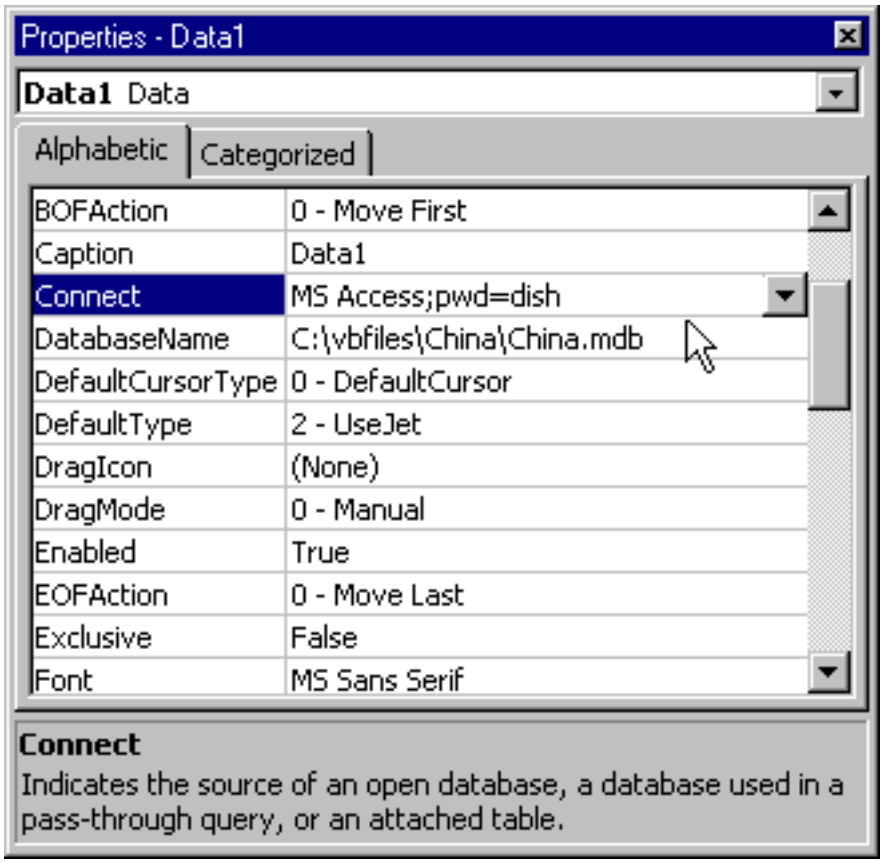
… set the RecordSource to the Users Table. However, when I try to select a Table from the RecordSource list, I receive this error message…

What's the problem? Access wants me to provide a password before it will allow me to select a table from the RecordSource Property (remember, the password we just set in Access is at the Database level---it applies equally to every table in the China Database). What confuses beginners here is that there is no Password Property, so how can we provide one.

The answer is the Connect Property. Ordinarily, by default, it just reads Access…

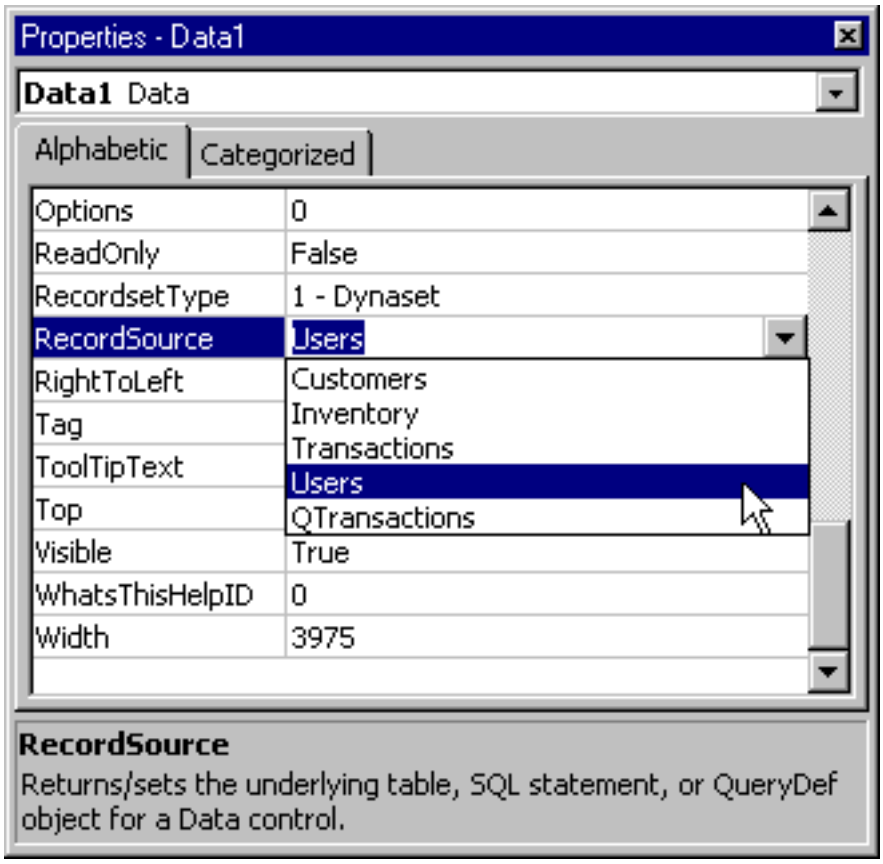


However, for a password protected Database, this is where the password goes. Like this…

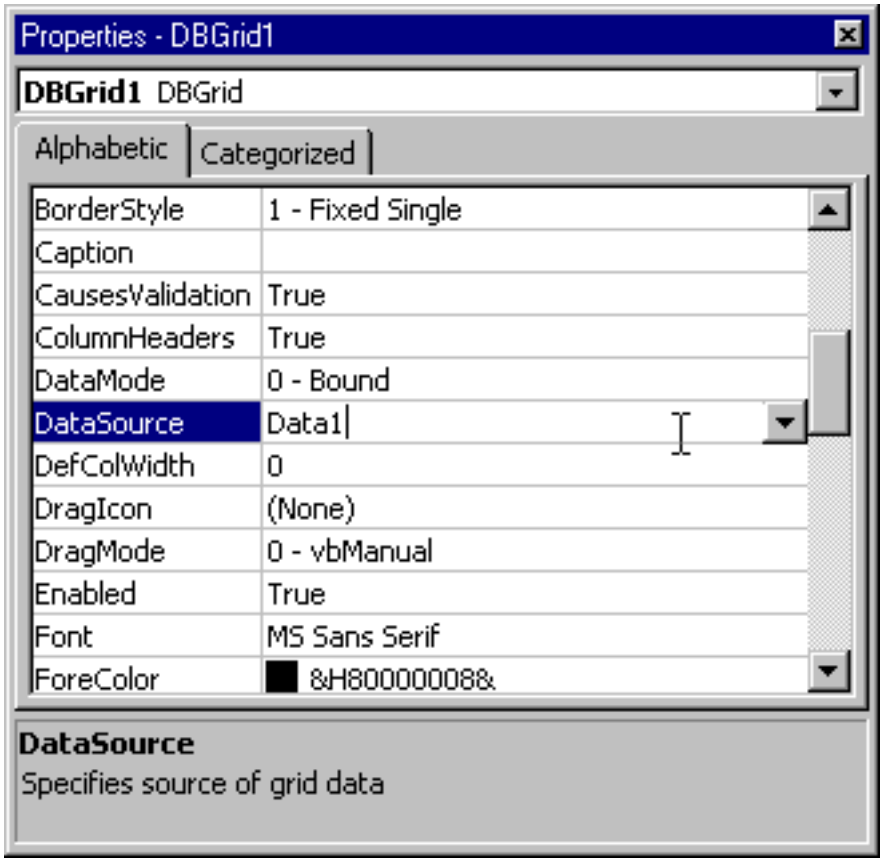


Once we set the Connect Property, a list of tables will appear in the RecordSource

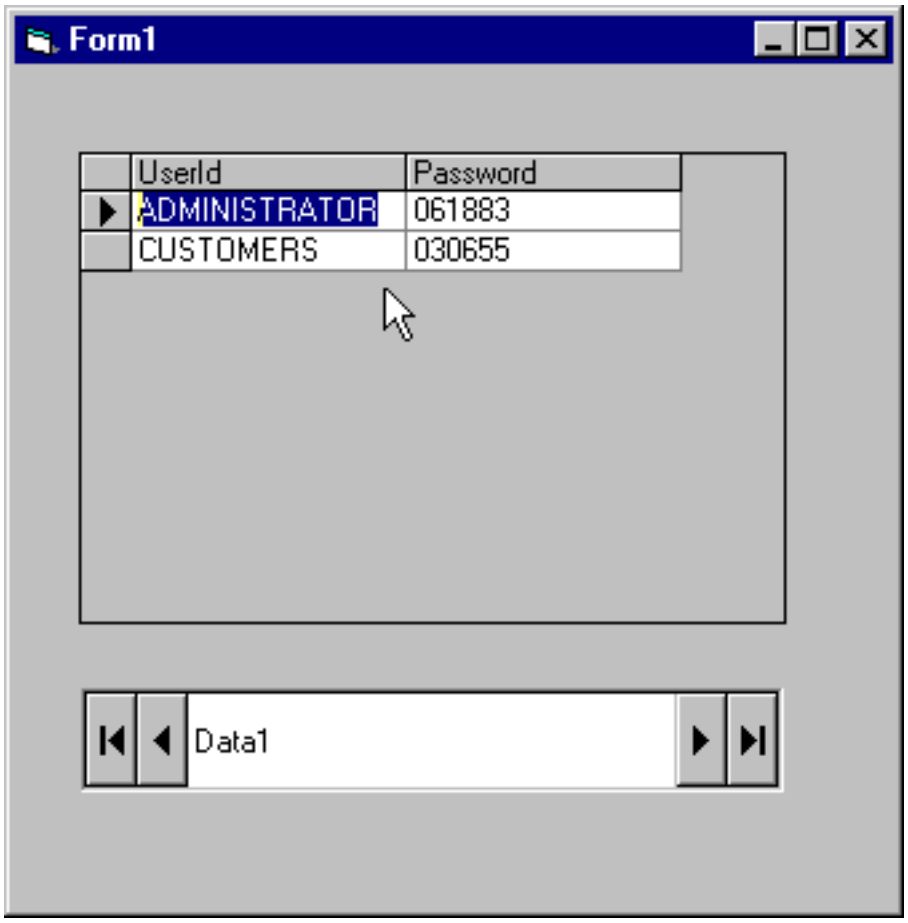
Property…



…and once we bind the DBGrid to the Data Control by setting its DataSource Property to point to the Data Control



…if we now run the program, we'll see the two records in the User table…

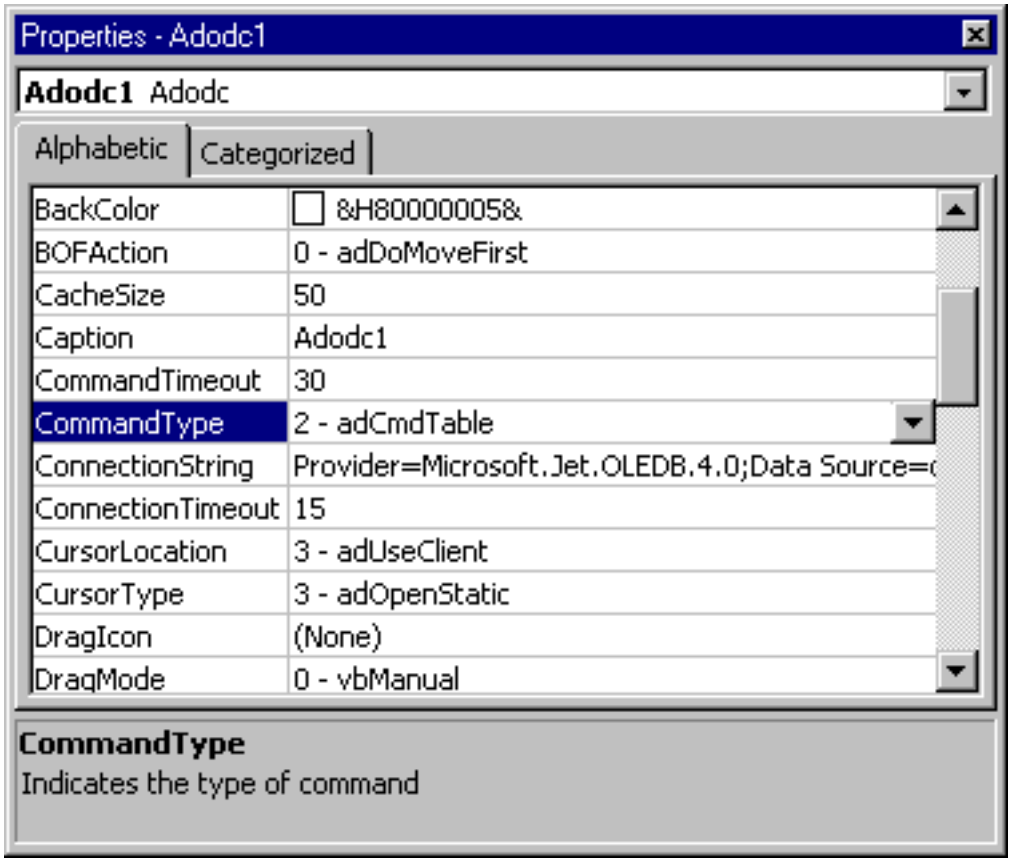


Success! That wasn't too bad was it? We've opened a password protected Access database from within Visual Basic.

The ADO Data Control

Working with the ADO Data Control is not quite as simple.

ADO Data Controls use a Connection String, and for a password protected Database, it should read like this…

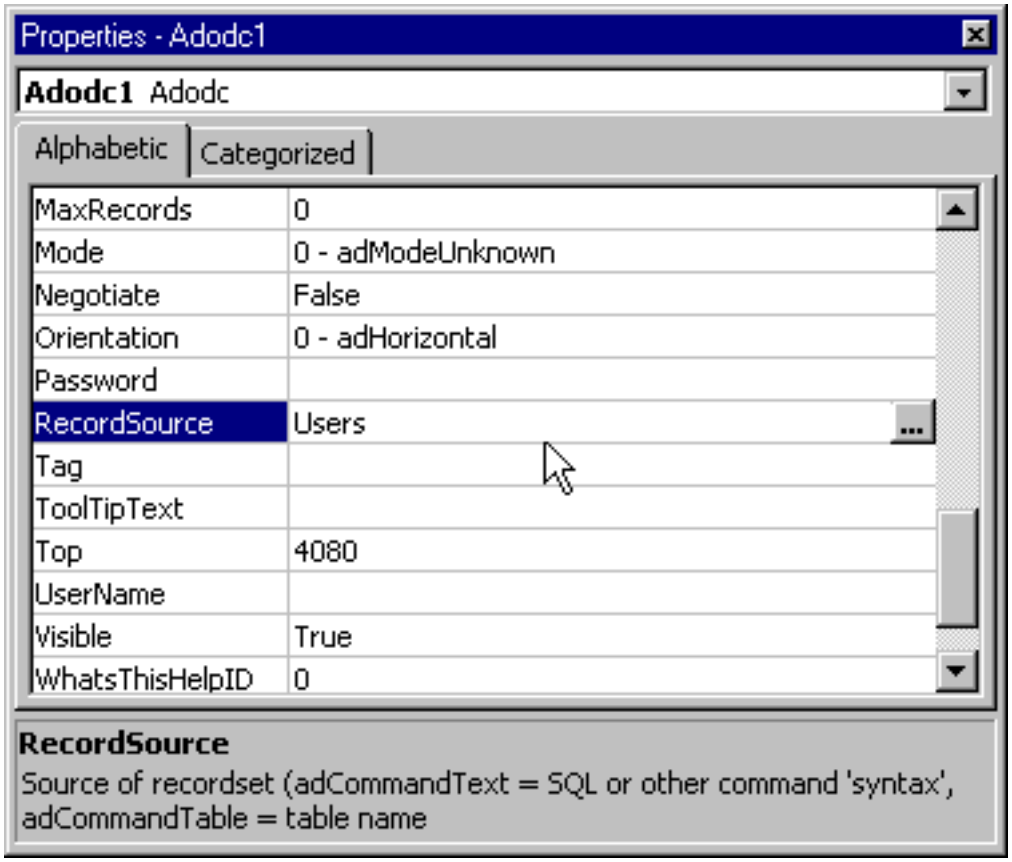


You can't see the full text of the ConnectionString Property, so I've included it below…

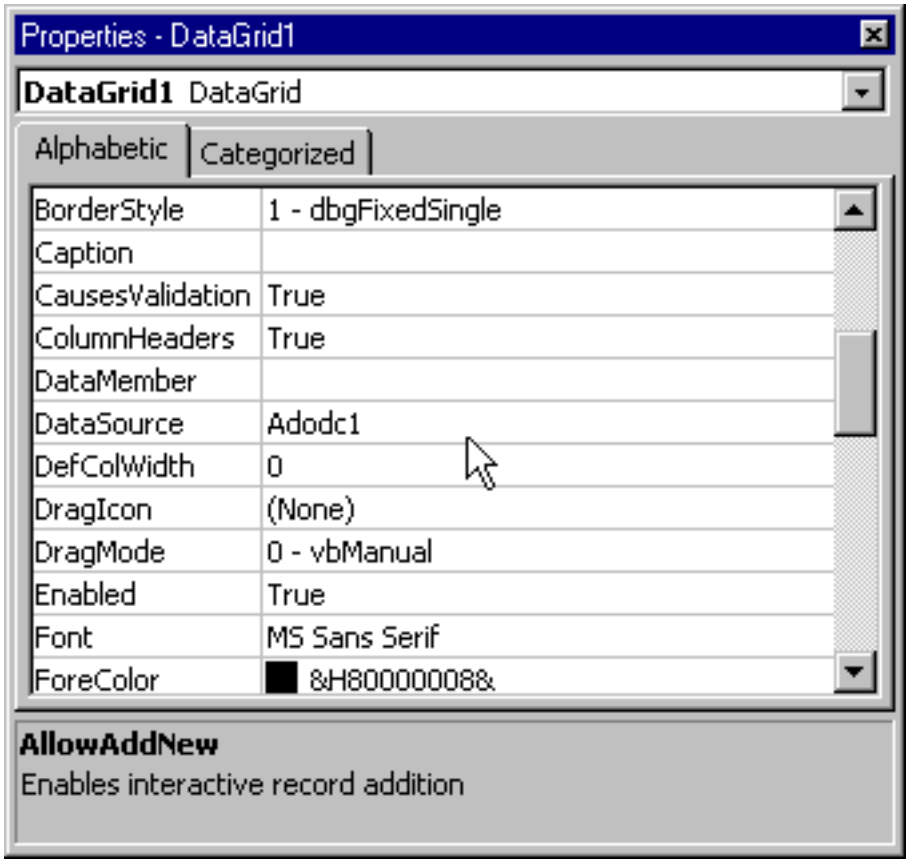
Provider=Microsoft.Jet.OLEDB.4.0;Data Source=c:\vbfiles\china\china.mdb;Jet OLEDB:Database Password=dish;

Notice how the ADO parameter is a little different---it's Database Password not pwd…

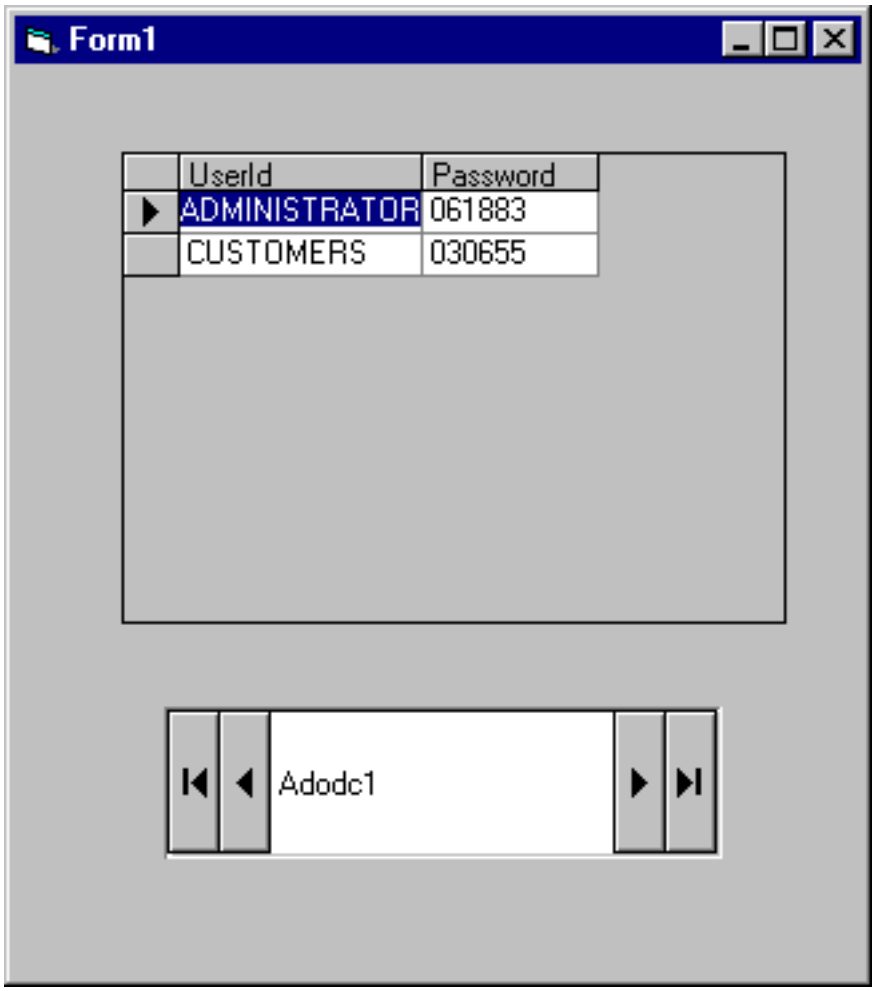
Now we need to specify a RecordSource Property of Users again…



…and as we did before, bind the Grid to the Data Control by setting its DataSource Property to point to the Data Control



If we now run the program, we'll see the two records in the User table…



What about Data Objects?

Some of you may be wondering about working without the Data Control---in other words, by using Data Objects. No problem! Let's look at DAO first.

DAO Objects

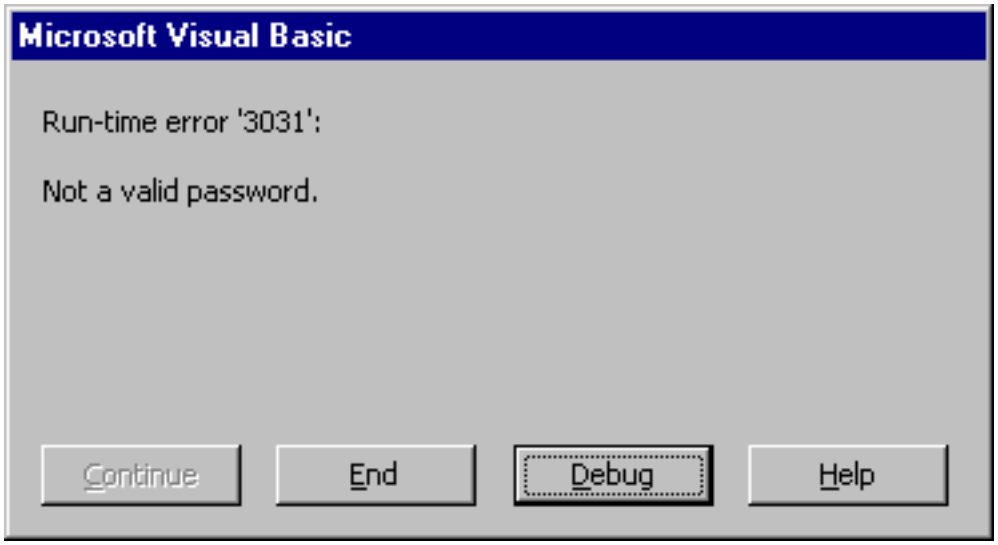
Let's remove the grid and the Data Control from this form, add a ListBox control and a Command Button, and access the Inventory table of the China Database from within Code. (Those of you who have read my Database book will recognize this code form Chapter 9). After we open the China Database, we'll use a SQL statement to list the unique china brand names in the ListBox.

In my Database book, we used this code to do the trick…

Private Sub Command1_Click()

```
Dim strSQL As String
Dim Ws As Workspace
Dim Db As Database
Dim Rs As Recordset

strSQL = "SELECT DISTINCT Inventory.Brand FROM Inventory"

Set Ws = DBEngine.Workspaces(0)
Set Db = Ws.OpenDatabase("c:\vbfiles\china\china.mdb")
Set Rs = Db.OpenRecordset(strSQL, dbOpenDynaset)

Do While Not Rs.EOF
  lstBrands.AddItem Rs.Fields("Brand").Value
  Rs.MoveNext
Loop

End Sub
```

The only problem is this code isn't supplying the password we need to open the China Database. If we place this code in the Click Event Procedure of our Command Button, and then run the program, it bombs with this error message.

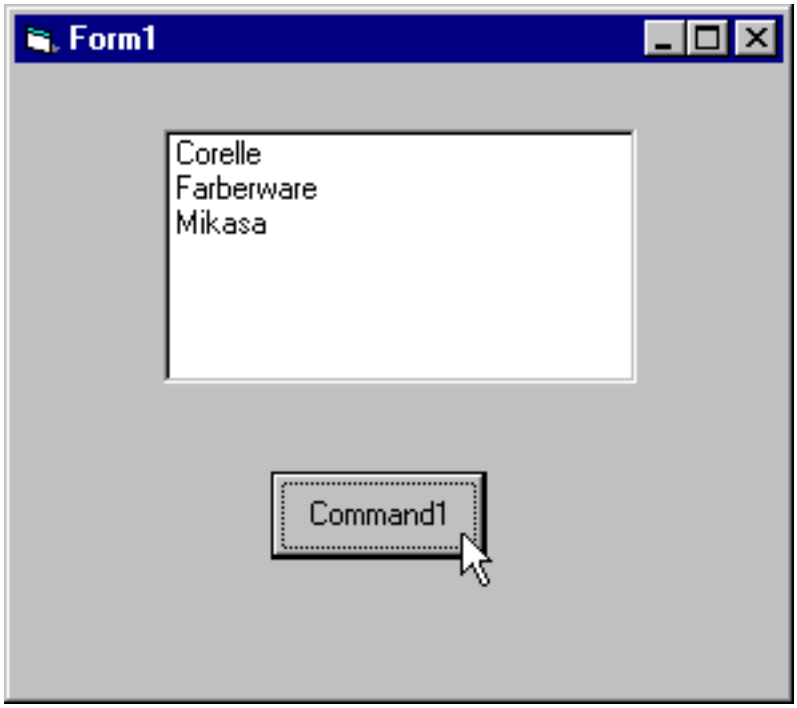


No problem though---all we need to do is change the line of code that opens the Database Object, like this…

```
Set Db = Ws.OpenDatabase("c:\vbfiles\china\china.mdb", False, False, ";pwd=dish")
```

It's a bit cryptic, I know, but essentially all we're doing is supplying a connection string (that's the fourth argument of the DAO OpenDatabase Method.

If we now run the program, and click on the Command button, you'll see the ListBox is populate with the unique brands from the Inventory table…



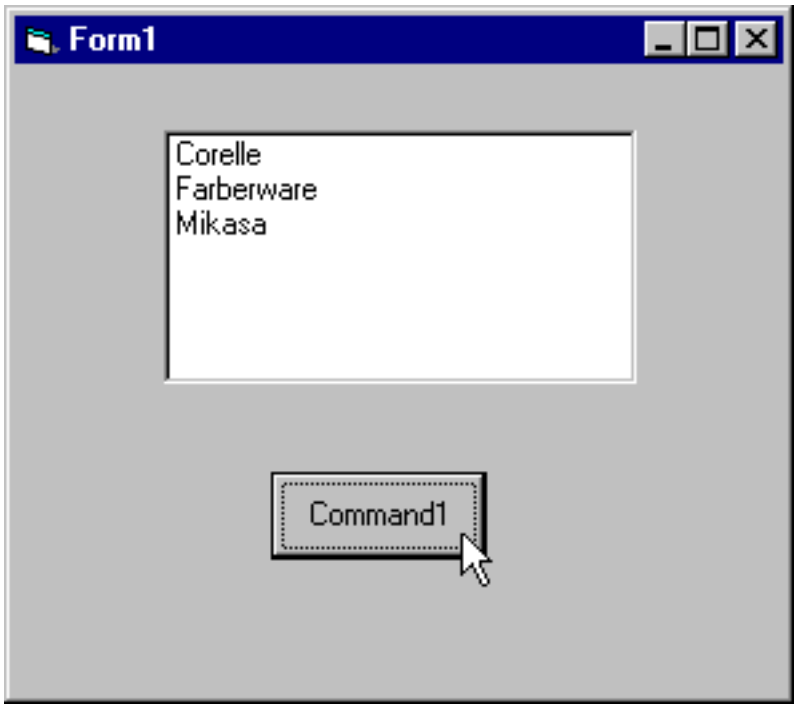
So far, so good. Now let's use ADO to make the connection, and populate the ListBox. Here's the code to do that…

```
Private Sub Command1_Click()

Dim strSQL As String
Dim cn As ADODB.Connection
Dim rs As ADODB.Recordset

Set cn = New ADODB.Connection
Set rs = New ADODB.Recordset

strSQL = "SELECT DISTINCT Inventory.Brand FROM Inventory"

cn.Open "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=c:\vbfiles\china\china.
mdb;Jet OLEDB:Database Password=dish;"

rs.Open strSQL, cn, adOpenForwardOnly, adLockReadOnly, adCmdText

Do While Not rs.EOF
  lstBrands.AddItem rs.Fields("Brand").Value
  rs.MoveNext
Loop
```

End Sub

Perhaps even more cryptic than the DAO version, this code uses the ADO Connection and Recordset Objects to open the China Database and, by using a SQL statement, creates a Recordset consisting of unique brand names.

If we now run the program, and click on the Command button, once again you'll see the ListBox is populated with the unique brands from the Inventory table…



Summary

I hope you now have the confidence to password protect your Databases!