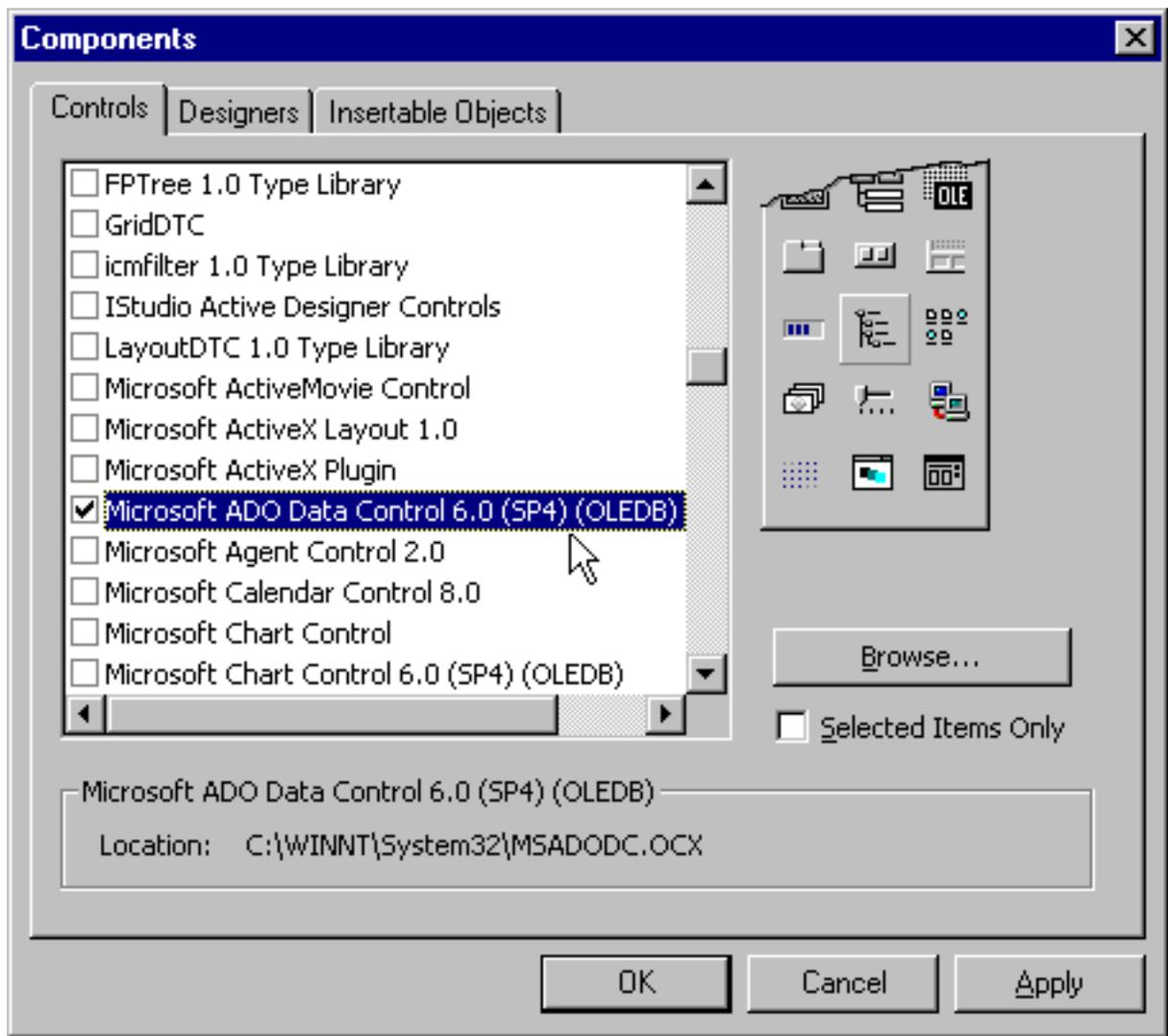


Use the ADO Control in your Visual Basic 6 projects

Aside from my Database book, I haven't done much writing concerning connection to a Database---and virtually nothing on ADO. In this article, I'd like to show you how you can connect to a Microsoft Database using the ADO Data Control and the ADO Data Grid.

Use the ADO Data Control to create a Connection

The first step in using the ADO Data Control to create a Connection to a database is to find the ADO Data Control. By default, the DAO Data Control is contained in the Visual Basic Toolbox, so you'll need to select Project-Components from the Visual Basic Menu Bar, and select the Microsoft ActiveX Data Control...

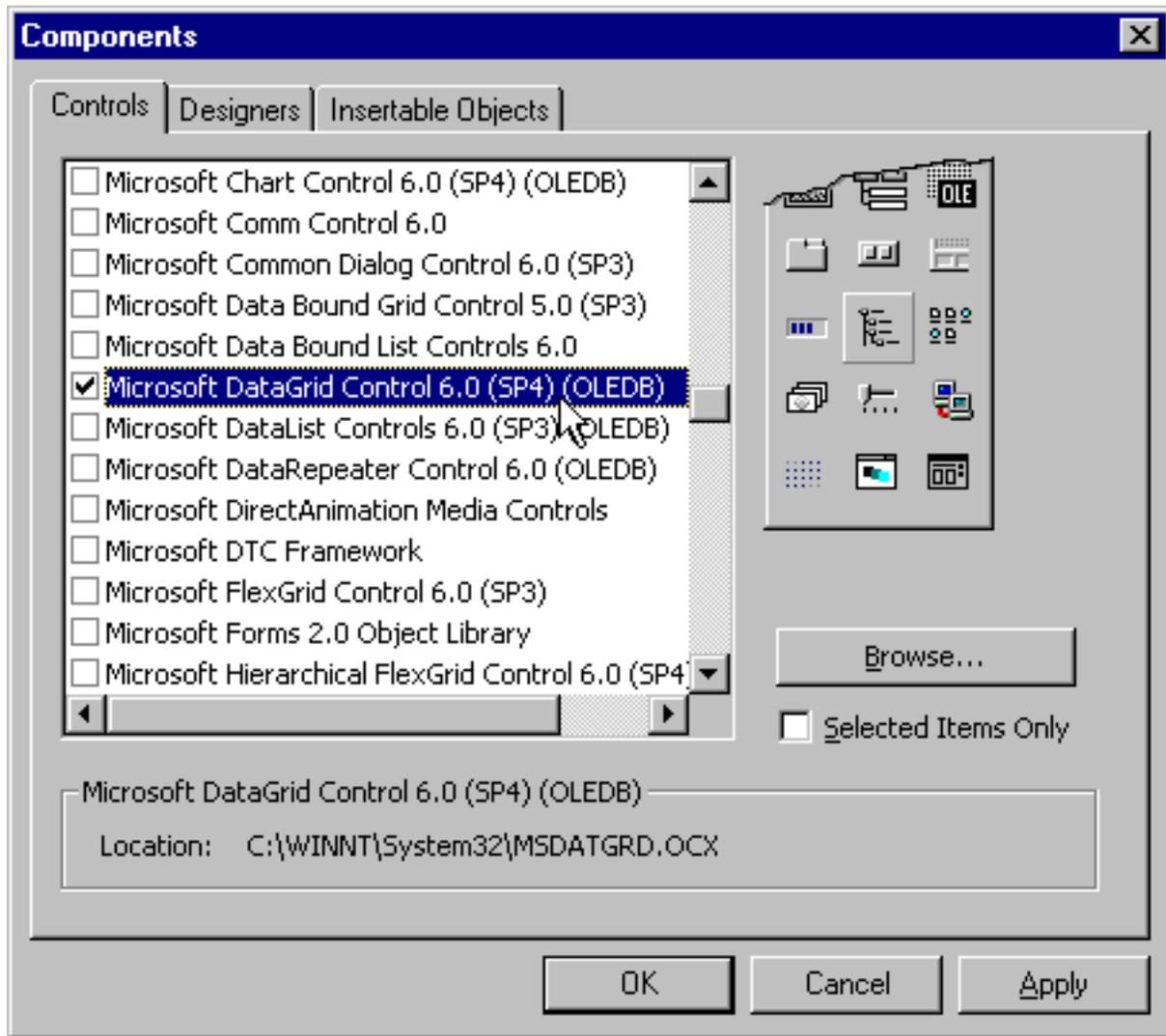


If you click on the OK button, the ActiveX Data Control will then appear in the

Visual Basic Toolbox...



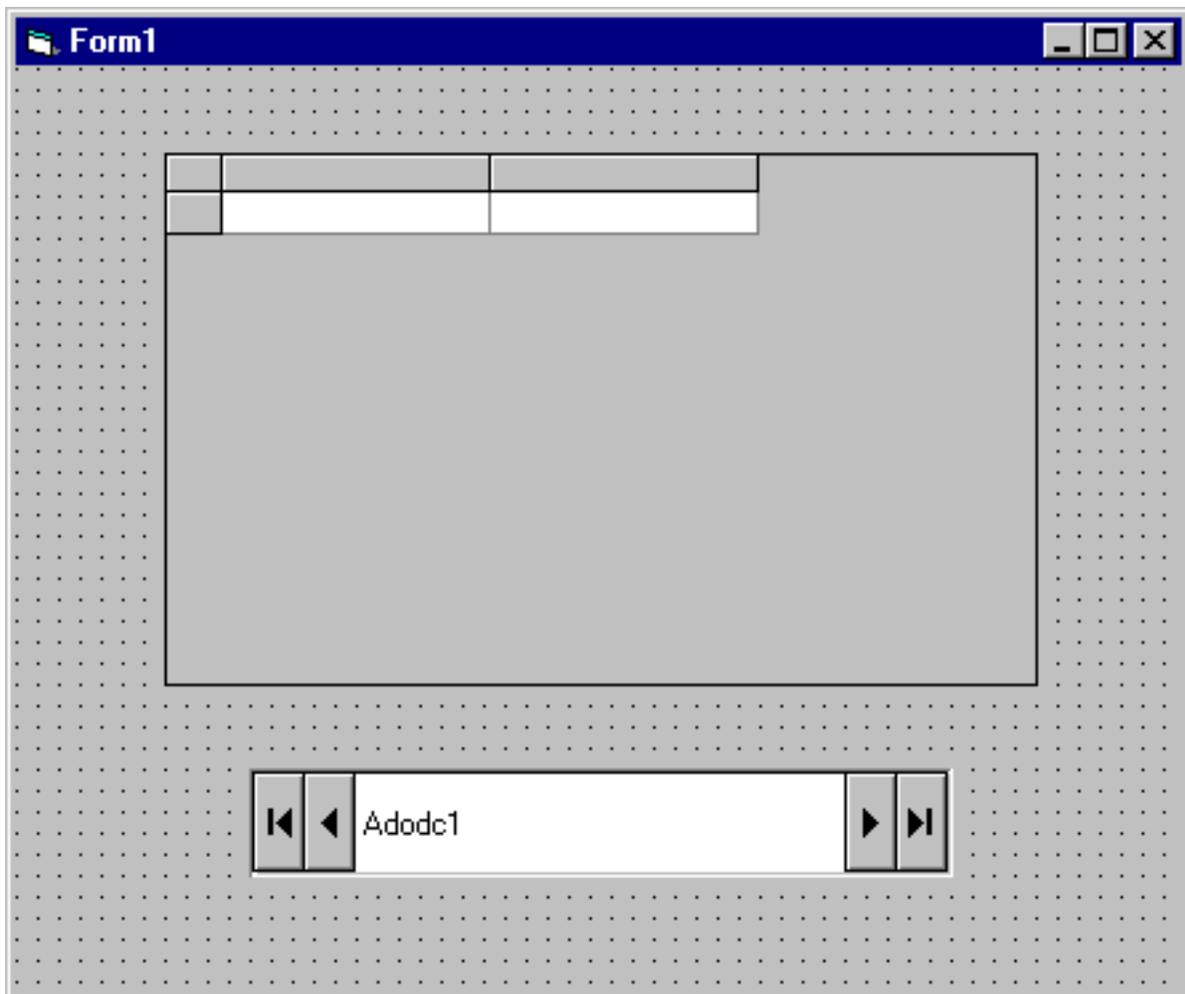
At this point, we may as well find the ADO DataGrid as well---we'll be needing it in just a few minutes to display the data retrieved from the Recordset (a virtual database table) that we'll be generating. Find the ADO DataGrid by selecting Project-Components and looking for Microsoft DataGrid Control...



If we click on the OK button, the DataGrid will now be added to the Visual Basic Toolbox also...

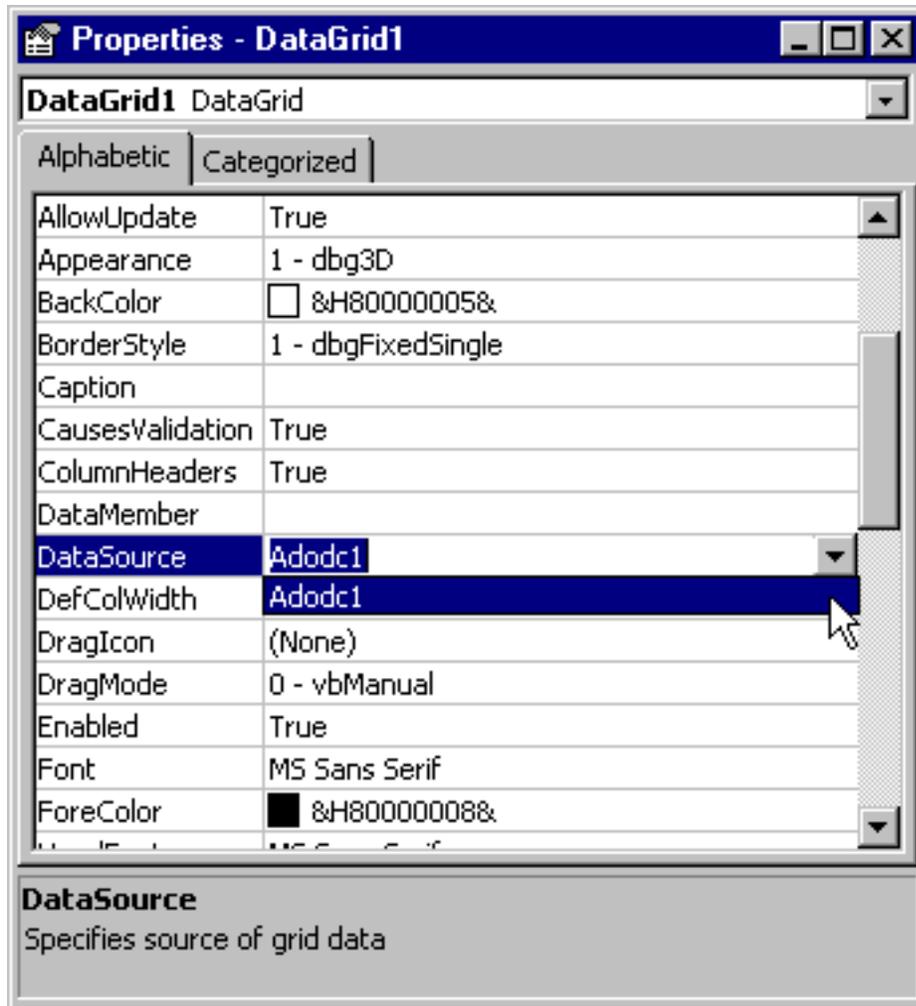


Now let's add both the Data Control and the DataGrid to our form...

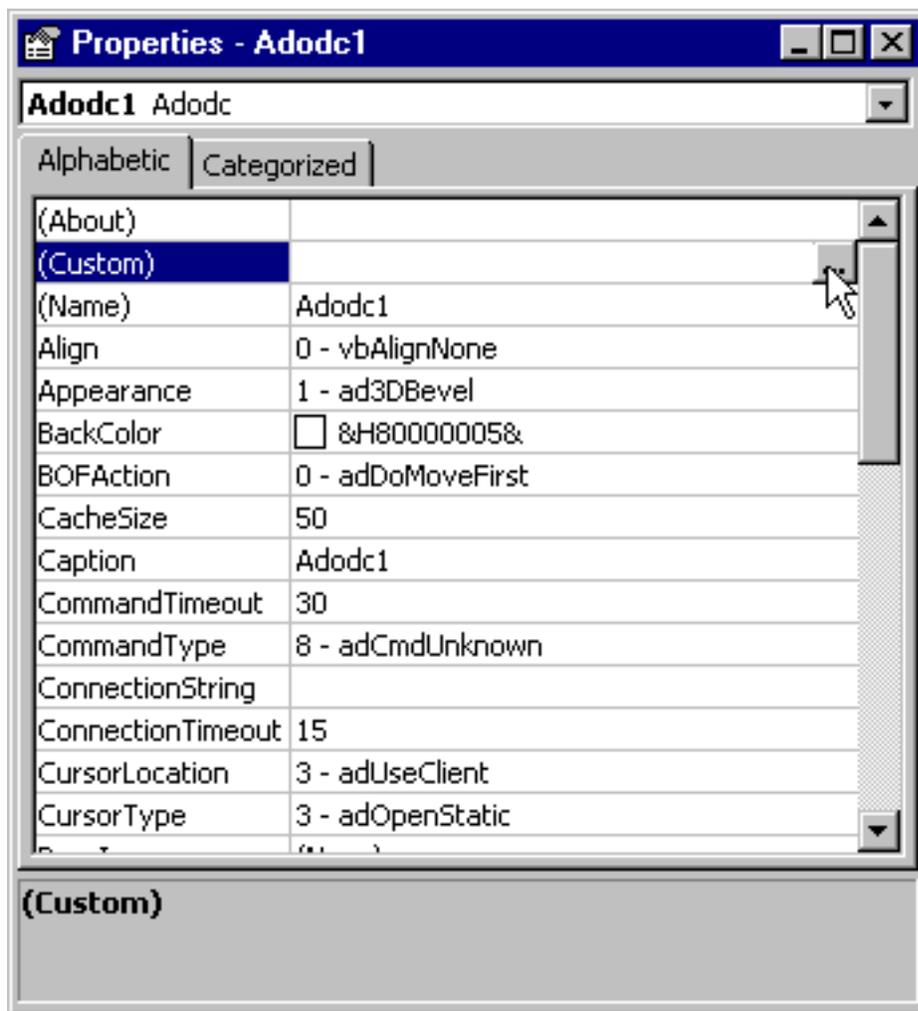


At this point, all we have is an empty DataGrid, and an uninitialized Data

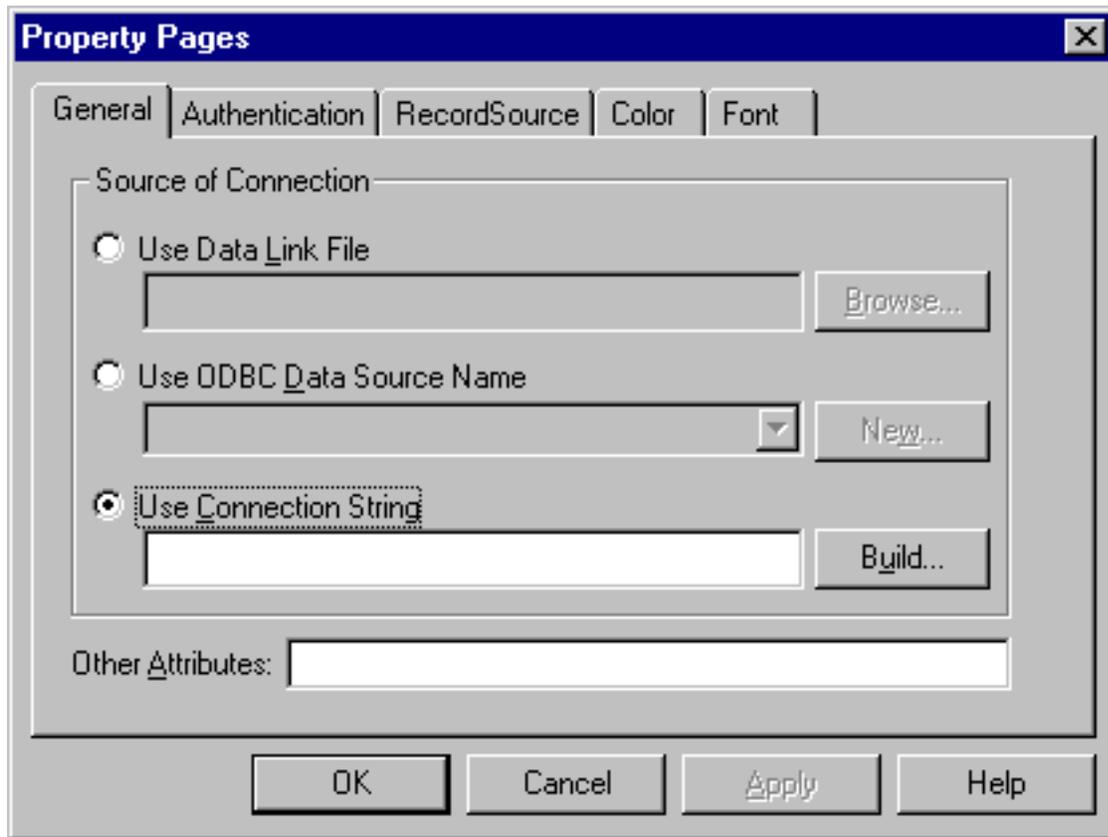
Control. Eventually (in just a few minutes), the Data Control will instantiate a Connection and Recordset, and the records from the Recordset will be 'bound' to the Data Grid. In fact, that's our next step---we need to bind the DataGrid to the Data Control, and we do that by bringing up the Properties Window for the DataGrid, and specifying its DataSource Property as Adodc1--the name of our Data Control...



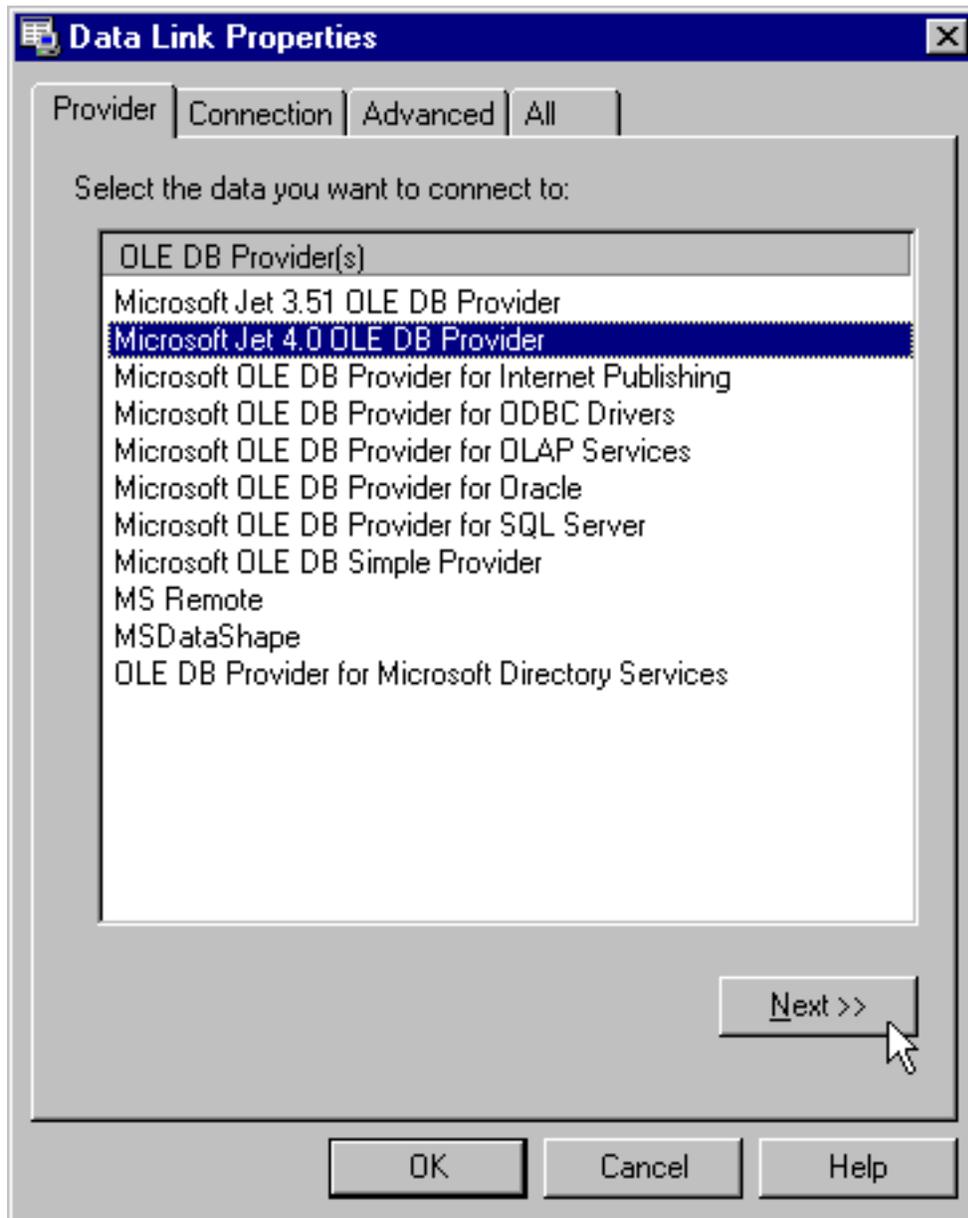
We still don't have a Connection or a Recordset built--but it won't be long. We need to provide values for the CommandType, the ConnectionString, and the RecordSource Properties of the Data Control. Coming up with values for these off the top of our head is too much trouble---Visual Basic will do it for us. All we need to do is click on the Custom Property of the Data Control...



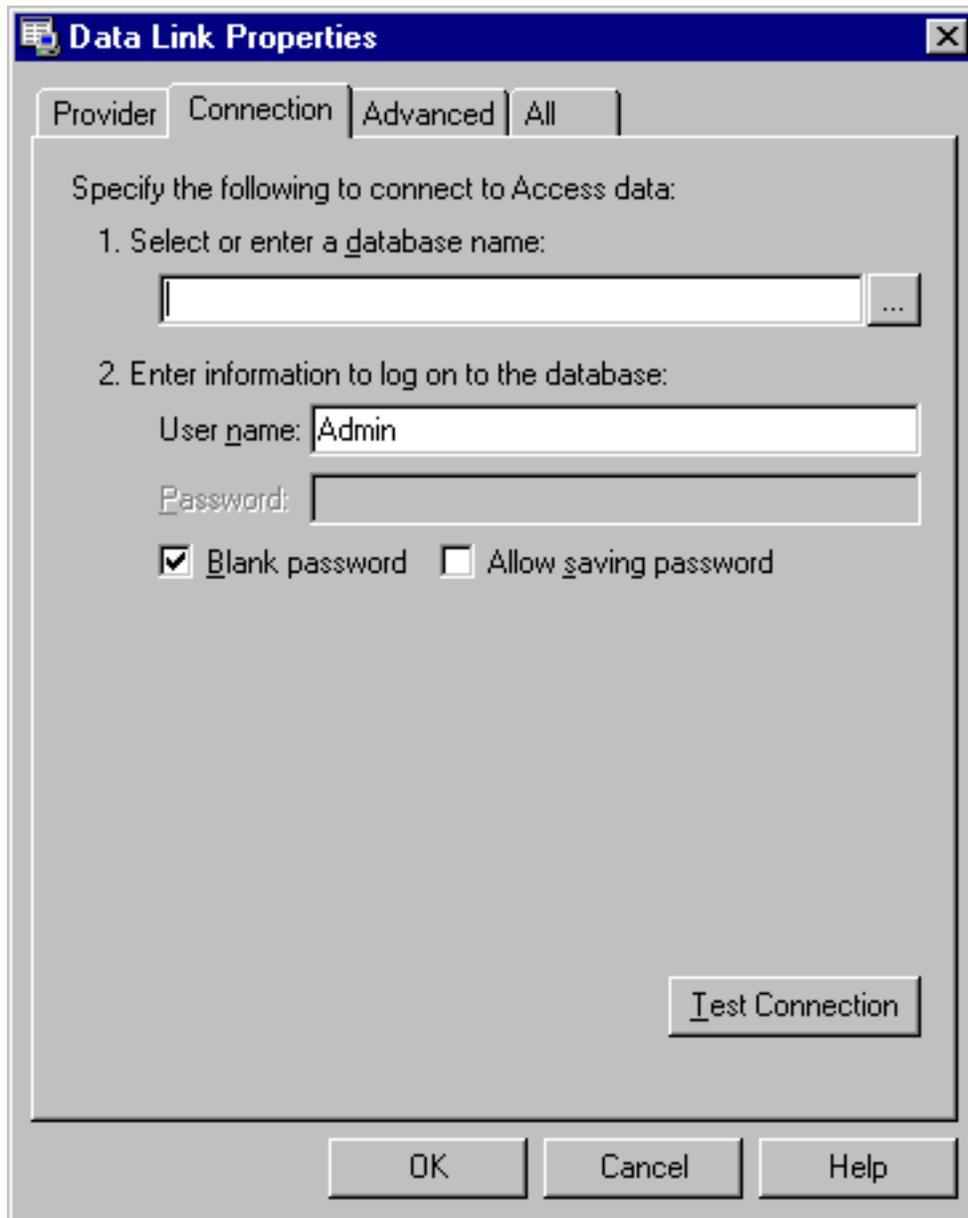
... and a Wizard to build the ConnectionString and the RecordSource Properties will appear...



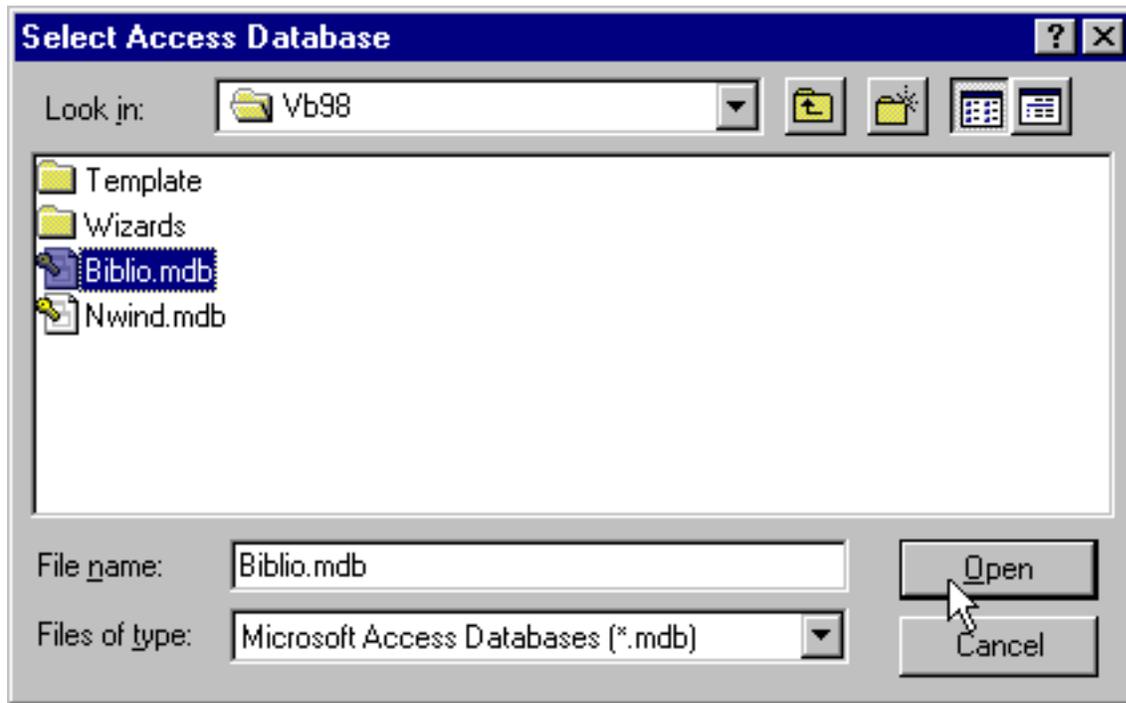
Click on the Build Button next to 'Use Connection String', and this window will appear...



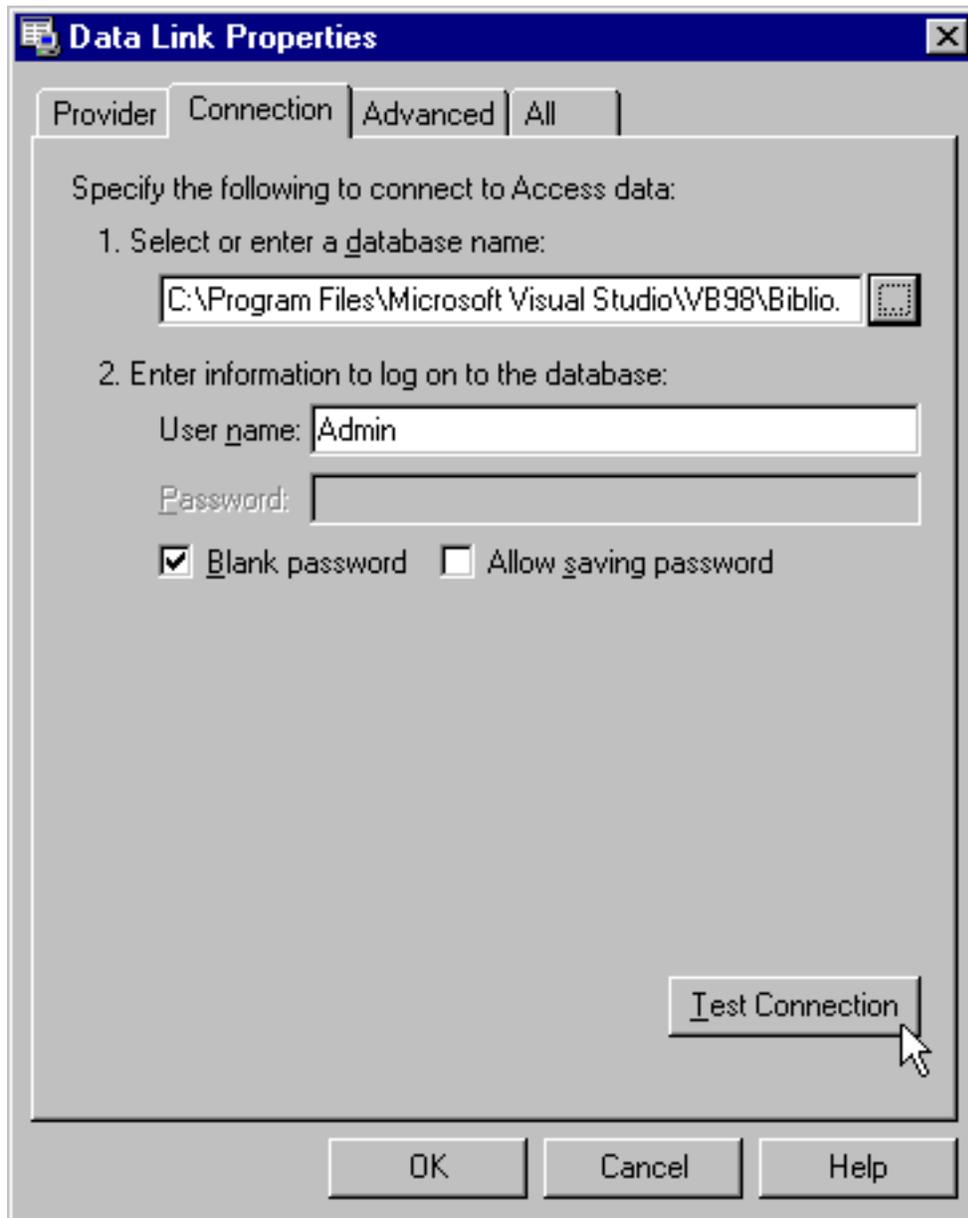
The names in this window may appear to be gibberish--but in order to connect to a Microsoft Access Database, we'll select Microsoft Jet 4.0 OLE DB Provider as our provider, and then click on the Next button. If we were connecting to an Oracle or a SQL Server database, we would select that type of Provider here. After clicking on the Next button, this window appear...



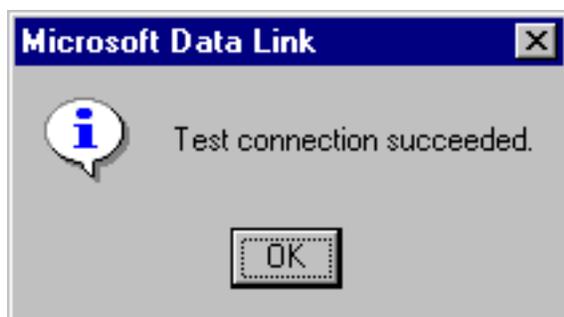
We'll need to specify a database name in this window, and we're going to select a sample database that Microsoft ships with both Access and Visual Basic called BIBLIO.MDB (it contains data about books, authors, publishers and titles). For Access Databases, the User Name is 'Admin' by default, and there is no password. If we had selected an Oracle or SQL Server Provider, this window would look somewhat different, but the idea is the same. For now, let's find and select the BIBLIO.Mdb database using this window...



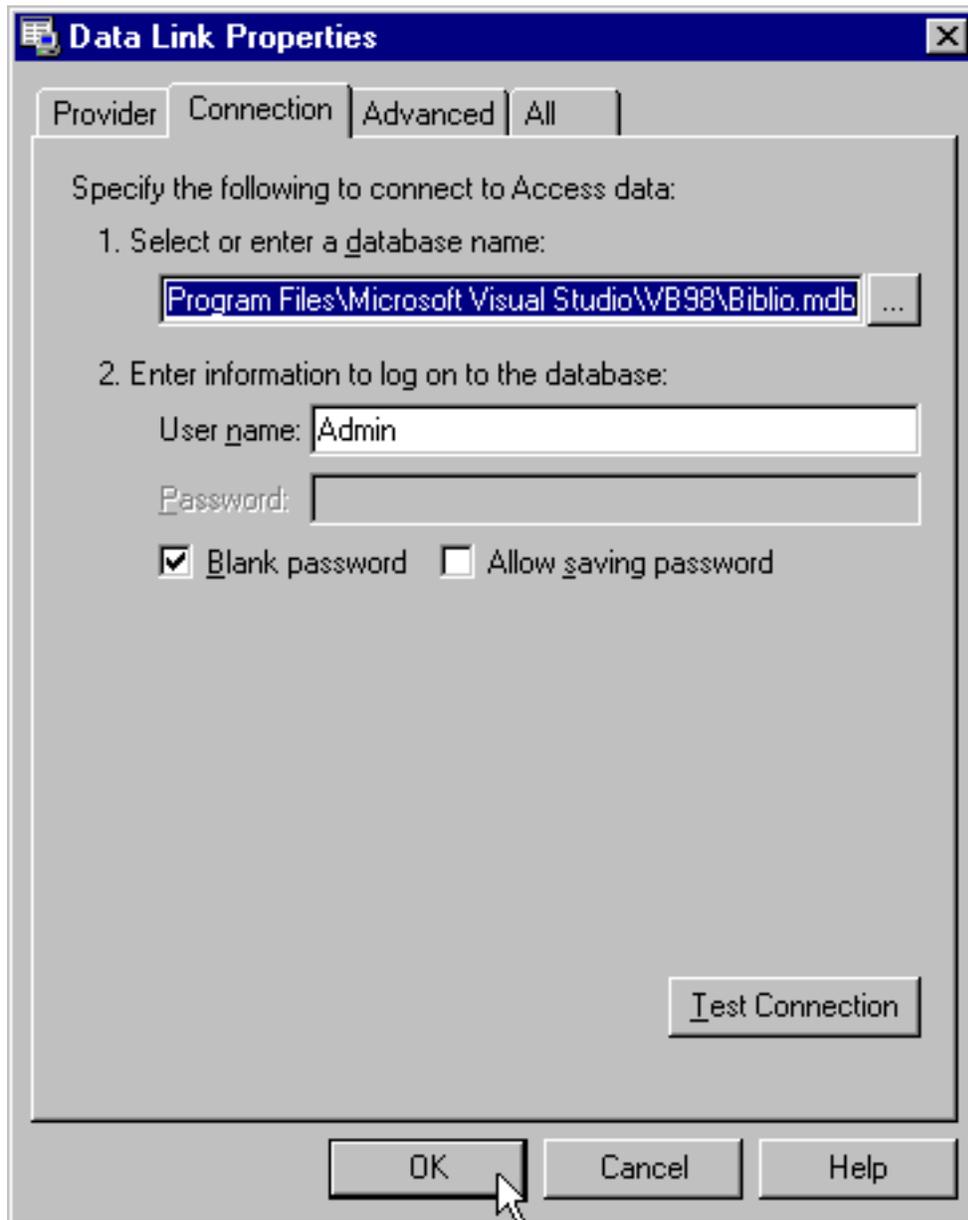
then click on the Open button...



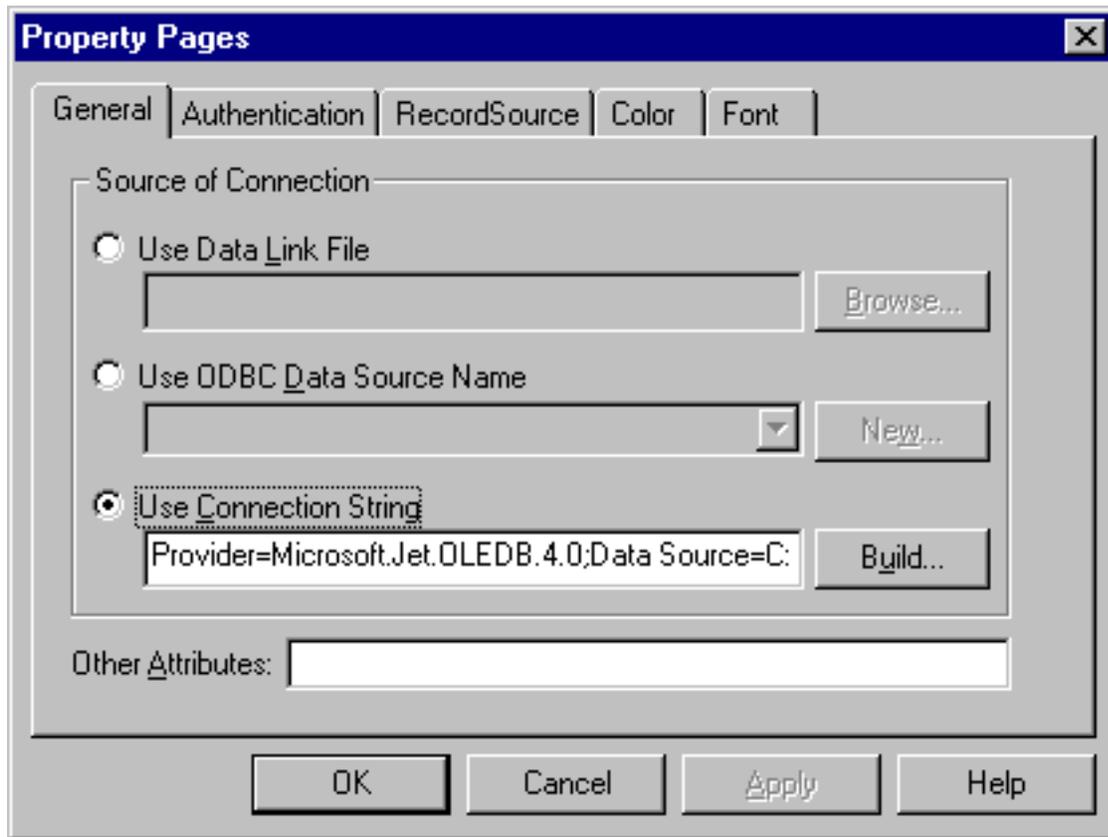
As you can see, the name of the Database now appears on the Connection Tab. At this point, we've established the parameters for the Connection. It's a good idea to click on the Test Connection button to ensure that the parameters are correct, and that the Database can be opened. A few seconds later, this window should appear...



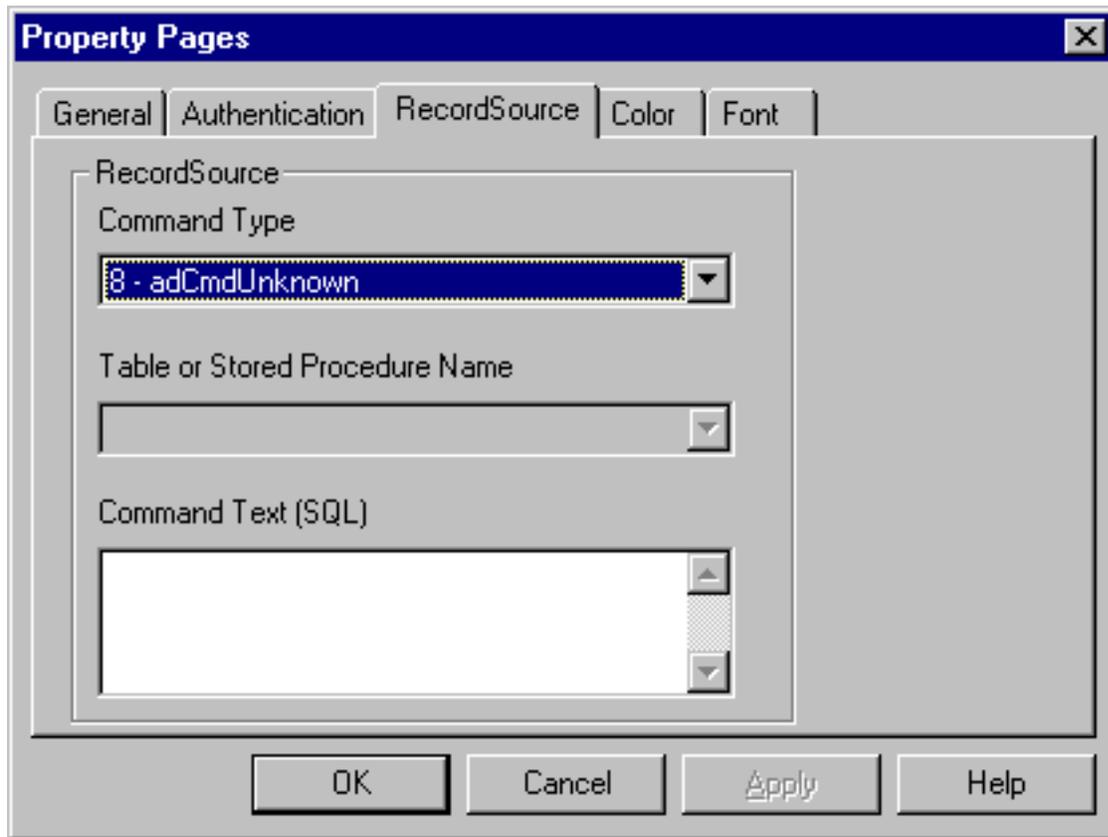
After clicking on this Message Box, you'll be back at this window....



Click on the OK button, and you'll be back here...

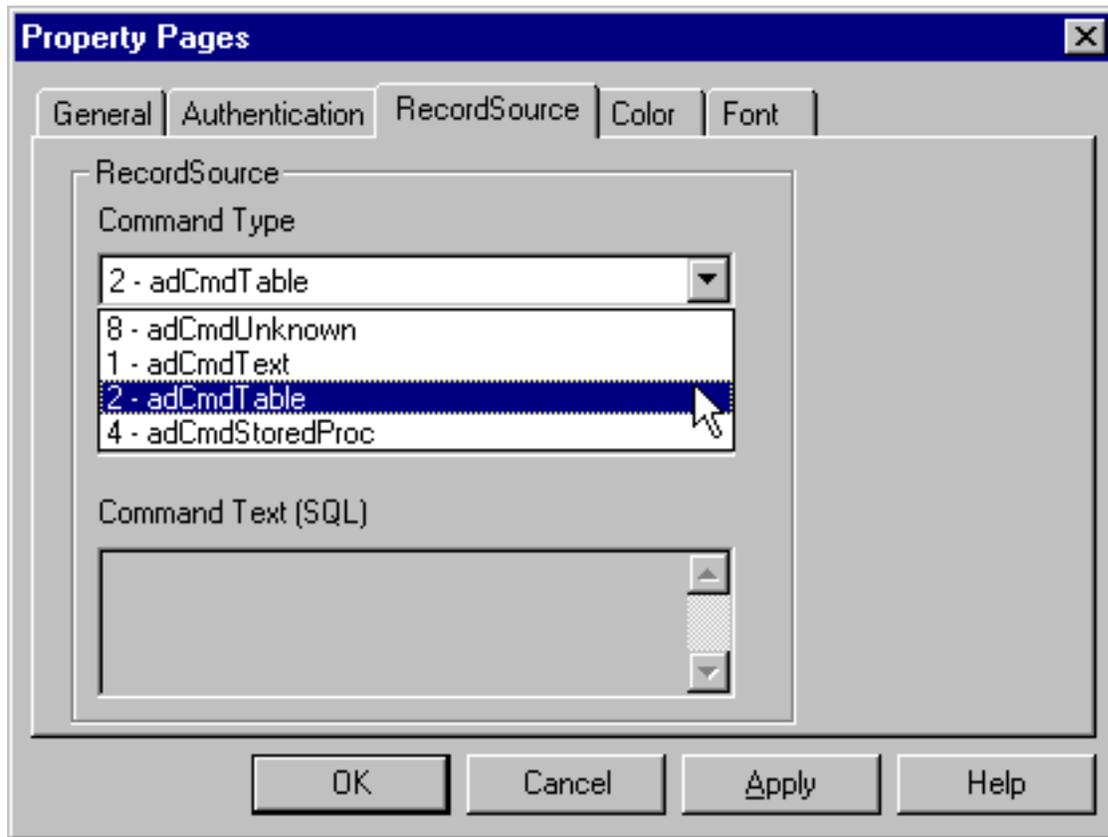


Notice how the Connection String has been built for you. Beginners at this point make the mistake of clicking on the OK button here, but we're not done--we need to specify a RecordSource, which we can do by selecting the RecordSource Tab at the top...

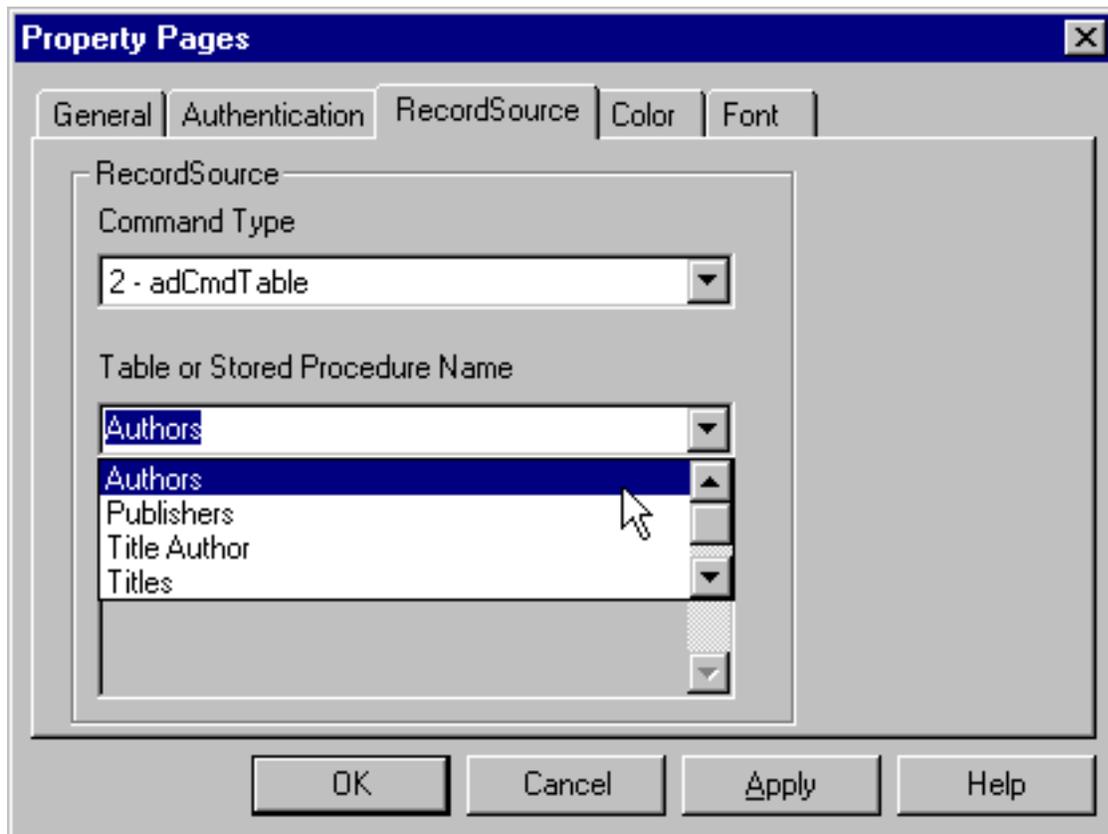


The RecordSource tab looks pretty imposing--but it's really a piece of cake. Just click on the Command Type drop-down ListBox, and select cmdTable as the Command Type...

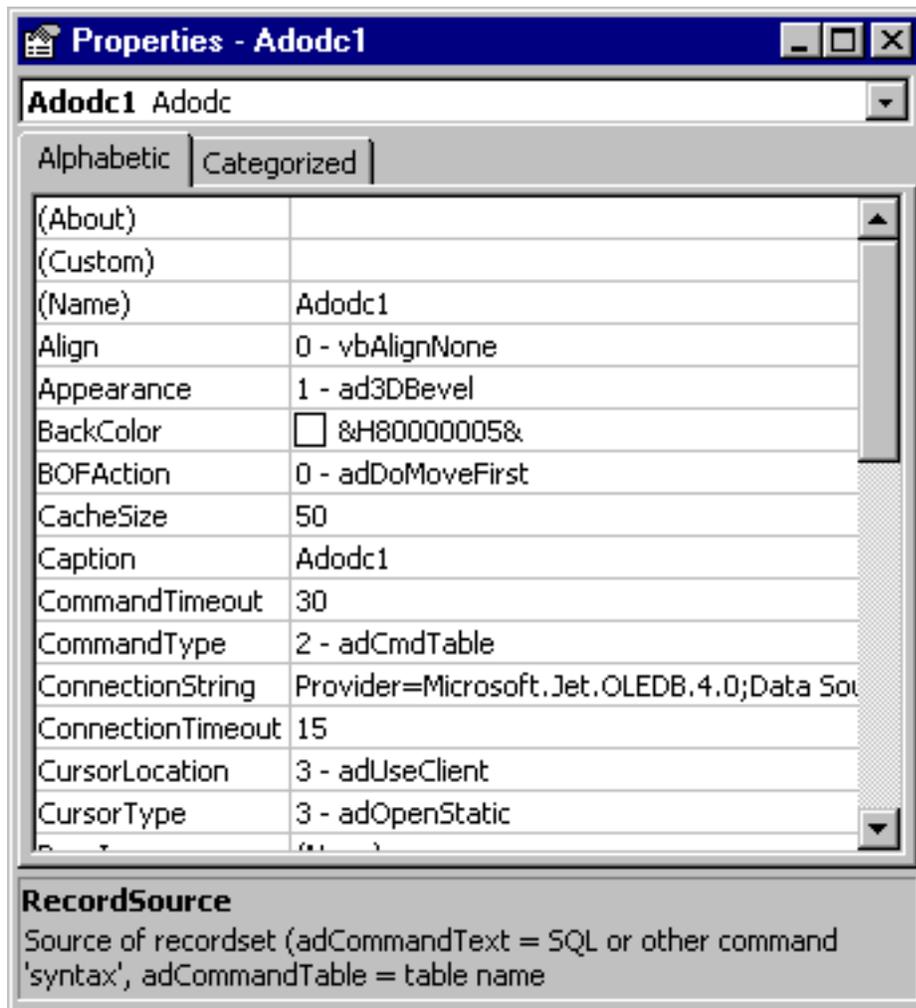
Selecting a Table type recordset is the easiest way to connect to a database and create a recordset. We could have selected adCmdText, but then we would have to specify a SQL (Structured Query Language) statement in the Command Text box. That may be the topic of a future article.



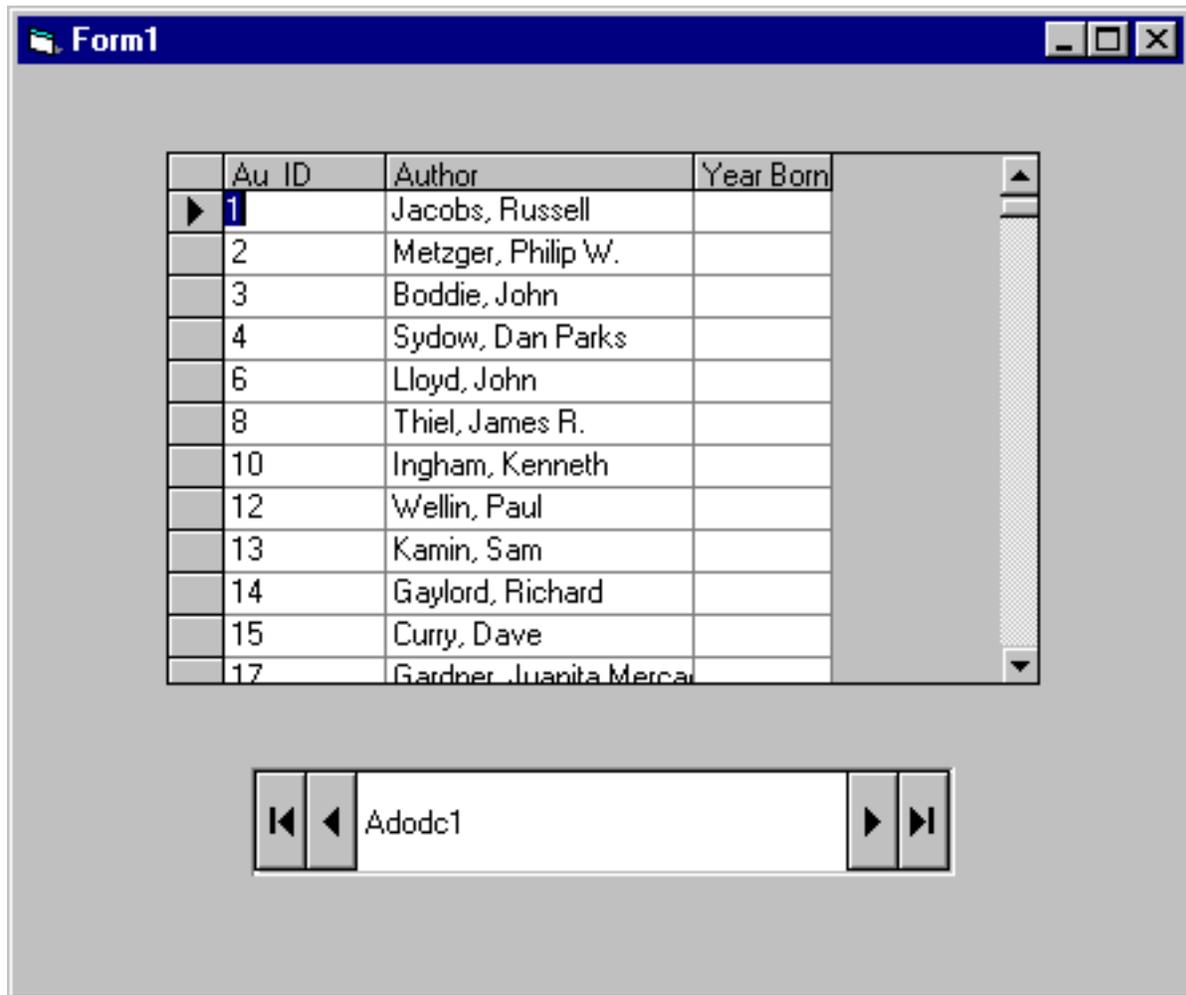
Once you've selected a table type Command Type, you'll see a list of the tables in the BIBLIO database displayed in the Table or Stored Procedure drop-down ListBox. Let's select Authors...



and then click on the Apply button. This will close the wizard, and you should see the Properties Window for the Data Control appear with the ConnectionString, Command Type and RecordSource Properties filled in

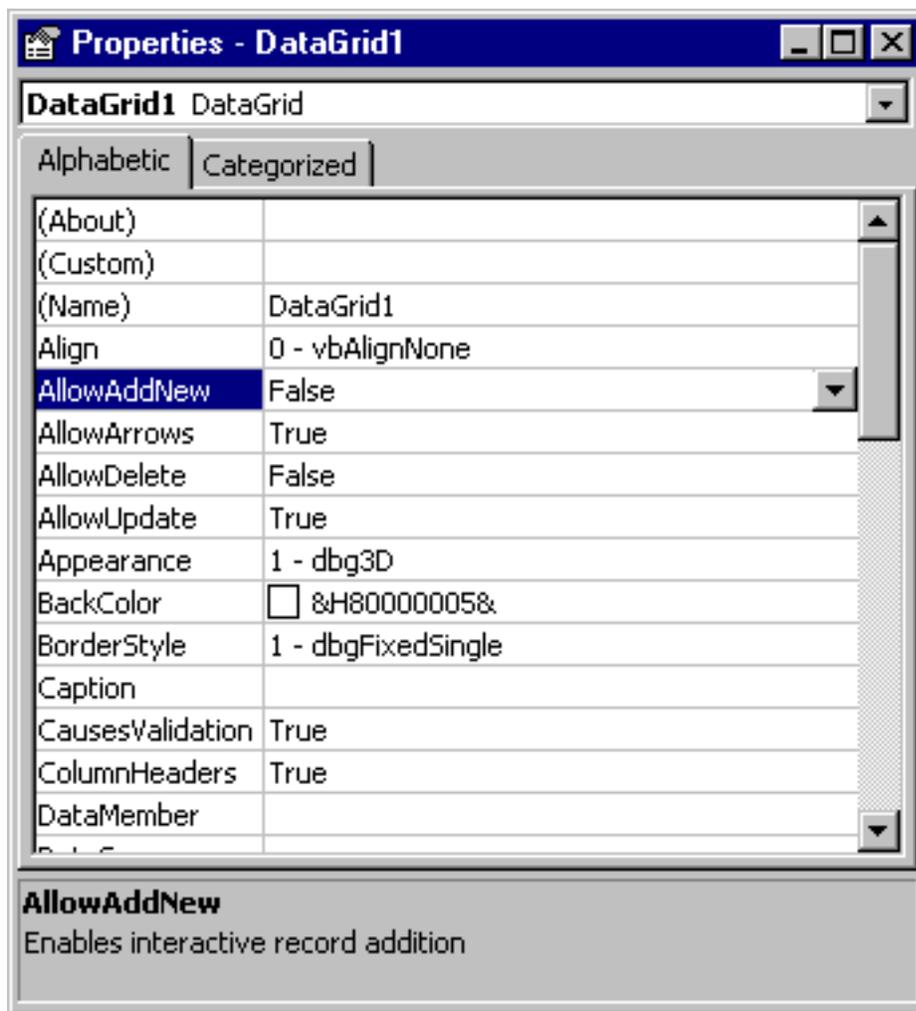


Having provided values for these properties, and 'bound' the DataGrid to our Data Control, pure magic will occur if we run this program. The DataGrid will be populated with values from the Authors table of the BIBLIO Database---without having written a single line of code!



---if you don't (and this can sometimes happen if you forget to click on the Apply button) just start the process again by clicking on the Custom Property of the Data Control.

With a connection to the database established, you can use the Grid to add new records, and update and delete existing records---provided the AllowAddNew, AllowDelete and AllowUpdate properties of the Data Grid are set to true.



Summary

I hope you enjoyed this article on using the ADO Data Control.