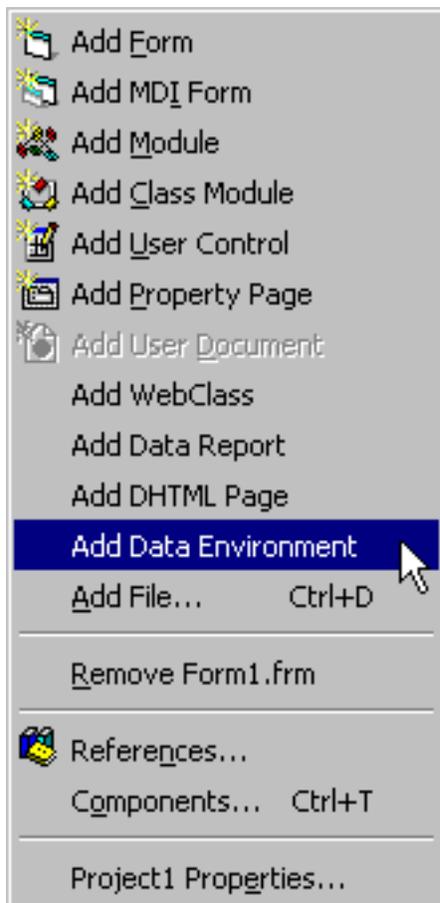
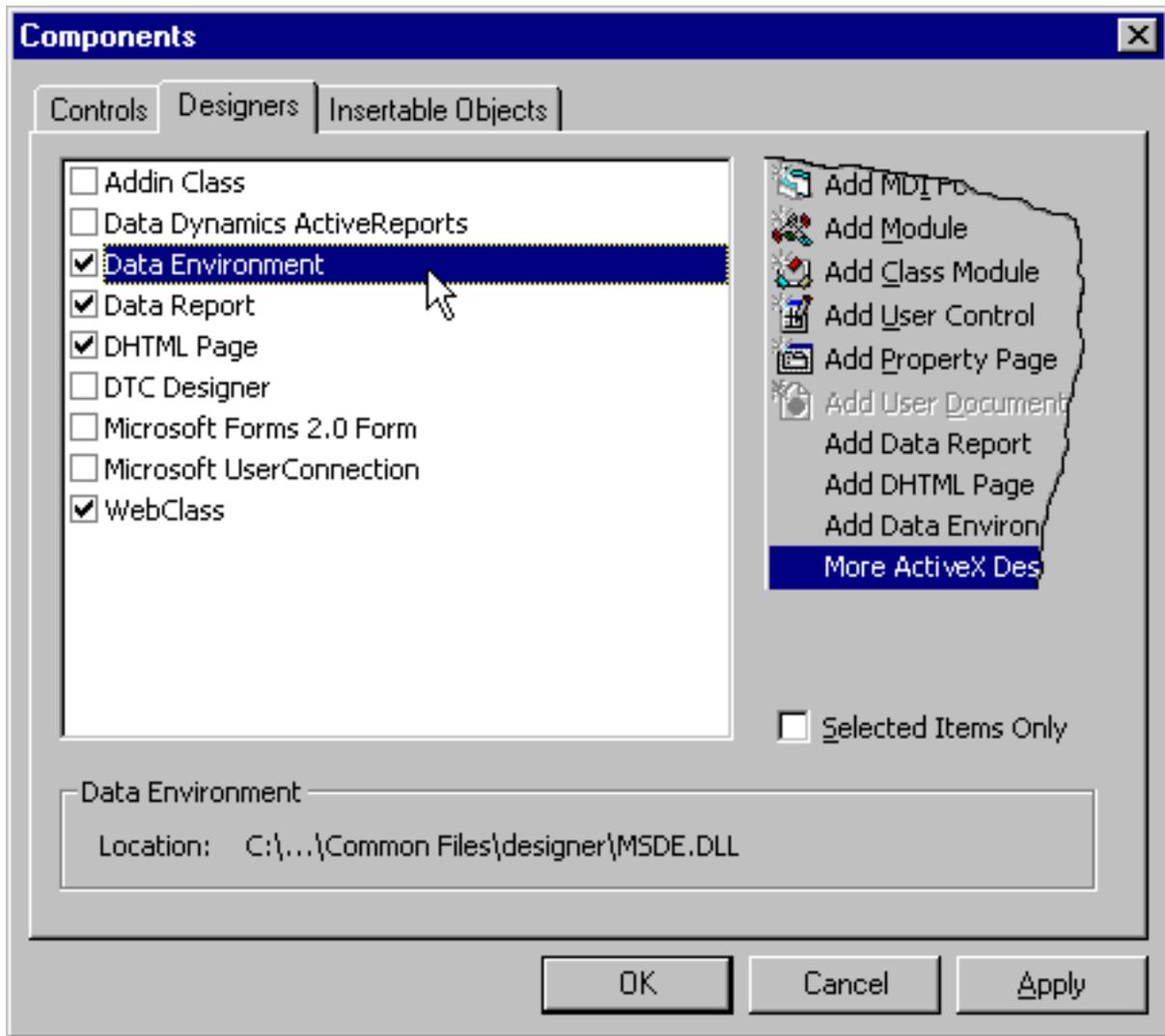


Use the ADO Data Environment Designer to create an ADO Connection in Visual Basic 6

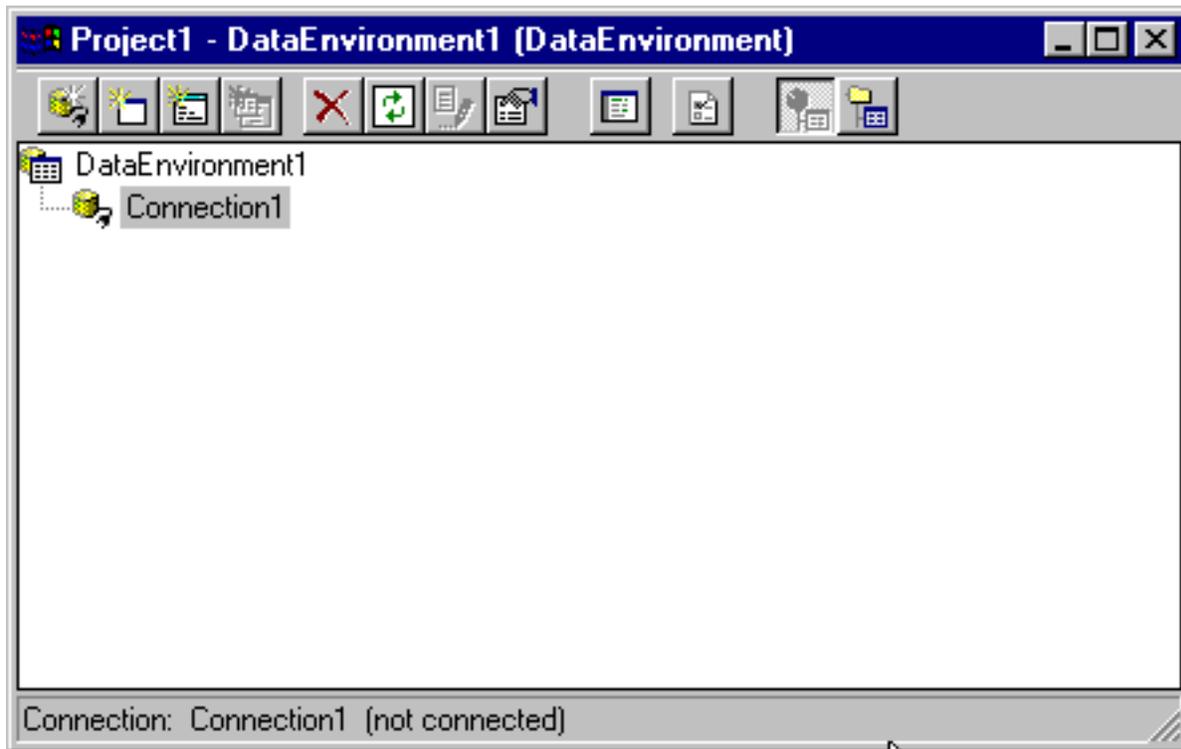
When we use the ADO Data Environment Designer to make our connection, we don't need a Data Control, so let's pretend that we've created a new project, placed the DataGrid in our toolbox, and placed it on our form. Now we need to find and startup the Data Environment Designer. To do that, select Project-Add Data Designer from the Visual Basic Menu Bar ...



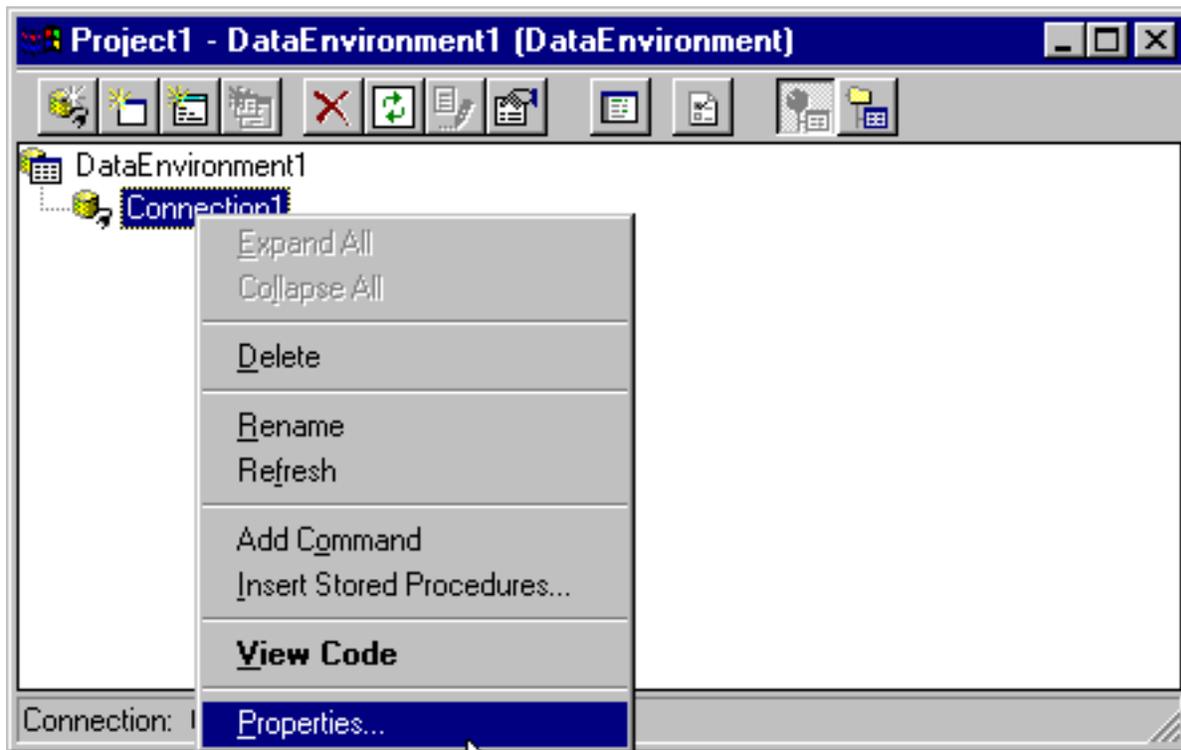
If you don't see Add Data Environment as an option, select Project-Components from the Visual Basic Menu Bar, click on the Designers Tab, and locate and select the Data Environment there...



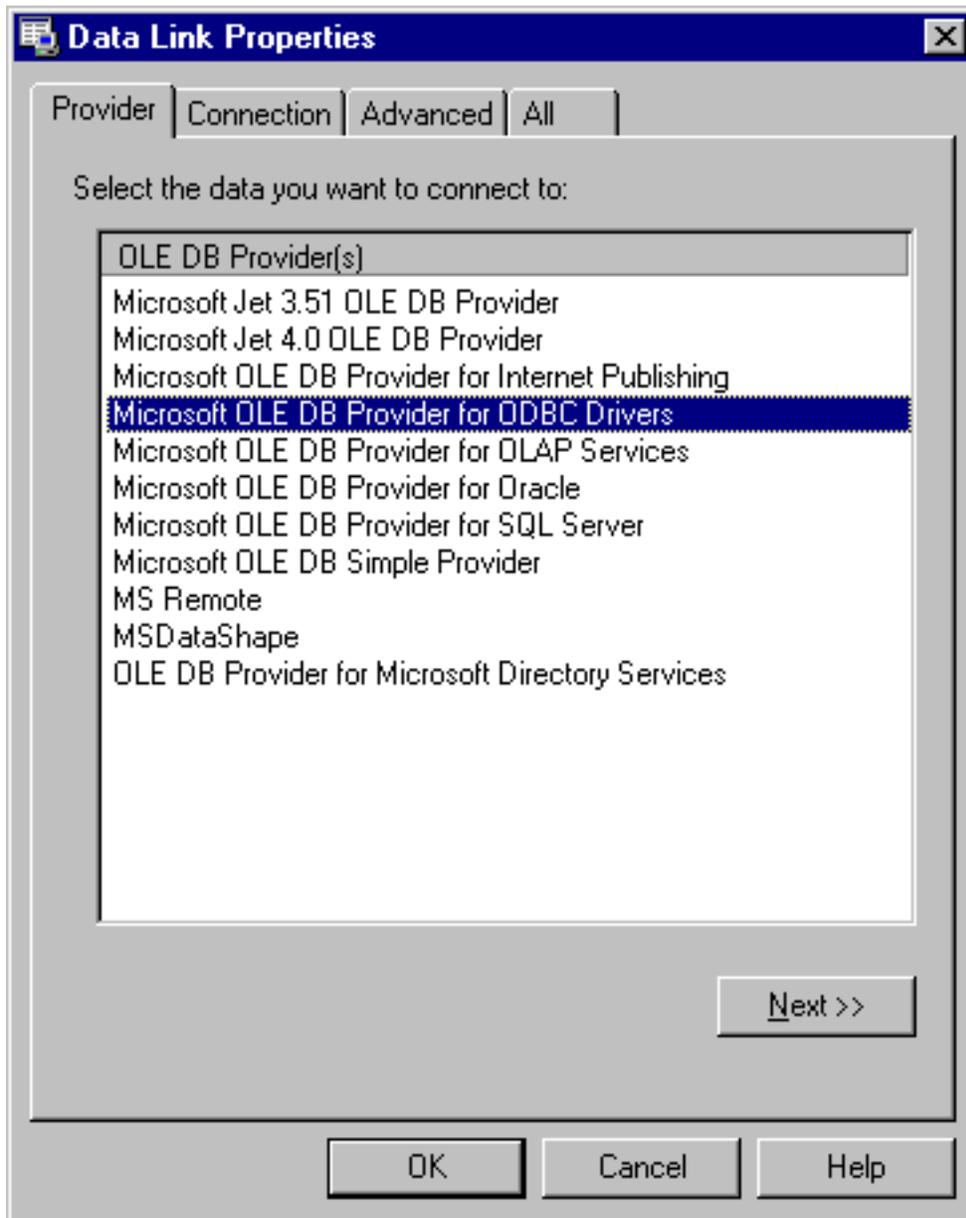
Once you select Add Data Environment from the Project menu, you should see this window...



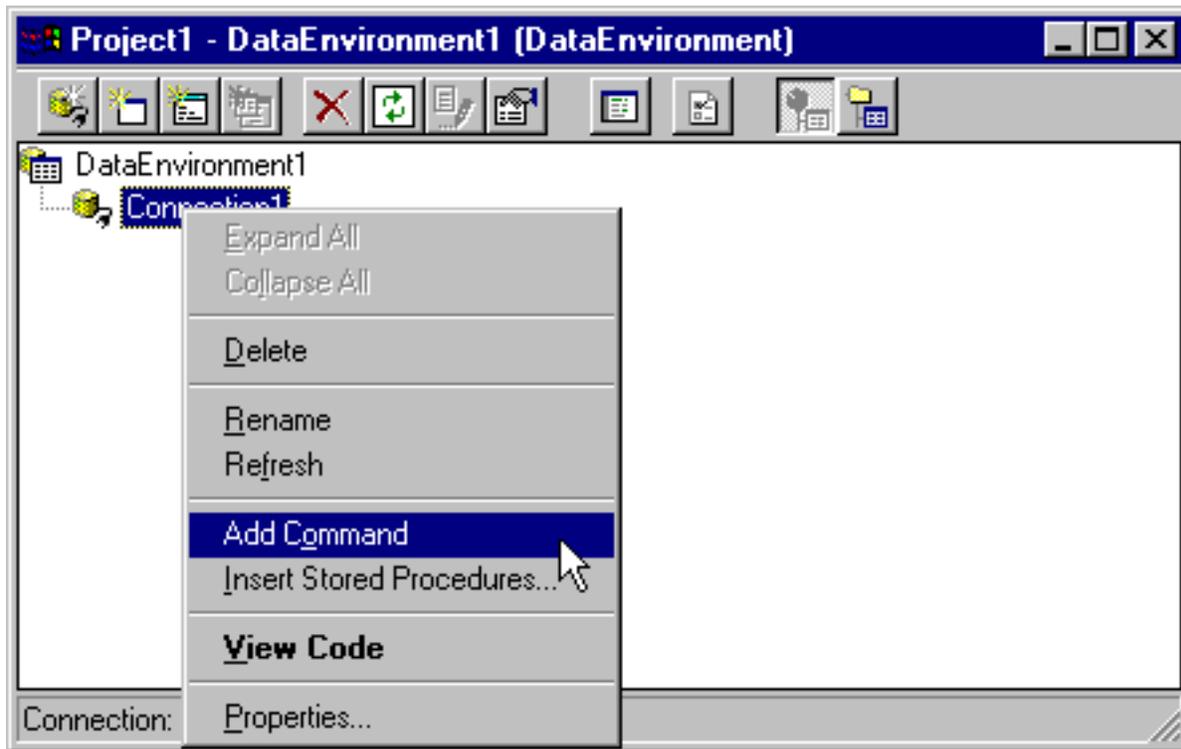
Right-click on the Connection object in the Data Environment Window and select Properties...



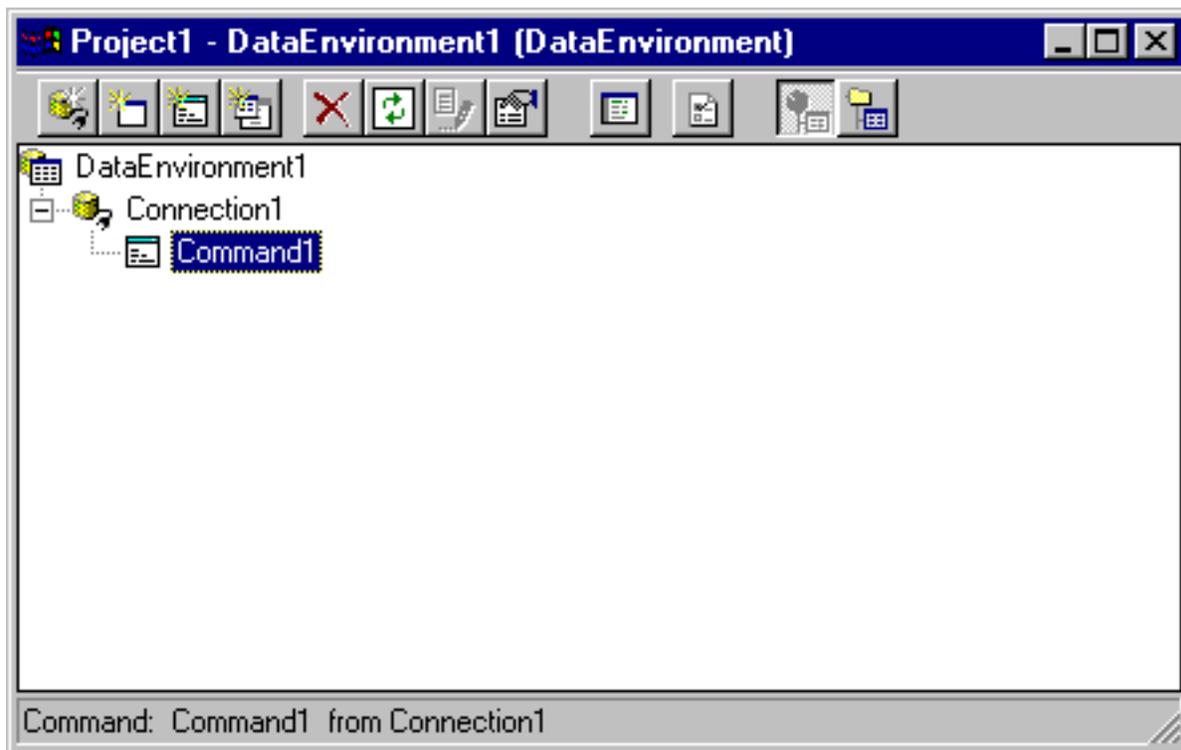
This window will appear...



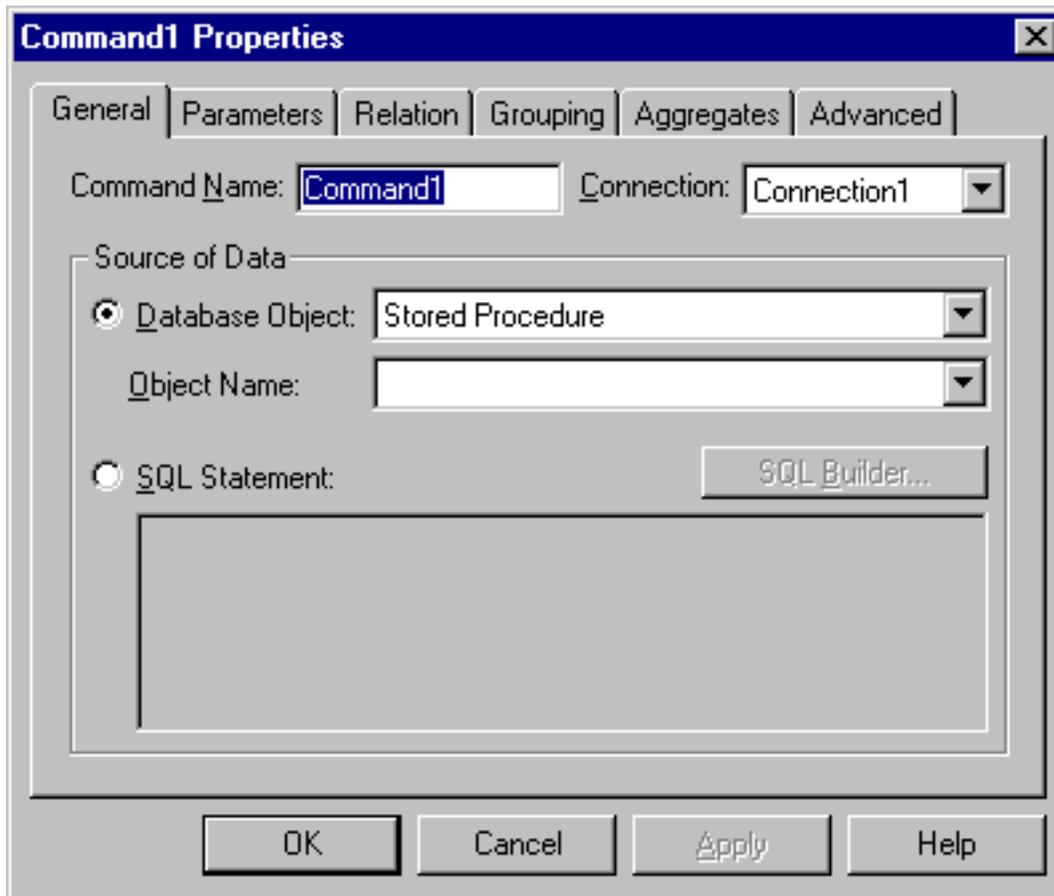
Look familiar? It should, it's the same wizard we saw when we created the connection using the Data Control. I'll repeat those same steps here (but in the interest of space I won't show you). Now we have a connection, but no recordset. Using the Data Environment, we must create a recordset via a Command Object. To create the Command Object, right-click on the Connection Object and select Add Command from the menu...



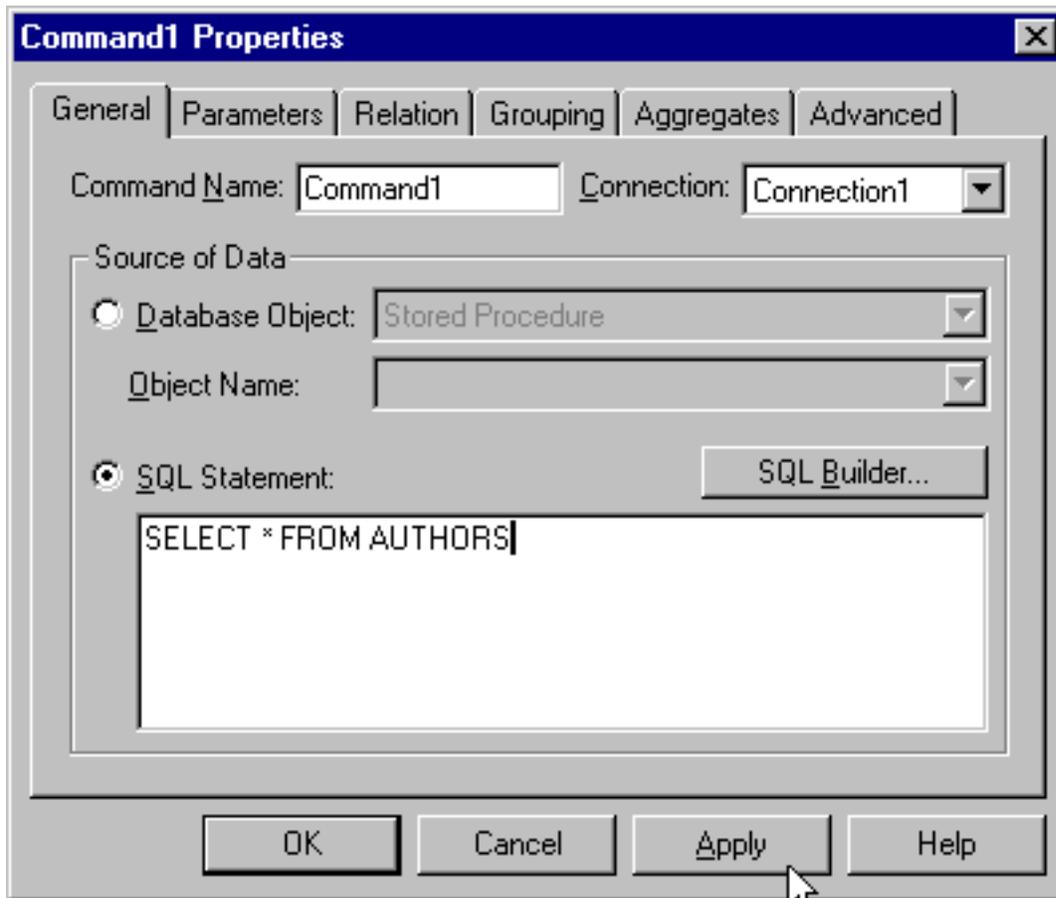
A Command object will appear in the Data Environment...



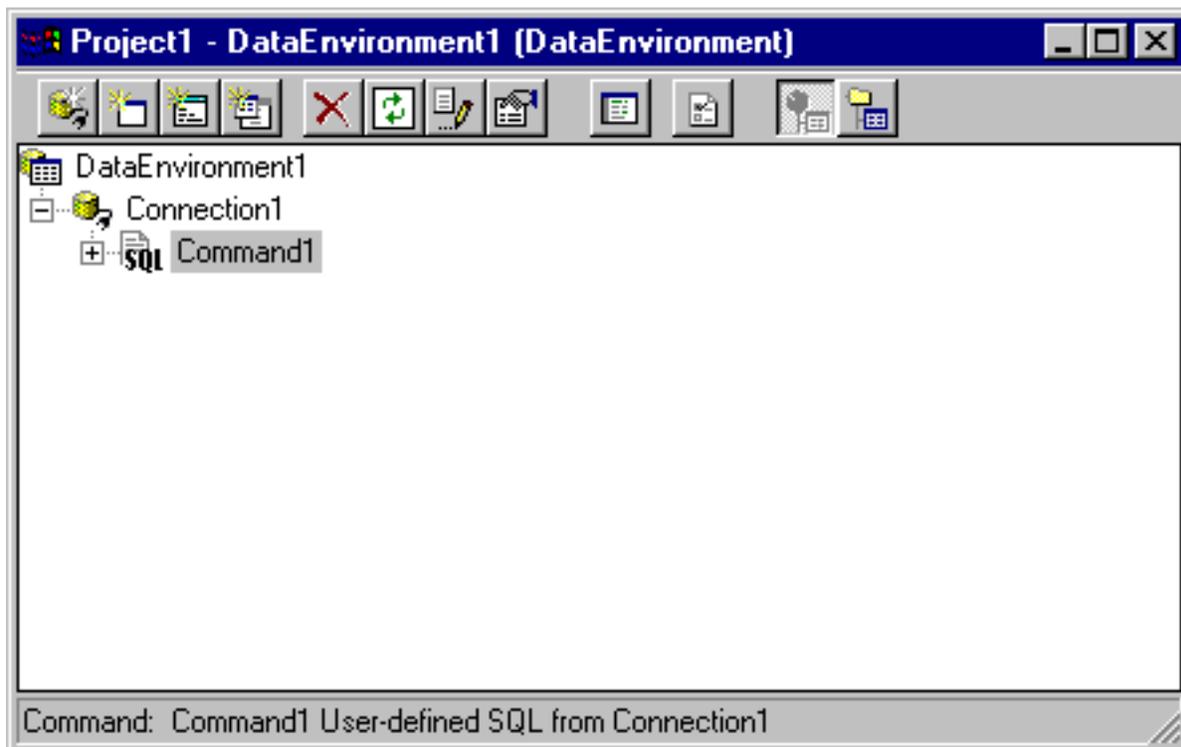
Right-click on it, and select Properties. This window will appear...



We'll use a SQL Statement to create the underlying recordset (if you are not familiar with SQL, click on the SQL Builder button)...

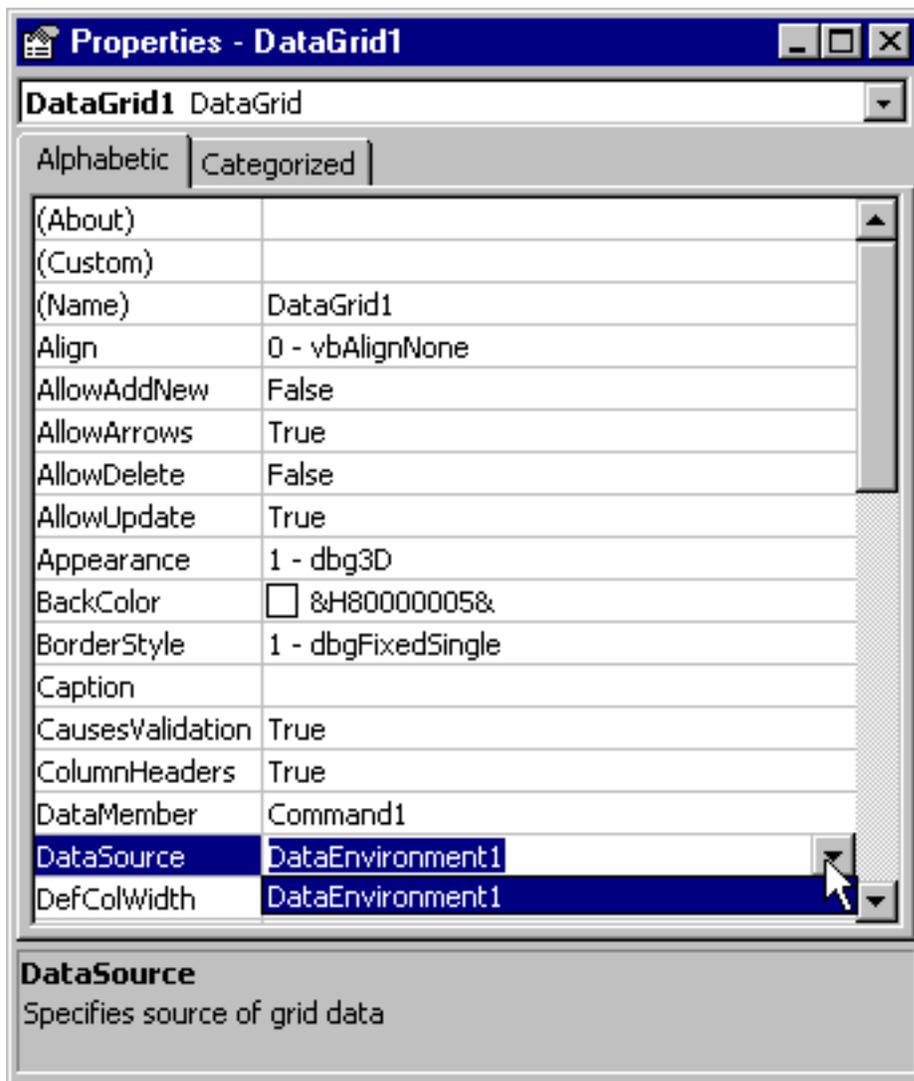


and after clicking on the Apply Button, we now have a properly defined Data Environment...



At this point, to 'bind' our DataGrid to the Data Environment Object that we've

just created, all we need to do open up the Properties Window for the Data Grid and specify the DataEnvironment as our Data Source and the Command Object in the DataMember property of the Data Grid...



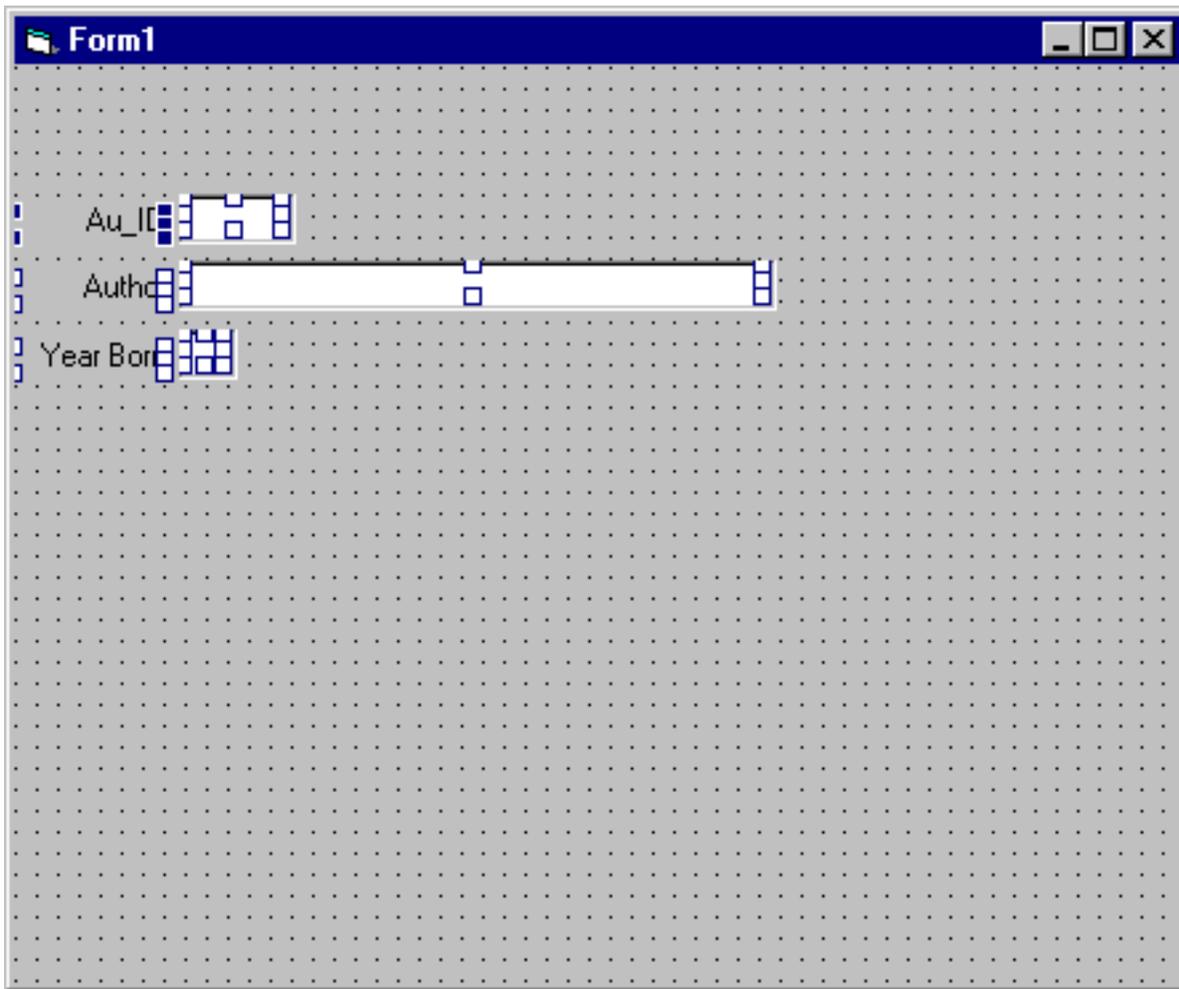
Now if we run the program, we'll find that the DataGrid is now 'bound' to our DataEnvironment...

	Au ID	Author	Year Born
▶	1	Jacobs, Russell	
	2	Metzger, Philip W.	
	3	Boddie, John	
	4	Sydow, Dan Parks	
	6	Lloyd, John	
	8	Thiel, James R.	
	10	, John	
	12	Wellin, Paul	
	13	Kamin, Sam	
	14	Gaylord, Richard	
	15	Curry, Dave	
	17	Gardner, Juanita Mercar	

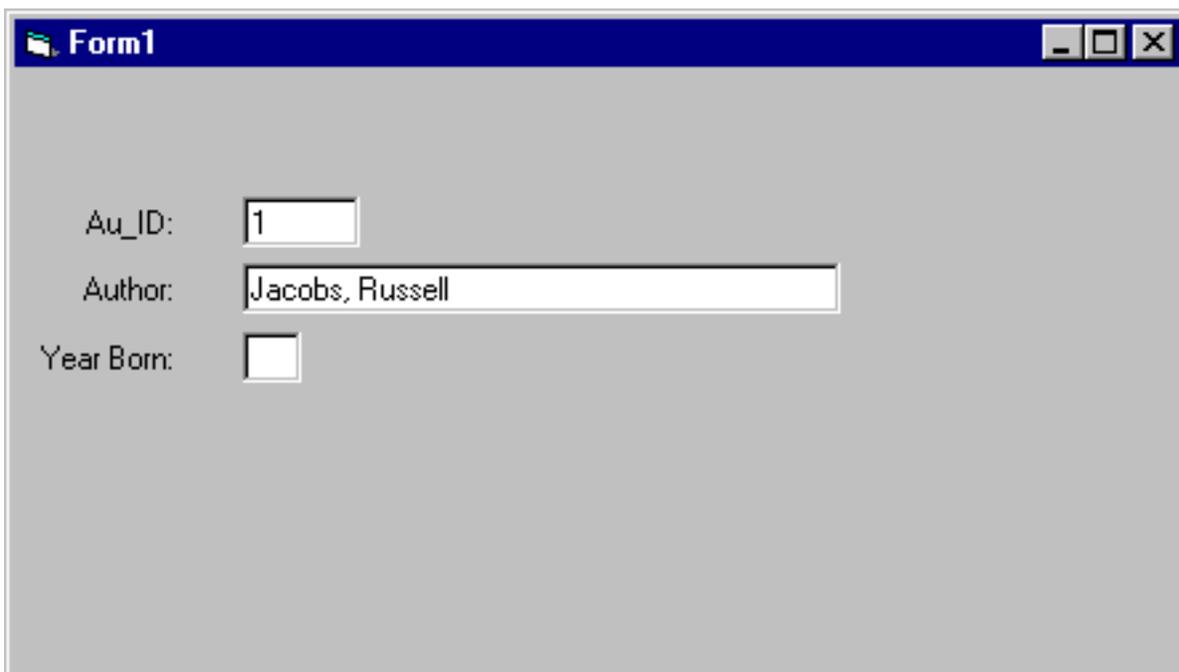
At this point, you may be thinking that the DataEnvironment really hasn't bought you all that much—after all, we can achieve the same functionality with the Data Control. That's true, but here's something the Data Control can never do for us. Suppose, instead of using a Data Grid to display the records in our Recordset, we'd like to display them in Textboxes?

It's pretty easy to display data in Textboxes using a Data Control—you just place labels and Textboxes on a form, and individually set the DataSource and DataField Properties of each Textbox to point to the Data Control. The DataEnvironment makes that even easier.

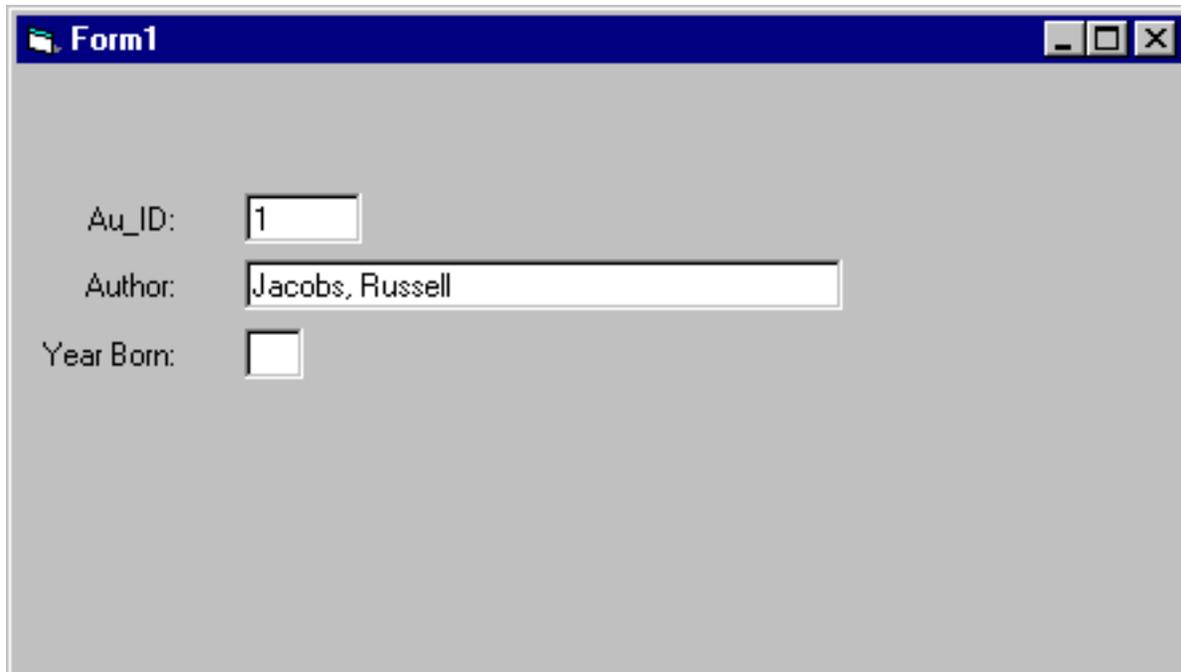
For now, let's delete the Data Grid from this form, and then click and drag the Command Object from the DataEnvironment Designer to our form---watch what happens!



As you can see, Visual Basic automatically creates Label and Textbox controls for each 'field' in the Recordset that is produced by the Command Object. This can be a tremendous time saver, especially if the Recordset contains a lot of fields. Let's take some time to make the form a bit 'tidier'...



...and then run the program. Voila, Data!

A screenshot of a Visual Basic form window titled "Form1". The form has a light gray background and contains three data entry fields. The first field is labeled "Au_ID:" and contains the number "1". The second field is labeled "Author:" and contains the text "Jacobs, Russell". The third field is labeled "Year Born:" and is currently empty. The form has a standard Windows-style title bar with minimize, maximize, and close buttons.

Not bad for the little work we had to do to get this—but there's a big problem—there's no way to move from record to record, as we have when we use a Data Control.

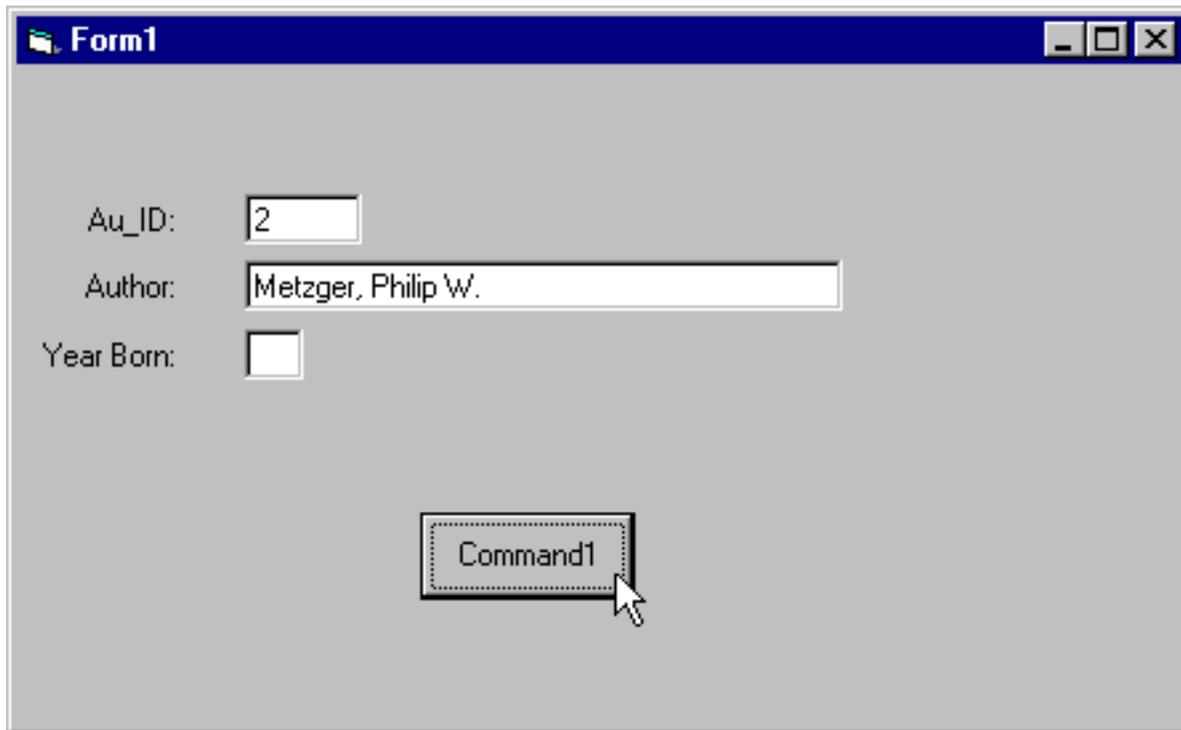
Let's place a command button on the form, and place this code in its Click Event Procedure—this will enable us to move to the next record in the Recordset...

```
Private Sub Command1_Click()
```

```
DataEnvironment1.rsCommand1.MoveNext
```

```
End Sub
```

Now if we run the program, and click on the Command Button, we'll easily be able to move to the next record in the Recordset...



The image shows a screenshot of a Visual Basic 6 form window titled "Form1". The form has a light gray background and a dark blue title bar with standard Windows window controls (minimize, maximize, close). There are three input fields on the left side: "Au_ID:" with a text box containing the number "2", "Author:" with a text box containing "Metzger, Philip W.", and "Year Born:" with an empty text box. In the center of the form, there is a button labeled "Command1" with a mouse cursor pointing at it.

Summary

I hope you enjoyed this article on using the Data Environment.