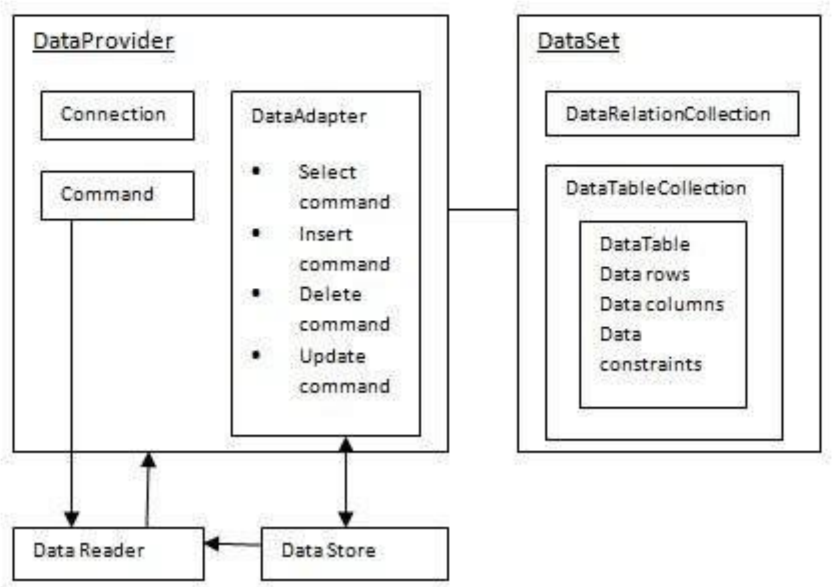
VB.Net - Database Access

Applications communicate with a database, firstly, to retrieve the data stored there and present it in a user-friendly way, and secondly, to update the database by inserting, modifying and deleting data.

Microsoft ActiveX Data Objects.Net (ADO.Net) is a model, a part of the .Net framework that is used by the .Net applications for retrieving, accessing and updating data.

ADO.Net Object Model

ADO.Net object model is nothing but the structured process flow through various components. The object model can be pictorially described as −



The data residing in a data store or database is retrieved through the **data provider**. Various components of the data provider retrieve data for the application and update data.

An application accesses data either through a dataset or a data reader.

- **Datasets** store data in a disconnected cache and the application retrieves data from it.
- **Data readers** provide data to the application in a read-only and forward-only mode.

Data Provider

A data provider is used for connecting to a database, executing commands and retrieving data, storing it in a dataset, reading the retrieved data and updating the database.

The data provider in ADO.Net consists of the following four objects −

Sr.No.	Objects & Description
1	**Connection** This component is used to set up a connection with a data source.
2	**Command** A command is a SQL statement or a stored procedure used to retrieve, insert, delete or modify data in a data source.
3	**DataReader** Data reader is used to retrieve data from a data source in a read-only and forward-only mode.
4	**DataAdapter** This is integral to the working of ADO.Net since data is transferred to and from a database through a data adapter. It retrieves data from a database into a dataset and updates the database. When changes are made to the dataset, the changes in the database are actually done by the data adapter.

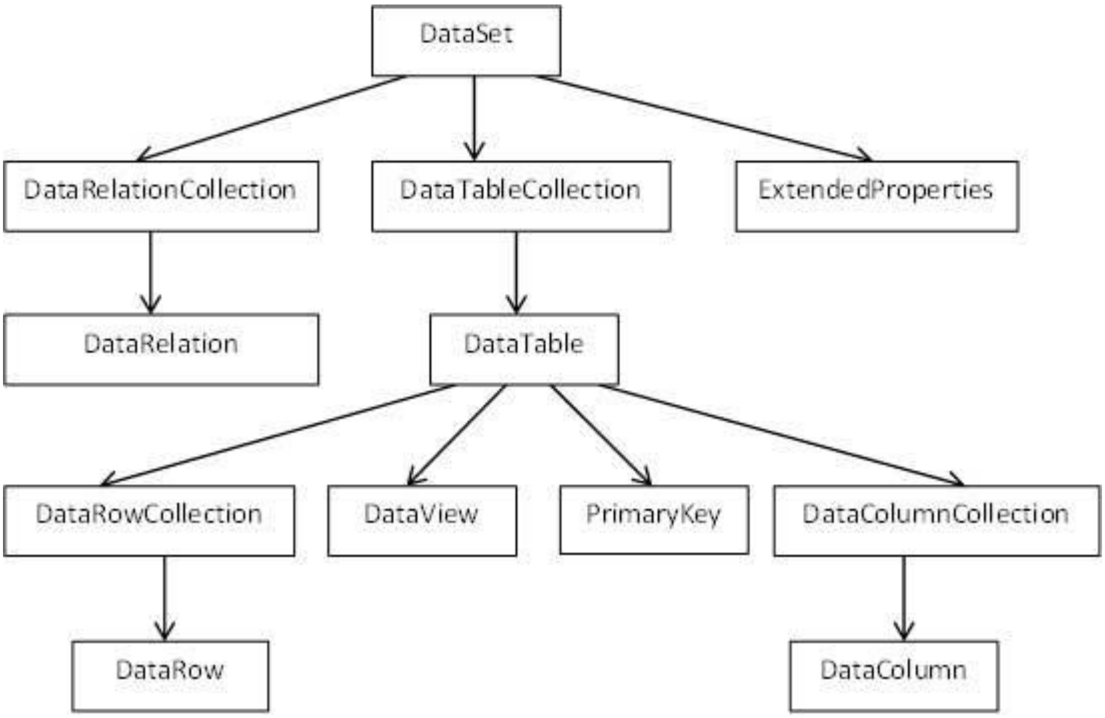
There are following different types of data providers included in ADO.Net

- The .Net Framework data provider for SQL Server - provides access to Microsoft SQL Server.
- The .Net Framework data provider for OLE DB - provides access to data sources exposed by using OLE DB.
- The .Net Framework data provider for ODBC - provides access to data sources exposed by ODBC.
- The .Net Framework data provider for Oracle - provides access to Oracle data source.
- The EntityClient provider - enables accessing data through Entity Data Model (EDM) applications.

DataSet

DataSet is an in-memory representation of data. It is a disconnected, cached set of records that are retrieved from a database. When a connection is established with the database, the data adapter creates a dataset and stores data in it. After the data is retrieved and stored in a dataset, the connection with the database is closed. This is called the 'disconnected architecture'. The dataset works as a virtual database containing tables, rows, and columns.

The following diagram shows the dataset object model −



The DataSet class is present in the **System.Data** namespace. The following table describes all the components of DataSet −

Sr.No.	Components & Description
1	**DataTableCollection** It contains all the tables retrieved from the data source.
2	**DataRelationCollection** It contains relationships and the links between tables in a data set.
3	**ExtendedProperties** It contains additional information, like the SQL statement for retrieving data, time of retrieval, etc.
4	**DataTable** It represents a table in the DataTableCollection of a dataset. It consists of the DataRow and DataColumn objects. The DataTable objects are case-sensitive.
5	**DataRelation**

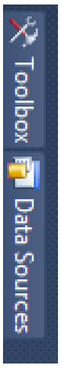
	It represents a relationship in the DataRelationshipCollection of the dataset. It is used to relate two DataTable objects to each other through the DataColumn objects.
6	**DataRowCollection** It contains all the rows in a DataTable.
7	**DataView** It represents a fixed customized view of a DataTable for sorting, filtering, searching, editing and navigation.
8	**PrimaryKey** It represents the column that uniquely identifies a row in a DataTable.
9	**DataRow** It represents a row in the DataTable. The DataRow object and its properties and methods are used to retrieve, evaluate, insert, delete, and update values in the DataTable. The NewRow method is used to create a new row and the Add method adds a row to the table.
10	**DataColumnCollection** It represents all the columns in a DataTable.
11	**DataColumn** It consists of the number of columns that comprise a DataTable.

Connecting to a Database

The .Net Framework provides two types of Connection classes −

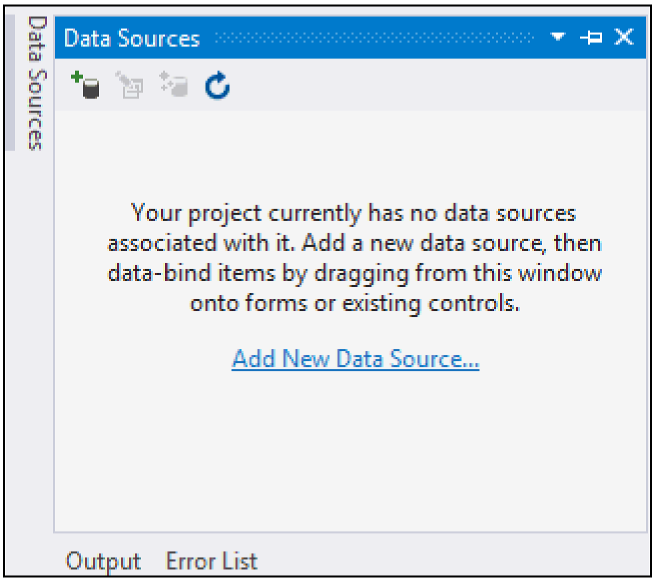
- **SqlConnection** − designed for connecting to Microsoft SQL Server.
- **OleDbConnection** − designed for connecting to a wide range of databases, like Microsoft Access and Oracle.

Let's make a start on our Database project. Create a new project and give it the name **AddressBook**.
We need to select a Data Source. So click on Data Sources on the left of the Toolbox (If you can't see the tab, click **View > Other Windows > Data Sources**):



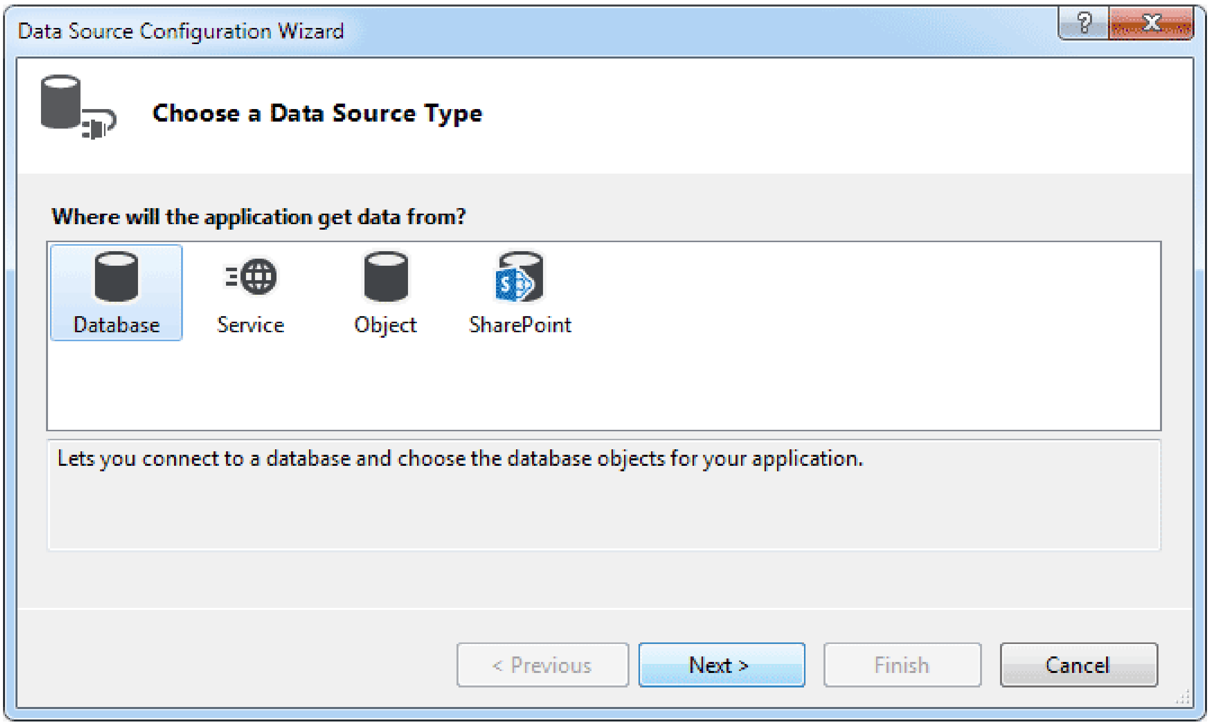
(In Visual Studion 2015/2017, click Data Sources again to get the Toolbox back.)

The Data Sources tab should look like this:



Your project currently has no data sources associated with it. Add a new data source, then data-bind items by dragging from this window onto forms or existing controls.

To Add a New Data Source, click on the link "Add New Data Source". When you do, you'll see the following:

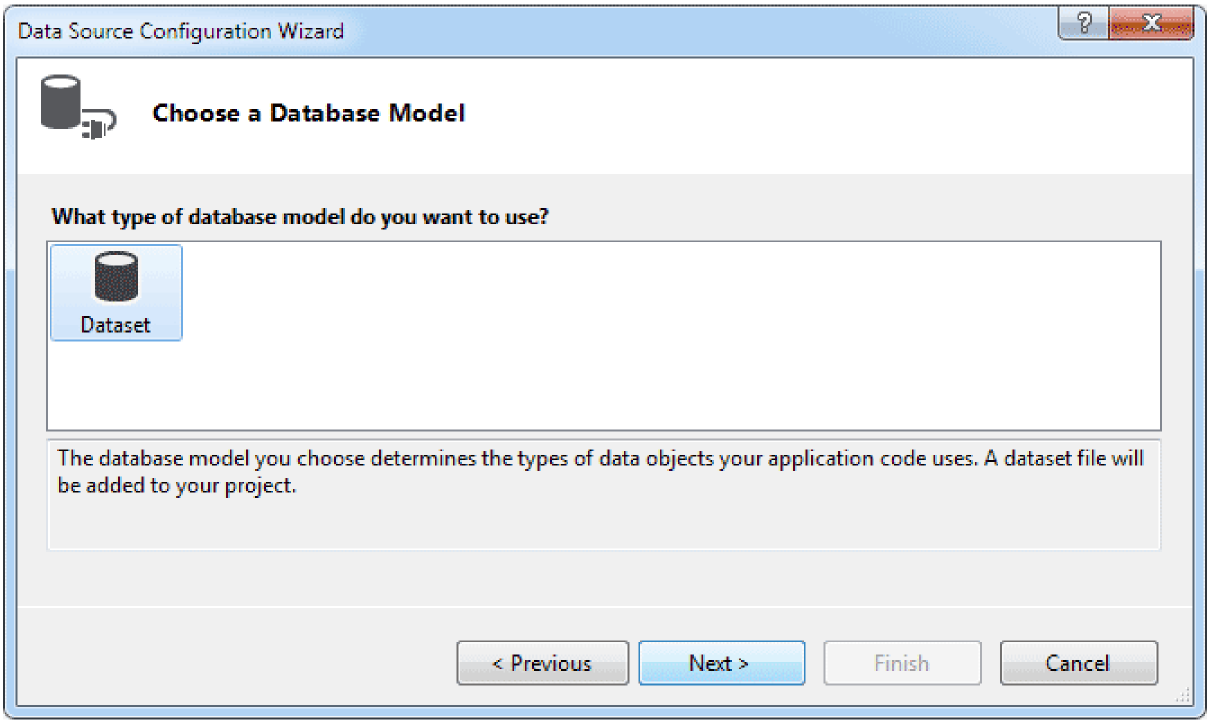


Choose a Data Source Type

Where will the application get data from?

Lets you connect to a database and choose the database objects for your application.

You want to connect to a Database. So select this option, and click Next. You'll see this screen appear:

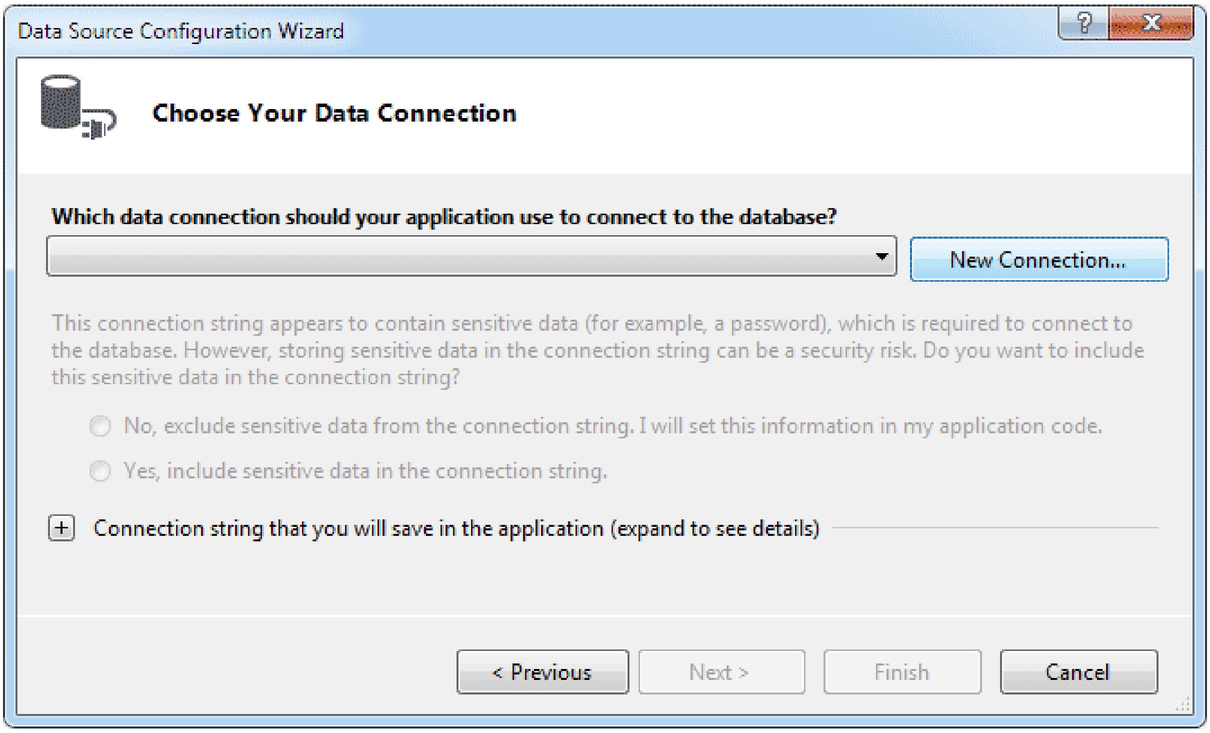


Choose a Database Model

What type of database model do you want to use?

The database model you choose determines the types of data objects your application code uses. A dataset file will be added to your project.

Select DataSet and click Next. You'll then see a Choose Your Data Conection screen:


Data Source Configuration Wizard

Choose Your Data Connection

Which data connection should your application use to connect to the database?

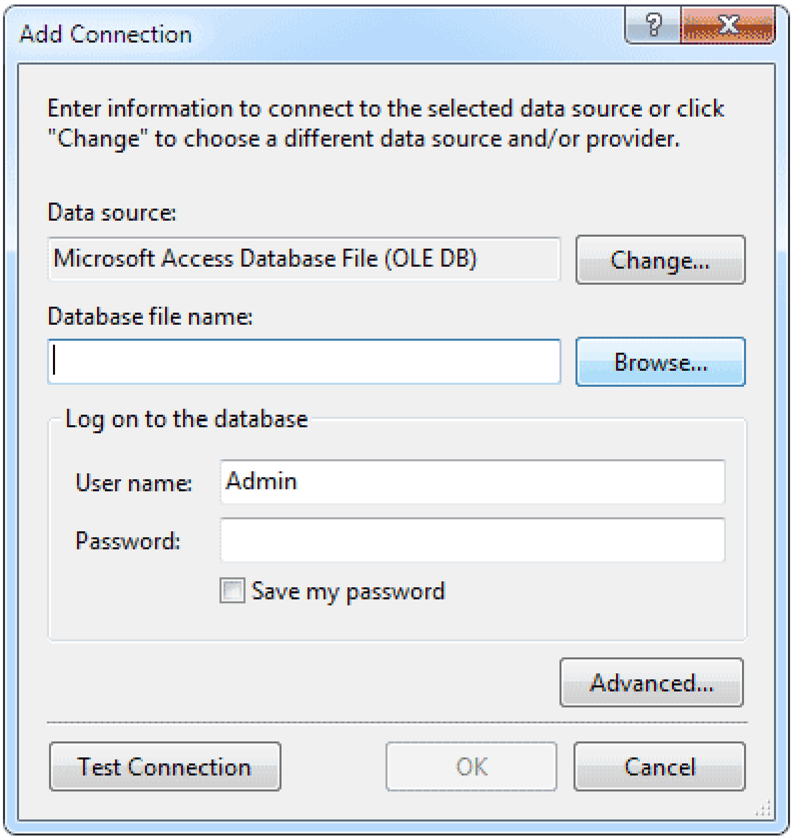
This connection string appears to contain sensitive data (for example, a password), which is required to connect to the database. However, storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?

○ No, exclude sensitive data from the connection string. I will set this information in my application code.

○ Yes, include sensitive data in the connection string.

⊞ Connection string that you will save in the application (expand to see details)

Click the **New Connection** button and another dialogue box pops up - - the Add Connection dialogue box:


Add Connection

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source: Microsoft Access Database File (OLE DB)

Database file name:

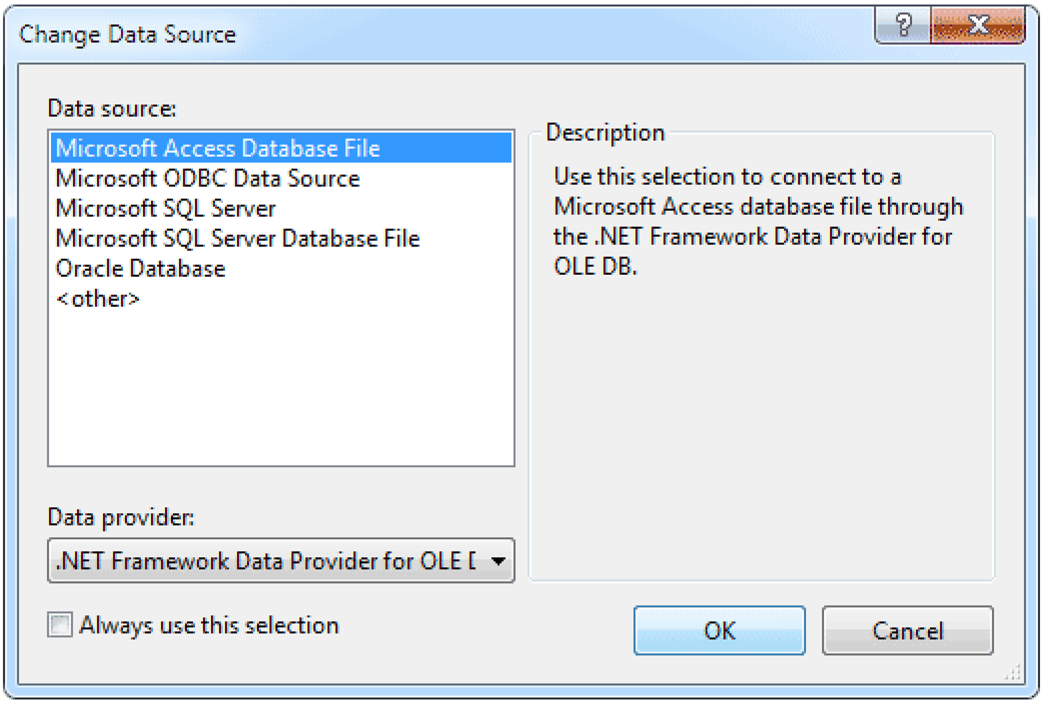
Log on to the database

User name: Admin

Password:

☐ Save my password

If your Data Source doesn't say Microsoft Access Dataabase File (OLE DB) then click the **Change** button to see this screen:



Change Data Source

Data source:

Microsoft Access Database File
Microsoft ODBC Data Source
Microsoft SQL Server
Microsoft SQL Server Database File
Oracle Database
<other>

Description

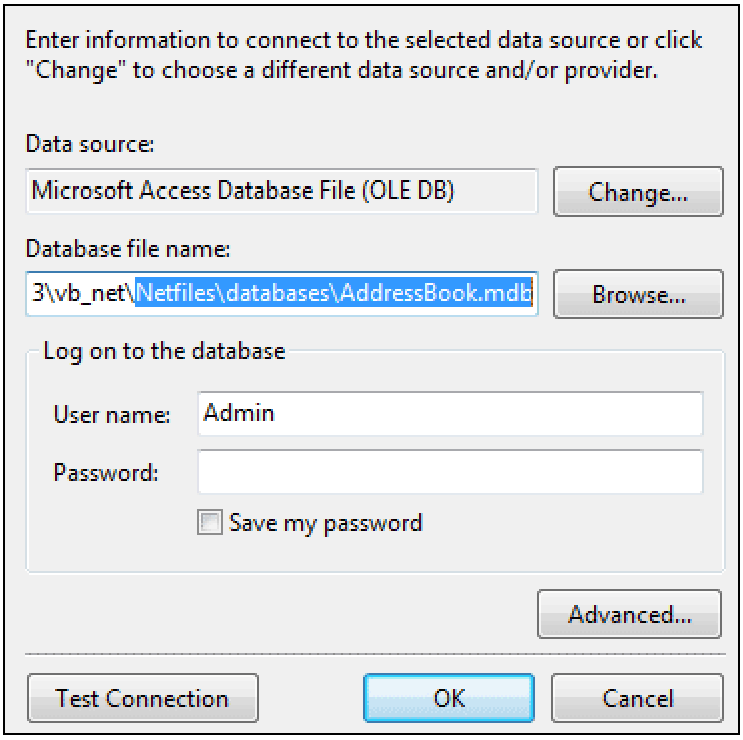
Use this selection to connect to a Microsoft Access database file through the .NET Framework Data Provider for OLE DB.

Data provider: .NET Framework Data Provider for OLE [

□ Always use this selection

Select Microsoft Access Database File, then click Continue (or OK in some version of Visual Studio). You'll be returned to the Add connection dialogue box.

Click the Browse button and navigate to where on your computer you downloaded our Access Database called AddressBook.mdb:



Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source: Microsoft Access Database File (OLE DB)

Database file name: 3\vb_net\Netfiles\databases\AddressBook.mdb

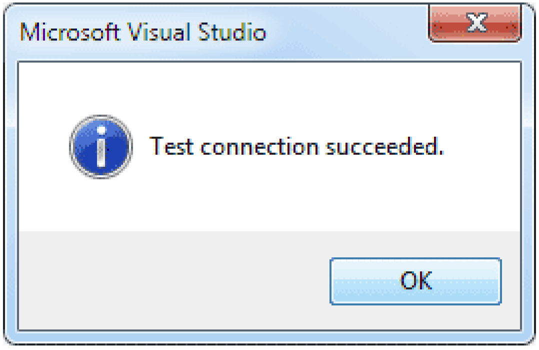
Log on to the database

User name: Admin

Password:

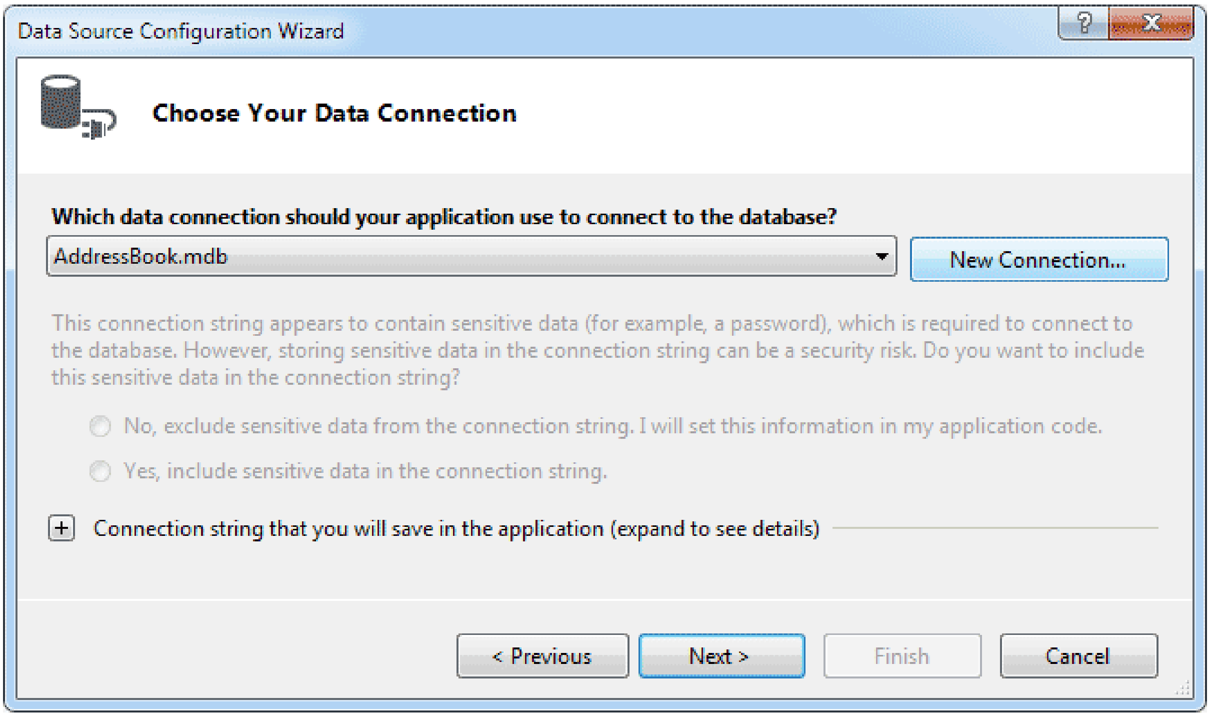
□ Save my password

Click **Test Connection** to see if everything is OK, and you'll hopefully see this:


Microsoft Visual Studio

Test connection succeeded.

Click the OK button, then click the OK button on the Add Connection dialogue box as well. You will be returned to the Data Source Configuration Wizard, which should now look like this:


Data Source Configuration Wizard

Choose Your Data Connection

Which data connection should your application use to connect to the database?

AddressBook.mdb

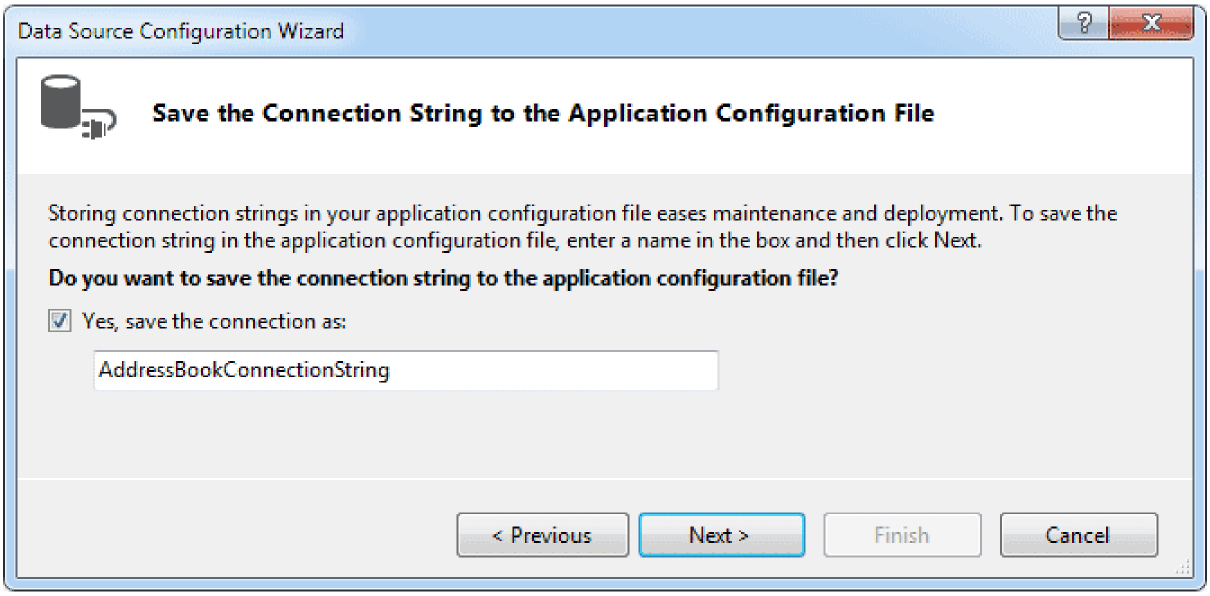
This connection string appears to contain sensitive data (for example, a password), which is required to connect to the database. However, storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?

○ No, exclude sensitive data from the connection string. I will set this information in my application code.

○ Yes, include sensitive data in the connection string.

+ Connection string that you will save in the application (expand to see details)

Click Next to move to the next step of the Wizard. You may see a message box appear, however. Click No on the message box to stop VB copying the database each time it runs. You should then see this:



Save the Connection String to the Application Configuration File

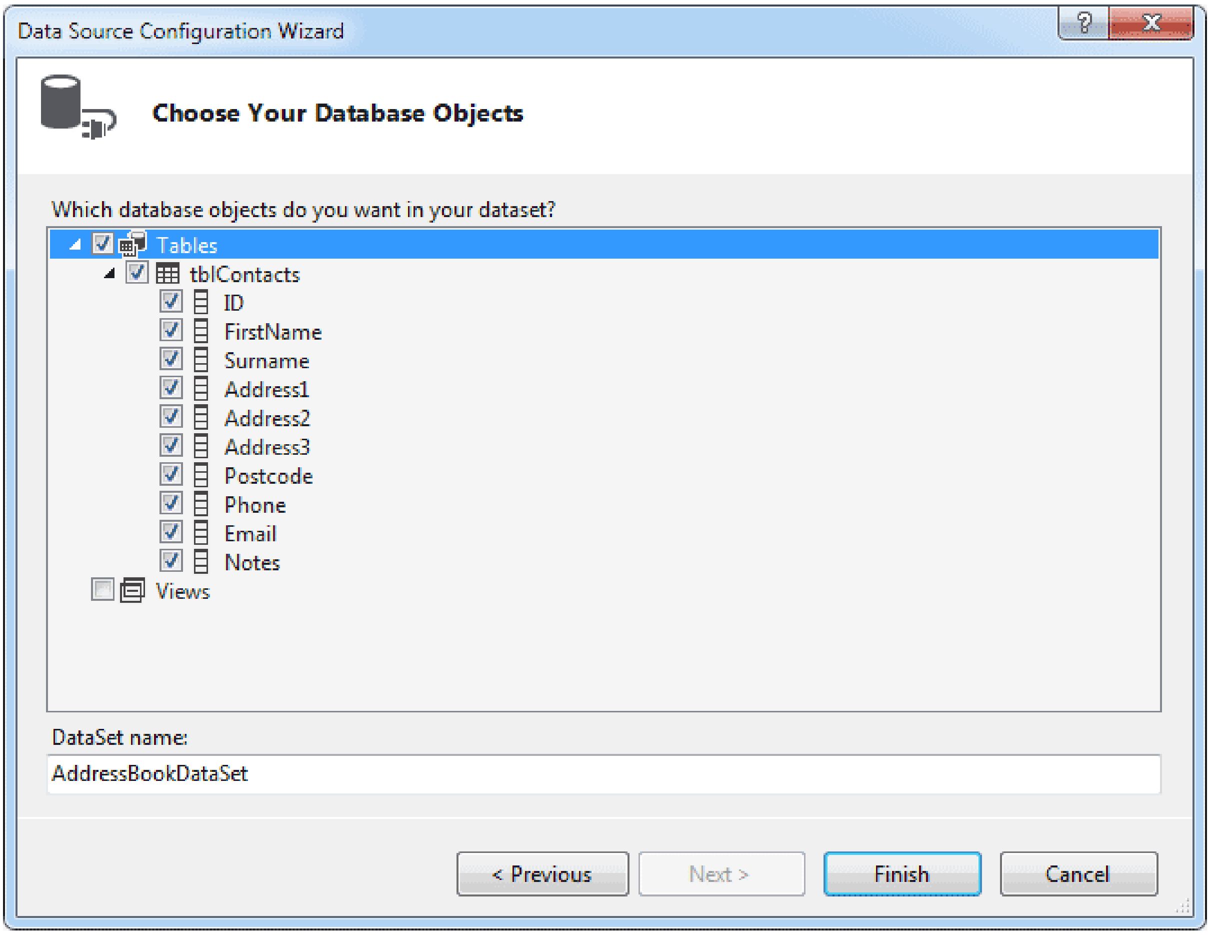
Storing connection strings in your application configuration file eases maintenance and deployment. To save the connection string in the application configuration file, enter a name in the box and then click Next.

Do you want to save the connection string to the application configuration file?

☑ Yes, save the connection as:

AddressBookConnectionString

Make sure there's a tick in the box for "Save the connection", and then click Next:



Choose Your Database Objects

Which database objects do you want in your dataset?

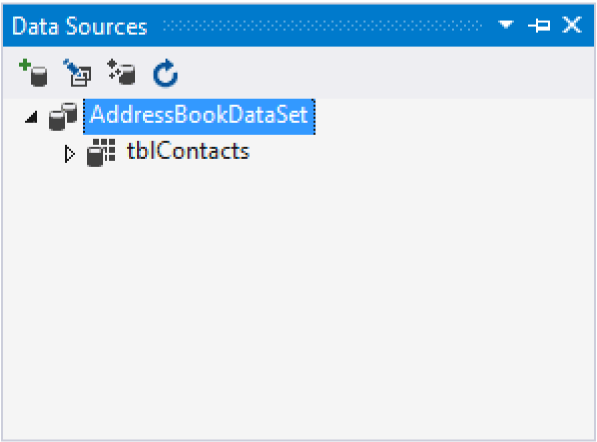
- ☑ Tables
  - ☑ tblContacts
    - ☑ ID
    - ☑ FirstName
    - ☑ Surname
    - ☑ Address1
    - ☑ Address2
    - ☑ Address3
    - ☑ Postcode
    - ☑ Phone
    - ☑ Email
    - ☑ Notes
- □ Views

DataSet name:

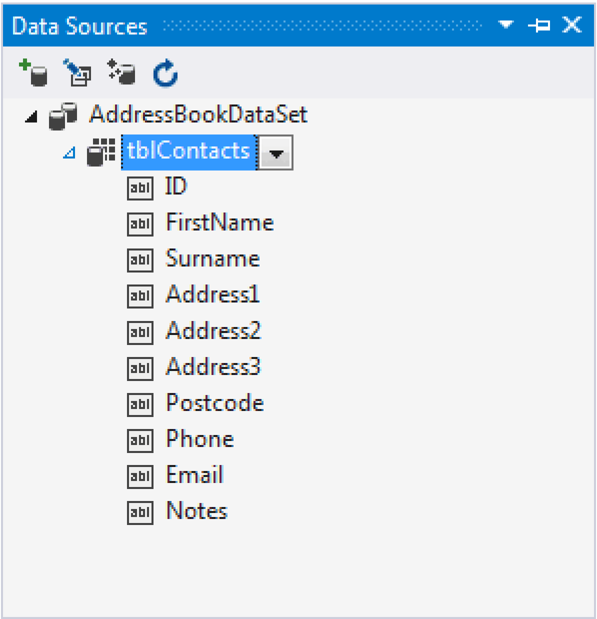
AddressBookDataSet

(If you can't see all the fields, click the arrow symbol on the left, next to Tables.)

Here, you can select which tables and fields you want. Tick the **Tables** box to include them all. (You can give your DataSet a different name, if you prefer. The name AddressBookDataSet is one the wizard comes up with.) Click Finish and you're done. When you are returned to your form, you should notice your new Data Source has been added:

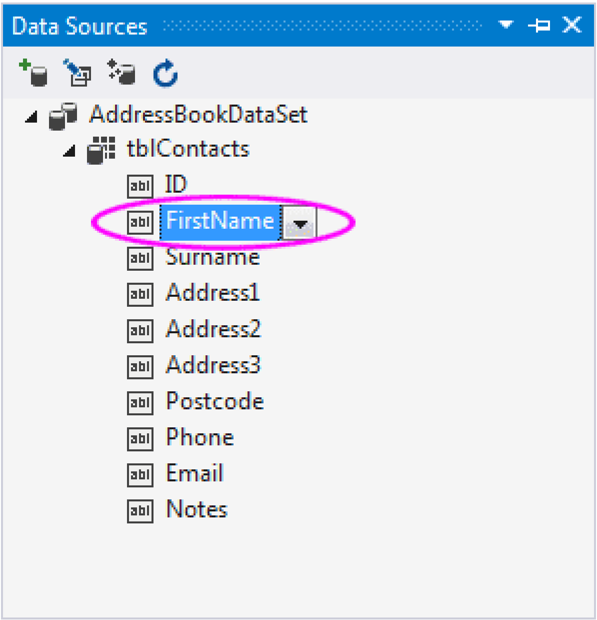


The Data Sources area now displays information about your database. Click the arrow symbol next to **tblContacts**:



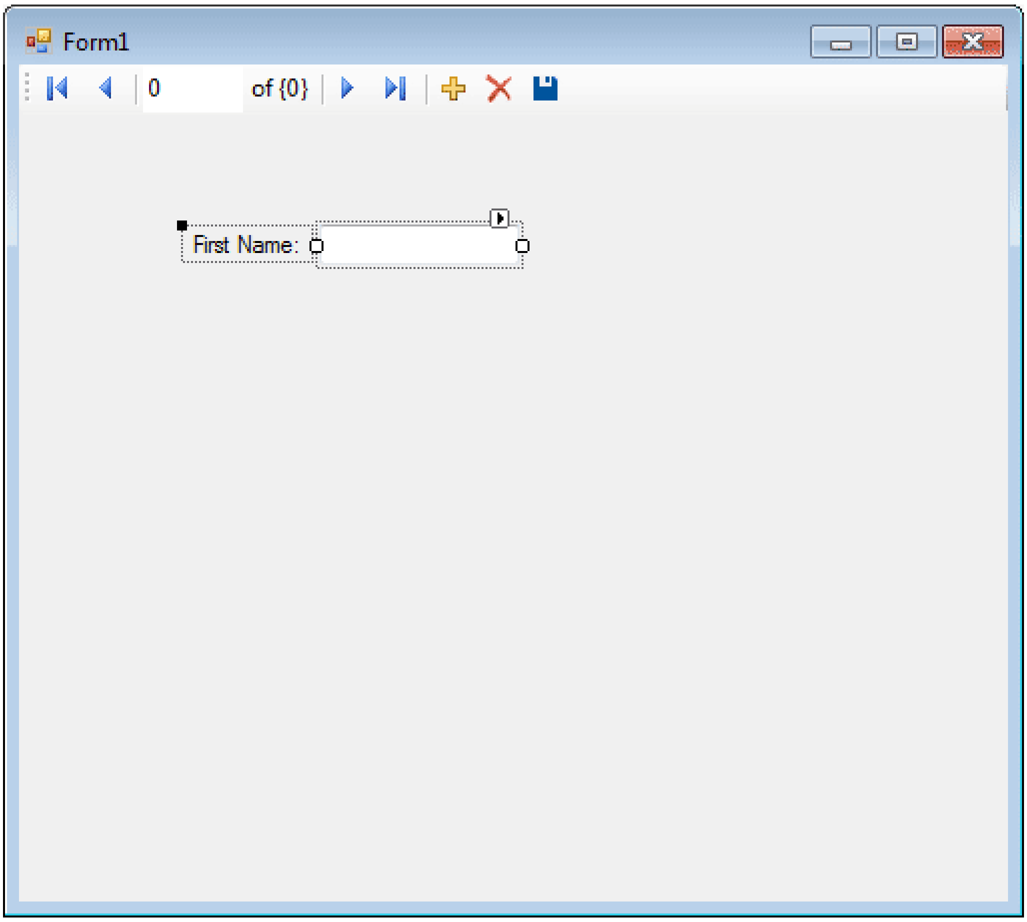
All the Fields in the Address Book database are now showing.

To add a Field to your Form, click on one in the list. Hold down your left mouse button, and drag it over to your form:

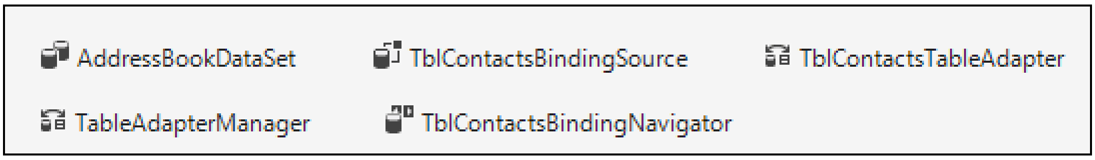


In the image above, the **FName** field is being dragged on the Form. Your mouse cursor will change shape.

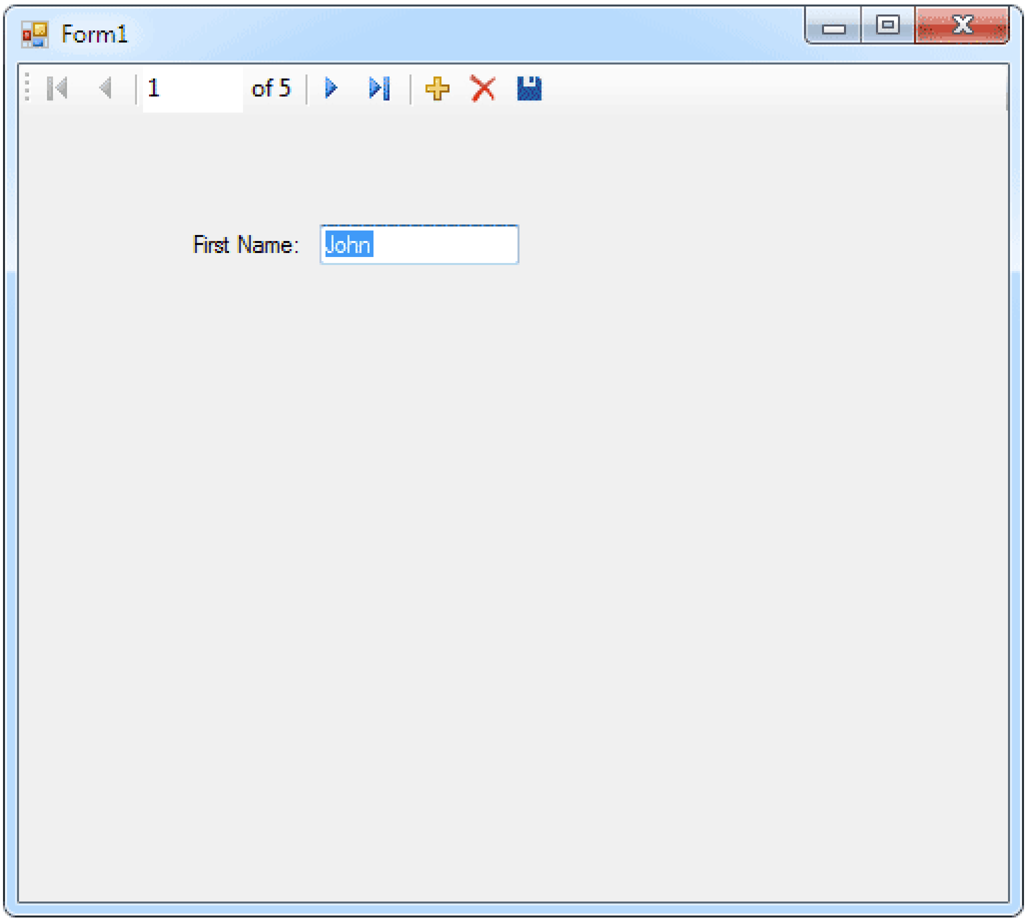
When your Field is over the Form, let go of your left mouse button. A textbox and a label will be added:



There are two other things to notice: a navigation bar appears at the top of the form, and a lot of strange objects have appeared in the object area at the bottom:



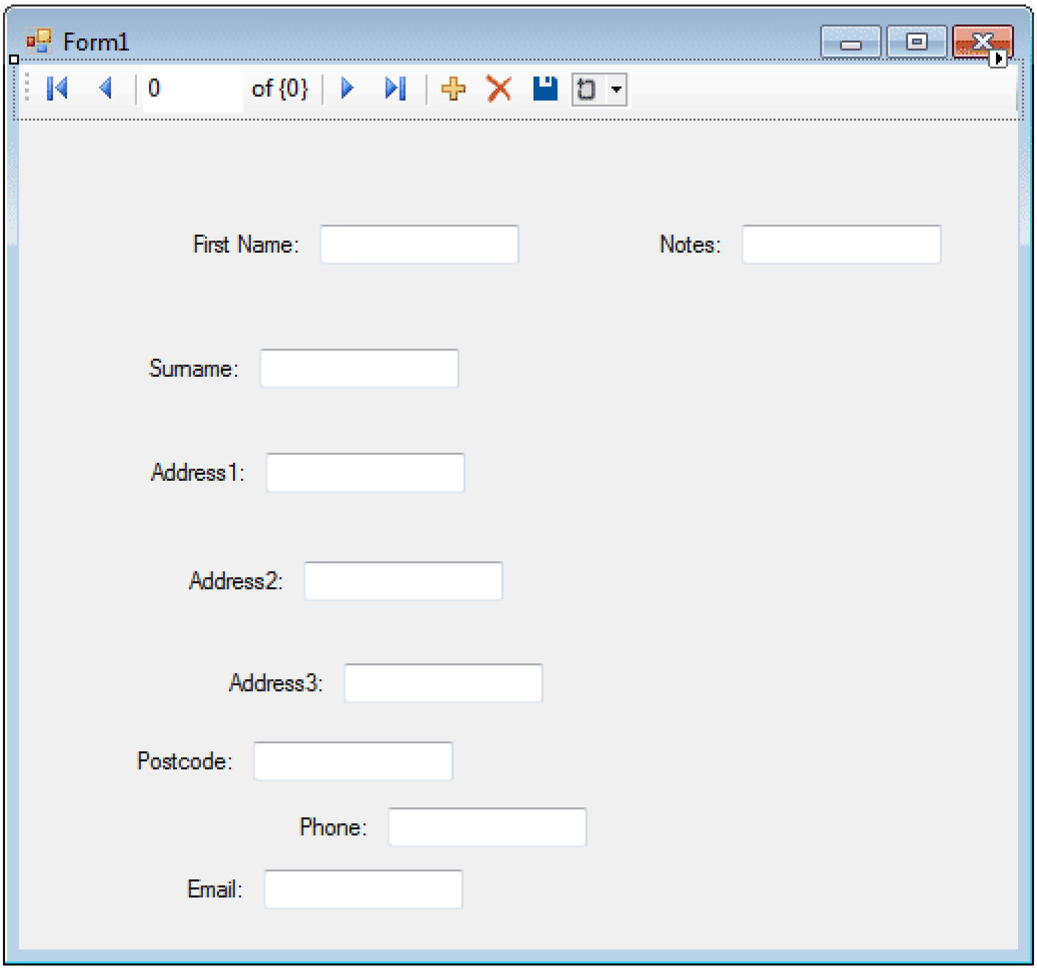
We'll explore the Objects in a later section. But notice the Navigation bar at the top of the form. Run your programme by hitting the F5 key on your keyboard. You should see this:



First Name: John

Click the Navigation arrows to scroll through the database, where it says 1 of 5. When you've played around with the controls, stop the form from running, and return to Design View.

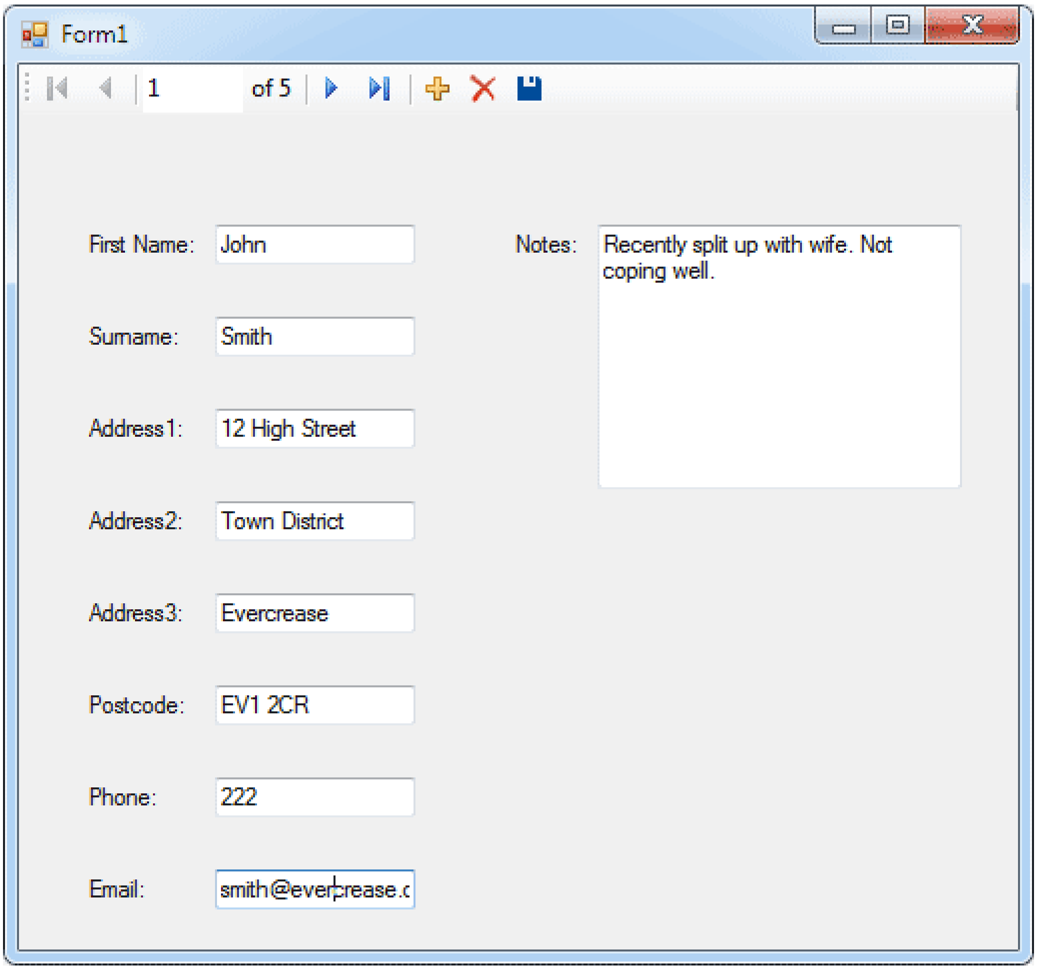
Drag and Drop more Fields to your form. But don't align them yet. We'll see an easy way to do this. But once you've dragged the fields to your form, it might look like this:



I'm sure you'll agree - that's a very untidy form. But there's a very easy way to align all your controls. Try this:

- **Click on a label with your left mouse button**
- **Hold down the Ctrl key on your keyboard, and select a second label**
- **With the Ctrl key still held down, click each label in turn (leave the Notes label unselected)**
- **When all the labels are selected, click on the Format menu at the top**
- **From the Format menu select Align > Lefts. The left edges of the labels will align themselves**
- **From the Format menu select Vertical Spacing > Make Equal. The space between each textbox will then be the same**
- **Now do the same with the textboxes (but leave the Notes textbox unselected)**

For the Notes Textbox, set the **MultiLine** property to **True** and resize the textbox. With your new controls added, and nicely aligned, press F5 to run your form. Your form might then be something like this:



First Name:	John	Notes:	Recently split up with wife. Not coping well.
Surname:	Smith		
Address1:	12 High Street		
Address2:	Town District		
Address3:	Evercrease		
Postcode:	EV1 2CR		
Phone:	222		
Email:	smith@evercrease.c		

Click the Navigation icons to move backwards and forwards through your database.