

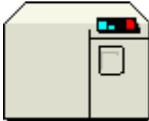


### Connecting to Databases

Envelop's **Data Control** is the primary link between Envelop controls and database fields. The Data Control is a **HyperControl** which was "Enveloped" using a standard form and buttons linked to a **RecordSet**. You place a Data Control on a form just like any other control from the controls palette. Envelop assigns a default name of DataControl1 to the first data control placed on a form. Once you've placed a Data Control to a form, you then "bind" other Envelop controls, such as TextBoxes and Labels to the Data Control. The process of binding controls to a data control involves the following three steps:

1. Connect the Data Control to a...
  - a. [ASCII text file.](#)
  - b. [ASCII text file through ODBC.](#)
  - c. [Relational Database file through ODBC.](#)
2. [Identify the format of the data fields \(ASCII text file only\)](#)
3. [Assign controls on the form to database fields.](#)

A **Database Wizard** is available to streamline the process of binding controls to a database. To help you better understand how easy it is to connect Envelop forms to various kinds of database fields, we will describe how the Data Control, on the Envelop Form shown below, links the (5) textboxes to fields in the DataBase Example. For step-by-step instructions, click one of the three "Connect" options (a, b or c) shown above to learn how to connect the Data Control to a corresponding data source. Click the "Assign" option (3.) to see how the textboxes are linked to the database fields.



The screenshot shows a window titled "Address Guide" with a grid background. It contains five textboxes arranged in a form layout:

- Name: TextBox1
- Address: TextBox2
- City: TextBox3
- State: TextBox4
- Zip: TextB

At the bottom of the form are five navigation buttons: <<, <, a large empty box, >, and >>.

[Envelop Form with Data Control](#)

[DataBase Example](#)

**Envelop Form with Data Control**

This Envelop Form contains five TextBoxes which are to be connected to five corresponding data fields (Name, Address, City, State, Zip). Each of the textboxes displays the default name of the textbox. The MaxLength property has been set to these values to limit the number of characters you may enter in each box. Double-clicking on the Data Control will activate the Database Wizard and start you down the path of connecting the textboxes to the data fields.

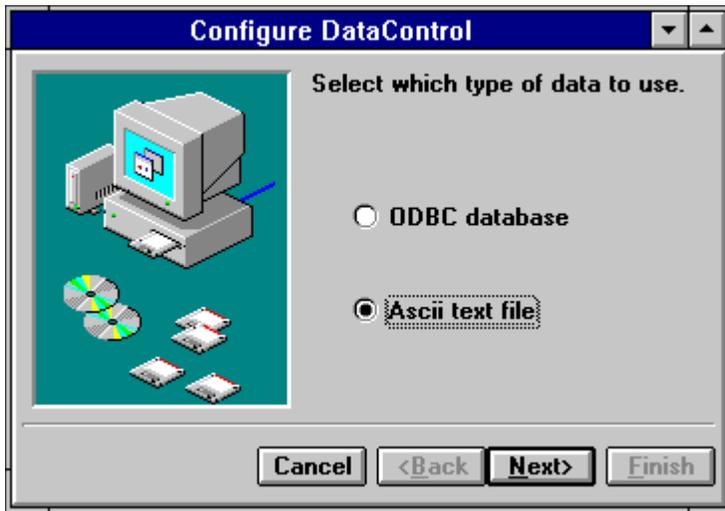
**DataBase Example**

| <b>Name</b>    | <b>Address</b>        | <b>City</b> | <b>State</b> | <b>Zip</b> |
|----------------|-----------------------|-------------|--------------|------------|
| CIMLINC Itasca | 1222 Hamilton Parkway | Itasca      | Illinois     | 60143      |
| CIMLINC Troy   | 1699 Stutz Drive      | Troy        | Michigan     | 48084      |

The data shown above will be referenced in other sections of the **Connection to Databases** help. The field names appear on the first row while the actual data begins on the second row. The size of each data field are as follows: Name(15), Address(25), City(12), State(12), Zip(5). The MaxLength properties of the corresponding textboxes on the Envelop Form have been set to these field sizes to limit input. The data shown above is from a standard ASCII text file.

### ASCII text file

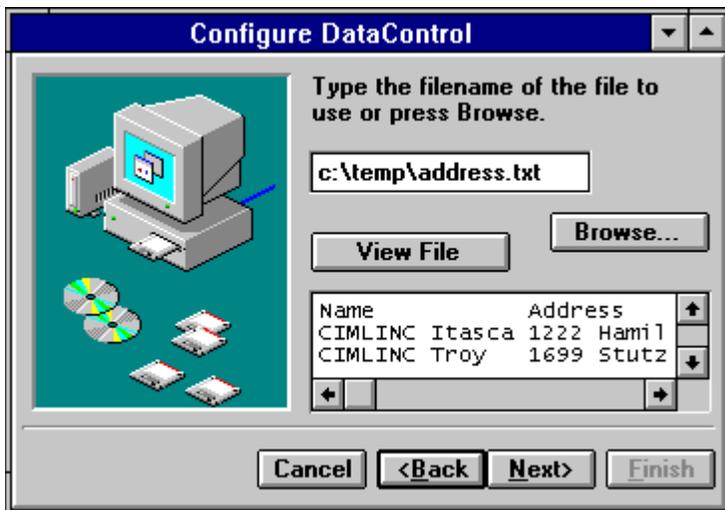
The three steps shown below will establish a connection between an ASCII text file and Envelop's Data Control. In addition, the format for how the data should be read will also be accomplished.



Step 1.

The first step is to click the **ASCII text file** option. This indicates that the Data Control will be connected directly to an ASCII file and that the individual textboxes will be linked to data fields in the text file. The text file we are using can be viewed on the previous help screen by clicking on the **Database Example** text.

[Behind the Scene...](#)



Step 2.

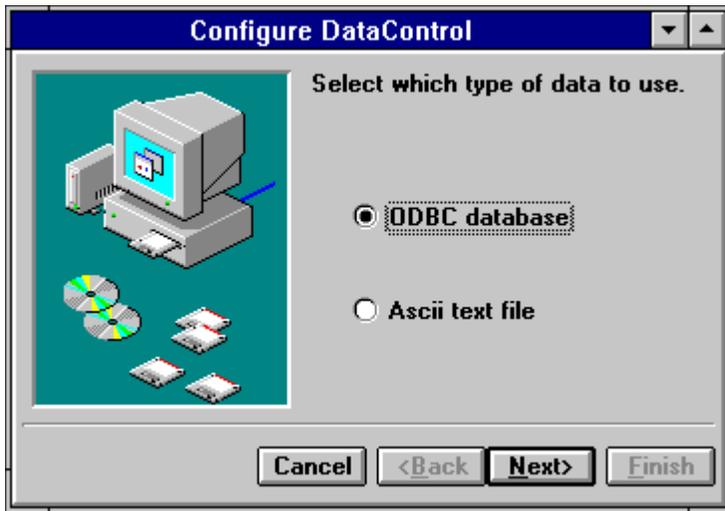
The second step is to enter an ASCII filename you wish to connect to. You may click the **Browse** button to use a file dialog for choosing a filename. If you manually enter a filename, please include the full path such as `c:\temp\address.txt`. To view the contents of the file, click the **View File** button. Use the scrollbars to view the contents of the file you have entered.

[Behind the Scene...](#)

**Note:** In the example above, we show the path of the file as being `c:\temp\address.txt`. In other examples, the path is `c:\envelop\bootcamp\basic\sampdata\address.txt`. We used the shorten `c:\temp` directory only as an example to show you a full path.

### ASCII text file through ODBC

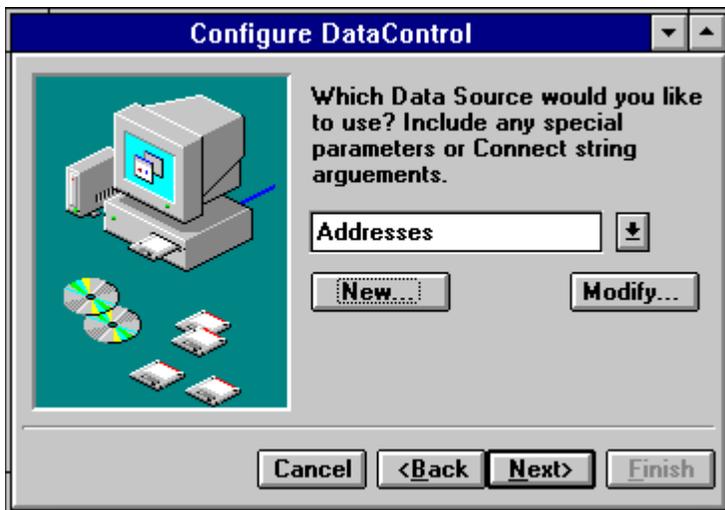
The three steps shown below will establish a connection between an ASCII text file through an ODBC Text Driver and Envelop's Data Control. In addition, the format for how the data should be read will also be accomplished.



Step 1.

Clicking the **ODBC database** option indicates that the Data Control will be connected directly to an ASCII file through an ODBC text driver and that the individual textboxes will be linked to data fields in the text file. The text file we are using can be viewed on the previous help screen by clicking on the **Database Example** text.

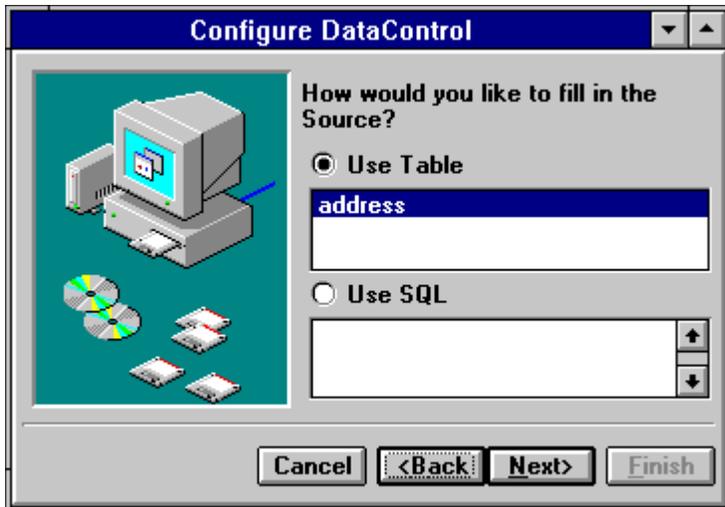
[Behind the Scene...](#)



Step 2.

The combobox contains a list of existing ODBC data sources. To add a new ASCII data source, enter a name then click the **New...** button to post an ODBC **Add Data Sources** dialog. To modify an existing data source, select a name and click the **Modify...** button to display an ODBC **Data Sources** dialog. See [ODBC Configuration](#) for setup information.

[Behind the Scene...](#)



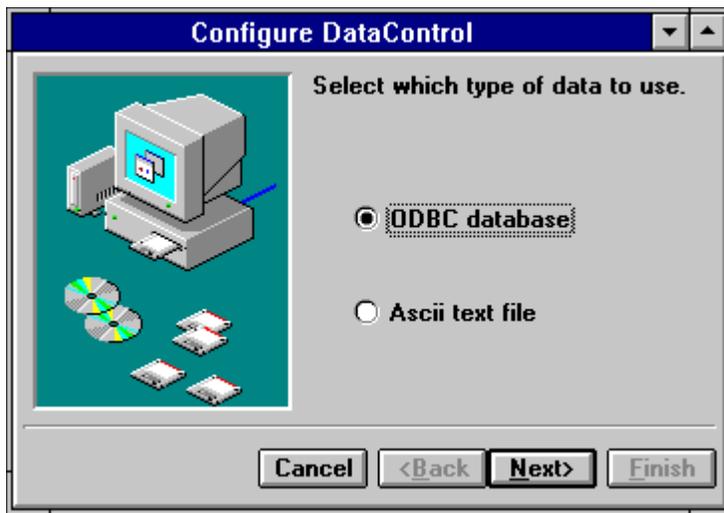
Step 3.

To retrieve the entire ASCII file, click the **Use Table** option then select a specific table entry from the tables list. To retrieve only part of the data, click the **Use SQL** option then enter an SQL select query in the textbox provided.

[Behind the Scene...](#)

## Relational Database file through ODBC

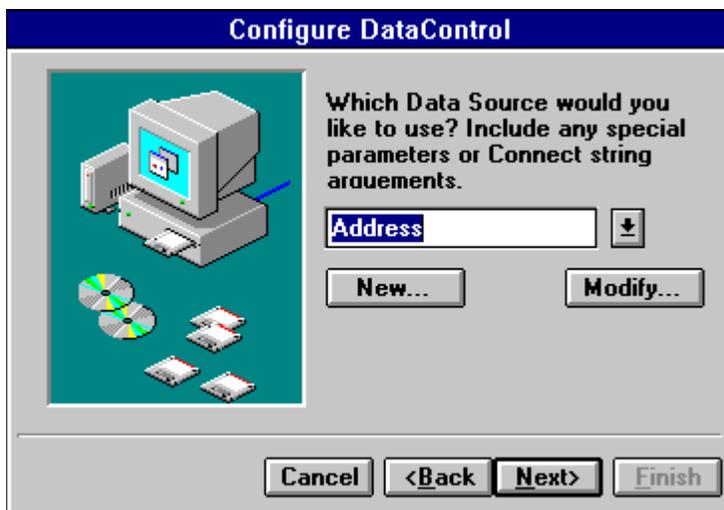
The three steps shown below will establish a connection between a dBase table file through an ODBC dBase Driver and Envelop's Data Control. In addition, the format for how the data should be read will also be accomplished.



Step 1.

Clicking the **ODBC database** option indicates that the Data Control will be connected directly to a database (dBase) file through an ODBC dBase driver and that the individual textboxes will be linked to data fields in the database file. The text file we are referencing can be viewed on the previous help screen by clicking on the **Database Example** text.

[Behind the Scene...](#)

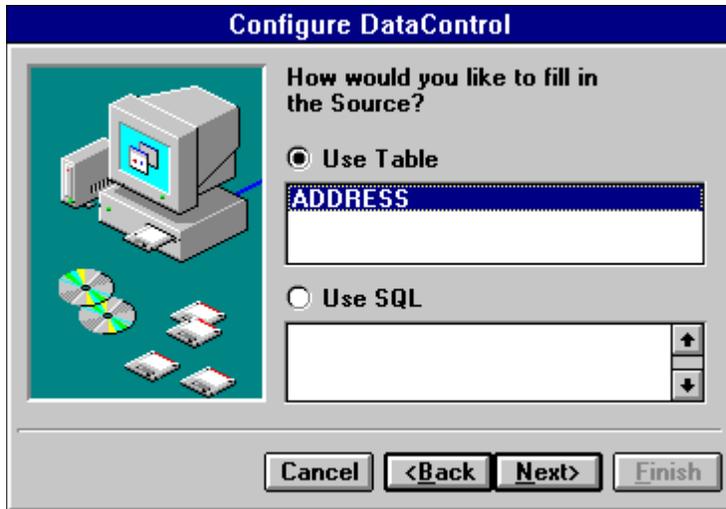


Step 2.

The combobox contains a list of existing ODBC data sources. To add a new dBase data source, enter a name then click the **New...** button to post an ODBC **Add Data Sources** dialog. To modify an existing data source, select a name and click the **Modify...** button to display an ODBC **Data Sources** dialog.

See [ODBC dBase](#) configuration for setup information.

[Behind the Scene...](#)



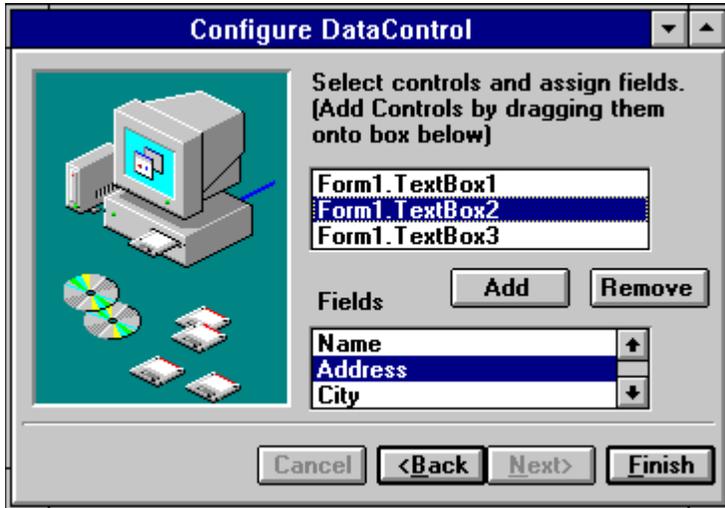
Step 3.

To retrieve the entire dBase table, click the **Use Table** option then select a specific table entry from the tables list. To retrieve only part of the data, click the **Use SQL** option then enter an SQL select query in the textbox provided.

[Behind the Scene...](#)

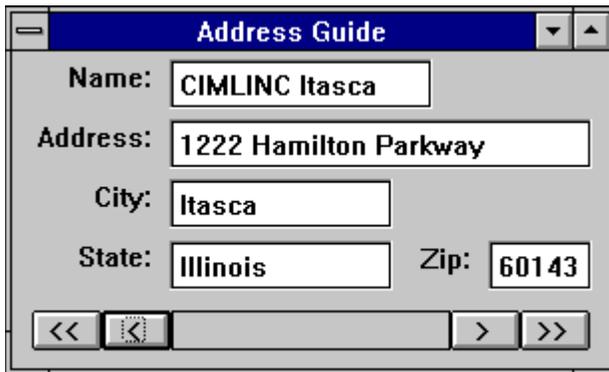
### Assign controls on the form to database fields

The last step of using the Database Wizard is to link controls on the form to corresponding data fields. At this point, it does not matter whether we are connected to an ASCII file or to a Database file through ODBC. Once a Data Control has been connected to a data source, "data fields" are established. In the case of connecting to ASCII files, the fields are based on the format of the data.



To bind controls on the form to data fields, you must place the form in "run" mode (i.e., deactivate the form editor) and using the right mouse button, click and drag each control on the form into the controls list. This list is located just above the Add and Remove buttons. As each control is dropped into this list, you will see the name of the control.

The next step is to click on the name of a control in the controls list, then click on the name of the data field you want to bind it to. When both control and field names are highlighted, click the **Add** button. This will set the **DataField** property of the control to the selected field. Just after clicking the Add button, you should see data from the first record or row appear in the control. To remove or break a link between a control and a data field, simply click the name of the control and click the **Remove** button. This will clear the DataField property on that control.



The form shown above is what the completed form would look like when all controls have been bound to the DataControl. The arrow buttons on the DataControl perform the following functions.

| Arrow | Function                    | Method    |
|-------|-----------------------------|-----------|
| <<    | Move to the first record    | MoveFirst |
| <     | Move to the previous record | MovePrev  |
| >     | Move to the next record     | MoveNext  |
| >>    | Move to the last record     | MoveLast  |

**Behind the Scene..**

The Database Wizard sets the **DatabaseType** property of the Data Controls **RecordSet** object. The default value set is **DelimitedAscii**.

**Behind the Scene...**

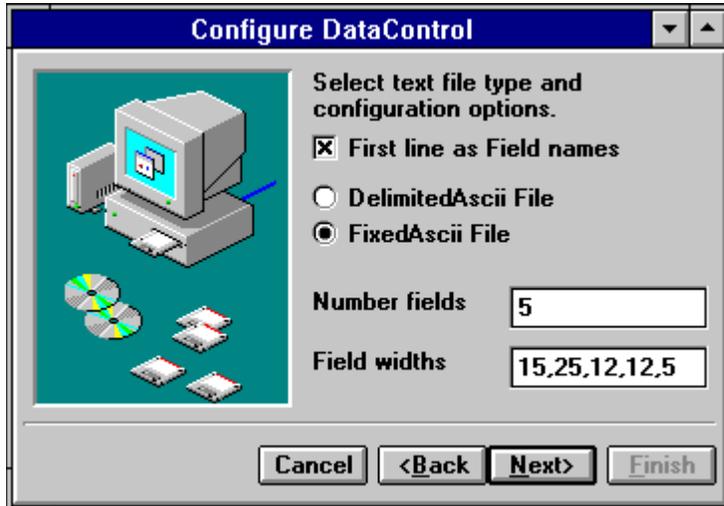
The Database Wizard sets the **Connect** property of the Data Controls **RecordSet** object. When the Connect property is set, the file will be automatically loaded into the RecordSet's **Database** object reference. A **DelimitedAscii** Database object is automatically created by default, to store the contents of the ASCII file identified in the Connect property. Additional RecordSet properties such as RowsRead and FieldList are also updated when the file is loaded.

### Behind the Scene...

Selecting the **FixedAscii file** option sets the RecordSet's **DatabaseType** property. As this is done, a corresponding **FixedAscii** Database object is automatically created. The RecordSet's **Database** property will now reference a FixedAscii Database object. Checking the **First line as Field names** option sets the RecordSet's referenced Database **FirstLineAsFieldNames** property to True. By setting this property to True, as the ASCII file is loaded, the first row is parsed and the names of each field are added to the RecordSet's **FieldList** property. Finally, the entries in the Field Widths textbox are used to set the RecordSet's referenced Database **FieldWidthList** property. As this property is set, the RecordSet's **FieldList** property is automatically updated to show the names of each data fields. Since the referenced Database's **FirstLineAsFieldNames** property was set to True, the actual names of the fields are used.

### Identify the format of the data fields (ASCII text file only)

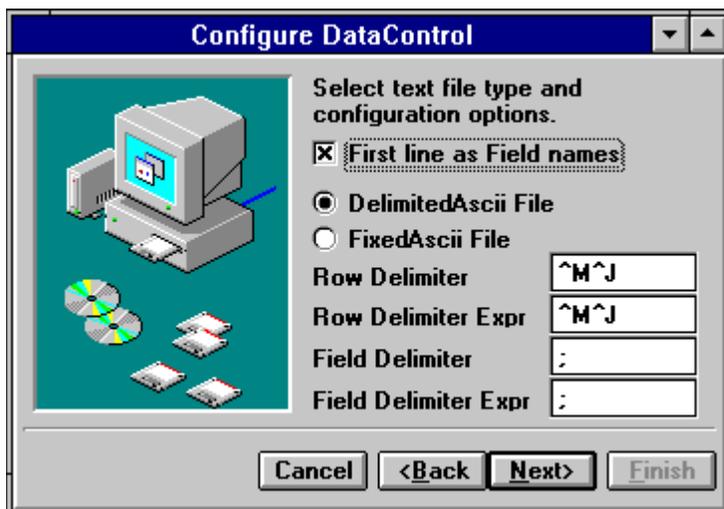
Once we have connected to an ASCII data source, the next step is to set the options for how the data is formatted. Data fields in ASCII files may be organized by **fixed width columns**, or may be organized by having a **delimiter character** such as a comma, separate its fields. In our example, the data is organized by fixed width columns, therefore we would select the **FixedAscii file** option.



Since the first row of our Database Example contains the names of each data field, we would also check the **First line as Field names** option.

To specify the lengths of each of the columns so we would enter "15,25,12,12,5" in the **Field Widths** textbox. As we enter each field width separated by a comma, the **Number Fields** entry will be updated automatically.

[Behind the Scene...](#)



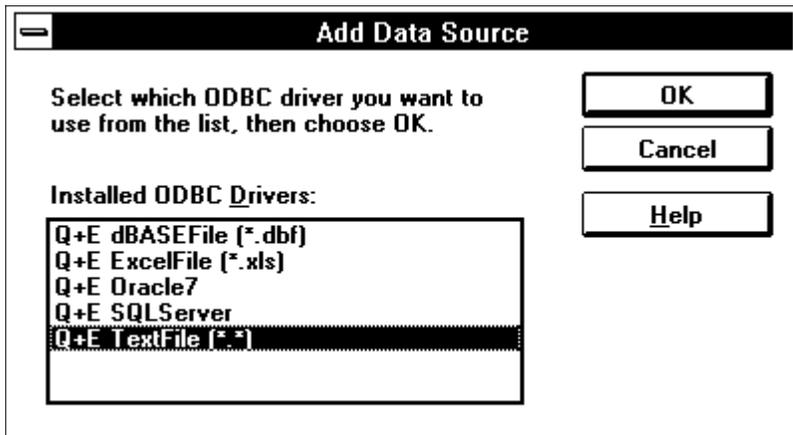
If our ASCII file had special characters which separated each data field, we would have used the **DelimitedAscii file** option.

In this option, we would specify the Field Delimiter character which separates each data field. In addition, we would specify Row Delimiters that separate each line. The ^M^J is a special character sequence for carriage return (^M) and linefeed (^J) characters.

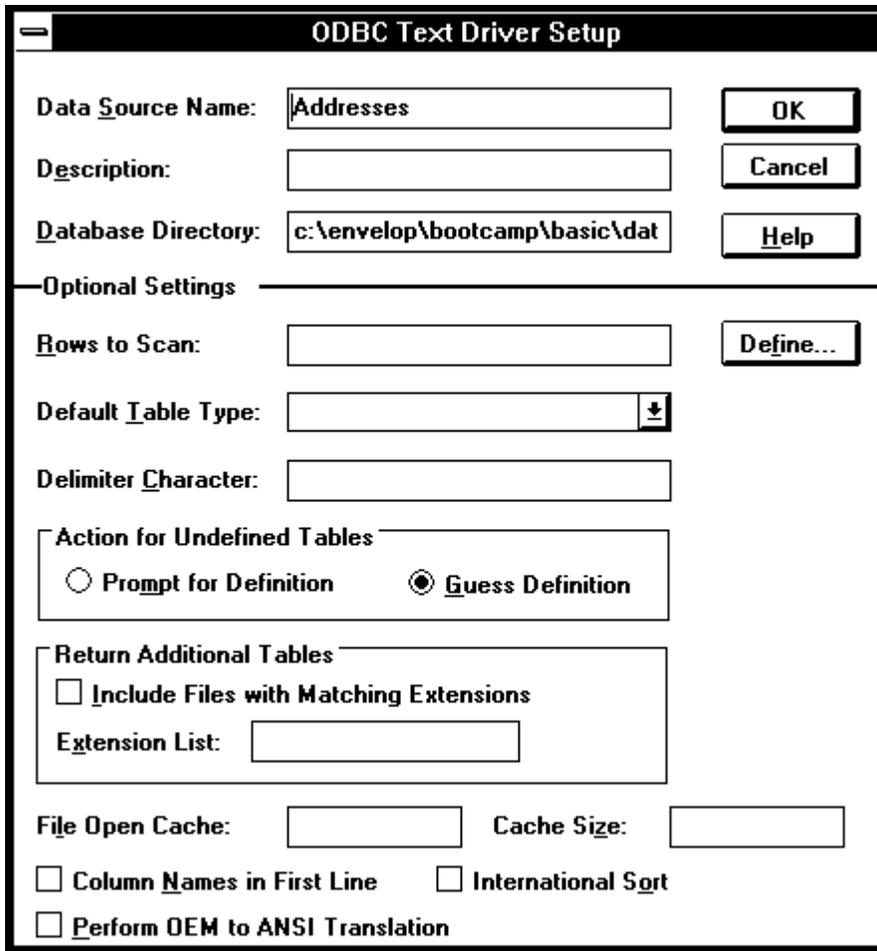
[Behind the Scene...](#)

### ODBC ASCII TextFile Configuration

Please consult your ODBC on-line help for detailed instructions for configuring the various ODBC dialogs presented below. We will outline how each dialog should be configured at each step.



At this step, we choose the type of driver based on the database you are accessing. In our case, we have selected the **QE TextFile (\*.\*)** option, then clicked the **OK** button.



When this dialog is displayed, the name Addresses appears as the Data Source. This is the name we had entered into the Data Wizard's Data Source textbox. Click the **Define...** button to go to the next dialog. A File dialog will appear on the screen. At this point, we would navigate to the c:\envelop\bootcamp\basic\datasamp directory and click the filename "address.txt".

**Define Table**

Database Name: C:\envelop\bootcamp\basic\datasamp

**Table Information**

File: address.txt

Table: address

Column Names in First Line

Table Type: Fixed

Delimiter Character:

DEM to ANSI Translation

**Column Information**

NAME  
ADDRESS  
CITY  
STATE  
ZIP

Name: [ ] Par

Type: Varchar Moc

Mask: [ ] Rem

Precision: [ ] Scale: [ ]

Length: [ ] Offset: [ ]

When this dialog is displayed, the **File:** entry shows the name "address.txt". Check the **Column Names in First Line** checkbox and select a **Table Type:** of **Fixed**. Now click the **Parse** button to view the contents of the first line. If our ASCII file did not have the names of the columns of data identified on the first line, generic names such as Field\_1, Field\_2 would be assigned to each of the data fields in the ASCII file.

**Parse Table**

Parse Line:

[Name ][Address ][City ][State ][Zip ]

The contents of the first line of the address.txt file is displayed above. When it initially appears, there are [square brackets] around the entire line. You must add square brackets, as shown above, to designate the start and finish of each column or "data field". When this is done, click the **OK** button and you will see the names of your data fields appear in the Column Information list. After viewing the updated column information in the Define Table dialog, click the **OK** button. This will return you back to the ODBC Text Driver Setup dialog. Click the **Cancel** button on this dialog to return to Envelop.

Two ODBC text files have been configured for you. In the same directory as where the ASCII file resides, a file named QETXT.INI has been created and contains format information. In addition, the ODBC.INI file in the windows directory has been configured.

## QETXT.INI

[Defined Tables]

address.txt=address

[address]

FILE=address.txt

FLN=1

TT=Fixed

Charset=ANSI

FIELD1=NAME, VARCHAR,15,0,15,0,

FIELD2=ADDRESS, VARCHAR,25,0,25,15,

FIELD3=CITY, VARCHAR,12,0,12,40,

FIELD4=STATE, VARCHAR,12,0,12,52,

FIELD5=ZIP, VARCHAR,5,0,5,64,

**ODBC.INI**

[Addresses]

Driver32=C:\WINDOWS\system32\QETXT04.DLL

Description=

Database=c:\temp\addresses

ScanRows=

TableType=Fixed

Delimiter=

UndefinedTable=GUESS

ExtraExtensions=

FileOpenCache=

CacheSize=

FirstLineNames=0

IntlSort=0

Charset=ANSI

### Behind the Scene...

Selecting the **DelimitedAscii file** option sets the RecordSet's **DatabaseType** property. As this is done, a corresponding **DelimitedAscii** Database object is automatically created. The RecordSet's **Database** property will now reference a DelimitedAscii Database object. Checking the **First line as Field names** option sets the RecordSet's referenced Database **FirstLineAsFieldNames** property to True. By setting this property to True, as the ASCII file is loaded, the first row is parsed and the names of each field are added to the RecordSet's **FieldList** property. Finally, the entries in the **Row Delimiter** and **Field Delimiter** textboxes are used to set the RecordSet's referenced Database Row Delimiter and Field Delimiter properties. As this property is set, the RecordSet's **FieldList** property is automatically updated to show the names of each data fields. Since the referenced Database's FirstLineAsFieldNames property was set to True, the actual names of the fields are used.

**Behind the Scene...**

The Database Wizard sets the **DatabaseType** property of the Data Controls **RecordSet** object. The value is set to **ODBC**.

### **Behind the Scene...**

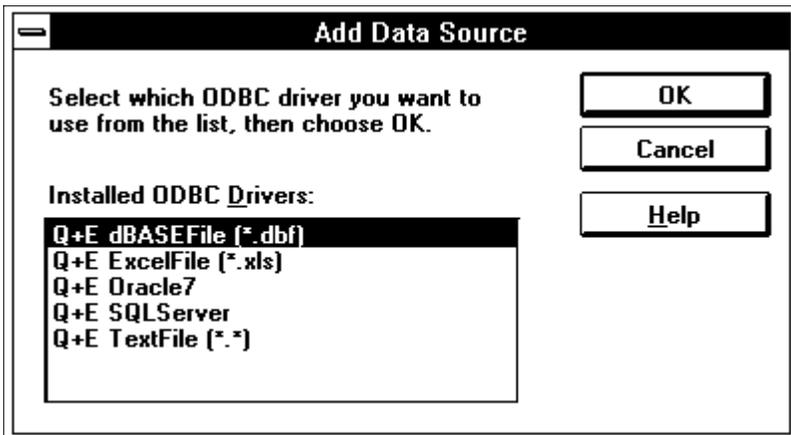
Basically this step interfaces to the ODBC object. The **New...** button executes the ODBC **CreateDataSource** method which displays a Add Data Source dialog. The **Modify...** button executes the ODBC **ManageDataSources** method which displays a Data Sources dialog. These dialogs belong to ODBC and are not part of Envelop. You should consult ODBC's help to understand the options presented in these dialogs as they are used to configure the ODBC Text file driver and establish data fields from the specified ASCII file. Once a data source has been established, the Data Control's **Connect** property is set to the name of the data source.

### **Behind the Scene..**

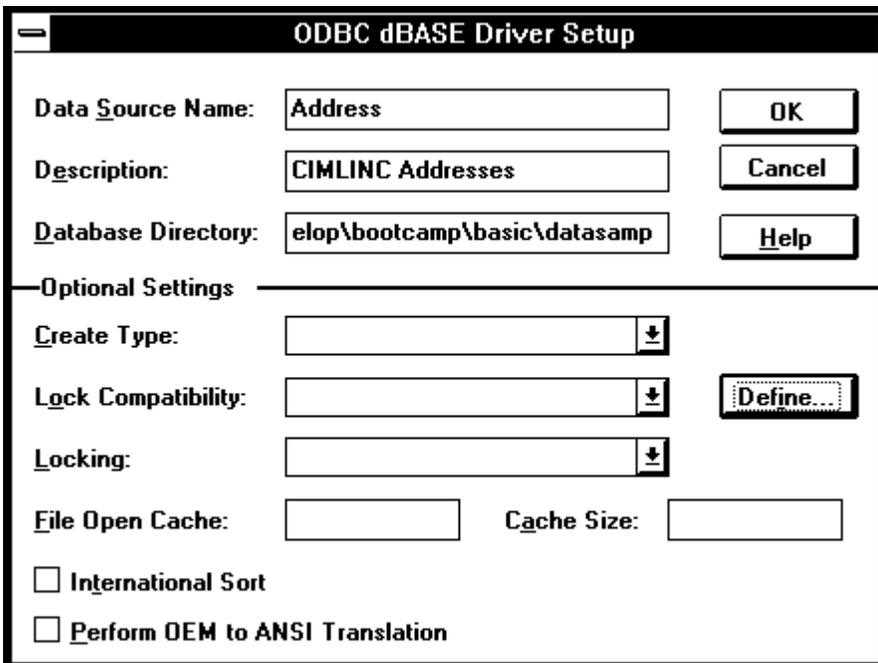
This step sets the **RecordSource** property of the Data Controls RecordSet object. This specifies the data that is actually read into Envelop. By default, the name of the ASCII textfile becomes the name of the table. By selecting the name of the table, you will load the entire ASCII file. By using an SQL statement such as: **select name, city, state from address** you would be loading data from the name, city and state columns in the ASCII file.

### ODBC dBase Table Configuration

Please consult your ODBC on-line help for detailed instructions for configuring the various ODBC dialogs presented below. We will outline how each dialog should be configured at each step.



At this step, we choose the type of driver based on the database you are accessing. In our case, we have selected the **QE dBASEFile (\*.dbf\*)** option, then clicked the **OK** button.



When this dialog is displayed, the name Address appears as the Data Source. This is the name we had entered into the Data Wizard's Data Source textbox. The **Data Source Name** identifies a single connection to a dBASE database. This can be any string. The **Description** is an optional long description of a data source name. The **Database Directory** identifies the directory that contains the database files. If none is specified, the current working directory is used. Click the **Define...** button to go to the next dialog. A File dialog will appear on the screen. At this point, we would navigate to the c:\envelop\bootcamp\basic\datasamp directory and click the filename "address.dbf".

**Define Table**

Database Name: C:\envelop\bootcamp\basic\datasamp OK

File: ADDRESS.DBF Cancel

OEM to ANSI Translation Help

---

**Indexes:**

|   | Unique                   | Maintain                 |
|---|--------------------------|--------------------------|
| Index File: <input style="width: 80%;" type="text"/> <span style="float: right;">↓</span> | <input type="checkbox"/> | <input type="checkbox"/> |
| Tag: <input style="width: 80%;" type="text"/> <span style="float: right;">↓</span>        |                          |                          |

The top section of the dialog displays the name and directory where the dBASE file is located. If this file uses the IBMPC character set, then check the OEM to ANSI Translation box. The bottom section of the dialog box displays the index information for the data file.

Select each index file that you want to associate with the dBASE file and select the Maintain check box. If the index file is unique, select the Unique check box to the right of the file name. If the selected index has an MDX or CDX extension, you cannot mark the index file as unique. Instead, you may mark the tags within the index as unique. To mark a tag unique, select the tag name within the Tag box and select the Unique check box to the right of the name. Click OK to save this information or press Cancel to abort the dialog.

Two ODBC text files have been configured for you. In the same directory as where the ASCII file resides, a file named [QEDBF.INI](#) has been created and contains format information. In addition, the [ODBC.INI](#) file in the windows directory has been configured.

**Behind the Scene...**

The Database Wizard sets the **DatabaseType** property of the Data Controls **RecordSet** object. The value is set to **ODBC**.

### **Behind the Scene...**

Basically this step interfaces to the ODBC object. The **New...** button executes the ODBC **CreateDataSource** method which displays a Add Data Source dialog. The **Modify...** button executes the ODBC **ManageDataSources** method which displays a Data Sources dialog. These dialogs belong to ODBC and are not part of Envelop. You should consult ODBC's help to understand the options presented in these dialogs as they are used to configure the ODBC dBase table driver and establish data fields from the specified dBase file. Once a data source has been established, the Data Control's **Connect** property is set to the name of the data source.

### **Behind the Scene..**

This step sets the **RecordSource** property of the Data Controls RecordSet object. This specifies the data that is actually read into Envelop. By selecting the name of the table, you will load the entire dBase table. By using an SQL statement such as: **select name, city, state from address** you would be loading data from the name, city and state columns in the dBase file.

**QEDBF.INI**  
[ADDRESS.DBF]  
CHARSET=ANSI

**ODBC.INI**

[Address]

Driver32=C:\WINDOWS\system32\QEDBF04.DLL

Description=CIMLINC Addresses

Database=c:\envelop\bootcamp\basic\datasamp

Locking=

LockCompatibility=

FileOpenCache=

CacheSize=

Charset=ANSI

IntlSort=0

CreateType=

