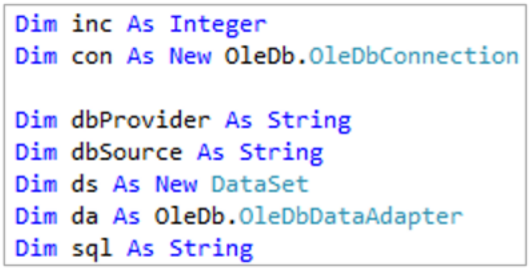
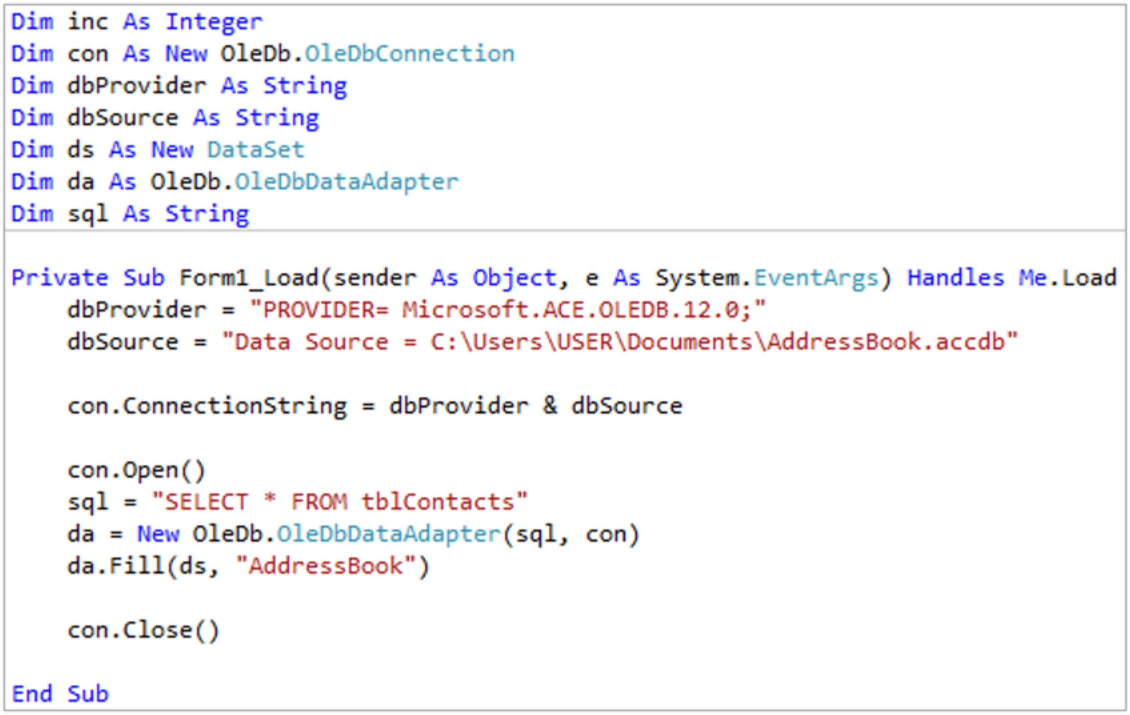
Lab 10: Navigate a Database with VB .NET

To navigate through the dataset, let's change our form by adding some navigation buttons. At the moment, all our code is in the Button we added to the form. We're going to delete this button, so we need to move it out of there. The variable declarations can be moved right to the top of the coding window. That way, any button can see the variables. So move your variables declarations to the top, as in the image below (don't forget to add the Dim inc As Integer line):



```
Dim inc As Integer
Dim con As New OleDb.OleDbConnection

Dim dbProvider As String
Dim dbSource As String
Dim ds As New DataSet
Dim da As OleDb.OleDbDataAdapter
Dim sql As String
```

We can move a few lines to the Form Load event. So, create a Form Load event, as you did in a previous section. Now move all but the textbox lines to there. Your coding window should then look like this (you can delete the message box lines, or just comment them out):



```
Dim inc As Integer
Dim con As New OleDb.OleDbConnection
Dim dbProvider As String
Dim dbSource As String
Dim ds As New DataSet
Dim da As OleDb.OleDbDataAdapter
Dim sql As String

Private Sub Form1_Load(sender As Object, e As System.EventArgs) Handles Me.Load
    dbProvider = "PROVIDER= Microsoft.ACE.OLEDB.12.0;"
    dbSource = "Data Source = C:\Users\USER\Documents\AddressBook.accdb"

    con.ConnectionString = dbProvider & dbSource

    con.Open()
    sql = "SELECT * FROM tblContacts"
    da = New OleDb.OleDbDataAdapter(sql, con)
    da.Fill(ds, "AddressBook")

    con.Close()

End Sub
```

For your button, all you should have left are these two lines:

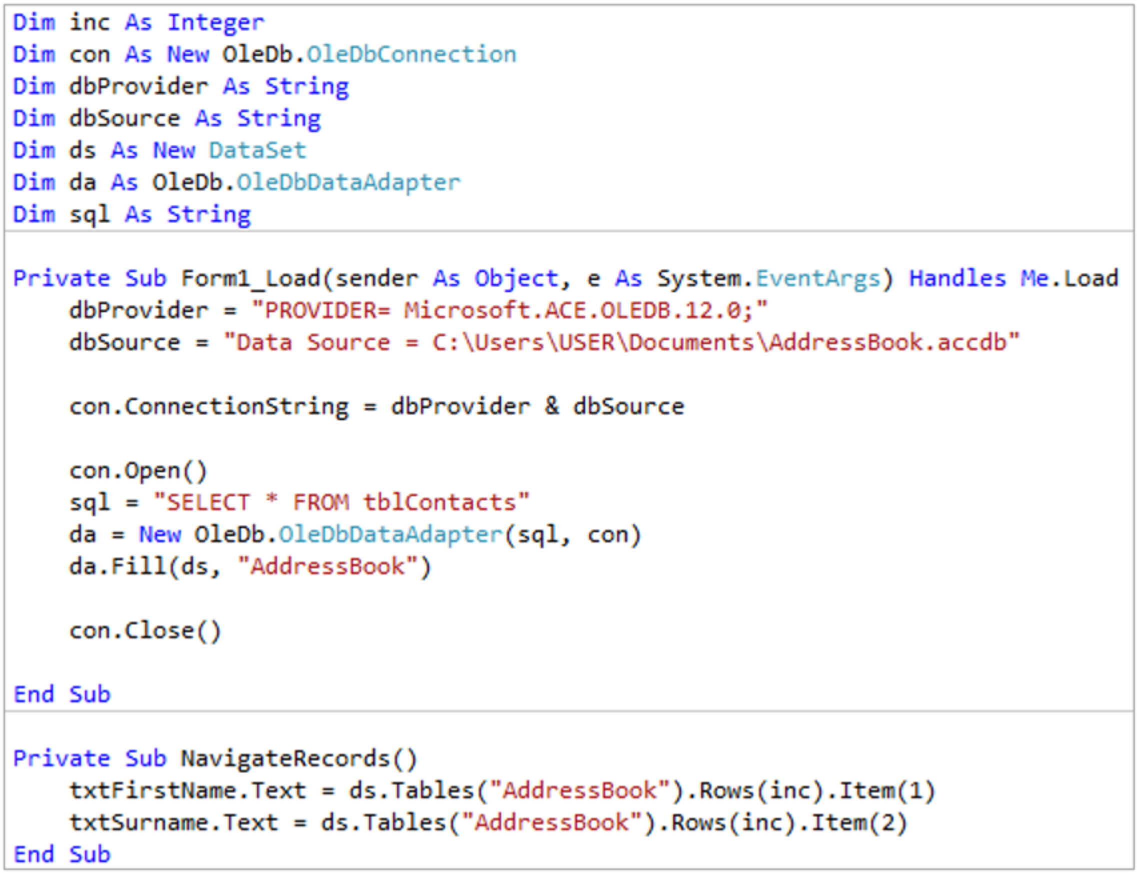
```
txtFirstName.Text = ds.Tables("AddressBook").Rows(inc).Item(1)
txtSurname.Text = ds.Tables("AddressBook").Rows(inc).Item(2)
```

Since we're going to be deleting this button, this code can be moved. Because all the buttons need to put something into the textboxes, the two lines we have left are an ideal candidate for a Subroutine. So add the following Sub to your code:

```
Private Sub NavigateRecords()

    txtFirstName.Text = ds.Tables("AddressBook").Rows(inc).Item(1)
    txtSurname.Text = ds.Tables("AddressBook").Rows(inc).Item(2)

End Sub
```

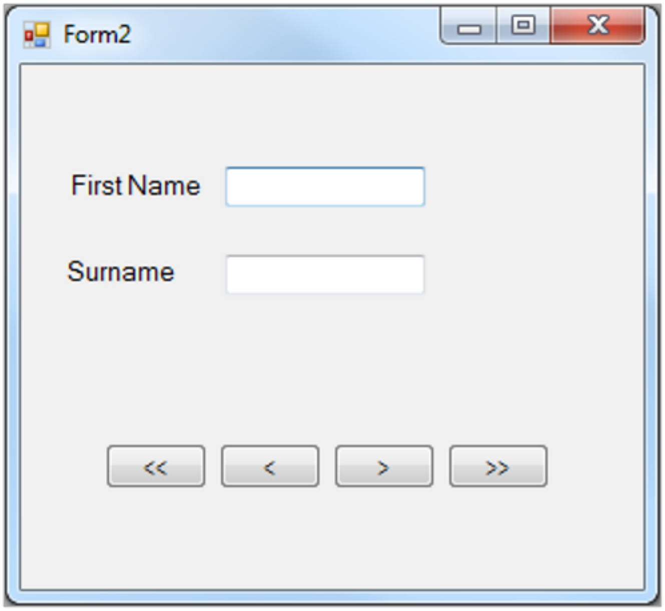
Now that all of your code has gone from your button, you can delete the button code altogether. Return to you form, click on the button to select it, then press the delete key on your keyboard. This will remove the button itself from your form.

Here's what your coding window should like:



```
Dim inc As Integer
Dim con As New OleDb.OleDbConnection
Dim dbProvider As String
Dim dbSource As String
Dim ds As New DataSet
Dim da As OleDb.OleDbDataAdapter
Dim sql As String

Private Sub Form1_Load(sender As Object, e As System.EventArgs) Handles Me.Load
    dbProvider = "PROVIDER= Microsoft.ACE.OLEDB.12.0;"
    dbSource = "Data Source = C:\Users\USER\Documents\AddressBook.accdb"

    con.ConnectionString = dbProvider & dbSource

    con.Open()
    sql = "SELECT * FROM tblContacts"
    da = New OleDb.OleDbDataAdapter(sql, con)
    da.Fill(ds, "AddressBook")

    con.Close()

End Sub

Private Sub NavigateRecords()
    txtFirstName.Text = ds.Tables("AddressBook").Rows(inc).Item(1)
    txtSurname.Text = ds.Tables("AddressBook").Rows(inc).Item(2)
End Sub
```

Now you can re-design the form. Add four new buttons, and change the Name properties to: btnNext, btnPrevious, btnFirst, and btnLast. Change the Text properties to >, <, <<, and >>. Your form will then look like this:



Add a new variable declaration to the top of your code, just under the Dim inc As Integer line. Add this:

```
Dim MaxRows As Integer
```

We can store how many rows are in the DataSet with this variable. You can get how many rows are in the DataSet with this:

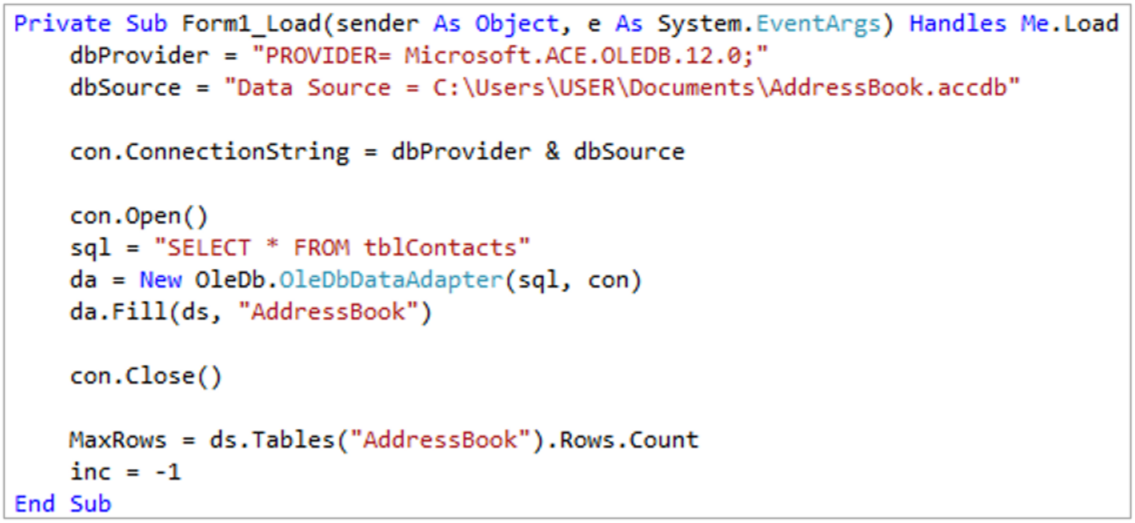
```
MaxRows = ds.Tables("AddressBook").Rows.Count
```

So the Rows property has a Count Method. This simply counts how many rows are in the DataSet. We're passing that number to a variable called MaxRows. You can then test what is in the variable, and see if the inc counter doesn't go past it. You need to do this because VB throws up an error message if try to go past the last row in the DataSet.

Add the following two lines of code to the Form Load Event of Form1:

```
MaxRows = ds.Tables("AddressBook").Rows.Count
inc = - 1
```

Your code should then look like this:



```
Private Sub Form1_Load(sender As Object, e As System.EventArgs) Handles Me.Load
    dbProvider = "PROVIDER= Microsoft.ACE.OLEDB.12.0;"
    dbSource = "Data Source = C:\Users\USER\Documents\AddressBook.accdb"

    con.ConnectionString = dbProvider & dbSource

    con.Open()
    sql = "SELECT * FROM tblContacts"
    da = New OleDb.OleDbDataAdapter(sql, con)
    da.Fill(ds, "AddressBook")

    con.Close()

    MaxRows = ds.Tables("AddressBook").Rows.Count
    inc = -1
End Sub
```

Notice the other line of code for the Form Load event:

```
inc = - 1
```

This line sets the inc variable to minus one when the form loads. When the Buttons are clicked, this will ensure that we're moving the counter on by the correct amount.

How to Move Forward One Record at a Time

Double click your Next Record button to access the code. Add the following If … Else Statement:

```
If inc <> MaxRows - 1 Then

        inc = inc + 1

        NavigateRecords()

Else

        MsgBox("No More Rows")

End If
```

We're checking to see if the value in inc does not equal the value in MaxRows - 1. If they are both equal then we know we've reached the last record in the DataSet. In which case, we just display a message box. If they are not equal, these two lines get executed:

```
inc = inc + 1
NavigateRecords()
```

First, we move the inc counter on by one. Then we call the Sub we set up:

NavigateRecords()

Our Subroutine is where the action takes place, and the values from the DataSet are placed in the textboxes. Here it is again:

```
Private Sub NavigateRecords()

        txtFirstName.Text = ds.Tables("AddressBook").Rows(inc).Item(1)
        txtSurname.Text = ds.Tables("AddressBook").Rows(inc).Item(2)

End Sub
```

The part that moves the record forward (and backwards soon) is this part:

Rows(inc)

Previously, we hard-coded this with:

Rows(0)

Now the value is coming from the variable called inc. Because we're incrementing this variable with code, the value will change each time the button is clicked. And so a different record will be displayed.

You can test out your Next button. Run your program and click the button. You should now be able to move forward through the DataSet. When you get to the end, you should see the message box display "No More Rows".

None of the other button will work yet, of course. So let's move backwards.

Move Back One Record at a Time

To move backwards through the DataSet, we need to decrement the inc counter. This means deducting 1 from whatever is currently in inc.

But we also need to check that inc doesn't go past zero, which is the first record in the DataSet. Here's the code to add to your btnPrevious:

```
If inc > 0 Then

        inc = inc - 1

        NavigateRecords()
```

```
Else

    MsgBox("First Record")

End If
```

So the If statement first checks that inc is greater than zero. If it is, inc gets 1 deducted from. Then the NavigateRecords() subroutine gets called. If inc is zero or less, then we display a message.

When you've finished adding the code, test your program out. Click the Previous button first. The message box should display, even though no records have been loaded into the textboxes. This is because the variable inc has a value of -1 when the form first loads. It only gets moved on to zero when the Next button is clicked. You could amend your IF Statement to this:

```
If inc > 0 Then

    inc = inc - 1

    NavigateRecords()

ElseIf inc = -1 Then

    MsgBox("No Records Yet")

ElseIf inc = 0 Then

    MsgBox("First Record")

End If
```

This new If Statement now checks to see if inc is equal to minus 1, and displays a message if it does. It also checks if inc is equal to zero, and displays the "First Record" message box.

Moving to the Last Record in the DataSet

To jump to the last record in the DataSet, you only need to know how many records have been loaded into the DataSet - the MaxRows variable in our code. You can then set the inc counter to that value, but minus 1. Here's the code to add to your btnLast:

```
If inc <> MaxRows - 1 Then

    inc = MaxRows - 1

    NavigateRecords()
```

```
End If
```

The reason we're saying MaxRows - 1 is that the row count might be 5, say, but the first record in the DataSet starts at zero. So the total number of records would be zero to 4. Inside of the If Statement, we're setting the inc counter to MaxRows - 1, then calling the NavigateRecords() subroutine.

That's all we need to do. So run your program. Click the Last button, and you should see the last record displayed in your textboxes.

Moving to the First Record in the DataSet

Moving to the first record is fairly straightforward. We only need to set the inc counter to zero, if it's not already at that value. Then call the Sub:

```
If inc <> 0 Then

    inc = 0

    NavigateRecords()

End If
```

Add the code to your btnFirst. Run your program and test out all of your buttons. You should be able to move through the names in the database, and jump to the first and last records.