Lab Guide for
Introduction to ADO.NET 2.0

Education and Research

Infosys®
POWERED BY INTELLECT
DRIVEN BY VALUES

Author(s)	Sachin Kumar, Pawan Kumar Deulkar
Authorized by	Srikantan Moorthy
Creation/Revision Date	May-2009
Version	1.2

COPYRIGHT NOTICE

Education and Research Department
Infosys Technologies Limited
Electronic City
Hosur Road
Bangalore - 561 229, India.

Tel: 91 80 852 0261-270
Fax: 91 80 852 0362
www.infy.com
mailto:E&R@infy.com

Document Revision History

Version	Date	Author(s)	Reviewer(s)	Description
1.0	Nov 2007	Nagasubramanya, Prakash Khude, Leena Patil	Pushpalatha Devendra, Komal Papdeja	Baselined
1.1	May 2008	Leena Patil, Mahesh M S, Nagasubramanya	Pushpalatha Devendra	Baselined
1.2	May-2009	Pawan Kumar Deulkar, Sachin Kumar	Sachin Kumar	Changed the flow of topics, included more good technology usage practices, rewritten content for data binding and extended/refined for few others.

Contents

Day-3 Assignments

All the assignments in this section must be completed on Day 2 of "Introduction to ADO.NET 2.0" Course.

Assignment 10: Implementing Data Binding for combo box

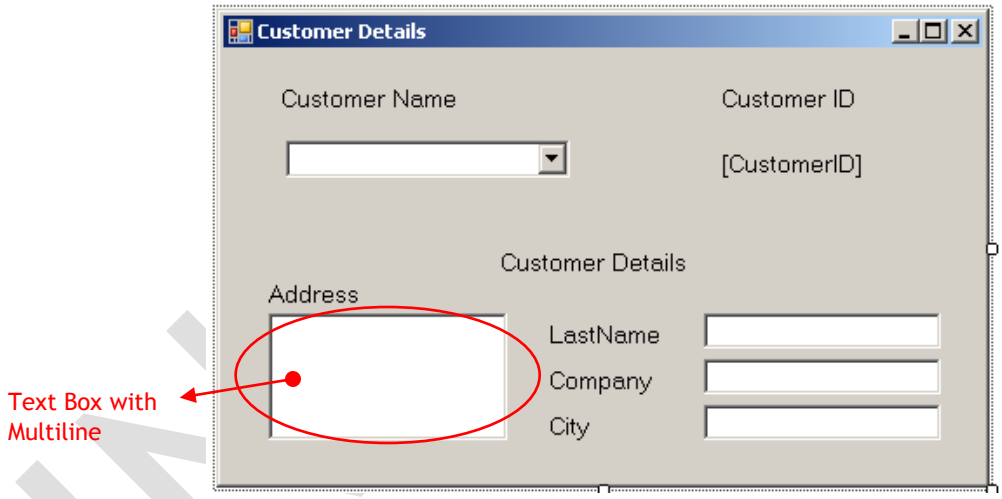
Objective: To learn how to use data binding to bind a combo box control and how to display the value of selected item from combo box.

Problem Description: Create and design the "FormCustomerDetails.cs" form.

Background: To have Customer information let us learn how to populate data to a combo box control.

Estimated time: 20 Min

Step 1: Open the project created in the previous assignment. Create FormCustomerDetails.cs form. Design the form as shown below:



Step 2: Open DataBaseLayer.cs file. Add a new method called "**LoadCustomerDetails**" with below mentioned input parameters and return value.

```
public DataSet LoadCustomerDetails()
{
     // Instantiating objects
   SqlDataAdapter DaCustomer;
   DataSet DsCustomer = new
   DataSet(); SqlConnection
   NET07TempDB;
   string Connection;
```

```
    Connection =ConfigurationManager.
      ConnectionStrings["MyConString"].ToString();

    NET07TempDB = new SqlConnection(Connection);
    DaCustomer = new SqlDataAdapter("SELECT * FROM CUSTOMER",
       NET07TempDB);

    // Fill the dataset using data adapter
    DaCustomer.Fill(DsCustomer, "Customer_ADO");

    return DsCustomer;
}
```

Step 3: Open FormCustomerDetails.cs in Design mode and double click on 'Form'. Type the following code in Load event:

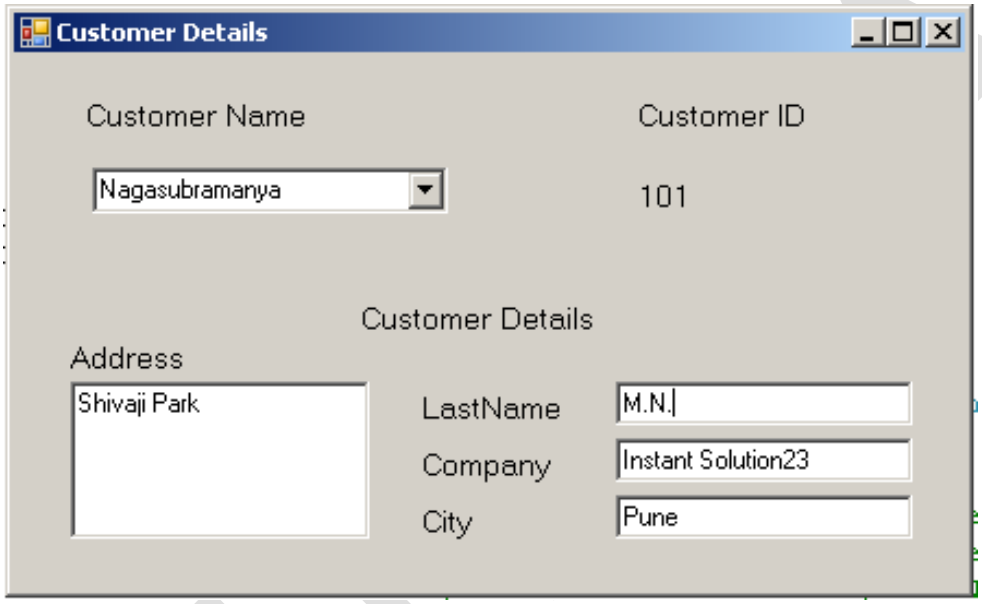
```
public DataSet DsCustomer = new DataSet();
private void FormCustomer_Load(object sender, EventArgs e)
{
      /* Write code to create an object of DataBaseLayer class.
         Call LoadCustomerDetails() method.
         Get the return value of LoadCustomerDetails() method in
DsCustomer(DataSet).
      */

      cmbCustomer.DataSource =
      DsCustomer.Tables["Customer_ADO"];
      cmbCustomer.DisplayMember = "FirstName";
      cmbCustomer.ValueMember = "CustomerID";
 }
```

Step 4: Write the code in SelectedIndexChanged event of combo-box control

```
private void cmbCustomer_SelectedIndexChanged(object
sender,EventArgs e)
{
      MessageBox.Show("Selected Customer ID is " +
            cmbCustomer.SelectedValue.ToString());

      lblCustID.Text = (dsCustomer.Tables[0].Rows
            [cmbCustomer.SelectedIndex]["CustomerID"]).ToString();

      txtAddress.Text = (dsCustomer.Tables[0].Rows
            [cmbCustomer.SelectedIndex]["Address"]).ToString();
```

```
        txtLastName.Text =  (dsCustomer.Tables[0].
             Rows[cmbCustomer.SelectedIndex]["lastname"]).ToString();
        txtCompany.Text = (dsCustomer.Tables[0].Rows
             [cmbCustomer.SelectedIndex]["company"]).ToString();

        txtCity.Text = (dsCustomer.Tables[0].Rows
             [cmbCustomer.SelectedIndex]["city"]).ToString();
  }
```

Step 5: Build and Run your application. Select one of the customer names from combo-box and check the output.



Summary of this exercise:
You have just learnt

- How to bind a combo-box control
- How to display the selected Data and Value of a combo-box control
- How to specify data binding for different controls like Label and TextBox using DataSet object

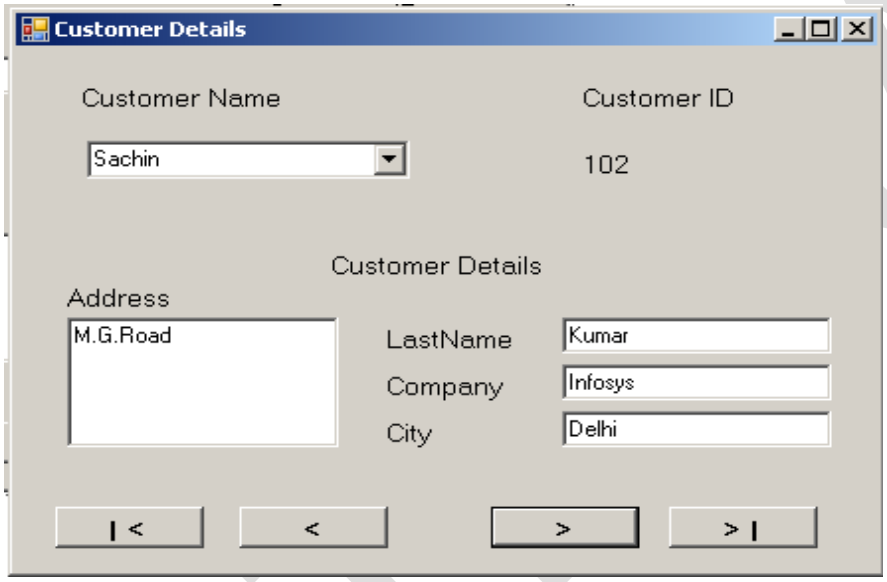
Assignment 11: Navigating records - Simple Data binding

Objective: Illustrate simple binding and navigation of records. Data binding allows binding control properties to data.

Problem Description: Application to browse customer data with the help of Simple binding context.

Estimated time: 5 Min

Step 1: Open the project and form created in the previous assignment. Alter the form (FormCustomerDetails.cs) as shown in figure below.



Step 2: Double click on the form and append the following code in the form Load event handler i.e. FormCustomerDetails_Load event.

```
lblCustID.DataBindings.Add("Text",DsCustomer.Tables[0],"CustomerID"
); txtFName.DataBindings.Add("Text",
DsCustomer.Tables[0],","FirstName");
txtLName.DataBindings.Add("Text",
DsCustomer.Tables[0],","LastName");
txtAddress.DataBindings.Add("Text",DsCustomer.Tables[0],","Address"
);
txtCompany.DataBindings.Add("Text",DsCustomer.Tables[0],","Comopany
"); txtCity.DataBindings.Add("Text",
DsCustomer.Tables[0],","City");
```

Step 3: Write the following code on btnFirst click event handler i.e. "|<<" click event.

```
// Showing First customer record
this.BindingContext[DsCustomer.Tables[0]].Position = 0;
```

Step 4: Write the following code on btnLast click event handler i.e. “>>|” click event.

```
// Showing Last customer record
this.BindingContext[DsCustomer.Tables[0]].Position =
dtCustomer.Rows.Count;
```

Step 5: Write the following code on btnPrevious click event handler i.e. “<” click event.

```
// Showing Previous customer record
this.BindingContext[DsCustomer.Tables[0]].Position -= 1;
```

Step 6: Write the following code on btnNext click event handler i.e. “>” click event.

```
// Showing Next customer record
this.BindingContext[DsCustomer.Tables[0]].Position += 1;
```

Step 7: Build and run the application and use the buttons to navigate through customer records.

Summary of this assignment:
You have just learnt

How to bind data to text box i.e. Simple Binding. How to use BindingContext property.
How to navigate records

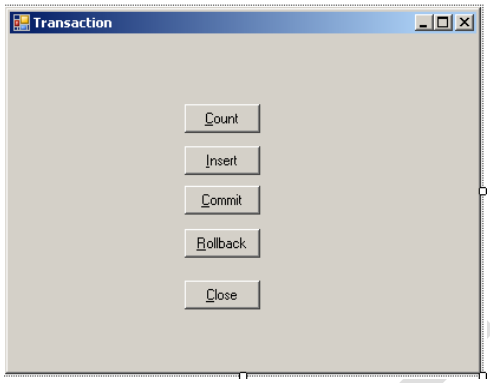
Assignment 12: Working with transaction

Objective: To learn how to work with transaction.

Background: Using Transaction one can commit or rollback all the operations that have been performed after starting the transaction. W can begin the transaction on a connection object and using either commit or rollback we can end the transaction.

Estimated time: **15 Min**

Step 1: Create a Windows form and name it as frmTransaction. Design the form as shown below.



Step 2: Include the SqlClient namespace and declare the connection object and initialize it.

Step 3: Double click on the form and type the following code.

```
namespace Infosys.Lab3
{
    public partial class frmTransaction : Form
    {
        SqlConnection
        NET07TempDB;
        SqlTransaction SqlTrans;
        …
        …
    }
}
```

Step 4: Double click on Form and type the following code in Load event.

```
private void FormTransaction_Load(object sender, EventArgs e)
{
     /* Write code to create NET07TempDB object using
        DataBaseLayer object. Refer Assignment 4.
```

```
        NET07TempDB .Open();
        MessageBox.Show("Connection open");

        SqlTrans = NET07TempDB .BeginTransaction();

        MessageBox.Show("Transaction started and Islation Level is "
       +sqlTrans.IsolationLevel.ToString());
 }
```

Step 5: Double click on 'Close' button and type following code.

```
 private void btnClose_Click(object sender, EventArgs e)
 {
        NET07TempDB .Close();
        MessageBox.Show("Transaction end & Connection Closed");
        this.Close();
 }
```

Step 6: Double click on 'Count' button and type the following code:

```
 private void btnCount_Click(object sender, EventArgs e)
 {
       int TotalRecords;
       SqlCommand CmdNextProductID =new SqlCommand("select count(*)
       from Product", NET07TempDB );

       CmdNextProductID.Transaction = Sqltrans;

       TotalRecords = (int)CmdNextProductID.ExecuteScalar();

       MessageBox.Show(" Total Records are ",
       TotalRecords.ToString());
 }
```

Step 7: Double click on 'Insert' button and type following code.

```
 private void btnInsert_Click(object sender, EventArgs e)
 {
       SqlCommand CmdInsert = new SqlCommand("insert into product
       values(1100,'LUMO',15000)", NET07TempDB);
       CmdInsert.Transaction = Sqltrans;
       CmdInsert.ExecuteNonQuery();
       MessageBox.Show(" Data Inserted to database");
 }
```

Step 8: Double click on 'Commit' button and type following code.

```
 private void btnCommit_Click(object sender, EventArgs e)
 {
```

```
      Sqltrans.Commit();
       MessageBox.Show(" Transaction has been Commmited and saved ");
}
```

Step 9: Double click on 'Rollback' button and type following code.

```
private void btnRollback_Click(object sender, EventArgs e)
{
      Sqltrans.Rollback();
      MessageBox.Show(" Transaction rolled back and not saved ");
}
```

Step 10: Build and Run the Application in the following way.

1. Execute → Count → Insert → Count → Commit → Close.
2. Execute → Count → Insert → Count → Rollback → Count → Close.
3. Execute → Count → Insert → Count.

Summary of this exercise:
You have just learnt
How to create transactions
How to use Commit or Rollback the operations using transaction

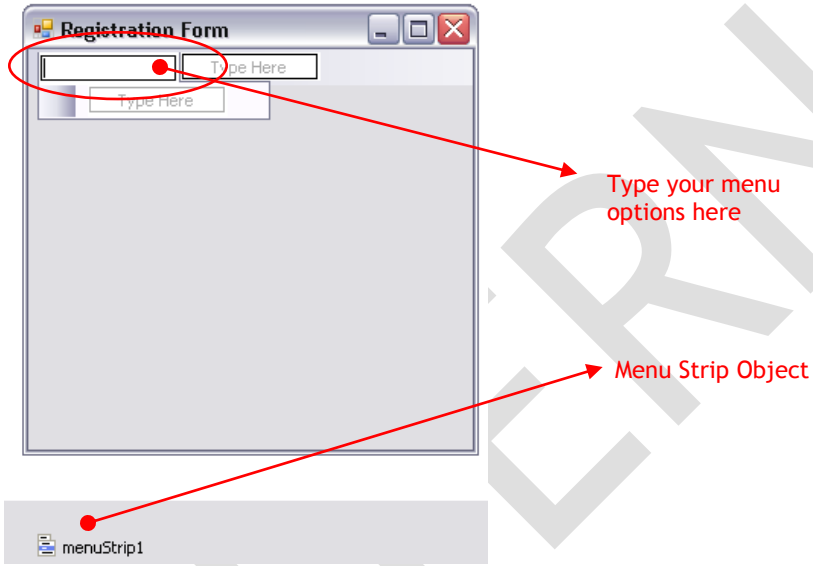
Assignment 13: Creation of MDI Form

Objective: To illustrate how to create MDI Form. Problem Description: Creation of MDI Form and Menu.

Estimated time: 10 Min.

Step 1: Create a new Windows application project as "MDI_Test" and rename the form as frmMDI.

Step 2: Drag and drop MenuStrip control from tool box in frmMDI form. Then form will get a MenuStrip object as shown below:



Step 3: Create following menu options:

- Customer
  - Customer Details
  - Fetch Customer
  - New Product
- Exit

Step 4: Select the form and go to its properties. Set **`IsMdiContainer`** property of frmMDI form to **`True`**.

Step 5: Go to the solution explorer, right click on References then select Add References and select the CustomerDetaiils.Exe (EXE file will be available on bin folder of the project)

Step 6: To display FormCustomerDetails form on click of 'Customer Customer Details' in MDI form, double click on 'Customer Details' menu.

Write the following lines of code in this:

```
FormCustomerDetails MdiCustomerChild = new
FormCustomerDetails(); MdiCustomerChild.MdiParent = this;
MdiCustomerChild.Show();
```

Step 7: Write code for the other menu options to display respective forms.

Summary of this assignment:
You have just learnt

- How to create MDI Parent Form
- How to instantiate the MDI-child forms and displays them.
- How to integrate different projects.