Sending Data from one Page to Another

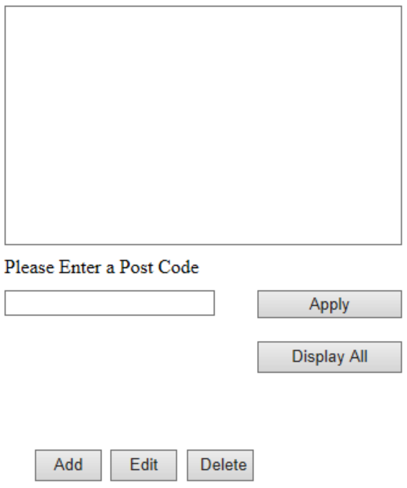
So far in building this application we have covered a great deal of ground.

We have an application allowing us to

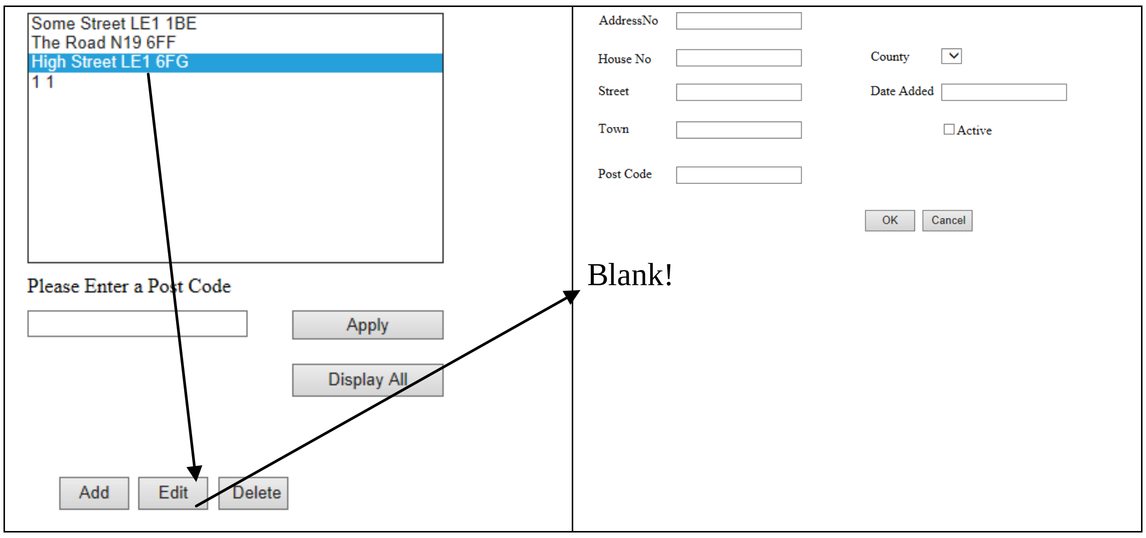
- Generate a list of records
- Add a new record
- Update an existing record
- Delete records

The big problem is that it is all a bit clunky.

The main menu is not populated with data from the database we have to click "display all" to do this.



If we want to edit a record the list on the main menu isn't linked to the edit page and existing data isn't displayed.



To delete a record we have to type the primary key of the record (which the user won’t ever know) the same being true for edit!

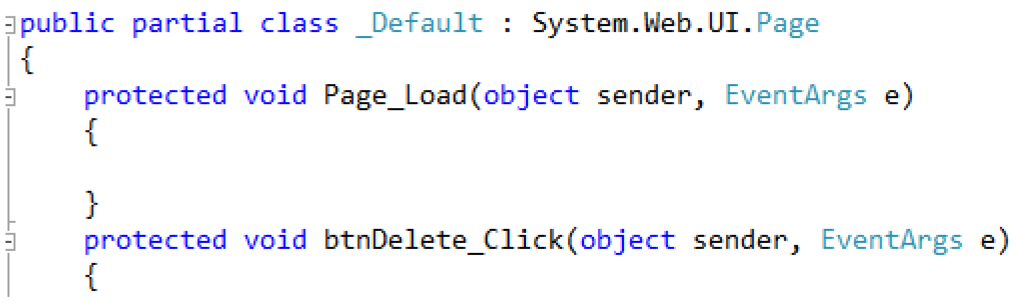
These are just a few of the “rough edges” on the system and we shall spend the next few weeks sorting these out.

Displaying the Data on Form Load

The first reasonable simple thing to fix is making the list box display the data when the form loads.

Each page has what is called a load event which is always triggered at the server <u>every time</u> an event is triggered on the page.

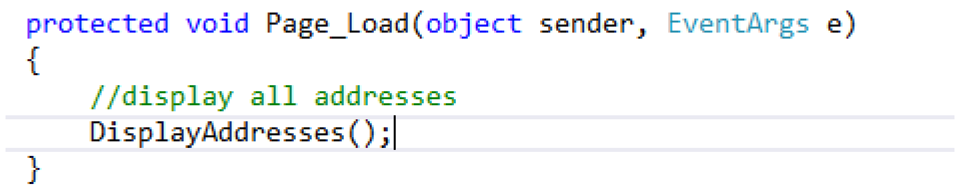
Open the form Default.aspx and view the code. There should be an event handler for the form load event created when you first made the form.



```
public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }
    protected void btnDelete_Click(object sender, EventArgs e)
    {
```

As things stand we have to press the display all button to populate the list box. The simple solution is to copy the same code to the load event handler.



```
protected void Page_Load(object sender, EventArgs e)
{
    //display all addresses
    DisplayAddresses();
}
```

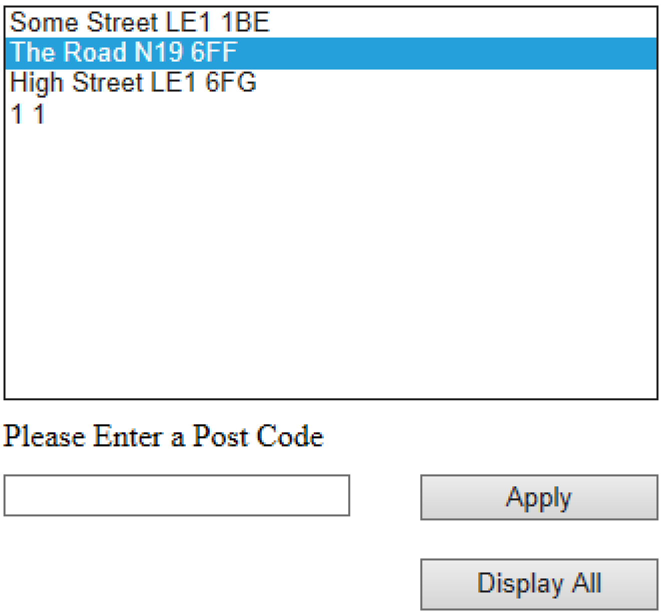
Run the program to see if this works.

There is a problem here though that you will get caught with at some point as a developer, no matter how experienced you think you are!

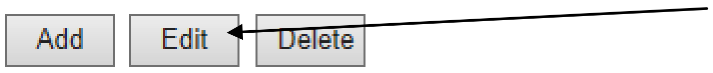
REMEMBER – the load event runs at the server EVERY TIME ANY EVENT IS TRIGGERED!

What does this mean?

OK. A user clicks on an item in the list...



They then press the button for edit or delete...



This sends the page back to the server to trigger the event handler for the edit button.

The problem is that before the server runs the edit event handler it always runs the load event handler.

This code is therefore triggered...

```
protected void Page_Load(object sender, EventArgs e)
{
    //display all addresses
    DisplayAddresses();
}
```

This code re-sets the list and removes the user selection.

By the time the event handler runs for the edit button the program has no idea what the user selected.

IsPostback

IsPostback is a fairly counter intuitive tool which returns a true or false value depending on if the form has been posted back from the server i.e. this is not the first time the form has been displayed.

IfPostback = False then this is the first time the form has been displayed

IfPostBack = True then this is not the first time the form has been displayed

We need to know this to avoid the above list box problem.

If this is the first time the form has appeared update the list box.

If this is NOT the first time for the form then leave the list box alone!

To implement this we modify the load event like so...

```
protected void Page_Load(object sender, EventArgs e)
{
    //if this is the first appearance of this form
    if (IsPostBack == false)
    {
        //display all addresses
        DisplayAddresses();
    }
}
```

Copying the Primary Key Value from the List to the Edit Form

We now have a list box that populates when the form loads but how do we get at the primary key value?

A long time ago we wrote the code that displays the data in the list.

```
Int32 DisplayAddresses()
{
    Int32 AddressNo;//var to store the primary key
    string Street;//var to store the street
    string PostCode; //var to store the post code
    clsAddressCollection AddressBook = new clsAddressCollection();//create an instance of the address collection class
    Int32 RecordCount;//var to store the count of records
    Int32 Index = 0;//var to store the index for the loop
    RecordCount = AddressBook.Count;//get the count of records
    while (Index < RecordCount)//while there are records to process
    {
        AddressNo = AddressBook.AddressList[Index].AddressNo;//get the primary key
        Street = AddressBook.AddressList[Index].Street;//get the street
        PostCode = AddressBook.AddressList[Index].PostCode;//get the post code
        ListItem NewEntry = new ListItem(Street + " " + PostCode, AddressNo.ToString());//create a new entry for the list b
        lstAddresses.Items.Add(NewEntry);//add the address to the list
        Index++;//move the index to the next record
    }
    return RecordCount;//return the count of records found
}
```

In this code we make use of the ListItem class like so...

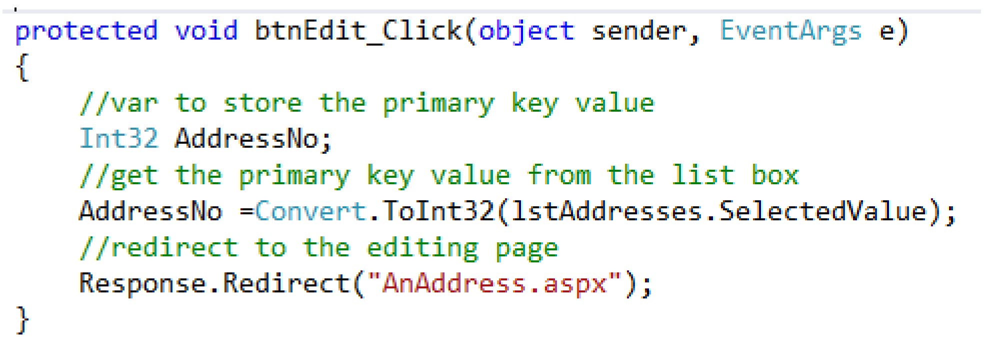
```
AddressNo = AddressBook.AddressList[Index].AddressNo;//get the primary key
Street = AddressBook.AddressList[Index].Street;//get the street
PostCode = AddressBook.AddressList[Index].PostCode;//get the post code
ListItem NewEntry = new ListItem(Street + " " + PostCode, AddressNo.ToString());//create a new entry for the list box
lstAddresses.Items.Add(NewEntry);//add the address to the list
```

We pass the list item class two parameters.

1. The data to be displayed in the list
2. The primary key value of the record we are processing

When the user clicks on an item in the list it sets the SelectedValue property allowing us to get the primary key value for the selected record.

Like so...



```
protected void btnEdit_Click(object sender, EventArgs e)
{
    //var to store the primary key value
    Int32 AddressNo;
    //get the primary key value from the list box
    AddressNo =Convert.ToInt32(lstAddresses.SelectedValue);
    //redirect to the editing page
    Response.Redirect("AnAddress.aspx");
}
```

Before we go any further it is also a good idea to add some validation to make sure that the user has clicked an entry in the list before pressing edit...

```
protected void btnEdit_Click(object sender, EventArgs e)
{
    //var to store the primary key value
    Int32 AddressNo;
    //check the list has been clicked first
    if (lstAddresses.SelectedIndex != -1)
    {
        //get the primary key value from the list box
        AddressNo = Convert.ToInt32(lstAddresses.SelectedValue);
        //redirect to the editing page
        Response.Redirect("AnAddress.aspx");
    }
    else
    {
        //display an error
        lblError.Text = "You must select an item off the list first to edit it.";
    }
}
```

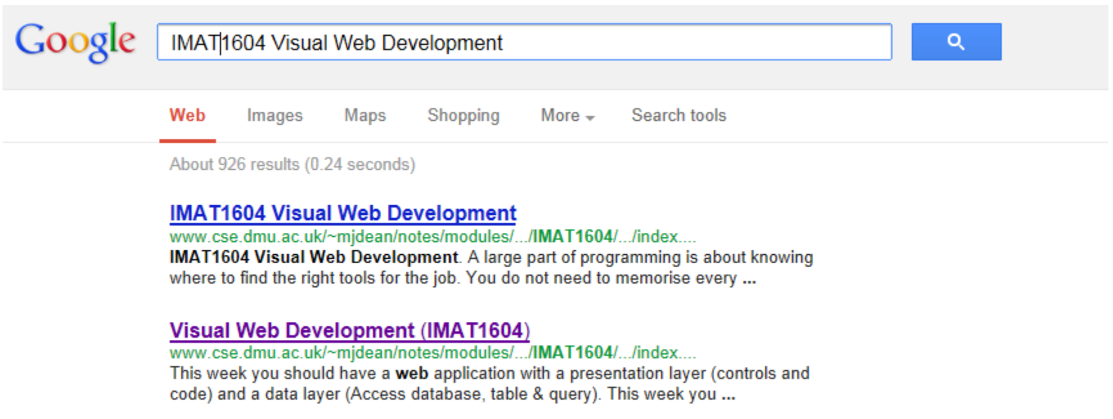
We now have one last question before this code is complete that is “how to send the primary key value to the form AnAddress.aspx?”

There are two techniques, one secure and the other not so secure. We shall look at the second one first and then come back to the secure technique in a later class.

Creating the Query String

Query strings are a useful mechanism for passing data between web pages. You will have seen query strings whilst browsing the web without really understanding what they are for.

If you perform a Google search like so...

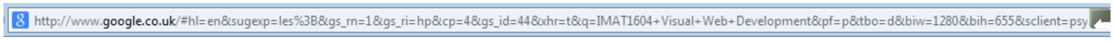


About 926 results (0.24 seconds)

IMAT1604 Visual Web Development
www.cse.dmu.ac.uk/~mjdean/notes/modules/.../IMAT1604/.../index....
IMAT1604 Visual Web Development. A large part of programming is about knowing where to find the right tools for the job. You do not need to memorise every ...

Visual Web Development (IMAT1604)
www.cse.dmu.ac.uk/~mjdean/notes/modules/.../IMAT1604/.../index....
This week you should have a **web** application with a presentation layer (controls and code) and a data layer (Access database, table & query). This week you ...

Notice what is going on in the URL...

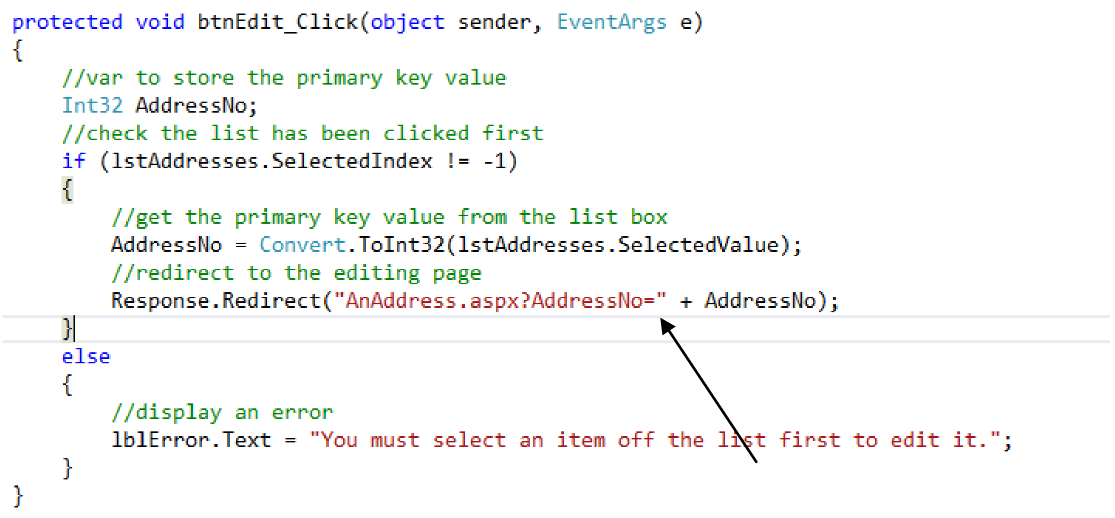


```
http://www.google.co.uk/#hl=en&sugexp=les%3B&gs_rn=1&gs_ri=hp&cp=4&gs_id=44&xhr=t&q=IMAT1604+Visual+Web+Development&pf=p&tbo=d&biw=1280&bih=655&sclient=psy
```

As well as the name of the web site (http://www.google.co.uk/) there are extra characters and codes embedded in the URL. If you read it you will see the search is part of this string.

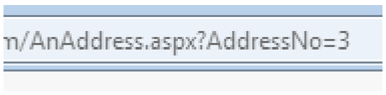
We may use Query Strings within our application to pass data between pages.

To create our query string here we need to concatenate the primary key value into the URL for the page like so...



```
protected void btnEdit_Click(object sender, EventArgs e)
{
    //var to store the primary key value
    Int32 AddressNo;
    //check the list has been clicked first
    if (lstAddresses.SelectedIndex != -1)
    {
        //get the primary key value from the list box
        AddressNo = Convert.ToInt32(lstAddresses.SelectedValue);
        //redirect to the editing page
        Response.Redirect("AnAddress.aspx?AddressNo=" + AddressNo);
    }
    else
    {
        //display an error
        lblError.Text = "You must select an item off the list first to edit it.";
    }
}
```

Run the program click an entry on the list and then press edit. When the edit form is displayed you should be able to see the query string in the URL.



```
n/AnAddress.aspx?AddressNo=3
```

Reading the Data from the Query String

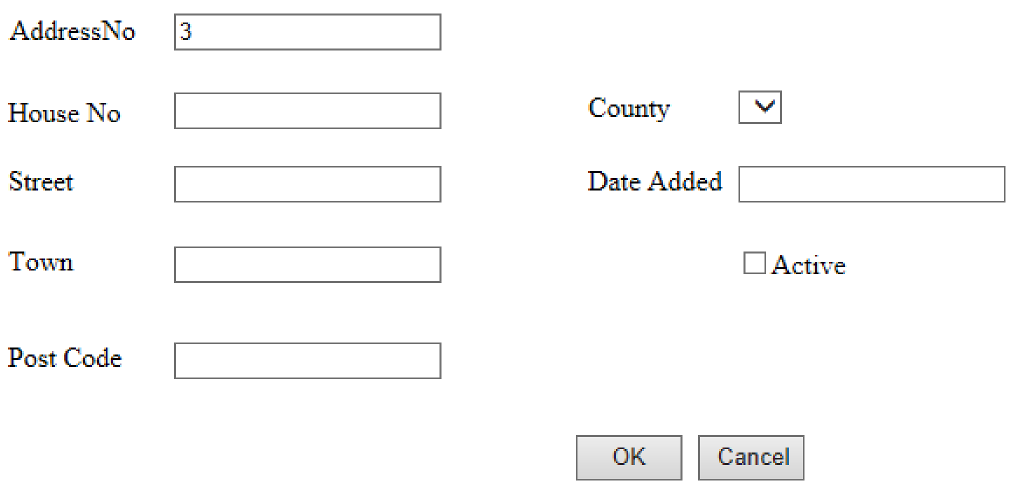
Now that we have a mechanism for sending data to the destination form we need some way of accessing that data in our code.

Modify the load event of the AnAddress.aspx form like so...

```
protected void Page_Load(object sender, EventArgs e)
{
    //copy the data from the query string to the text box txtAddressNo
    txtAddressNo.Text = Request.QueryString["AddressNo"];
}
```

Run the program to see what happens.

You should find that the primary key value is copied into the text box...



AddressNo	3		
House No		County	
Street		Date Added	
Town			☐ Active
Post Code			

While we are at it set up the event handler for the add button on Default.aspx like so...

```
protected void btnAdd_Click(object sender, EventArgs e)
{
    //redirect to the add new page
    Response.Redirect("AnAddress.aspx?AddressNo=-1");
}
```

Using the code for edit set up the delete button so that the primary key value is passed to the page Delete.aspx.

Test the buttons to see if they now send the correct value from one form to another.