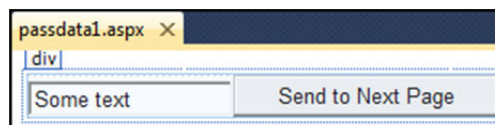


## Lesson 3-11: Send data between pages

You now know how to retrieve data from a page, access the properties of controls and transfer between pages, but quite often you'll want the site to remember something the user has entered on a previous page after they move to a new page.

In this lesson you'll learn a few of the ways to send data from one page to another.

- 1 Open *CSharpTest* from your sample files folder.
- 2 Open *passdata1.aspx* in *Design* view.



You'll see that there is a text box and a button on the page. You're going to write code that will send the contents of the text box to another page, which will display it.

- 3 Send data using *PreviousPage*.

If you use *Server.Transfer* to move between pages, you can actually access the *Page* object of the previous page.

1. Open the code-behind file of *passdata1.aspx*.
2. Add the following code to the *ButtonSend\_Click* event handler:

```
Page.Server.Transfer("passdata2.aspx");
```

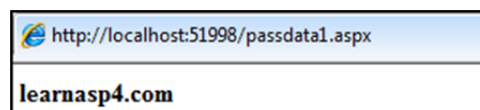
```
protected void ButtonSend_Click(object sender, EventArgs e)
{
    Page.Server.Transfer("passdata2.aspx");
}
```

As mentioned in *Lesson 3-10: Move between pages using C#, Server.Transfer* will keep a link open to the previous page.

3. View *passdata1.aspx* in your browser.



4. Change the text in the text-box if you wish and then click the *Send to Next Page* button.



You'll see that *passdata2.aspx* has picked up the data and displayed it, but because you used *Server.Transfer*, you still appear to be on *passdata1.aspx*.

5. View the code-behind file of *passdata2.aspx*.

You will see the following code:

```
LabelReceivedData.Text =
((TextBox)Page.PreviousPage.FindControl("TextBoxText")).Text;
```

### note

#### Advantages and disadvantages of *Page.PreviousPage*

The *Page.PreviousPage* method isn't widely used for passing data between pages. Most developers prefer the *QueryString* method.

The main disadvantage of the *Page.PreviousPage* method is that it only works if you use *Server.Transfer* to move to the target page.

Using the *Server.Transfer* method means that the person using your site won't be aware that they've moved between pages.

Sometimes that is exactly what you want, but most of the time you will need to use another method of passing data.

### note

#### Other ways of passing data

There are other ways of moving data between pages than the ones mentioned in this lesson.

You could store values in cookies or a database, or even use plain HTML forms as you did in: *Lesson 2-12: Work with HTML Forms*.

## note

### Advantages and disadvantages of *QueryString*

Sending values with *QueryString* is one of the best ways to send values between pages when you don't mind exposing them to the user.

The major disadvantage of *QueryString* is that the values can be very easily tampered with by the user. If you have, for example, a *documentid* value in the query string, the user could very easily change it in order to bypass security and access any document in the system.

You could create a secure system with *QueryString*, but you would have to incorporate a way to check that the user has access to the resource they have requested.

This is the code that retrieves *TextBoxText* from the previous page and sets the *Label* on this page to its *Text* property. It probably looks a little confusing at the moment. You won't be able to fully understand this code until you've completed: *Lesson 5-9: Convert variables using cast and ToString*.

*Page.PreviousPage* is a link to the *Page* object you came from.

*FindControl("TextBoxText")* looks for the *TextBoxText* control on the previous page.

#### 4 Send data using *QueryString*.

The *QueryString* method is the most widely used method of sending data between pages.

While browsing the Internet, you might have noticed that sometimes web addresses look something like:

*www.[site].com/[page].aspx?page=10&product=13*

The values after the question mark are *QueryString* values, and you can use them to send data between pages.

1. Open the code-behind file of *passdata1.aspx*.
2. Replace the *Page.Server.Transfer* line of code with:

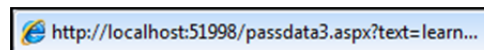
```
Page.Response.Redirect
("passdata3.aspx?text=" + TextBoxText.Text);
```

```
protected void ButtonSend_Click(object sender, EventArgs e)
{
    Page.Response.Redirect("passdata3.aspx?text=" + TextBoxText.Text);
}
```

learnmicrosoftexcel.com

Send to Next Page

3. View *passdata1.aspx* in your browser.
4. Change the text in the text-box if you wish and then click the *Send to Next Page* button.

  
learnmicrosoftexcel.com

You'll see that *passdata3.aspx* has retrieved the text you entered and displayed it.

If you look at the address bar, you can see how the text was sent to *passdata3.aspx*.



5. Close your browser and view the code-behind file of *passdata3.aspx*.

You'll see that the text was retrieved using *Page.Request.QueryString*.

```
if (Page.Request.QueryString["text"] != null)
{
    LabelReceivedData.Text = Page.Request.QueryString["text"];
}
```

You'll fully understand this code later in the course, after completing: *Lesson 7-1: Use the if statement*.