

## Lesson 3-10: Move between pages using C#

You should now have a pretty good idea of how ASP.NET works on a single page. You know how to change the properties of elements on the page and capture the data that the user has entered.

Although you learned how to create HTML hyperlinks in *Lesson 2-7: Display images and links on a page*, there are also ways to move between pages using C#.

- 1 Open *CSharpTest* from your sample files folder.
- 2 Open the code-behind file of *movepage1.aspx*.
- 3 Move to *movepage2.aspx* with *Page.Response.Redirect*.

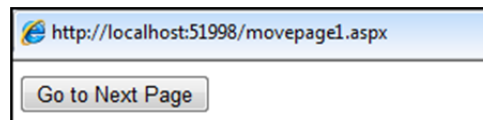
You learned about the *Page.Response* object in *Lesson 3-7: Understand Request and Response*. By using its *Redirect* method, you can send the browser to a different page.

1. Add the following code to the *ButtonMove\_Click* event handler:

```
Page.Response.Redirect("movepage2.aspx");
```

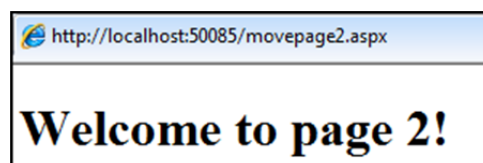
```
protected void ButtonMove_Click(object sender, EventArgs e)
{
    Page.Response.Redirect("movepage2.aspx");
}
```

2. View *movepage1.aspx* in your browser.



3. Click the button.

You'll see that you are redirected to *movepage2.aspx*.



Note that the address bar shows that you are on *movepage2.aspx* now.

- 4 Move to the next page with a hyperlink.
  1. Close your browser and open *movepage1.aspx* in *Source* view.
  2. Add an HTML link after the button using the code:

```
<a href="movepage2.aspx">Next Page</a>
```

```
<div>
  <asp:Button ID="ButtonMove" runat="server"
    onclick="ButtonMove_Click" />
  <a href="movepage2.aspx">Next Page</a>
</div>
```

3. View *movepage1.aspx* in your browser.

### note

#### Advantages and disadvantages of Response.Redirect

The *Page.Response.Redirect* method is the most often-used method of moving between pages using C#.

When you use *Response.Redirect* the new page is loaded exactly as if the user had clicked on a link or typed the address into their address bar. The page they are taken to has no ties to the page they came from. This can be an advantage or a disadvantage, depending on what you are trying to do.

One disadvantage of using *Response.Redirect* or any method of transferring between pages in C# is that you cannot open the new page in a new window.

Links that open in new windows can only be made with HTML or JavaScript.

## note

### Advantages and disadvantages of `Server.Transfer`

`Page.Server.Transfer` is useful if you don't want site visitors to know that they are changing pages. You might want to do this if you have a multi-step process split into several pages.

In *Lesson 3-11: Send data between pages*, you'll also see that `Server.Transfer` can also be used to pass data between pages.

The disadvantage of `Server.Transfer` is that without knowing the address of the actual page they are on, your visitors won't be able to bookmark the page.

As with `Response.Redirect`, you can't use `Server.Transfer` to open a new window.

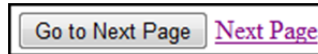
## note

### `Page.Server.TransferRequest`

If you try `Server.TransferRequest`, it will seem to work in exactly the same way as `Server.Transfer`. The difference is in how `ASP.NET` is working in the background.

With `Server.Transfer`, `ASP.NET` doesn't create a completely new `Request`, which means it doesn't re-check things like authentication. You'll learn about authentication in: *Session Nine: Authentication*.

`TransferRequest` creates an entirely new request for the page. Although this is more complete and secure, it is also slower.



- Click the *Next Page* link.

You'll see that this has exactly the same result as `Response.Redirect`.



- Move to the next page with `Page.Server.Transfer`.

You haven't looked at the `Server` object yet. The `Server` object contains methods to tell the web server to carry out certain operations. For now you're interested in the `Transfer` method.

- Close your browser and return to the code-behind file of `movepage1.aspx`.
- Replace your `Page.Response.Redirect` line of code with:

```
Page.Server.Transfer("movepage2.aspx");
```

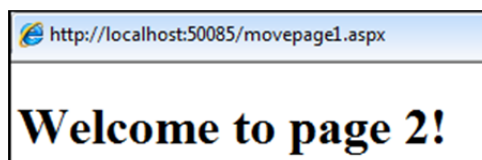
```
protected void ButtonMove_Click(object sender, EventArgs e)
{
    Page.Server.Transfer("movepage2.aspx");
}
```

- View `movepage1.aspx` in your browser.



- Click the button.

You'll see that once again you are redirected to `movepage2.aspx`, but look at the address bar:



The browser still thinks you're on `movepage1.aspx`! This is one of the major differences with `Server.Transfer`.

`Server.Transfer` switches to the new page internally, but doesn't tell the browser. From the perspective of the person viewing your site, they haven't changed pages at all.

Using `Server.Transfer` also keeps a reference to the previous page, as you'll see in: *Lesson 3-11: Send data between pages*.

- Close your browser and Visual Studio.