

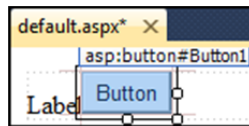
Lesson 3-2: Add event handlers to Controls

An *event handler* is a piece of code that is triggered by something happening. In the last lesson you worked with the *Page_Load* event, which runs whenever a page loads.

In this lesson you'll add some more events to the page.

- 1 Open *CSharpTest* from your sample files folder.
- 2 Open *default.aspx* in *Design* view.
- 3 Add a *Button* control from the *Standard* category of the *Toolbox*.

Place it after the *Label* you added in: *Lesson 3-1: Change properties with C#*.



- 4 Set the *ID* property of the new button to: **ButtonChangeText**
- 5 Add a *Click* event handler to the button.

1. Make sure you are in *Design* view.
2. Select the *Button* so *ButtonChangeText* appears in the *Properties* window.

3. Click the *events* button  in the *Properties* window.

It can be found just below the drop-down menu.

You will see a list of the possible events for the *Button* control.

4. Double-click in the empty box next to *Click*.



You will be automatically taken to the code-behind file of *default.aspx* and will see that some code has been automatically added:

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

This is the *event handler* for the button's *click* event. When the button is clicked, any code between the *{* and *}* will be run.

5. Add the following line of code to the new event handler:

LabelOutput.Text = "Text changed!";

```
protected void ButtonChangeText_Click(object sender, EventArgs e)
{
    LabelOutput.Text = "Text changed!";
}
```

note

Changing events in Source view

Although you can use the *Properties* window to change properties in *Source* view you are unable to add events in this view.

If you select an ASP.NET control in *Source* view, you'll see that the *events* icon disappears from the *Properties* window.

note

The order of events

In this lesson you have two events both trying to set the *Text* property of a label control, but the *click* event 'wins'. This is because the events are always processed in a certain order.

When the button is clicked, the page reloads and the *Page_Load* event sets the label's text to *Hello World!* At a later point in time the *Click* event occurs and sets the label control's text property to *Text Changed!*

On a real web site you'd probably try to avoid this and make sure that only one piece of code affects the label at a time.

- View the page in your browser and click the button.


You'll see that the text in the label is changed. This is because the *click* event comes after the *Page_Load* event (see sidebar).




- Close your web browser.

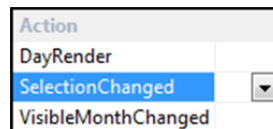
- Add a *Calendar* control to the page from the *Standard* category of the *Toolbox*.

- Change the *ID* property of the *Calendar* control to: **CalendarSelect**

If your *Properties* window is still displaying *Events* you will have to switch back to *Properties* by clicking the *Properties* icon. 

- Add an event handler to the *Selection Changed* event of the *Calendar* control.

- In the same way as you did for the button, select the calendar in *Design* view and then click the *events* button  in the *Properties* window.



As you can see, *Calendar* has a different set of events to *Button*. You'll learn more about the events of different controls in: *Session Four: ASP.NET Controls*.

- Double-click in the empty box next to *SelectionChanged* in the *Properties* window.

The *SelectionChanged* event happens when the user selects a date from the calendar.

- Add the following code to the event handler that is automatically generated:

LabelOutput.Text = CalendarSelect.SelectedDate.ToString();

```
protected void CalendarSelect_SelectionChanged(object sender,
{
    LabelOutput.Text = CalendarSelect.SelectedDate.ToString();
}
```

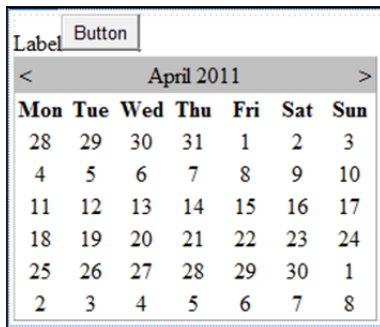
This code will set the *Text* property of your label to the date that was selected in the calendar.

You will learn about *ToString* in: *Lesson 5-9: Convert variables using cast and ToString*.

- View the page in your browser.
- Click on a date in the calendar.

You will see that the event handler code runs and the label is filled in with the date that was selected.

- Close your web browser and Visual Studio.



note

'Hidden' events

Although the vast majority of ASP.NET events can be accessed through *Design* view, you might be wondering where you can click to set page events like *Page_Load*.

Unfortunately, it's not possible to access those events through the designer so you have to type them in manually if you want to use them.

The important 'missing' events are:

Page_Init

Happens before *Page_Load*.

Page_PreRender

Happens after *Page_Load*, just before the page's controls are converted into HTML code.

Page_Unload

The last event in the page life cycle. Happens after the page has been sent to the user.

You won't be using any of these in this book, but it is good to be aware of their existence.

