

Lesson 3-9: Work with ViewState

important

ViewState and ControlState

You might expect *ViewState* to store the state of all of your controls, but many controls use *ControlState* instead.

Most controls that contain information that is input by the person visiting the page will automatically store their state in *ControlState*, regardless of whether *ViewState* is enabled or disabled.

ControlState is similar to *ViewState*, but can't be switched off.

If you want to see *ControlState* in action, try putting some text in the text box in this lesson and notice that the text persists, even after a post-back.

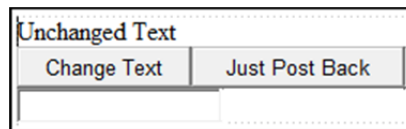
Don't worry too much about switching off *ViewState*, as ASP.NET will only use it if it isn't already storing the state in *ControlState*.

Generally, you should only need to worry about *ViewState* if you are setting the properties of controls using code.

Have you ever used an online form and become really annoyed when all of the contents of the controls disappeared after a post-back? This is the problem that *ViewState* elegantly solves.

In this lesson you'll see *ViewState* in action and discover how you can also use *ViewState* to store values of your own

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



You'll see that there is a label control, two button controls and a text box control.

- 3 See *ViewState* in action.

1. View *viewstatetest.aspx* in your browser.
2. Click the *Change Text* button.



You'll see that the label at the top of the screen changes to say: *Text Changed!*

3. Click the *Just Post Back* button.



You'll see that the label continues to say *Text Changed!* even after the page is posted back; the *state* of the label has been maintained thanks to *ViewState*.

4. See the hidden *ViewState* data.

As you've done before, view the source of the page by right-clicking (in Internet Explorer) and then clicking *View Source* from the shortcut menu.

5. Look for the *aspNetHidden* div near the top of the page.

```
<div class="aspNetHidden">
<input type="hidden" name="__VIEWSTATE"
</div>
```

The middle line contains all of the *ViewState* data. It looks like nonsense because it's encrypted, but it is easy to decrypt using tools available on the Internet so don't rely on it to be secure.

- 4 Disable *ViewState* and see the difference.

1. Close your browser and return to the *Design* view of *viewstatetest.aspx*.



2. Change the *EnableViewState* property of the *LabelText* control to: **False**
3. View *viewstatetest.aspx* in your browser.
4. Click the *Change Text* button.



You'll see the text change as before.

5. Click the *Just Post Back* button.



This time the state of the label hasn't been maintained by *ViewState*, so the label's text reverts to its original value.

5 Store a value in *ViewState*.

As well as letting ASP.NET automatically store the values of your controls in *ViewState*, you can also use *ViewState* to store values of your own.

1. Open the code-behind file of *viewstatetest.aspx*.
2. Put the following code in the *Page_Load* event handler:

```
ViewState["MyText"] = "Hello World!";
```

```
protected void Page_Load(object sender, EventArgs e)
{
    ViewState["MyText"] = "Hello World!";
}
```

This code will store the text *Hello World!* in *ViewState*, next to a *key* of *MyText*. The *key* is used to retrieve the text from *ViewState*.

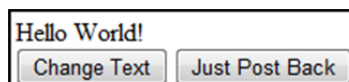
3. Replace the code in the *ButtonChangeText_Click* event handler with:

```
LabelText.Text = ViewState["MyText"].ToString();
```

```
protected void ButtonChangeText_Click(object sender, EventArgs e)
{
    LabelText.Text = ViewState["MyText"].ToString();
}
```

This sets the *Text* property of the *LabelText* control to the value you stored under *MyText* in *ViewState*.

4. View the page in your browser.
5. Try clicking the *Change Text* button.



You'll see that the text *Hello World!* is retrieved from *ViewState* and displayed in the *Label*.

Storing values in *ViewState* is useful when you have a value that you need to keep after a post-back but don't want to put it in a visible control on the page.

anecdote

ViewState troubles

I was once called upon to urgently solve a problem while another developer was away on holiday. His web application had suddenly stopped working and nobody knew why.

The page in question had a large table that was populated with data from a database and *ViewState* had been left on.

Because *ViewState* makes the browser send back the state of the control, the browsers were sending back a huge amount of data every time the page posted back. This made the page run very slowly, but didn't actually stop it from working.

Over time, as the database grew, more and more data was being sent back until, on that day, it exceeded the request limit of 4 megabytes and ASP.NET stopped it from working.

After simply disabling *ViewState* on the table, everything worked perfectly again.