

Lesson 3-13: Edit the Web.config file

note

XML

XML isn't actually a specific programming language; it's really a standard for storing information. The HTML code you wrote earlier was a form of XML.

XML always works as a series of nested tags enclosed in < and > symbols, which open and close in the same way HTML tags do.

The *Web.config* file recognizes different tags to HTML, but basically works the same way.

Because XML can be used to store any kind of value in a hierarchy it's an extremely versatile way to store data.

You might have been wondering what the *Web.config* file does, as you've seen it in every project you've worked on so far.

Web.config is an XML file (see sidebar) which contains settings to tell the web server how to handle your web site. By changing the settings in *Web.config* you can modify many important settings.

Changing settings in Visual Studio often automatically makes changes to your *Web.config* file.

- 1 Open *CSharpTest* from your sample files folder.
- 2 Open *Web.config*.

You will see that there is actually very little code inside the *Web.config* file. This is because you are using ASP.NET 4. ASP.NET 4 automatically manages a lot of things that had to be in *Web.config* in previous versions.

```
<?xml version="1.0"?>
<!--
For more information on how to configure your ASP.NET app
http://go.microsoft.com/fwlink/?LinkId=169433
-->
<configuration>
  <system.web>
    <compilation debug="true" targetFramework="4.0" />
  </system.web>
</configuration>
```

- 3 Use *Web.config* to switch *customErrors* on.

customErrors is a setting that determines whether ASP.NET displays the details of error messages to visitors to the site.

On a live site, the details of errors should be hidden from visitors as they contain glimpses into the site's code which may be a security risk.

1. Add the following line of code to *Web.config* inside the *system.web* tag:

```
<customErrors mode="On"></customErrors>
```

```
<configuration>
  <system.web>
    <compilation debug="true" targetFramework="4.0" />
    <customErrors mode="On"></customErrors>
  </system.web>
</configuration>
```

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

Either do this by right-clicking on it and clicking *View in Browser* from the shortcut menu or by opening it and clicking *Debug* → *Start Without Debugging*.

You need to view the page without debugging, as otherwise Visual Studio will stop execution before the error is shown.

note

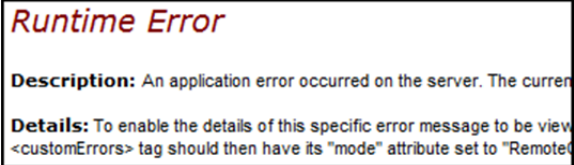
IIS and Web.config

As mentioned in the sidebar in *Lesson 2-12: Work with HTML Forms*, ASP.NET sites run on Microsoft IIS servers.

Microsoft IIS has its own administration interface which can also change the settings in *Web.config*.

Be aware that if you or your web administrator change settings using the IIS admin interface, the *Web.config* file on the server may no longer match the one you have on your own machine.

You will see that a *Runtime Error* dialog is displayed, but it offers no further details of the error itself. In fact, it shows instructions of how to modify your *Web.config* file to show the error details.

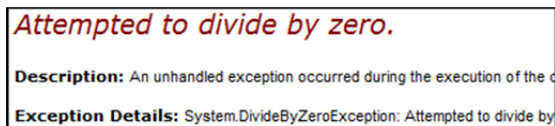


4 Switch *customErrors* off.

1. Close your browser and return to *Web.config*.
2. Change the *customErrors* line to:

```
<customErrors mode="Off"></customErrors>
```
3. View *crashme.aspx* in your browser.

This time you will see the full details of the error message, as you have switched off *customErrors*.



You will also see a snippet of code that shows which line caused the error. This is the reason *customErrors* shouldn't be switched off on a live web server, as it might show code that could help someone break into your system.

```
Line 13:      {
Line 14:          int Zero = 0;
Line 15:          int Crash = 1 / Zero;
Line 16:      }
Line 17:  }
```

There is a third setting for *customErrors* called *RemoteOnly* (see sidebar).

5 Extend the length of sessions.

In *Lesson 3-11: Send data between pages*, you learned that a user's session on an ASP.NET site lasts for 20 minutes after their last request, but that is only the default. Using *Web.config*, you can make sessions last longer.

1. Close your browser and open *Web.config*.
2. Add the following line inside the *system.web* tag:

```
<sessionState timeout="60"></sessionState>
```

```
<system.web>
  <compilation debug="true" targetFramework="4.0" />
  <customErrors mode="Off"></customErrors>
  <sessionState timeout="60"></sessionState>
</system.web>
```

This will make sessions last for 60 minutes instead of 20.

There are many, many other settings that can be changed in *Web.config*, but for now it is just important that you understand what *Web.config* does and how you can edit its settings.

6 Close Visual Studio.

note

customErrors RemoteOnly

The default setting for *customErrors* is actually *RemoteOnly*.

In *RemoteOnly* mode, users who are viewing the site from the server itself are able to see the details of error messages, while people viewing the site normally can't.

If you don't set *customErrors* at all, it will default to *RemoteOnly*.