

Changes to the final release of ASP.NET 2.0 that affect the *Upgrader's Guide: VB Edition*

The web.config file is added to a web site when it's created.

With Beta 2, the web.config file wasn't added to a web site until you ran the application with debugging. This is mentioned in the last paragraph on page 34 of the *Upgrader's Guide*. With the final release, the web.config file is added when you create the web site.

The default validation schema for rendering HTML has changed.

With Beta 2, the code in the aspx file was validated using XHTML 1.1. With the final release, the code is validated using XHTML 1.0 Transitional. This results in several changes. First, the DOCTYPE directive of each new web page you create indicates that XHTML 1.0 Transitional is used instead of XHTML 1.1. In addition, warning messages are displayed for features that are considered outdated, and error messages are displayed for features that will not be supported in future releases of HTML. For example, if you specify the height or width attribute for a td element, you'll get a warning message indicating that the attribute is considered outdated and that a newer construct is recommended. Or, if you specify the height attribute for the tr element, you'll get an error message indicating that it is not a valid attribute. These attributes, as well as others, should now be coded as properties of the style attribute.

XHTML 1.0 Transitional also marks attribute values not enclosed in quotes as errors. Because Visual Studio 2005 generates attribute values with quotes, you don't have to worry about any code you generate using the Web Forms Designer. If you enter code directly using the HTML editor, however, you'll want to be sure to code quotes around all attribute values.

Even though Visual Studio identifies some of these constructs as errors, you should realize that your applications will still work correctly without any changes. That means that all of the applications we developed for the *Upgrader's Guide* will still work. Because these errors will cause problems with future releases of HTML, however, you should correct them in any new applications you create.

The change to XHTML 1.0 Transitional also changes the way you force ASP.NET 2.0 to revert to HTML rendering. This may be necessary if you run applications created with ASP.NET 1.x under ASP.NET 2.0 as described on pages 480 and 481 of the *Upgrader's Guide*. The element you use to specify the rendering has been changed from xhtml11Conformance to xhtmlConformance. In addition, instead of coding the enableLegacyRendering attribute with a value of "true," you code the mode attribute with a value of "Legacy" like this:

```
<xhtmlConformance mode="Legacy" />
```

The default value for the OldValuesParameterFormatString attribute has changed.

The OldValuesParameterFormatString attribute is used to determine the names that are given to parameters that hold original column values for a row in an Access, SQL, or object data source. With Beta 2, the default value of this attribute was original_{0}, which means that the parameters were given the names of the columns prefixed with "original_". This is reflected in the aspx code

for the data sources shown on pages 129, 174, and 208 and 209 of the *Upgrader's Guide* and the update and delete methods shown on page 265 for the data source shown on page 259.

With the final release, the default value of the `OldValuesParameterFormatString` attribute has been changed to `{0}`. This indicates that the parameters for original column values should just be given the name of the column. That means that the names of the delete and update parameters that are generated for the SQL data source shown on page 129 will be `@CategoryID` instead of `@original_CategoryID`. However, the names of the parameters that are generated for the data sources shown on pages 174, 208, and 259 will remain unchanged. That's because these data sources use optimistic concurrency. In that case, the code that's generated for the data source will include an `OldValuesParameterFormatString` attribute that's set to a value of `original_{0}`. In other words, the default value for this attribute will be automatically overridden if you specify that you want to use optimistic concurrency.

If you've used Beta 2 to develop an application with a data source that updates the underlying database, you'll need to modify the application so that it works with the final release. To do that, just add the `OldValuesParameterFormatString` attribute to the data source and assign it a value of `original_{0}`. That way, the data source will work with the parameters that were generated for it. We've made this change to the applications that you can download from our web site, so these applications should run without any trouble.

The schema of the database used to store profile data has changed.

In chapter 10 of the *Upgrader's Guide*, you learn how to use profiles to store information related to a user in a persistent data store. If you have used this feature to store data, you should realize that the schema of the database that's used to store the profile data has changed. This change means that you will not be able to use an existing profile database (`AspNetDB.mdf` by default) without modification.

One way to correct this problem is to delete and recreate the database. If you don't want to reenter the information you've added to the database, however, you can use a tool that comes with the .NET Framework to reconfigure the database.

Before you can use this tool, you'll need to attach the database you want to reconfigure to SQL Server Express. Then, you can run the program named `aspnet_regsql.exe`. You'll find this program in the `C:\Windows\Microsoft.NET\Framework\version` directory, where *version* is the version number of the .NET Framework you're using. Finally, you should detach the database. Note that to attach and detach the database, you can use batch and SQL files like the ones provided for attaching and detaching the Halloween database that's used by the applications in the *Upgrader's Guide*. For more information on these files, see appendix A of the book or the Readme file that's included with the applications download.

You should also realize that we don't include the `AspNetDB.mdf` database with the applications that you can download from our web site. Because of that, these applications should work without any problem.

Some of the menus have changed slightly.

On pages 36 and 37 of the *Upgrader's Guide*, we indicate that you use the `File→New→Web Site` command to create a new web site. This command has changed in the final release. It is now just `File→New Web Site`. Similarly, the `File→Open→Web Site` command has changed to `File→Open Web Site`.

The techniques for adding folders to a web site have changed.

On pages 38 and 39 of the *Upgrader's Guide*, we explain how to add a regular folder or a special folder to a web site. These techniques have changed in the final release. Now, to add a regular folder, you right-click the web site in the Solution Explorer, select the New Folder command, and then enter the name of the folder. To add a special folder, you right-click the web site in the Solution Explorer, point to the Add ASP.NET Folder command, and then select the folder you want to add from the list that's displayed.

Some of the options have been deleted from the Create User page of the Web Site Administration Tool.

On page 299 of the *Upgrader's Guide*, the page that the Web Site Administration Tool displays for creating a user includes options for sending the password to the new user and for generating the password automatically. These options aren't available with the final release.

Data formats aren't applied to bound fields of the GridView and DetailsView controls.

In the Product List application shown on page 161 of the *Upgrader's Guide* and in the illustrations on pages 155 and 157, a currency format is applied to the unit price field of the GridView control. Similarly, in the Product Maintenance application shown on page 205, a currency format is applied to the unit price field of the DetailsView control. This doesn't work in the final release because, by default, bound fields are HTML-encoded. That means that the fields are converted to strings that are displayed by the browser rather than being interpreted as HTML. That way, any malicious scripts that may have been stored in the data source aren't executed.

To format the unit price fields properly, you can turn off HTML encoding. To do that, just set the `HtmlEncode` property of the field to `False`. This also prevents an error that occurs if you try to delete a product using the Product Maintenance application. This error occurs because, before a product is deleted, the `DetailsView1_ItemDeleting` procedure shown on page 211 is executed to remove the dollar sign from the value of the unit price field. But if a dollar sign hasn't been applied, this procedure removes the first digit of the unit price, which causes a concurrency error to occur. If you apply the dollar sign by turning off HTML encoding, this procedure will work as intended.