

# Introduction to Visual Studio 2005

**V**isual Studio 2005 is the Integrated Development Environment (IDE) for the .NET platform. It was designed with the productivity of the developer and teams of developers in mind. There are different versions of Visual Studio available, ranging from Visual Studio Express to Visual Studio Team System. Each version has a different target market in mind.

The goal of the IDE is to make it as easy as possible for developers to create productive, bug-free applications. To

accomplish this goal, Visual Studio offers a wide range of visual development tools and debugger interfaces. The IDE utilizes a series of movable and dockable windows that give developers quick access to information about the project or solution they are developing. Some of the standard windows include the Solution Explorer, Properties, Server Explorer, Class View, and ToolBox.

## Visual Studio Project Types

Visual Studio gives developers the ability to create a variety of application projects, including Web sites, WinForm applications, class libraries, Console applications, as well as the new .NET 3.0 projects and custom templates exported by the developer. Projects can be developed in VB.NET, C#, J#, and C++. Support for mobile and compact development is also supported, but requires additional free SDK downloads from the Microsoft site.

## Visual Studio File Types

Visual Studio offers the developer the ability to edit all kinds of files, including Web pages, WinForms, class files, style sheets, resource files, XML documents, databases, and JavaScript, as well as just about any file needed to build any of the supported applications. Files can be added as needed to a project at any time. If a file is added to a project and there is a special folder associated with that file type, Visual Studio prompts you to place the file in that folder. For example, if you add a new class file to a project it asks if you want to place the file in the App\_Code folder.

## Visual Studio Windows

Visual Studio 2005 offers most of the standard windows available in previous versions with some new additions. A new feature is the Error Notification Assistance window, which is available from most syntax errors. It offers an explanation of the error and some suggested corrections.

Another new feature is the Tag Navigator bar, which appears at the bottom of a design page. Clicking on a tag name in the bar sets the focus to that HTML element.

## Visual Studio and Controls

The use of controls is vital to the development process on the Windows platform. Visual Studio provides full visual experience in both design and source view of Web pages and WinForms. Controls can either be dragged from the control ToolBox or created by hand in source view. If the control has a designer associated with it, you can also interact with the control at designtime and often invoke custom editors provided with the control.

## Running and Testing Web Applications

Executing applications from the development environment is important, and Visual Studio provides several methods for you to execute the Web site you are developing. You can define a start page for the site that loads when you start the application either with or without debugging. You can also open any page or handler directly in a browser. Visual Studio 2005 ships with a built-in personal Web server that requires no configuration by the developer to conveniently serve the current Web site.

Code Snippets, Intellisense, and Code Reuse

Productivity is the primary reason development environments were created. Visual Studio 2005 introduces Code Snippets, the easy ability to export and import templates and files, and improved Intellisense capabilities. Code Snippets are an extendable library of code patterns that can be quickly embedded in an application from a context menu or set of keystrokes. Entire application projects or individual files from a project may be exported to be reused for other projects in the future. Intellisense has always been one of the best features of Visual Studio, and the 2005 edition has a much improved Intellisense feature.

Exporting Web Sites

Another new feature to Visual Studio 2005 is the ability to export projects, Web sites, and pages. In the past, creating a custom project type was a very complicated process. Now you can create a base project or Web site and choose to export the project or site as a template from the File menu. After starting the export process, a wizard walks you through the steps to export. After this is done, the project or site is added as a project type to the New Project or New Web Site choices. Similarly, a single file or page can be exported for consistent reuse in future projects.

Publishing Web Sites

Posting a finished Web site can be done right from Visual Studio. The Web Site Copy utility allows you to post and synchronize content from the development workstation to a remote server through FTP, XCopy, the local instance of IIS, or a remote site with FrontPage Extensions installed.

Debugging in Visual Studio

Visual Studio offers a rich set of debugging tools, enabling you to step through application code, SQL Server Stored Procedures, and even through JavaScript. Breakpoints can easily be set by pressing F9 on the line to initiate debugging. Stepping through the application is done by pressing F10 to move to the next line of code or F11 to step into a method or property.

Debugging is initiated by pressing F5 or attaching to an existing process. The application does not necessarily have to be a .NET application. Remote debugging can even be done on a machine where your account has the appropriate credentials.

New features to Visual Studio are the DataTips and Design-Time Syntax notifications. DataTips are available when the mouse cursor is placed over the variable. This allows you to drill into complex data types in a modeless window. Syntax notifications indicate when code has been added to a page that violates proper syntax for the language, and works both in markup and code-behind pages.

VISUAL STUDIO VERSION	DESCRIPTION
Visual Studio Express	This is a lightweight, free version of Visual Studio targeted at the novice, student, and hobbyist. While there are limitations to the capabilities, developers will find with this version that they can develop just about any application from Web sites to games.
Visual Studio Standard	This version of Visual Studio is targeted at the individual developer who builds n-tier applications and Web sites.
Visual Studio Professional	This is a step up from the Standard edition. It is targeted at developers in small teams and offers a slightly improved user experience, full XML editing, Crystal reports, remote debugging, SQL 2005 integration, and more.
Visual Studio Team System	This version is targeted to large development teams. It offers a full range of lifecycle, communication, and collaboration tools vital to the large team development and testing process.

# Create a Web Site with Visual Studio 2005

To create a new Web site in Visual Studio 2005 you can either choose File and then New Web Site from the menu or click the New Web Site icon on the application toolbar. This displays the New Web Site dialog box. You will see a variety of Web site projects, depending on what is installed on your system.

Your options will include ASP.NET Web Site, ASP.NET Web Service, and Empty Web Site. In the New Web Site dialog box you will need to type the name of your Web site and the location on your computer that you want the site created.

One of the great new features of ASP.NET 2.0 is that you place the actual Web site anywhere you want and are not limited to a virtual Web site inside of IIS. Visual Studio 2005 ships with Personal Web Server, which is a virtual Web server built into the development environment. When

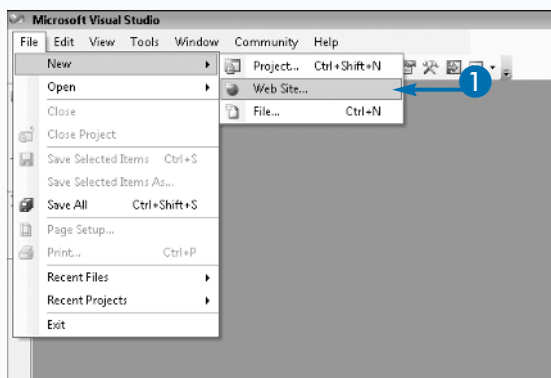
you launch a Web site from within Visual Studio 2005 it opens the site in a new browser, but you see a path similar to `http://localhost:5465/MyWebSite/default.aspx`.

After you complete the creation of your new Web site, you will see a default.aspx, a web.config, global.asax, and style sheet files in your new site. You can now proceed in developing your new Web site, starting with the default.aspx page. You can add new pages, style sheets, themes, Master pages, and handlers by either right-clicking on the location in the solution explorer and selecting the file type or by typing the handler signature manually.

Visual Studio 2005 first shipped without a Web site project. Microsoft eventually released an update that included the concept of a Web site project. This allows developers to add additional projects to a Web site solution. You can do this by choosing File and then Add. Typically, a class library will be added to the solution.

## Create a Web Site with Visual Studio 2005

1 Click File → New → Web Site.

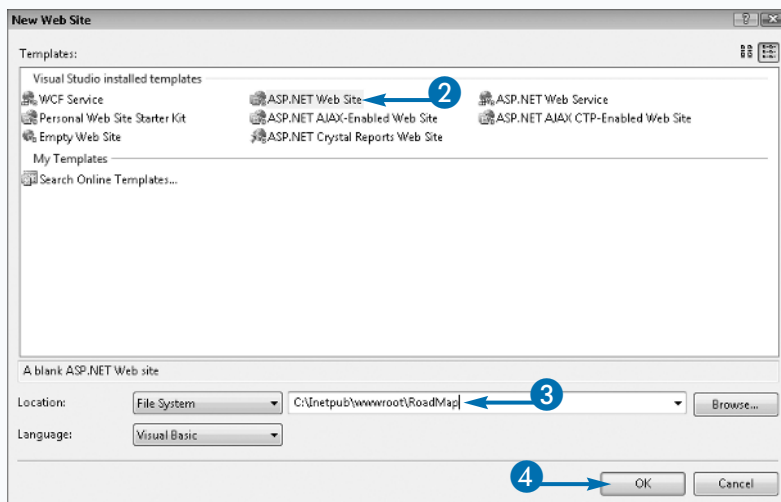


2 Select ASP.NET Web Site from the New Web Site dialog box.

3 Set the path to the new Web site.

4 Click OK to create the new Web site.

**Note:** You may have more or less project types to choose from depending on what has been installed on your workstation.



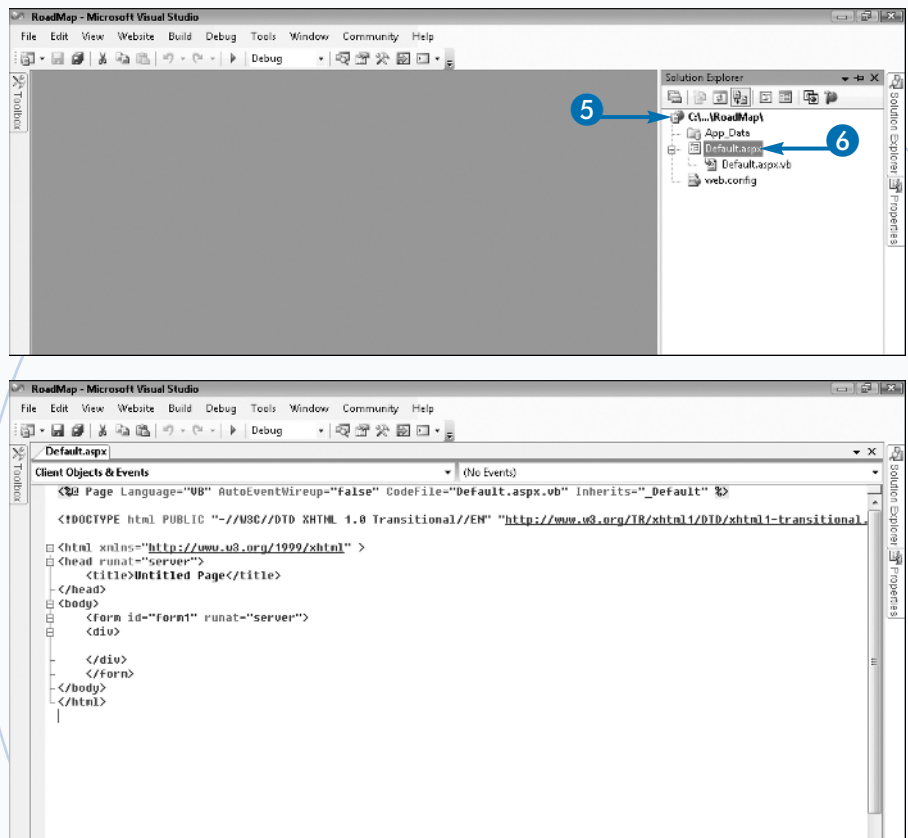
- 5 Open the Solution Explorer window.

**Note:** Visual Studio leverages dockable and hideable windows. If a window is hidden, there will be a tab on the side of Visual Studio that will expand when you move the mouse over it.

The new Web site will have a default.aspx page and a web.config file.

- 6 Double-click the default.aspx Web page to see the design code.

The Web page opens displaying its design code.



## Extra

If you have installed any Starter Kits, Templates, ASP.NET AJAX, or possibly the ASP.NET 3.0 SDK, you will see more options. Starter Kits are specialized prebuilt projects that can be created as a base framework application to build your new Web site. Templates are very similar to Starter Kits, but are exported from Visual Studio, which is detailed later in this chapter. Recently, Microsoft released the ASP.NET 3.0 SDK, which is the latest version of the .NET framework, but is beyond the scope of this book.

You will notice a randomly assigned port, 5465 in this example, used by Personal Web Server to differentiate it from any standard install of IIS. The default installation of IIS will typically use port 80, the RFC-defined port for Web or HTTP activity. An RFC is a Request for Comment and is used to define standards for the Internet and many other network-related protocols. Protocols are executed over ports that range from 1 to 65555. Port 80 is the standard port for HTTP traffic. Another reason a nonstandard port is used is to make it harder for hackers to attack a development machine.

# Add a Web Page to a Web Site

**A**fter you create a Web project, you need to add pages to the site. New pages or files may be added either through keystrokes or menu selections. Most developers opt for displaying the context menu from the root node of the Solution Explorer window. But the same can be accomplished for root pages of a site by pressing CTRL+N.

Adding new pages to a site through the context menu in the Solution Explorer gives you the ability to add the page to a subdirectory, whereas the keystroke only allows you to add a page to the Web site's root folder. To add a new page at any location in the site, right-click the folder to which you want to add the new page. In the context menu that appears, choose Add New Item.

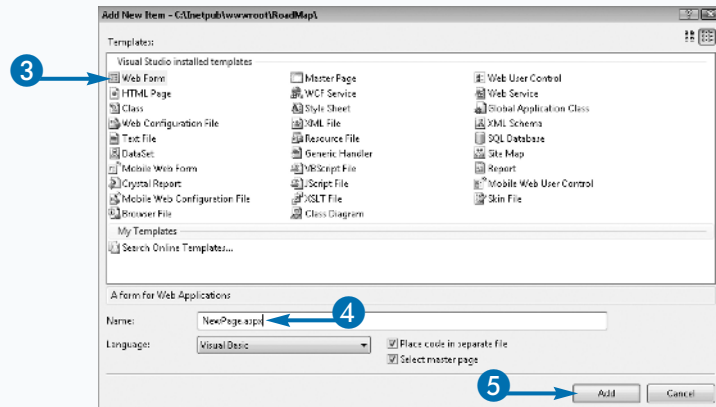
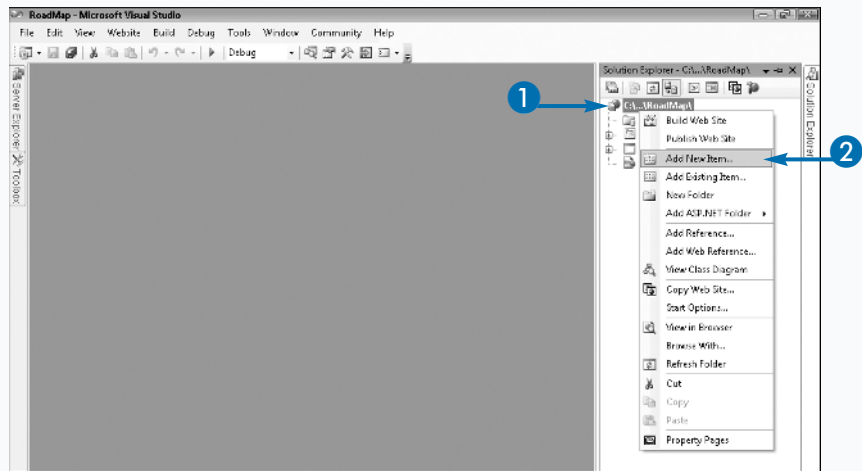
The Add New Item dialog box that appears contains a long list of file types that may be added in this location.

For example, if you display this dialog box in the App\_Code folder you only see file types for classes and other nonvisual items. There will generally be three file types you will add to a Web site: Web form (commonly referred to as a page), Master page, and Web user control.

If you are adding a new Web form, you are also given the option of selecting the associated Master page in the form of a check box in the lower portion of the dialog box. If you select this option you are prompted to select an existing Master page to associate with the new Web form. A *Master page* is a common template that can be applied to multiple pages in a Web site for a consistent layout. This is convenient because it creates the new Web form with all the code needed to leverage the Master page.

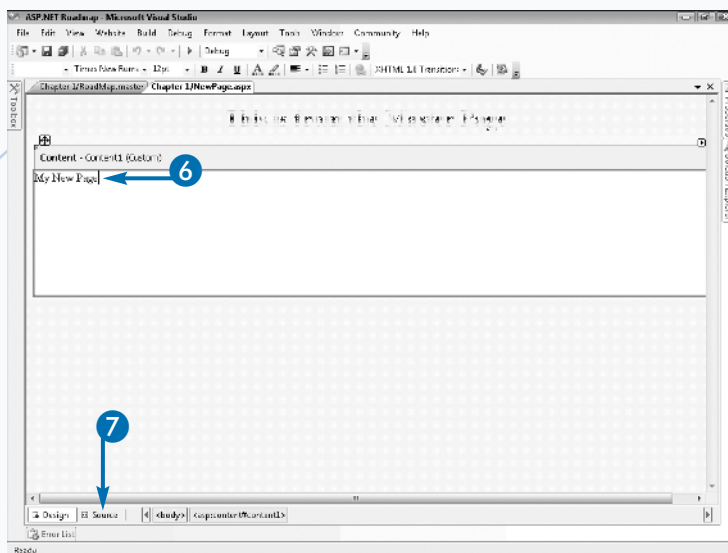
## Add a Web Page to a Web Site

- 1 Right-click the root of the Web site in the Solution window.
- 2 Click Add New Item from the context menu.
- 3 Select Web Form in the template list.
- 4 Type a name for your new file.
- 5 Click the Add button.

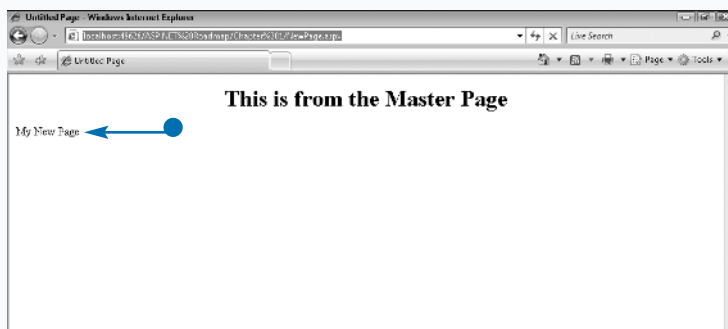


**Note:** Keep the Place code in a separate file option checked. You should also leave the Select Master Page option checked for this example.

- 6 Type some text to indicate the page you added.
- 7 Click the Source tab at the bottom to view the page's markup.



- 8 Open the page in a browser by right-clicking and selecting View in Browser.
- The new page appears showing the text you added in design mode.



## Extra

There is a long list of potential files you can add to most folders in a Web site from the Add New Item dialog box. The dialog box is smart enough to filter this list to only allow file types that are supported in the folder. Some folders are marked with special attributes by Visual Studio, such as the `App_Code` and `App_Data` folders. There is a smaller subset of potential files for these folders; for example, `App_Data` only allows you to add SQL databases, XML files, Reports, Text files, and Class Diagrams.

The Add New Item dialog box offers two different views of the file types — Large Icons and Small Icons. The only differences are the icon size, large and small, and where the page type is displayed, below or to the right. This can be changed by selecting the corresponding icon in the top right corner of the dialog.

If you have installed any custom file templates in Visual Studio they are listed at the bottom of the Templates window in the My Templates section. You can learn more about these in the Export Template section.



# Edit a Web Page

Editing a Web page in Visual Studio 2005 involves two distinct paths — page design and program code, typically in the code-behind. By default, Visual Studio 2005 creates a Web page as two distinct files, the .aspx or design page and the .vb or .cs file (.vb is for a VB.NET file and .cs corresponds to C#). The .aspx file contains the HTML markup that defines how the page will be rendered to the browser. It contains standard HTML markup tags and attributes as well as the tags that define Web controls, such as a `TextBox`.

The Web page itself can be edited in either design or source views. The design view is a WYSIWYG visual format that allows the developer the ability to drag and drop elements and visually edit a page. You can change to source view to edit the page in the HTML markup. Each has distinct advantages. In the source view, for

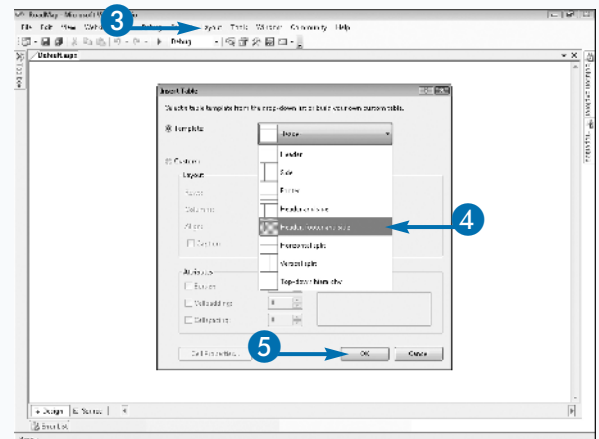
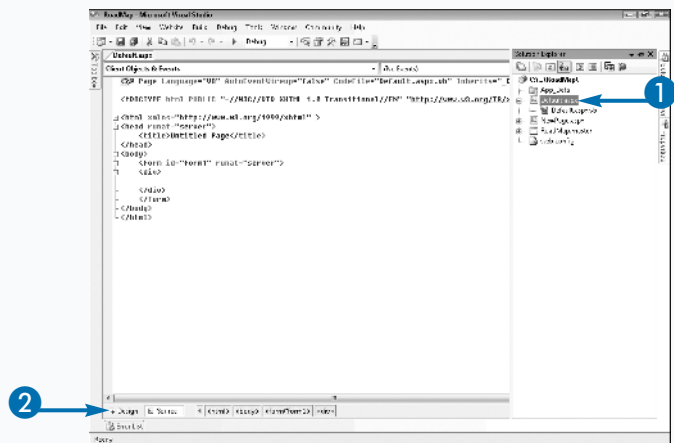
example, most developers find it easier to manage finite details of a page. Design view allows the developer to see real time how a page may layout.

The code file is used to enter application logic to process data and inputs. It contains a class for the page and generally matches the name of the page with either a .vb or .cs extension. Each ASP.NET page has a life cycle marked by a series of events. These events can have handlers added to the code file that is executed when that event is raised. For more information about these events, see Chapter 2.

Methods, properties, and variables (known as class members) can be added to the page's class. These members can be called from event handlers to branch into the logical flow of the application to render the page or process user input as needed.

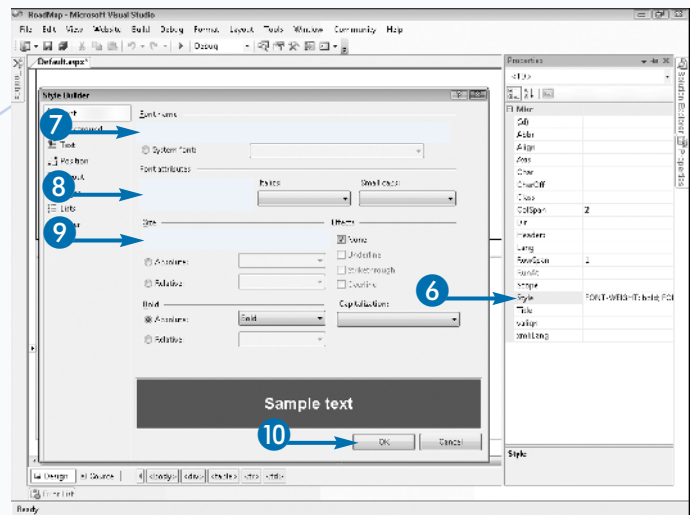
## Edit a Web Page

- 1 Open an existing Web page.
- 2 Click the Design View tab to change to Design mode.
- 3 Click Layout → Insert Table to open the Insert Table dialog box.
- 4 Click the Template drop-down arrow and select Header, footer, and side.
- 5 Click OK.



A preformatted table inserts into the document.

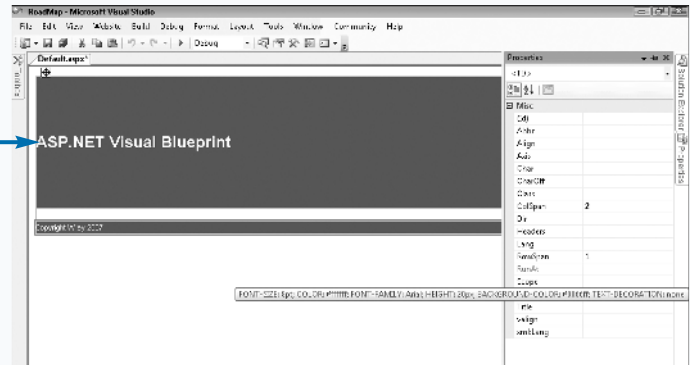
- 6 Select a cell (TD tag) in the table and then click the Ellipse button to open the Style Builder dialog box from the Properties window.
- 7 Set the Font Family to Arial.
- 8 Set the Font Color to White.
- 9 Set the Font Size to 24 px.
- 10 Click OK to store the settings.



- 11 Type **ASP.NET Visual Blueprint** in the header cell.

You can now open the page in a browser to see the results.

11 → ASP.NET Visual Blueprint



## Extra

Most ASP.NET Web controls as well as Web user controls will have a SmartTag located in the top-right corner of the control in design view. The tag is a small square with a little arrow. Each control can have its own SmartTag functionality, but all are designed to give the programmer easier access to common programming or formatting functionality.

The SmartTag for a control will be a list of links or drop-down lists that have selections for common properties. Clicking one of the links typically displays a configuration dialog box for a particular property, such as setting the `datasource` for data control or a predefined style for many template controls. You can almost always access these same properties through the Properties window, in design view as attributes, or at runtime, but often the SmartTag interface optimizes the management of these properties.



# Explore the Visual Studio 2005 Windows

**V**isual Studio 2005 was designed with productivity in mind. The IDE utilizes a series of content windows that can be moved and pinned to suit the developer's style while providing easy access to the tools and information needed to efficiently work on projects.

The main window used to edit files is called the document window. This window takes on slightly different views based on the file type being edited. A Web page has both a Design view and Source view. The Document window also includes a Tag navigator at the bottom of the window when editing a Web page, giving quick access to the underlying HTML tags. A code file has one view, but it has a pair of drop-down lists at the top to help the developer access and create event handlers for the page, controls, or base classes.

Other windows support the development process by providing quick access to information and project

management to the developer. The Solution Explorer window contains a tree of projects, files, and folders that comprise an overall solution. Pages, files, and other project-related resources can be added and interacted with through simple mouse clicks. To open a file from the Solution Explorer, double-click the file.

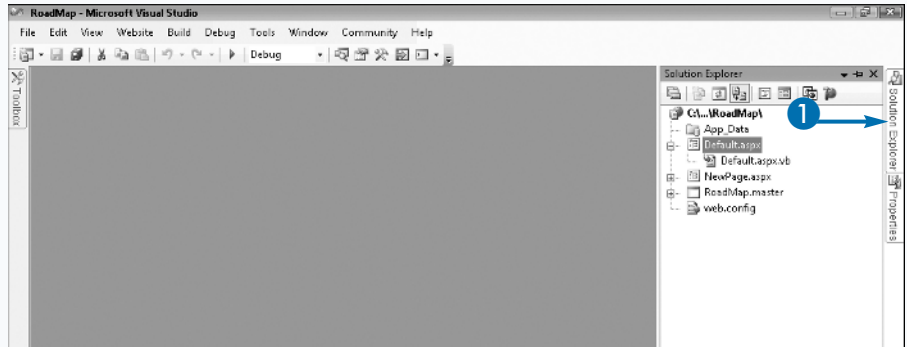
The Properties window lists properties available to the developer from a control or HTML tag. Properties can be set inside the window and any custom property dialog box can be displayed. Some examples of a custom property dialog box are a color picker, style sheet builder, or items manager.

At the bottom of Visual Studio is a series of windows. These windows can vary, but often include Error List and Task List. These windows display a list of current solution errors and tasks defined by the developer. Other windows can be opened from the View menu and positioned at the left, right, or bottom of the IDE.

## Explore the Visual Studio 2005 Windows

- 1 With a Web site loaded, move your mouse over the Solution Explorer tab to make it visible.

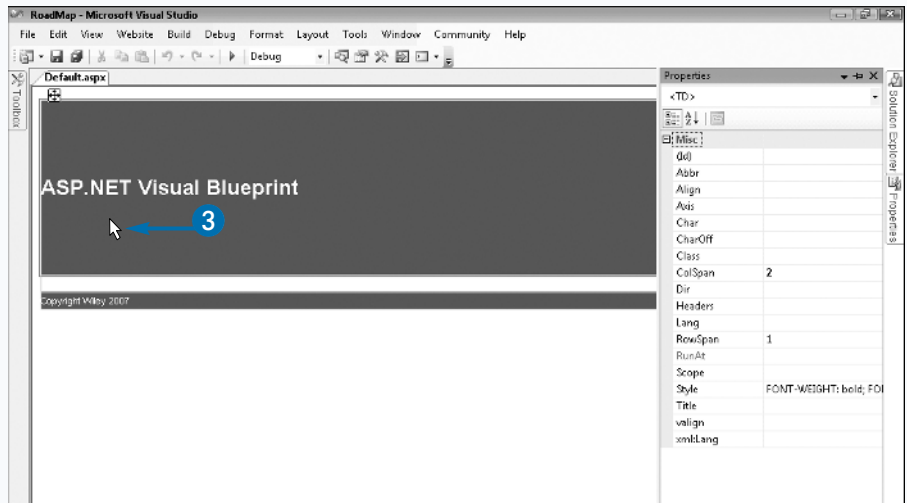
**Note:** The Solution Explorer window may be pinned open.



- 2 Open an existing Web page.
- 3 Place the mouse cursor in a table cell.

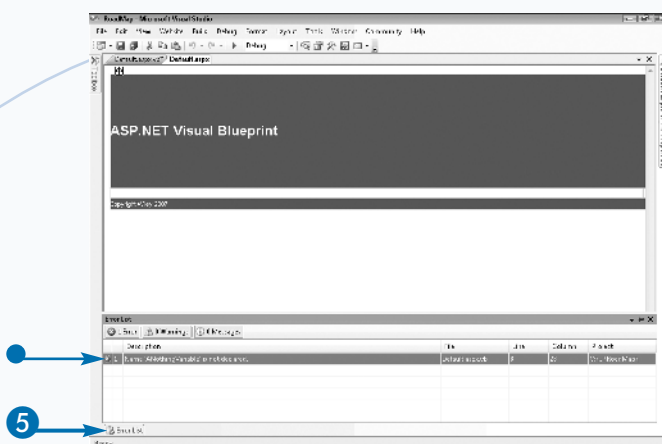
**Note:** This can vary based on the page you open. The properties for the selected HTML element or control will be displayed.

- 4 Press F4 to see the Properties window for the page.



5 Position the mouse over the Error List tab.

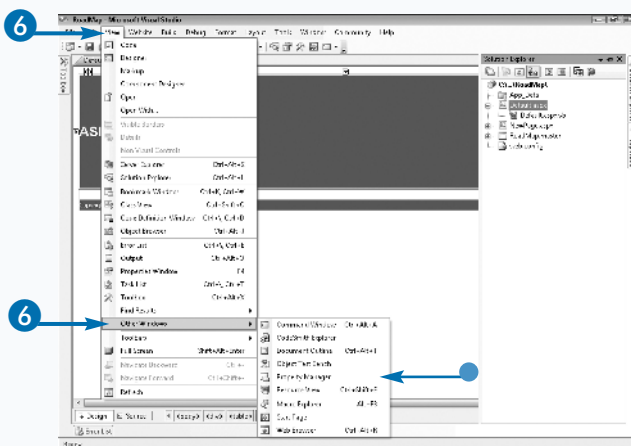
- A list of errors currently acknowledged by Visual Studio appears.



6 Click View → Other Windows.

- A list of available windows in Visual Studio appears.

**Note:** The list of available windows on your desktop will depend on the third-party add-ins you have installed.



## Extra

Visual Studio windows can be closed, moved, and pinned as needed. In the upper-right corner of each window, except the Document window, there is a down arrow, a push pin, and an X. The down arrow opens a menu with docking and hiding options for the window. The push pin allows you to either pin or keep the window open or allow it to collapse to the edge of the IDE. If a window is pinned, the Document window resizes to accommodate; otherwise the Document window stays beneath the unpinned window. Clicking the X closes the window. If you close the window you can still open it if you need it later, typically through the View menu or a keyboard shortcut.

While Visual Studio comes with a strong set of tools at the disposal of the developer, there are many third-party add-in tools that can be integrated into the development environment. Some of the common tools are source control, code generation, testing, and utility packages. Vendors for these utilities and tools often integrate an interface directly in Visual Studio to make it more convenient to work with their tool.

# Add a Web Control to a Web Page

**A** Web control is an object that is typically used to collect information or display information to a user. It can also be used to manage extra attributes of controls and control groups. Web controls are analogous to Controls in a Smart Client application because they provide the programmer an object that can be programmed both on the Web form and in the application code.

Generally, a Web control inherits a series of base properties from the `WebControl` class that give you the ability to adjust various display settings, such as font color, bold, and background color. Some controls, such as the `Placeholder` control, have these properties suppressed. Other controls are composite controls, meaning they are composed of several controls but appear programmatically as one control.

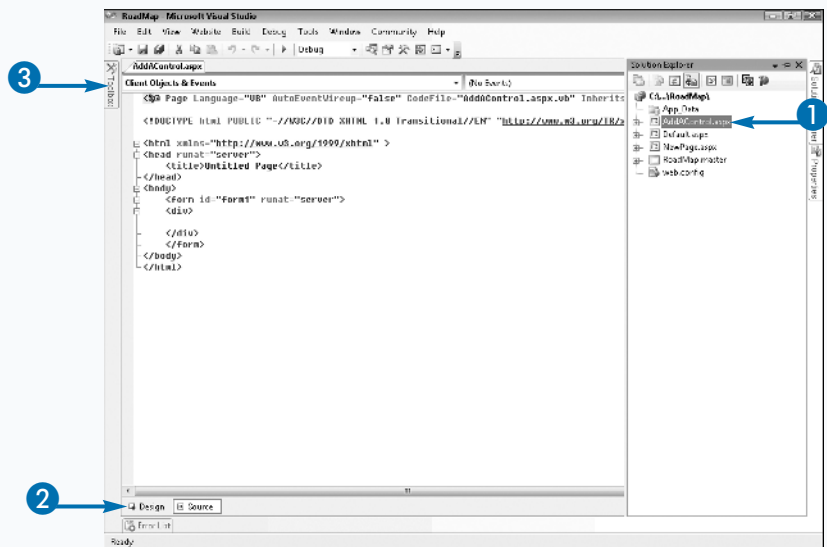
Properties can be set either declaratively (in design mode) or at runtime (in the application code). Web controls all have an ID property that is by default created as the name of the control type and the current number of the controls without unique names on the page. For example `TextBox1` is the initial ID of the first `TextBox` placed on a page. You can change the name of any control to make it represent what it does; for example, `txtFirstName`. All controls should also have the `runat=server` set in the control definition.

```
<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
```

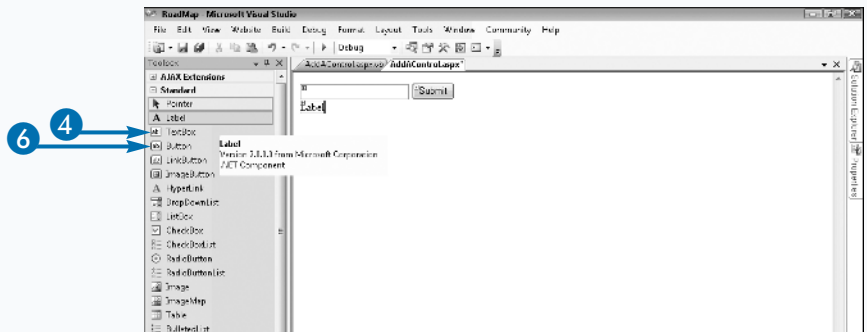
You can quickly add controls to a page by selecting a control in the Visual Studio ToolBox and either dragging it to the location on the page you want the control displayed or double-clicking the item. If you double-click, it is added at the current location of the cursor in the page.

## Add a Web Control to a Web Page

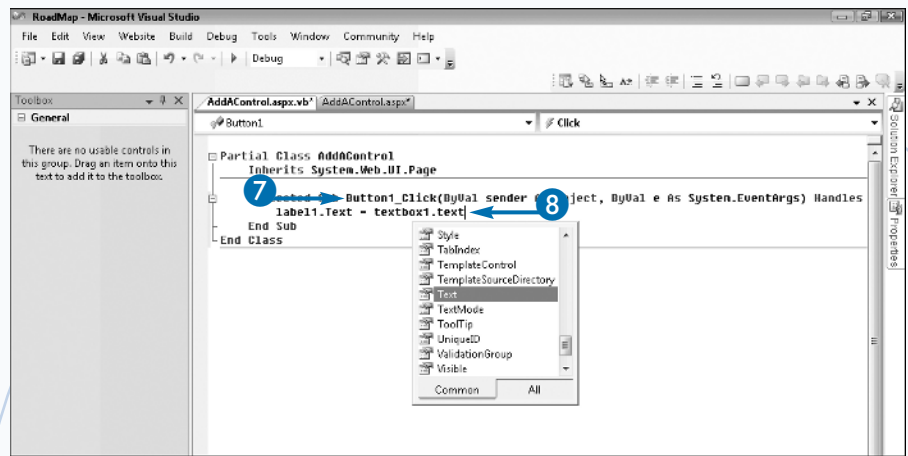
- 1 Create a new Web page.
- 2 Click the Design tab to switch to Design mode.
- 3 Open the ToolBox window in Visual Studio.



- 4 Double-click `TextBox` in the Standard tab list.
- 5 Press the Spacebar to add separation from the `TextBox`.
- 6 Position your mouse over the `Button` control in the Toolbox and double-click the `Button` control to add it to your page.



- 7 Double-click the Button control to add a click event handler.
- 8 Type code to echo the value entered in the TextBox to the Label control.



- 9 Open the page in a browser by right-clicking and selecting View in Browser.
- 10 Type some text in the TextBox control.
- 11 Click Submit.
  - You should now see the value entered in the TextBox echoed in the Label control.



## Extra

You can also create your own custom Web controls. The ASP.NET framework allows you to create your own controls from scratch or derive your controls from the `CompositeControl` class. If you create a control from scratch, you should inherit the class from `WebControl` to gain access to many core properties. Consider creating your own Web controls when you consistently reuse a set of markup and logic repeatedly in your Web sites.

You can also create a user control, which is much like a Web control, but is more like a small Web page. These are built in `.ascx` files that enable you to quickly build a set of reusable code in the same way you would build a Web page. User controls can be dragged onto the surface of a Web page just like a Web control. They will actually be a file in the Web solution and are not capable of being compiled into a reusable library.

# Add a Reference to a Class Library

**M**any applications, including Web sites, rely on third-party class libraries as well as internal business libraries. These libraries are typically packaged as dynamic link libraries, or DLLs. Class libraries may contain a series of related classes, .NET controls, or resources.

You can add a reference to these libraries to your Web site by right-clicking the root node of the Web site in Solution Explorer to display the context menu. Choose Add Reference from the menu. In the Add Reference dialog box that appears, you will see a series of tabs — .NET, COM, Projects, Browse, and Recent.

The .NET tab contains a list of .NET DLLs that have been registered in the global assembly cache (GAC). The COM tab contains a list of COM objects registered on the machine. The Projects tab contains a list of the projects in

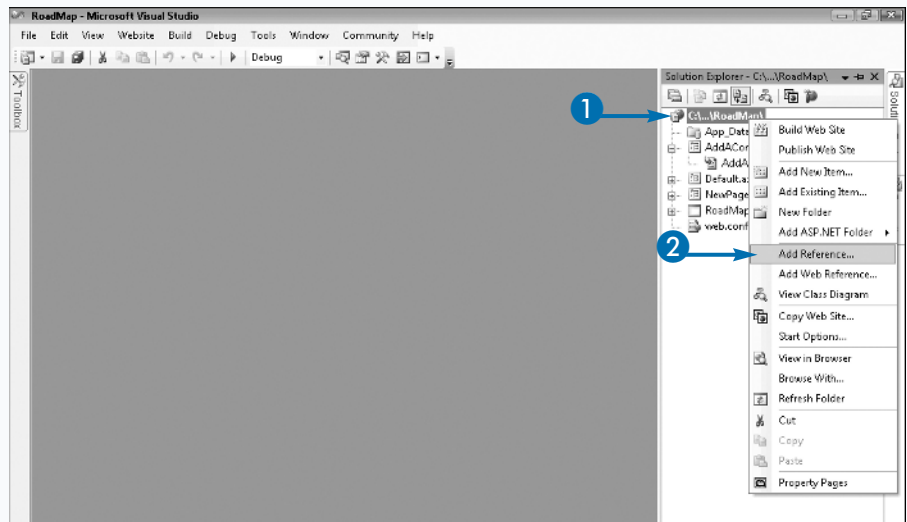
the current solution along with the Web site. The Browse tab resembles a File Open dialog box and lets you search the local computer and any network shares for the library you want to add. The Recent tab lists all the libraries that have been recently added to the project.

You can select any of the libraries from any of the tabs. In the case of tabs with a list, you can select multiple libraries to add to the Web site. For example, you can Ctrl+click items in the list box before clicking OK to add the references to your project.

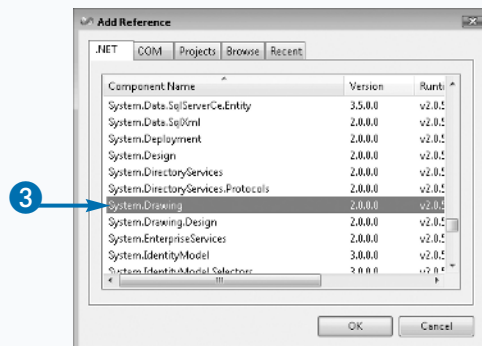
A Visual Studio Solution can contain more than one project. For example, an enterprise application may contain a series of business logic libraries, a custom control library, and a Web site project. Each of these projects can reference each other as needed. Utilizing a library is a great way to package reusable code to reuse on multiple projects.

## Add a Reference to a Class Library

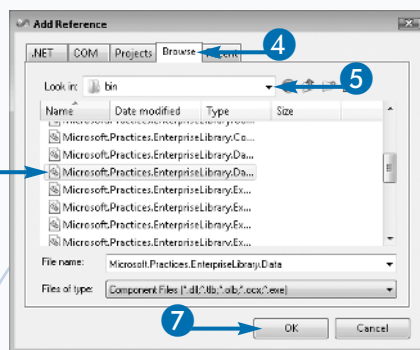
- 1 Right-click the root of the Web site in Solution Explorer.
- 2 Click Add Reference.



- 3 Scroll through the list of class libraries registered in the .NET tab to the System.Drawing component.

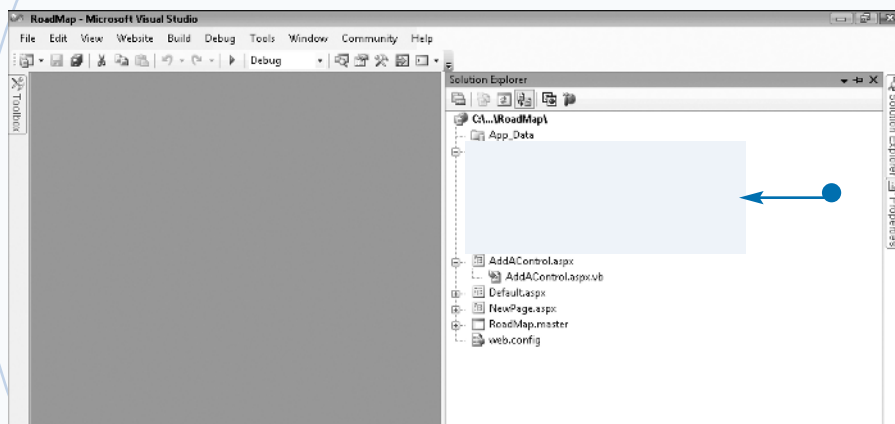


- 4 Click the Browse tab.
- 5 Browse your local computer to find an available component, a .dll, .exe, .ocx, or .tlb.
- 6 Select the component in the list.
- 7 Click OK to add the reference to the Web site.



**Note:** If you are not aware of the location of a local component, you can add a reference to one of the components in the .NET tab.

- The component you selected appears in the bin folder of the Web site. In this example, components were added on which the selected component is dependent upon.



## Apply It

The web.config file contains a pages section that contains a namespaces section. Each namespace can be added to the namespaces section to make it available by default on each code page in the site. If you do not need a global reference to a namespace you can add it to your code page by adding an Imports (VB) or Using (C#) line to the header of the file.

### Namespaces:

```
<pages>
  <namespaces>
    <clear/>
    <add namespace="System" />
    <add namespace="System.Drawing" />
    <add namespace="System.Data" />
  </namespaces>
</pages>
```

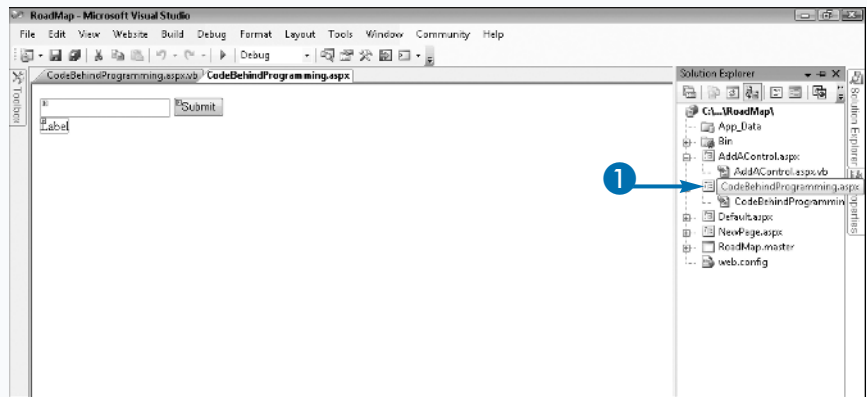
# Program a Page Using Code-Behind

Classic ASP programming uses an inline programming model, where any sort of programming logic is performed between a set of delimiters such as `<% [Programming Here] %>`. Several modern languages still utilize this style of programming: PHP, for example. ASP.NET offers a newer method of programming a Web application known as the code-behind model. This model separates design code from application logic, making it easier to debug and maintain application code. This allows the design portion of the site to be easier to maintain as well as define.

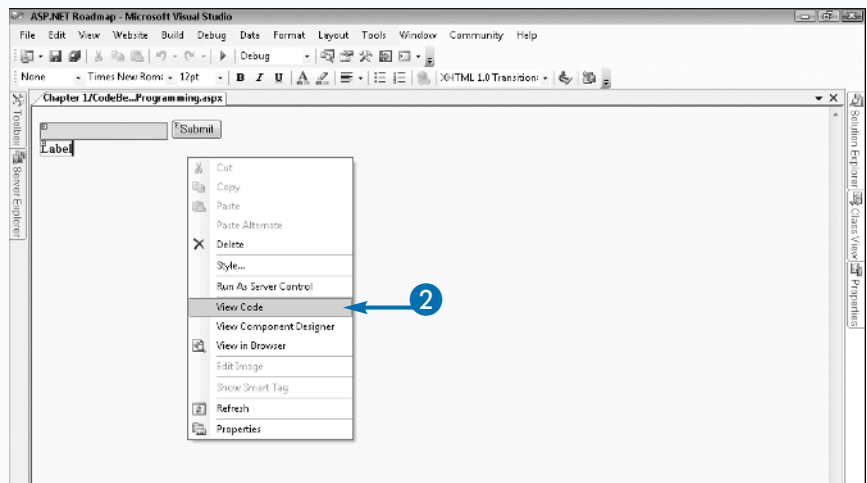
To help programmers transition from Classic ASP, ASP.NET supports the old inline method of coding. An intermediate model is also supported where you can define methods in the page header that are executed on the server. Again this model has limitations that make the code harder to maintain and scale as it grows.

## Program a Page Using Code-Behind

- 1 Open an existing Web page by double-clicking it in Solution Explorer.

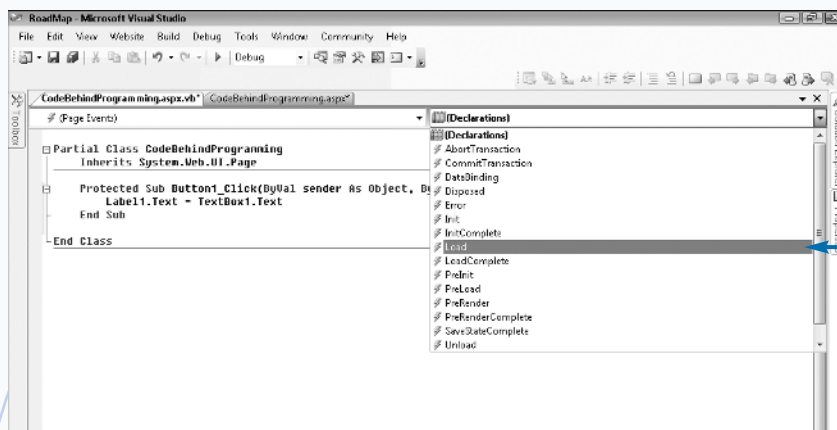


- 2 Right-click anywhere on the page and choose View Code from the context menu.





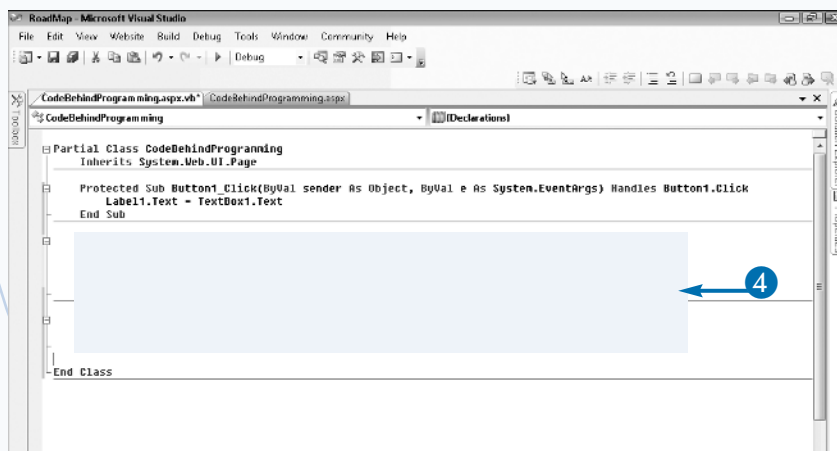
- 3 Add a Handler for the Page Load event.



- 4 Add some code to set the initial value in the TextBox control.

**Note:** In this example, a page with the Add A Control to a Web Page section is selected.

You can now open the page in a browser to see the results.



## Extra

Most applications leverage the Page Load event handler to begin processing local logic, but you are not limited to this event as a place to begin processing. You can create handlers for any of the other events in the ASP.NET page life cycle. Other important events are PreRender, Init, and Error.

Always check if the page is being posted back to the server through the IsPostBack property. A post back occurs when an input control such as a Button is displayed, causing the form on the page to post back to the server. Not every piece of code should be executed each time a page is loaded. Data binding, for example, should only be done the first time a page is loaded or is explicitly invoked after a change in the dataset.

You can create any code needed in the page's class file, including functions that return values, sub routines, properties, and enums. Instances of controls on a page are created in the page's Init event handler. If a control is created at runtime you will need to create a new instance of this control each time a page is processed.

# Open a Page in a Browser

**D**eveloping a Web site involves checking the pages in a browser to verify they work and render correctly. You can quickly execute the site by pressing F5 (debug mode) or Ctrl+F5 (without debugging). This opens the default or start page for the site, which is typically the default.aspx page added to the site when it is first created. If you do not have a default page set for the Web site, you will be prompted to do so before executing the site.

Any page in the site can be set as the default page to open. There are two ways to define the start page for a Web site project. The easiest way to open an individual page is to right-click the page in the Solution window, then select Set as Start page from the context menu.

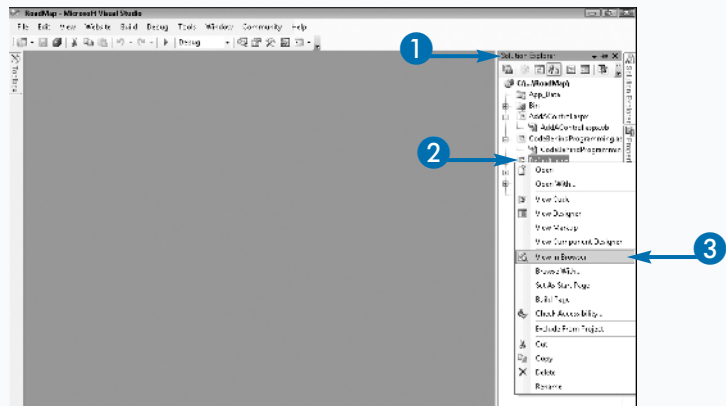
Alternatively, you can also open the properties for the Web site and open the Start Option section of the Properties dialog box. In the Start action section, you can choose to use the current page, a specific page, start an external program, a specific URL, or not to open a page at all. This gives you great flexibility in the start-up debug options for an ASP.NET 2.0 Web site.

If you are working the page file (.aspx), you can also right-click anywhere in the code or design surface to open that page in a browser to begin validating how it is rendered and functions.

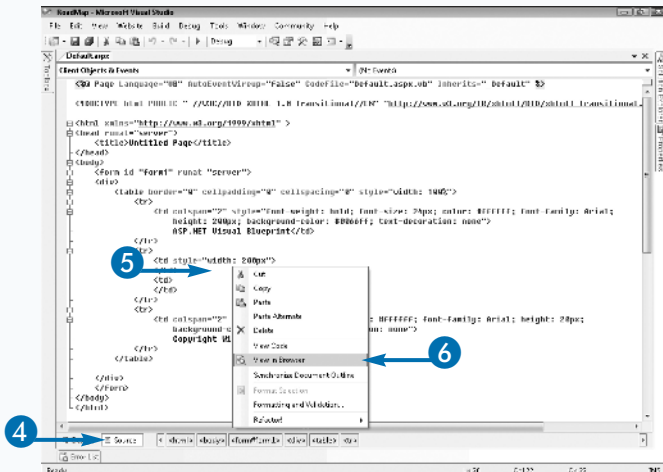
You are not limited to using the built-in Personal Web Server; you can also configure your application to run in a Virtual Web under a local instance of IIS or even on a remote machine running IIS.

## Open a Page in a Browser

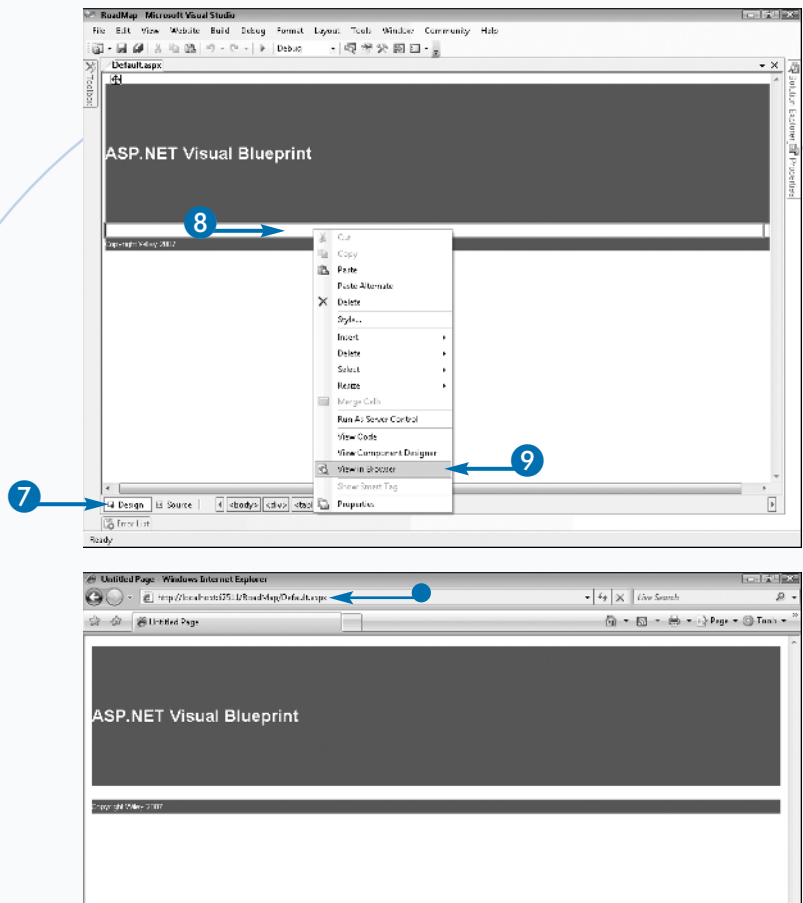
- 1 Open the Solution Explorer window.
- 2 Right-click any page in the solution.
- 3 Click View in Browser.



- 4 Open a page in Source view.
- 5 Right-click anywhere on the page.
- 6 Click View in Browser.



- 7 Click the Design tab to switch to Design mode.
- 8 Right-click anywhere on the page.
- 9 Click View in Browser.



- The page opens in the browser.

## Extra

Traditionally, a Web developer's workstation will have a local version of the Web server installed. For classic ASP and ASP.NET developers, this is traditionally IIS. But as corporate security and support policies have evolved, developers are finding it harder to have a local copy of Web servers running on their development workstations because of the security exposure it offers. Visual Studio 2005 ships with the Personal Web Server, which is a small instance of an IIS Web server that is only invoked while the developer actively runs the Web site from Visual Studio. The nice thing about the Personal Web Server is it runs on any current edition of Windows XP, Windows Server, and Vista.

The Personal Web Server does not run over the traditional Web service HTTP port 80. Instead, it randomly selects a port number to run in the valid range of ports. Selecting a random port number makes it more difficult for a potential hacker to attack the local Web server instance. Personal Web Server, previously known as Casini, is available for free download at [www.asp.net](http://www.asp.net).

# Use Code Snippets to Quickly Add Common Code

**R**apidly developing applications is one of the most compelling attributes of the ASP.NET and Visual Studio 2005 experience. Visual Studio 2005 provides a new feature, Code Snippets, which make it very easy for a developer to add predefined code to an application. A Code Snippet is a little piece of code that performs a specific task or is a repeatable pattern, such as an `If Then` statement.

There are hundreds of Code Snippets that ship with Visual Studio and vary by language, project, and file type. The IDE is intelligent enough to offer only applicable snippets based on the code file's use. To add a Code Snippet, simply right-click in the code-behind file to display the context menu and click Insert Snippet.

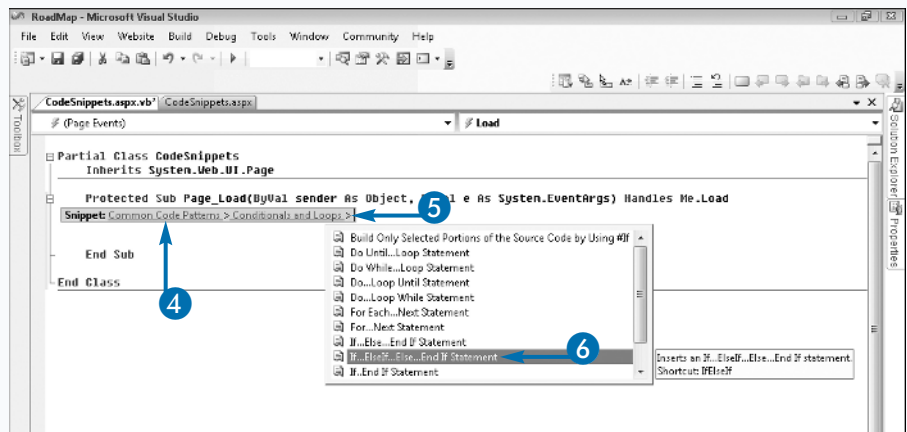
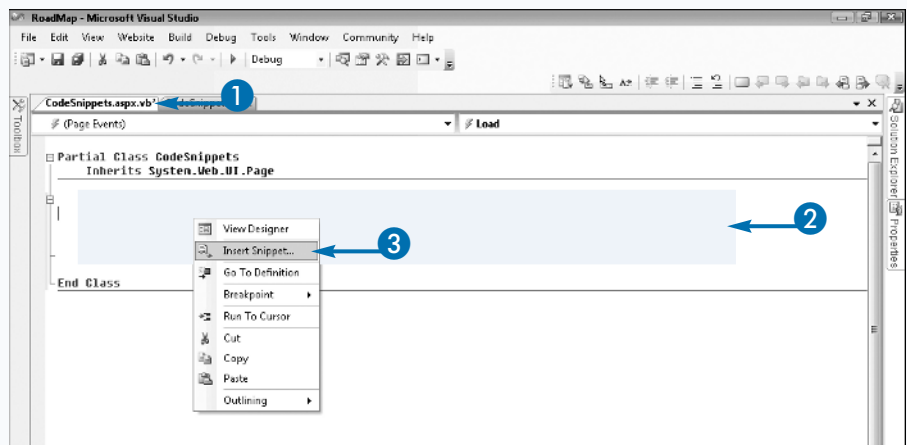
A drop-down list appears right where your cursor is positioned with a top-level list of snippet categories such

as networking, system, data, File IO, math, and common structures. Each time you select an item in the list, another drop-down list with subcategories is displayed or the code is placed in the file. After the code is in the file you can clean it up to fit your application. There may also be some sections that require you to type values or variables in order for the snippet to function. These are highlighted in green by default. For example, an `If Then` statement needs a conditional statement to evaluate.

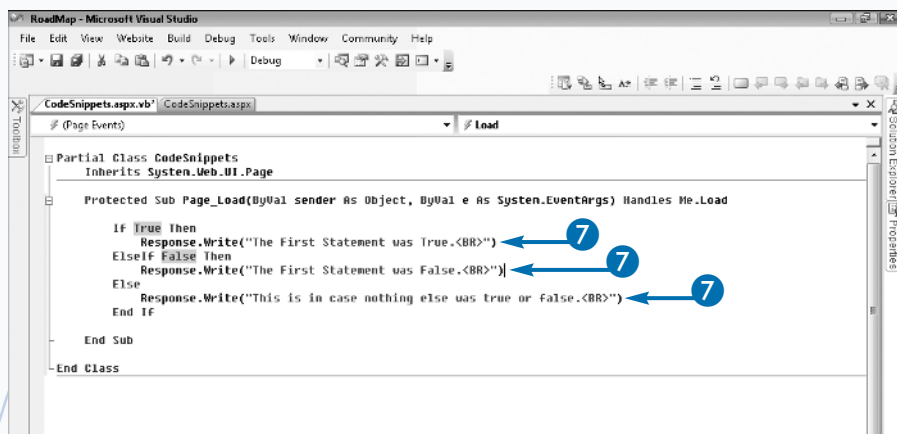
Each snippet also has a shortcut to add it to a file. The shortcut for a snippet can be seen in the Tooltip window when the snippet is selected in the drop-down list. The shortcut is a value that can be typed in a file followed by pressing `Tab` to add the section. For example, `If Then` has a shortcut of `If+Tab`. You can improve your programming efficiency by learning these shortcuts.

## Use Code Snippets to Quickly Add Common Code

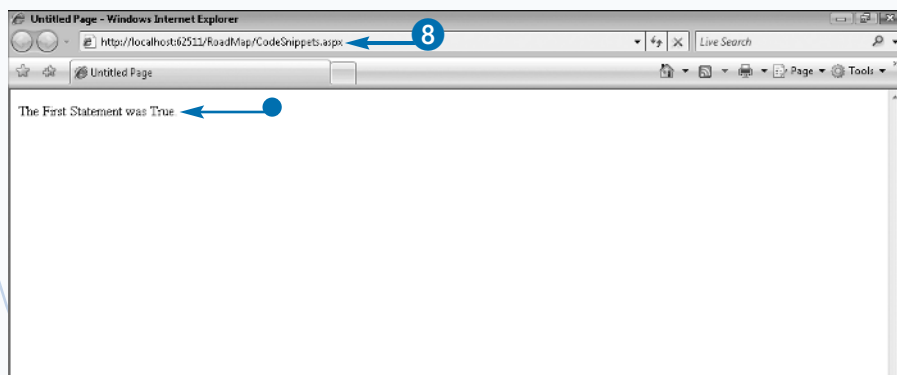
- 1 Create a new Web page and then open the code file.
- 2 Create a `Load` event handler for the page.
- 3 Right-click inside the `Load` event handler and click Insert Snippet.
- 4 Select Common Code Patterns from the snippet drop-down list.
- 5 Select Conditionals and Loops from the next drop-down list.
- 6 Select `If...ElseIf...Else..EndIf` Statement from the final drop-down list.



- 7 Add Response.  
.Write calls to  
indicate the different  
cases of the If  
conditional statement.



- 8 Open the file in a  
browser to see the text  
echoed from the If  
conditional statement.



- In this example, you should see the first string because you did not set any values in the conditional section and used the default of True.

## Extra

A Code Snippet is an XML text file with the .snippet extension that follows the Code Snippet XML schema. A snippet can display an input dialog box or message box, add import (VB) or using (C#) statements to a code file, and highlight areas that require a user to customize.

You can manage your code Snippet library with the Code Snippet Manager. The Code Snippet Manager can be opened from the Tools menu of Visual Studio or by pressing Ctrl+K, Ctrl+B. It allows you to add, remove, and import Code Snippets as well as navigate through all the installed snippets.

The syntax that defines a Code Snippet allows you to define a name, author, and description for the snippet. You can also add any namespace imports needed for the code. If the namespace is already included in the file, this is ignored. A snippet can also have variables or declarations that need to be customized by the developer.

# Export Projects and Files

Visual Studio comes with several prebuilt projects, including Web, WinForm, Class Library, console applications, and so on. It has been possible for developers to create similar templates in previous versions of Visual Studio, but it was too difficult a task for the average developer or team to accomplish. Visual Studio 2005 includes a useful Export feature to assist a developer create a core or skeleton project with all the files, pages, and resources needed to provision a vertical-oriented site. The same Export feature can also create individual file templates that may be reused on future projects.

For example, a team that specializes in e-commerce solutions might create a core solution or set of customized solutions it can use to quickly get a project under way. The main advantage of a custom project template is that it includes all base pages, classes, and

resources consistently needed for all similar projects. You can think of it as a base class library to inherit your next set of projects from, saving hours of common programming.

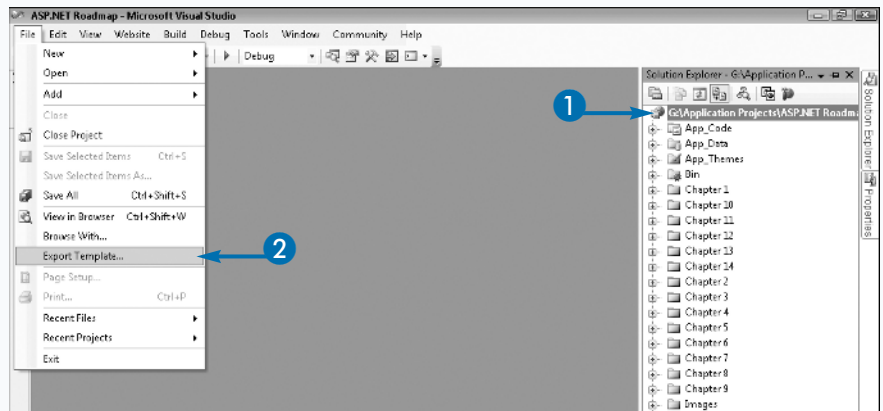
Exporting a project is fairly straightforward. First, create a new Web site or class library project, or another base project you may want to customize. After you add all the core elements, such as pages, classes, and style sheets, you can export it intact. The Export Wizard can be launched from the File menu.

You are not limited to just exporting projects and Web sites; you can also export files. For example, most Web sites have a contact form. You can create a standard contact form and export it. Each time you create a new Web site, you can just add a new prebuilt contact form to the site.

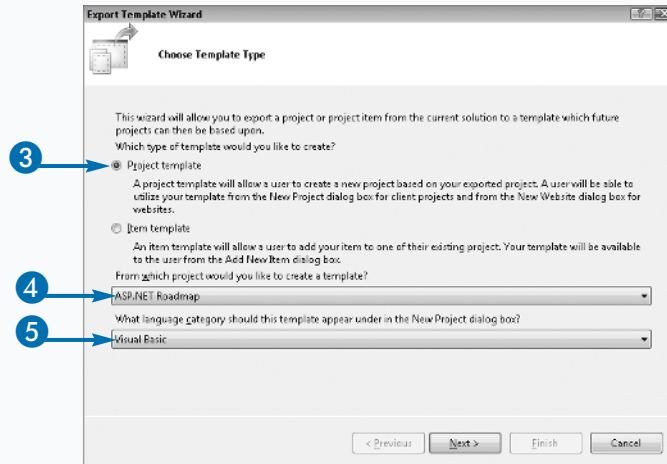
## Export Projects and Files

- 1 Open a Web site in Visual Studio.
- 2 Click File → Export Template.

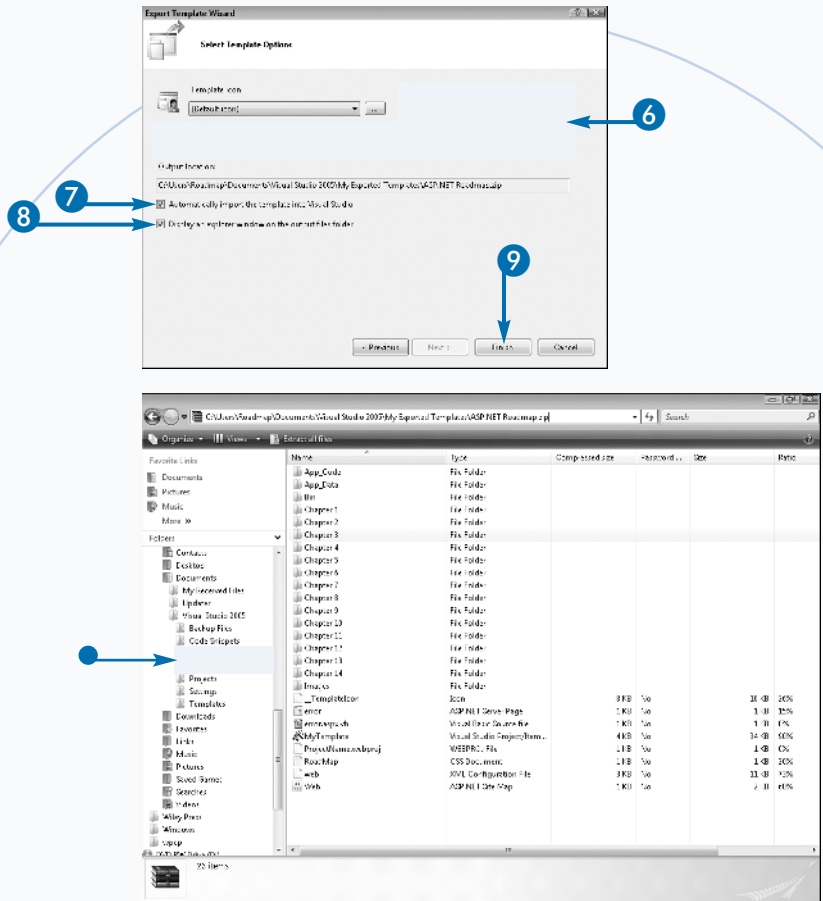
**Note:** You may need to add this to your menu by clicking *Tools → Customize*.



- 3 Click the Project template option.
- 4 Click here and select the Web project you want to export.
- 5 Click here and select the language for which the Web site is developed.



- 6 Change any of the Template option values you need to customize.
  - 7 Select the Automatically import the template into Visual Studio option.
  - 8 Select the Display an explorer window on the output files folder option.
  - 9 Click Finish to export the Web site.
- Explorer opens to the Zip file that was exported. Depending on your system, the contents of the Zip file may display.



## Extra

Visual Studio 2005 includes the ability to export and import settings for Visual Studio. This can be very helpful to a developer who works on a variety of projects. In this case, certain windows and layouts of the IDE can be more productive for each type of project.

Environment settings may be imported and exported from the Tools menu. Open the Tools menu and choose Import and Export Settings. The Import and Export Settings Wizard opens, and you have a choice to Export, Import, or Reset all Settings. The two screens allow you to select the settings to export and the name and location of the settings file. Similarly, the Import Settings will prompt you to indicate if you want to save the current settings and which set of settings you want to import. Resetting the settings returns Visual Studio to the initial state after install.



# Publish a Web Site

**P**ublishing a Web site to a production or development and quality assurance environments from a developer workstation can be done in several ways: XCopy deployment, the Copy Web Site option, Precompiling, or by building an Installer program.

XCopy deployment is simply copying all the files from one location to the other through a standard file copy operation. XCOPY is a command-line tool that can be used to copy entire directory or drive structures from one location to another. The term XCOPY deployment simply refers to the process of copying the Web application files from one location to another. This could also include a simple FTP operation to the server.

The Copy Web Site option refers to a built-in GUI inside of Visual Studio 2005 that helps manage the copying and synchronization of files from the development machine to another location. This GUI can be displayed by clicking

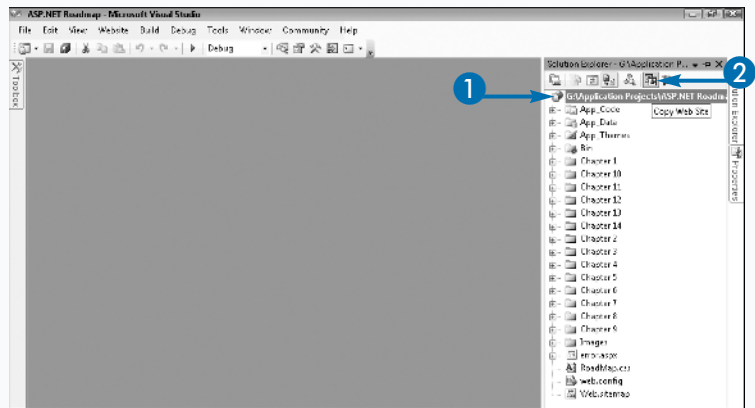
on the Copy Web Site icon on Solution Explorer or by choosing Website and then Copy Web Site. This GUI allows you to select the destination location. There are several destination options: File System, Local IIS, FTP Site, and Remote Site. After you connect to the destination you will see a file and directory structure for both the source and destination.

Synchronizing between the two locations can be done by selecting the files and directories you want to synchronize with the destination, then using the arrow buttons in the center of the window to perform the synchronization process. The first two buttons copy a file from the workstation to the destination, or the vice versa. The next option synchronizes between the two locations, selecting the most current version for copying to the other location. The remaining button allows you to terminate or cancel any operation while it is running.

## Publish a Web Site

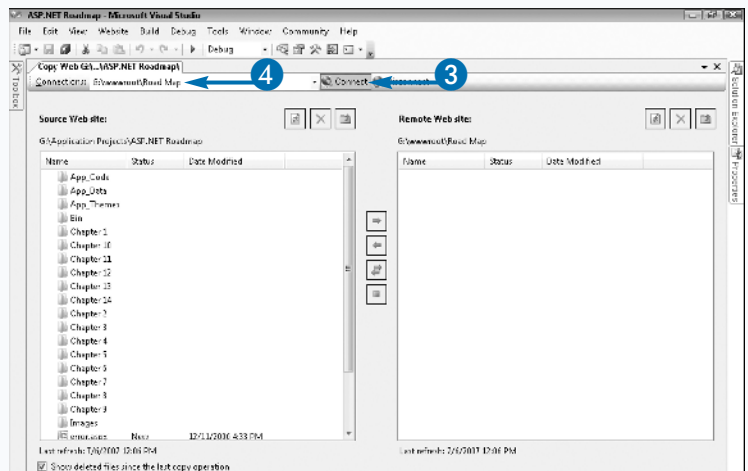
- 1 Open a Web site in Visual Studio.
- 2 Click the Copy Web Site icon at the top of the Solution Explorer window.

**Note:** You must have a node in the Web site selected for this icon to be available.



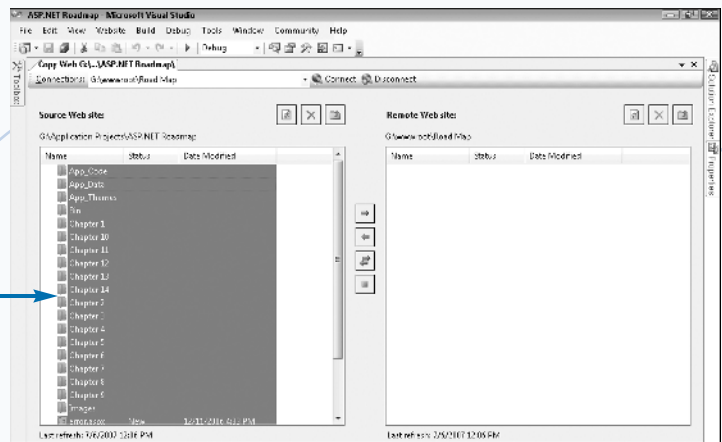
- 3 Click Connect at the top of the Copy Web Site window.
- 4 Select the destination Folder.

**Note:** There are four possible destination types, each of which requires a set of connection values.



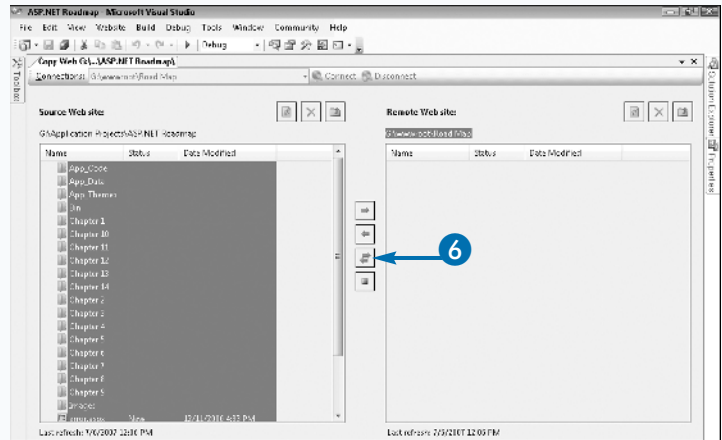
- 5 Select a group of files in the Web site to synchronize with the destination.

5



- 6 Click the Synchronize button to copy the selections to the destination location.

6



## Extra

The Copy Web Site utility allows you to select from four destination types: File System, Local IIS, FTP Site, and Remote Site. Each destination type has a set of information required to make a connection.

File System is easiest because you just need to indicate the path to the destination folder on the local machine. This can also be a mapped drive. The Local IIS option provides a tree list of the local Web servers and their virtual webs. You can select any of these nodes to which you have write permission to synchronize your site.

The FTP Site option handles connecting over FTP protocol to a remote server. Note that you may need to type the destination folder because the Copy Web Site tool assumes the initial folder is the destination folder. The Remote Site option allows you to type a destination URL that is configured with FrontPage Server extensions.

# Precompile a Web Site

**B**y default, ASP.NET performs batch compiling on a Web application. This means that a site is compiled as it is used. The first time a page is requested it is compiled, which causes an initial performance hit. Every subsequent request for that page bypasses the compilation process and instead serves up the compiled page. If you make a change to the file it goes through the compilation process again.

ASP.NET 2.0 introduces precompilation, which gives you the ability to compile the Web site before it is deployed, thus bypassing the batch compilation penalty. There are two ways in which a Web site can be compiled: through the command line with the `aspnet_compiler.exe` utility or from the Build menu Publish Web Site option.

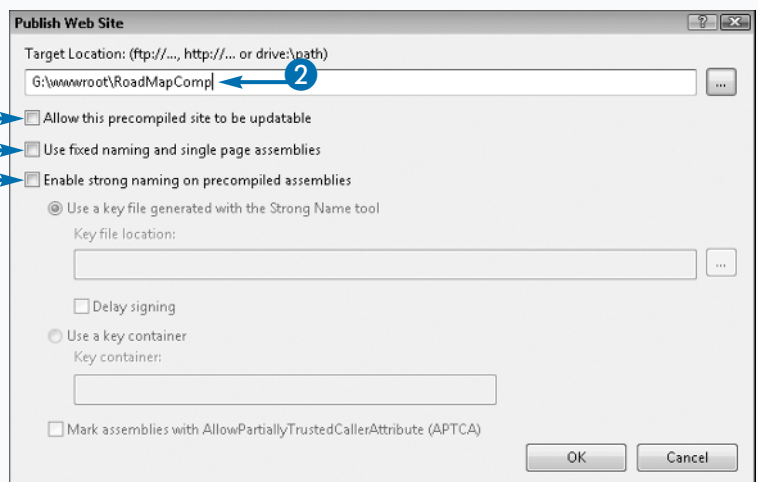
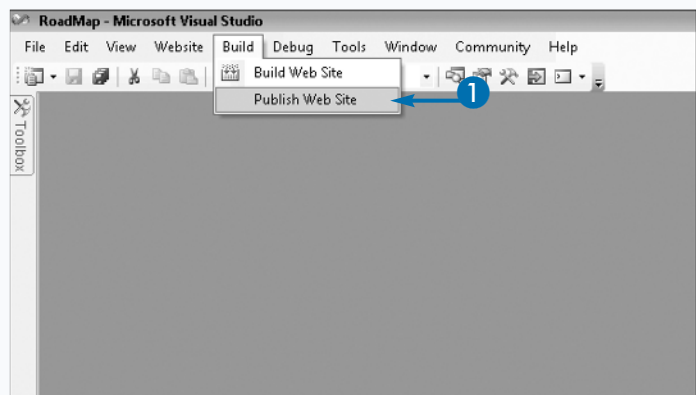
The `aspnet_compiler` accepts a few command-line switches to indicate the source Web site, either from the

IIS metabase or a physical path on the workstation and the destination folder. The Publish Web Site option displays a dialog box asking for the target location, and several compilation options. You can allow the compiled site to be updateable, use fixed naming and single page assemblies, or enable a strong name.

The output of the compilation will resemble the original site, but if you examine the compiled Web pages you will see they are merely a placeholder file to mark the original file. In the site's Bin folder are a series of randomly named DLL files that the application was compiled into. You can recognize the files because they will be named `App_Web_ABCXYZ.dll`, where ABCXYZ are random characters. Some files, such as XML, config, and HTML files, are not compiled.

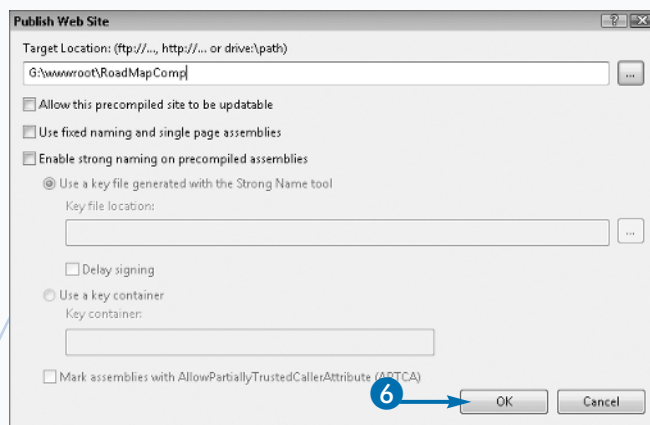
## Precompile a Web Site

- 1 Click Build → Publish Web Site.
- 2 Type the destination for the compiled Web site.
- 3 Deselect the Allow this precompiled site to be updateable option.
- 4 Deselect the Use fixed naming and single page assemblies option.
- 5 Deselect the Enable strong naming on precompiled assemblies option.



- 6 Click OK to compile the site.

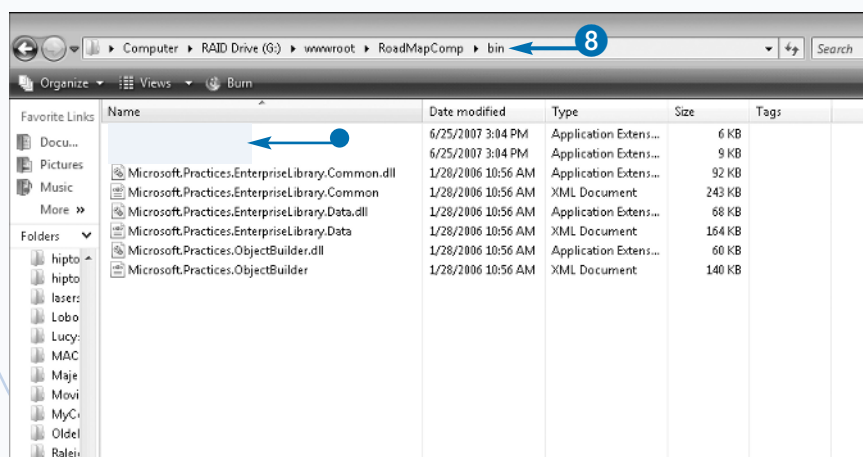
**Note:** You will not see anything happen, but if you examine the Output window you can see the result.



- 7 Open your Web browser.

- 8 Type in the destination's Bin folder.

- The compiled DLL files appear.



## Extra

The Publish Web Site utility allows you to precompile the site to a destination location. It also gives you the ability to control certain aspects of how your site is compiled. The first choice, Allow this precompiled site to be updateable, retains the original .aspx files to enable them to be changed at a later time. This means you can make small changes to the design of a page and replace the file on the destination at any time.

Deselecting the first option means the site must be recompiled to apply any changes. Each page is still represented on the site, but with a place-holder file. Each of these files contains the following statement: "This is a marker file generated by the precompilation tool, and should not be deleted!"

The Use fixed naming and single page assemblies option causes the compiler to create a DLL file for each page in the Web site. The third option is to use a Strong name on the assembly, signed with the key of your choice.