

# Visual Studio .NET Tips and Tricks

---

Minh T. Nguyen

## Preview

ISBN: 1-4116-1396-1

<http://www.lulu.com/content/78560>

[http://www.enderminh.com/minh/vsnet\\_tt.aspx](http://www.enderminh.com/minh/vsnet_tt.aspx)

# Contents

<b>Chapter 1: Editing Code</b>	<b>1</b>
Adding XML Commenting	2
Commenting Web Pages	3
Using IntelliSense Across Projects	3
Inserting Comment Tokens	4
Commenting Code Blocks	5
Creating Regions	6
Hiding the Current Selection	7
Selecting a Single Word	8
Selecting an Entire String Literal	8
Switching to the Front and End of a Selection	8
Placing Code into the Toolbox	8
Using the Clipboard Ring	9
Transposing a Single Character or Word	9
Cutting, Copying, Deleting, and Transposing a Single Line	10
Editing XML in Tabular Mode	10
Pasting Text as HTML	12
Adding Class Field Members	12
Formatting Entire Blocks	12
Toggling Word-Wrapping	14
Forcing IntelliSense for Field Members	15
Forcing IntelliSense for Parameter Information	15
Completing a Word	16
Implementing Methods of an Interface	16
Overriding Methods	18
Creating GUIDs	19
Creating Rectangular Selections	19
Switching Between Views	20
Going to a Line Number	20
Searching for a Word	21
Finding and Highlighting Matching Tokens	21
Performing an Incremental Search	21
Searching or Replacing with Regular Expressions or Wildcards	22
Performing a Global Search or Replace	23
Using Bookmarks	25
Going to the Definition of a Method	27
Using Browser-Like Navigation	27
Inserting External Text File	28
Inserting JavaScript Tags	28
Outlining HTML/Form Hierarchy	29

## Chapter 2: Exploring the IDE \_\_\_\_\_ 31

Opening with Default Action _____	32
Showing Extra Files _____	32
Setting Project Dependencies _____	33
Embedding Files As Resources _____	34
Changing the Font Size of IDE Windows for Demos _____	34
Dragging Files to Obtain a Full Path _____	35
Moving Any Window Around _____	36
Creating Split Screens in the Same File _____	37
Customizing the VS.NET Menu and Toolbars _____	38
Adding External Programs to the VS.NET Menu _____	39
Dragging Files from Windows Explorer into VS.NET _____	40
Accessing the Command Window _____	41
Aliasing Your Favorite Commands _____	42
Switching to Immediate Mode from the Command Window _____	42
Using the Command Window in Find a Drop-Down List _____	42
Using the Built-in Web Browser _____	43
Using Full-Screen Mode _____	44
Copying the Fully Qualified Name of a Class _____	45
Recording and Replaying a Temporary Macro _____	45
Saving, Editing, and Debugging Macros _____	46
Assigning Shortcuts and Menu Items to Macros _____	46
Changing Properties of Several Controls _____	48
Locking Controls _____	49
Toggling the Description in the Properties Window _____	49
Change Drop-Down List Values in the Properties Window _____	50
Adding and Removing Event Handlers Through the IDE _____	50
Selecting Control Through a Drop-Down List _____	51
Adding an Installer Through the Designer for Windows Services _____	52

## Chapter 3: Compiling, Debugging, and Deploying \_\_\_\_\_ 53

Linking Files Instead of Copying Them into a Project _____	54
Setting the Default Namespace and Assembly Name _____	54
Generating Compiler Warnings and Errors _____	55
Generating Compiler Warnings Through the Obsolete Attribute _____	56
Setting Pre/Post Compile Build Steps _____	56
Setting the Assembly Output Path _____	57
Setting the .NET Framework Version for Your Assembly _____	57
Deploying ASP.NET Web Applications _____	59
Moving the Next Statement During Debugging _____	60
Changing Variable Values in the Watch Window _____	61
Executing SQL Procedures Through the Server Explorer _____	61
Using the Immediate Window to Display Variables and Execute Methods _____	62
Customizing the Call Stack _____	63
Placing Program Command-Line Arguments into Project Properties _____	64
Attaching VS.NET to an Already Running Process _____	65
Debugging Several Projects Inside the Solution _____	66
Breaking Only for Certain Exception Types _____	67

Breaking Only When Certain Conditions Apply	69
Debugging ASP.NET Web Application Through Trace.axd	71
Saving Any Output Window	73

## **Chapter 4: Using VS.NET 2005** **75**

Refactoring Code	76
Generating Method Stubs	77
Using Error Correction Suggestions	79
Using Predefined Code Snippets	79
Aligning UI Elements Automatically	82
Adding a Standard Menu Strip	82
Editing UI Element Properties with the Property Editing View	83
Controlling C# Code Formatting Precisely	84
Setting the Tab Order of Controls	85
Performing a Class, Instance, and Method Search	85
Viewing Code Definitions	85
Editing Web Controls in the HTML View	86
Validating HTML Code for Accessibility	87
Working with Different .NET Languages	88
Opening Web Projects Through FTP	88
Importing and Exporting IDE Settings	89
Closing All Other Windows	91
Showing Shortcuts for All Buttons	91
Building Selected Subset of Projects	92
Using Edit-and-Continue in VB.NET	92
Expanding Variable Members While Debugging	93
Using Data Visualization	94

## **Chapter 5: Other .NET Tips and Tricks** **97**

Obfuscating Your .NET Assemblies	98
No Rebuilding After HTML Changes	98
Iterating Over Strings on a Char per Char Basis	98
Using Inline Strings as Object Instances	99
Adding App.config to Your Application	99
Using Intermediate Language Disassembler to Inspect a .NET Assembly	100
Using Windows Class Viewer to Reference a Class	101
Running aspnet_regiis to Fix an IIS Installation	102
Precompiling Your ASP.NET Web Application	103
Setting ASP.NET Versions for Web Applications	104
Clearing the Assembly Cache Manually	104
Using Unicode for Strings, Labels, and Inline .NET Code	105
Rethrowing the Same Exception	106

## **Index** **109**

# Introduction

When Microsoft introduced C# and Visual Basic .NET upon the release of the .NET Framework in 2002, they also gave the developer community a brand-new integrated development environment, Visual Studio .NET. Over the past few years, while developers argued over which .NET language was better or whether .NET will hold its promise of being “the next big thing,” the vast majority of developers have come to agree on one thing: Visual Studio .NET is one of the most scalable, capable, and flexible development tool that Microsoft has ever released.

Visual Studio .NET is filled with thousands of features and functionalities that make our lives as developers more efficient. Not only is it a code editor, compiler, and debugger, but it contains features to stress-test, analyze, and optimize your code, and to allow easy integration with code documentation, reporting, or smart-device programming, such as the Pocket PC.

The number of features that Visual Studio .NET contains is immense. I doubt that many .NET developers know all of its features, shortcuts, and functionalities. Being a fan of keyboard shortcuts and tidbits that make my life as a developer better, or simply faster, I began to collect Visual Studio .NET tips and tricks. What started out as a small list became bigger and bigger, leading to workshops I gave in Visual Studio .NET. Those workshops finally evolved into this book.

*Visual Studio .NET Tips and Tricks* is a compilation of my favorite, or the most popular, tips and tricks that exist in this great IDE. This is not a comprehensive list, however, because that would have resulted in a book probably twice this size. I also consciously skipped a few obvious tips and tricks that I assume all developers should know. This book features the most common, most popular, or most effective tips and tricks that developers will find very useful. Beginners will find a box of treasures, while advanced Visual Studio .NET users will be amazed by how many new features and improvements the new Visual Studio .NET 2005 brings.

The majority of these tips and tricks are not undocumented. On the contrary, most of them are accessible through the VS.NET main menu or context menus. However, given the vast amount of features with which VS.NET is equipped, I often find that developers do not know them, or do not use them, as much as I think they should.

This book is based on the Beta 1 version of Visual Studio .NET 2005. As such, the final product might vary from how I describe it.

I welcome you to dive into this book. I hope you enjoy reading about these tips and tricks as much as I enjoyed discovering them.

Minh T. Nguyen  
September, 2004  
Bellevue, Washington

## Creating GUIDs

As you develop new classes and components, you often need to create so-called Global Unique Identifiers (GUIDs) which are 128-bit values often represented by 32 hexadecimals. In the beginning, component developers used GUIDs to assign their components with unique names because the likelihood of two components sharing the same GUID was extremely small. These days, developers use GUIDs for virtually anything that requires a unique identifier. Manually creating GUIDs by randomly selecting 32 hexadecimals is rather tedious.

Fortunately, VS.NET is preshipped with a utility that creates GUIDs for you whenever you need one. Select Tools > Create GUID to open the Create GUID dialog box (see Figure 22). Here you can generate identifiers in various formats, including common code snippets often used in COM development.

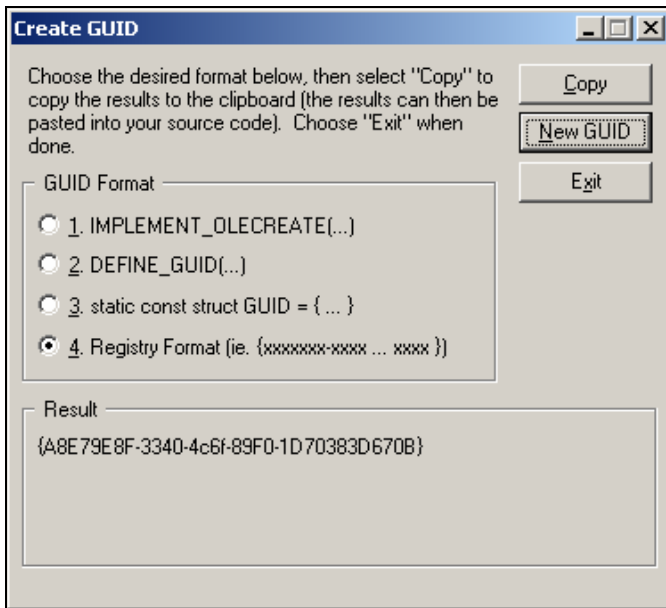


Figure 22 - Create GUIDs

Unfortunately, this utility does not offer a bare-basic format in which the 32 hexadecimals are simply provided in a sequence. This would be very useful as well.

## Creating Rectangular Selections

Did you know that you can make rectangular selections in VS.NET by pressing the Alt key while dragging the mouse over a rectangular area? This technique allows you to

create a rectangular selection that does not include the intervening lines (see Figure 23). Copying, cutting, and pasting rectangular blocks can be done very easily this way.

```
/// Summary description for WebForm1.
/// </summary>
public class WebForm1 : System.Web.UI.Page
{
    protected System.Web.UI.WebControls.Button Butt
    protected System.Web.UI.WebControls.DataGrid Da
    protected System.Web.UI.WebControls.Button Butt
    protected System.Web.UI.WebControls.LinkButton
    protected System.Web.UI.WebControls.Repeater Re
    protected System.Web.UI.WebControls.Calendar Ca

    private void Page_Load(object sender, System.Ev
    {
```

Figure 23 – Press and hold the Alt key to create rectangular selections

You might wonder why anyone would ever need this esoteric feature. In fact, I have used it often to make targeted query-replaces in certain parts of a document that otherwise prevented me from doing so because of normal line-wrapping selections. I assure you that you will find this technique very handy.

## Switching Between Views

When you develop ASP.NET applications, you often need to switch between the Designer view and the HTML view. Instead of moving the mouse to the appropriate view button, you can switch between them much faster by pressing the Ctrl-PgUp and Ctrl-PgDn keyboard shortcuts. (For web forms, it really doesn't matter which shortcut you use because either one will switch to the other view.) This also works for switching between the XML view and the Data view when reading XML documents.

For Windows forms, press F7 to switch to the Code view and press Shift-F7 to switch to the Designer view.

In VS.NET 2005, Ctrl-PgUp and Ctrl-PgDn work for web forms but do not work for XML documents. For Windows forms, F7 toggles the designer between both views. (Shift-F7 is reserved for another feature entirely in VS.NET 2005.)

# Deploying ASP.NET Web Applications

Whenever you deploy an ASP.NET web application to the web server, you have to choose the files carefully that you want to send over FTP. One question that newbies often ask is what files they have to deploy. For instance, you do not need to—and it's recommended that you don't—deploy any of the code-behind source code files (because they are already compiled into the DLL file) or the project and solution files. In most cases, all you need to deploy are Web.config, Global.asax, and any files ending with .aspx, asmx, ascx, or .dll. However, you can allow VS.NET to make this tedious selection for you.

Selecting Project > Copy Project allows you to copy your web project to a web server using FrontPage extensions or through a network share (see Figure 54). I usually copy the project to a temporary virtual directory using the File Share option because, by default, it's set up for the local development computer. At the bottom of the Copy Project dialog box I specify to copy “only files needed to run this application.” By selecting this option, VS.NET finds out which files need to be deployed and copies only those. Now I just need to FTP the complete content of the temporary directory to the web server.

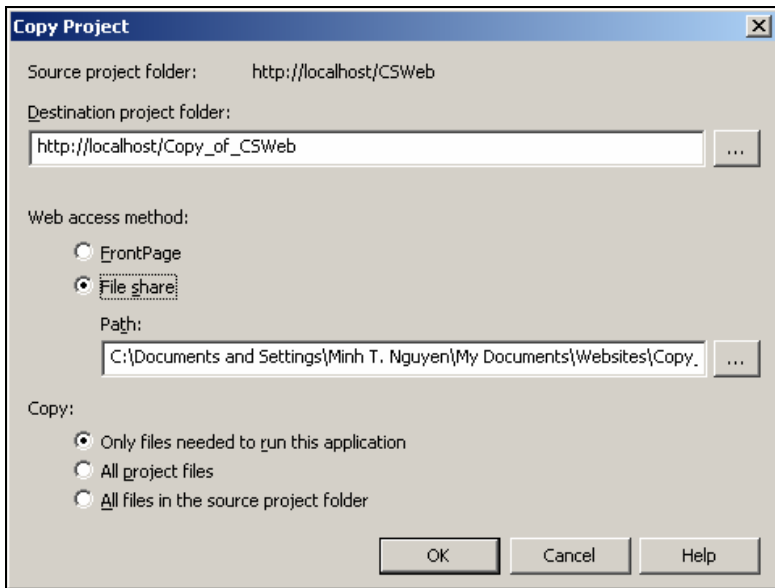


Figure 54 - Letting VS.NET decide which files to deploy

## Setting the Tab Order of Controls

The tab order is the order by which controls on the form receive focus as you press the Tab key. You can control this order by setting the Tab Index property of each control to a number that corresponds to the position in this order. This can prove difficult at times because you don't know—and can't see—the other controls' tab index unless you select them. VS.NET 2005 introduces a new way to set the tab order: the Tab Order button on the Layout bar (see Figure 77).



Figure 77 - Tab Order button on the left of the Layout bar

Clicking the Tab Order button displays the tab index for all UI elements on the form. Here's the beauty of this mode: Not only do you now see all the tab indices, but you can repeatedly click on each UI element to set the tab order in linear fashion. The first element you select has a tab index of zero. The next one you select has a tab index of one, and so on. As you set the index for each control, the background color of the tab index caption switches from blue to white, so you can keep track of which UI elements you have already tagged. To prevent you from accidentally selecting a wrong UI element, a gray rectangle surrounds the element you mouse over for better identification.

When you are done setting the tab order, click the Tab Order button again or simply press the Escape key. Unfortunately, this tab order feature works only for Windows forms, not web forms.

## Performing a Class, Instance, and Method Search

In a solution with many projects, it is sometimes hard to locate a class, instance, or method. If you know the name of a class, instance, or method, but don't know where it is located, you could always use the global find feature.

A better way is to use the new search feature in the Class view. Open up the Class view (click the tab usually right next to the Solutions Explorer or press Ctrl-Shift-C) and note the search window at the top. You don't have to type the full name of the item you are looking for; a partial search term works too. The Class view filters the items to show only those that match your partial search key.

## Viewing Code Definitions

The Code Definition view is a window new to VS.NET 2005 that allows you to view the definition of a class as you move your cursor over a type. View this window by selecting View > Other Windows > Code Definition View or pressing Ctrl-Shift-D. By

default, it appears as another window below the main text editor. As you move your cursor around and inside a word that defines a class (see Figure 78), the Code Definition view displays all the properties, methods, indices, and other members of the class—all of which are fully equipped with XML comments (if they exist). This tool is similar to the output of the WinCV utility that accompanied the initial version of VS.NET.

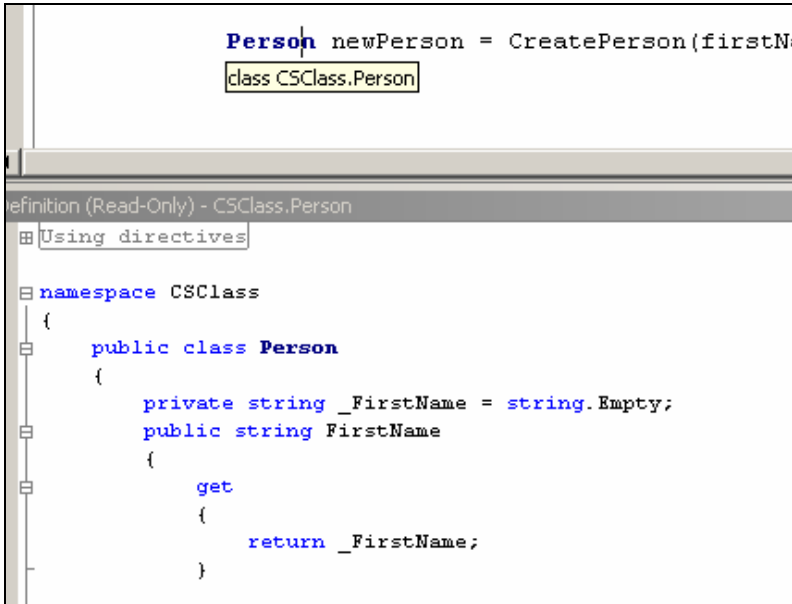


Figure 78 - Code Definition view displaying class information

The Code Definition view also works for user-defined classes. In fact, if the class is part of your solution, it displays the actual source code of that class—not just what it can get through reflection.

## Editing Web Controls in the HTML View

In VS.NET 2002 and 2003 you can modify web controls only in the Designer view. This is not a pleasant way to work because of the additional rendering process and often incorrect auto-formatting of your HTML code. This has now been greatly improved. For one, your HTML code is no longer formatted unless you specifically request it. Second, anything that you can do in the Designer view, you can now do in the HTML view.

For instance, you can drag controls from the Toolbox directly into the HTML editor. The necessary HTML or ASP.NET tag will be inserted for you. While the cursor is located in a certain web control, you can also modify the control's properties in the Properties window. Changes immediately appear in the HTML view as attributes. Setting a property to blank removes the attribute from the HTML or ASP.NET tag.

# Index

## A

- Accessibility, validating, 87–88
- Accessibility Validation dialog box, 87
- Add dialog box and class members, 12
- Add New Item dialog box, 54–55
- Aliases, 42
- App.config, adding, 99–100
- Application configuration file, modifying, 58
- Arguments, command line, 64
- Aspnet\_regiis, running, 102–103
- ASP.NET Web applications
  - debugging, 43, 71–73
  - deploying, 59
  - FTP and, 88–89
  - precompiling, 103
  - tracing, 71–73
  - versions, setting, 104
  - VS.NET, attaching, 65–66
- Assemblies
  - .Net Framework
    - inspecting, 100–101
    - obfuscating, 98
    - version, setting, 57–58
  - output path, setting, 57
- Assembly cache, clearing, 104–105
- Assembly names, setting, 54–55
- Attributes
  - localOnly described, 73
  - Obsolete and compiler warnings, 56
- Autocomplete, 16

## B

- Blocks, formatting, 12–14, 86
- Bookmarks, 25–26
- Bookmarks window, 26
- Breaking, 67–71

- Breakpoints window, 69, 70, 71
- Browsers, 27–28, 43–44
- Build Action property and file embedding, 34
- Build Events dialog box, 57
- Build menu, 92
- Build order, controlling, 33–34, 56–57
- Build Order tab, 33
- Buttons
  - creating, 47
  - shortcuts, showing, 91–92
- Bytes and file embedding, 34

## C

- C#
  - blocks, formatting, 12–14
  - build order, controlling, 56–57
  - code blocks, commenting, 5
  - code snippets, inserting, 81
  - compiling in, 55–56
  - Edit-and-Continue function, 93
  - errors correction suggestions, using, 79
  - event handlers, adding/removing, 50–51
  - formatting, controlling, 84
  - interface methods, implementing, 16–18
  - object member fields, viewing, 2
  - string literals, selecting, 8
- Call stack, customizing, 63–64
- Class Definition view, 102
- Classes
  - adding, 54–55
  - definitions, viewing, 85–86
  - field members, adding, 12
  - names, copying, 45
  - referencing, 101–102
  - resolving, 79
  - searches, 85, 101–102

- Class Library projects, 58
- Class view and searching, 85
- Client-side scripting and JavaScript tags, 28–29
- Clipboard Ring described, 9
- Code
  - behind file, showing, 32–33
  - blocks, commenting, 5
  - collapsing, 6
  - compatibilities, checking, 58
  - definitions, viewing, 85–86
  - error compiler directives in, 55–56
  - refactoring, 76–77
  - and regions, 7
  - snippets, using, 79–81
  - and the Toolbox, 8–9
- Code Comment Web Report, contents of, 3
- Code Definition view, 85–86
- Code Snippet Manager, 81
- Code view and event handlers, 51
- Commands, 42
- Command window
  - accessing, 41
  - drop-down lists, finding, 42–43
  - and Immediate mode, 42
- Comments, adding, 2–5
- Compatibilities, checking, 58
- Compiler functions
  - ASP.NET Web applications, 103
  - automating, 103
  - build steps, setting, 56–57
  - directives in, 55–56
  - obfuscating, 98
  - warnings, generating, 55–56
- Conditions for breakpoints, setting, 69–71
- .config files, managing, 99–100
- Configuration file, modifying, 58
- Control Library projects, naming, 55
- Controls
  - editing, 48, 86
  - locking, 49
  - properties, changing, 48
  - selecting via drop-down list, 51
  - tab order of, 85

## D

- DataSet and data display, 94–95
- DataTable and data display, 94–95
- DataTables, viewing contents of, 62–63
- Data visualization, 94–95
- Debugging
  - ASP.NET Web applications, 43, 71–73
  - command-line arguments, placing, 64
  - and Immediate mode, 42
  - macros, 46
  - multiple projects, 66–67
  - next statement, moving, 60
  - and SQL Server, 66
  - and the stack trace, 64
  - variable members, expanding, 93–94
  - variable values, changing, 61
  - VS.NET, attaching, 65–66
- Dependencies, setting, 33–34
- Description, toggling, 49
- Designer view, 52
- Documents, formatting, 12–14. *See also* Text
- Dotfuscator, 98
- Drop-down lists
  - controls, selecting, 51
  - finding, 42–43
  - values, changing, 50

## E

- Edit-and-Continue function, 92–93
- Editor, changing, 32
- Elements, UI
  - aligning, 82
  - properties, editing, 83–84
- Encapsulate Field method described, 77
- Errors
  - correction suggestions, using, 79
  - generating, 55–56
- Event handlers, adding/removing, 50–51

Exceptions  
  adding/removing, 69  
  rethrowing, 106–107  
  runtime, 20–21  
  types, breaking for, 67–69  
Exceptions dialog box, 68, 69  
Expressions, evaluating, 62  
Extract Interface method described, 77  
Extract Methods process described, 76

## F

Favorites folder window, 43  
Field members and IntelliSense, 15  
Files  
  configuration, modifying, 58  
  dragging, 35–36, 40–41  
  embedding, 34  
  extra, showing, 32–33  
  linking, 54  
  macros, saved location of, 46  
  opening, 32, 41  
  split screens, creating, 37–38  
  text file, inserting, 28  
  windows, closing, 91  
Find dialog box, 26  
Font size, changing, 34–35  
Formatting standards defined, 84  
Forms, Windows  
  command-line arguments, placing, 64  
  controls, 49, 51  
  downloads, forcing, 105  
  hierarchy, outlining, 29–30  
  menus, adding, 82–83  
  properties, changing, 48  
  views, switching, 20  
FromFirst column, 72  
FromLast column, 72  
FTP and Web projects, 88–89  
Full-screen mode, 44

## G

Global unique identifiers (GUIDs),  
  creating, 19  
GUIDs (Global unique identifiers),  
  creating, 19

## H

Hierarchy, outlining, 29–30  
HTML  
  hierarchy, outlining, 29–30  
  rebuilding and, 98  
  tags, collapsing, 7  
  text, pasting as, 12  
  validating for accessibility, 87–88  
HTML Editor  
  blocks, formatting, 12–14  
  files, dragging in, 35–36  
  Web controls, editing, 86  
HTML Visualizer, 95

## I

Icons, managing, 38, 40  
IDE restarts and task shortcuts, 5  
IDE settings, importing/exporting,  
  89–91  
IIS installation, fixing, 102–103  
Images, embedding, 34  
Immediate mode from the Command  
  window, 42  
Incremental searches, 21–22  
Installers, adding, 52  
Instances, searching, 85  
IntelliSense  
  across projects, applying, 3–4  
  code snippets and, 79–80  
  and the Command window, 41  
  described, 2  
  and field members, 15  
  and the Immediate window, 63  
  inline strings and, 99

- methods, overriding, 18
- parameter information and, 15

Interfaces, 16–18, 77

Intermediate Language Disassembler, 100–101

## J

JavaScript tags, inserting, 28–29

## L

Labels and Unicode, 105–106

Languages and the .Net Framework, 88

Language tool and Unicode strings, 105

Line numbers, going to, 20–21

localOnly attribute described, 73

## M

Macro Explorer window, 46

Macros

- and the build order, 57
- recording/replaying, 45
- saving/editing/debugging, 46
- shortcuts, assigning, 46–48

Menus

- customizing, 38–40, 46–48
- standard, adding, 82–83

Methods

- definitions, going to, 27
- executing, 62–63
- extracting, 76–77
- interface, implementing, 16–18
- overriding, 18
- and regions, 6
- searches, 85
- stubs, generating, 77–78

## N

Namespaces, 45, 54–55

Navigate-Backward button, 27, 28

Navigate-Forward button, 27, 28

Navigation, browser-like, 27–28

.Net Framework

- assemblies
  - inspecting, 100–101
  - obfuscating, 98
  - version, setting, 57–58
- exceptions and, 69
- languages and, 88
- and Unicode, 105–106
- version, setting, 57–58

Next statement, moving, 60

## O

Objects, 2, 99

Obsolete attribute and compiler warnings, 56

Open With dialog box, 32

Outlining menu described, 7

Output path, setting, 57

Output window, saving, 73

Override command defined, 18

## P

Paragraphs, commenting, 5. *See also* Text

Parameter information and IntelliSense, 15

Placeholders, identifying, 80

Procedures, editing stored, 61

Processes dialog box, 65

Programs, external adding to menus, 39–40

Project properties and command-line arguments, 64

Projects

- dependencies, setting, 33–34

- subsets, building, 92
- Promote Local Variable To Parameter
  - method described, 77
- Properties
  - changing, 48, 83–84, 85
  - command-line arguments, placing, 64
  - creating, 77
- Properties window
  - controls
    - changing, 48, 86
    - locking, 49
  - description, toggling, 49
  - drop-down list values, changing, 50
- Property Editing View button, 83
- Public override command defined, 18

## R

- Rectangular sections, creating, 19–20
- Regions
  - collapsed and incremental searches, 22
  - creating, 6–7
  - temporary, 7
- Reminders, listing, 4
- Remove Parameters method described, 77
- Rename method described, 77
- Reorder Parameters method described, 77
- Replace dialog box, 22
- Replacing, 22, 23–25
- Runtime exceptions and line numbers, 20–21

## S

- Search dialog box, 22
- Searches
  - class/instance/method, 85, 101–102
  - global, 23–25
  - incremental, 21–22
  - wildcards, 22–23

- words, 21
- Selections, hiding, 7
- Server Explorer and SQL procedures, 61
- Services, Windows, 52, 66
- Shortcuts
  - assigning, 40, 46–48
  - IDE restarts and, 5
  - viewing, 91–92
- Solution Explorer
  - files
    - dragging, 35–36, 40–41
    - embedding, 34
    - linking, 54
  - startup project listing, 66
  - validating HTML for accessibility, 87–88
- Split screen, creating, 36–38
- SQL Server, 61, 66
- Stack frame described, 63
- Stack trace
  - clearing, 106–107
  - debugging, 64
  - described, 64
- Strings
  - inline as object instances, 99
  - iterating over, 98–99
  - literals, selecting in C#, 8
  - Unicode, 105–106
- Symbols, 77, 105–106
- Synchronization, forcing, 29

## T

- Tab Index property, setting, 85
- Task shortcuts, placing, 5
- Text
  - autocomplete, 16
  - blocks, formatting, 12–14
  - characters ignored, 10
  - files, inserting, 28
  - font size, changing, 34–35
  - front/end switching, 8
  - pasting as HTML, 12
  - rectangular sections, creating, 19–20

- single line functions, 10
- single words, selecting, 8
- strings, iterating over, 98–99
- transpositions of, 9–10
- words, searching for, 21
- word-wrapping, toggling, 14
- XML editing, 10–11
- Text Editor toolbar and bookmarks, 25–26
- Text Visualizer, 95
- Toolbars
  - customizing, 38–39, 46–48
  - shortcuts, viewing, 91–92
- Toolbox, 8–9
- Trace.axd, 71–73
- TraceContext class described, 72
- Trace logs, saving, 73

## U

- UI elements, 82, 83–84
- Undo feature and replacing, 24
- Unicode, 105–106

## V

- Validation for accessibility, 87–88
- Variables
  - displaying, 62–63
  - members, expanding, 93–94
  - values, changing, 61
- View mode, customizing, 32
- Views, switching, 20
- VS.NET, attaching, 65–66

## W

- Warn() method described, 72
- Watch window, 61, 62
- Web applications
  - debugging, 43, 71–73
  - deploying, 59
  - FTP and, 88–89
  - precompiling, 103
  - tracing, 71–73
  - versions, setting, 104
  - VS.NET, attaching, 65–66
- Web browser, using, 43–44
- Web Content Accessibility Guidelines, 87
- Web controls, editing, 48, 86
- Web pages, 3, 59
- Windows, 36, 91
- Windows Explorer and file dragging, 40–41
- Windows Forms. *See* Forms, Windows
- Windows Services. *See* Services, Windows
- Words. *See* Text
- Word-wrapping, toggling, 14

## X

- XML
  - blocks, formatting, 12–14
  - comments, adding, 2–3
  - editing
    - in tabular mode, 11–12
    - in text mode, 10–11
  - nodes, inserting, 11
- XML Visualizer, 95
- XSLT transformations, creating, 3