

Coding Faster:

Getting More Productive with Microsoft®

Visual Studio®

Covers Microsoft Visual Studio 2005, 2008, and 2010



Zain Naboulsi
Sara Ford

Coding Faster: Getting More Productive with Microsoft® Visual Studio®

Expert advice to help you work smarter and increase your productivity with Visual Studio.

Dig in—and learn practical techniques to help you work more efficiently with Visual Studio. This fully revised and expanded version of *Visual Studio Tips: 251 Ways to Improve Your Productivity* includes a comprehensive collection of tips and shortcuts for working with the code editor, visual designers, searches, debugger, and other features in Visual Studio 2005, 2008, and 2010. You'll gain valuable insights for using this IDE—no matter what your experience level.

Discover how to:

- Change development settings and keyboard mappings to optimize your efficiency
- Save time initiating a new project by creating custom templates
- Uncover shortcuts and tips for working with the toolbox, commands, and window layouts
- Discover better ways to work with files, such as the File Tab Channel
- Use the Editor more effectively with tips on selecting and manipulating code
- Apply techniques to help reduce the time you spend debugging code
- Create Visual Studio extensions to increase your development productivity



Includes online appendix of additional tips

Ready to download at
<http://go.microsoft.com/fwlink/?Linkid=223758>

For **system requirements**, see the Introduction.

microsoft.com/mspress

ISBN: 978-0-7356-4992-7



U.S.A. \$39.99
Canada \$45.99
[Recommended]

Programming/Microsoft Visual Studio



About the Authors

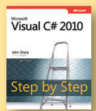
Zain Naboulsi is a senior developer evangelist at Microsoft, and frequently lectures on Visual Studio topics. He maintains the Visual Studio Tips and Tricks blog, and created Online Community Evangelism in an effort to build communities on social networks such as LinkedIn and Facebook.

Sara Ford is the author of the award-winning *Visual Studio Tips: 251 Ways to Improve Your Productivity*. During her nine years with Microsoft, she was a software tester for Visual Studio and managed CodePlex, the open source project hosting site.

RESOURCE ROADMAP

Developer Step by Step

- Hands-on tutorial covering fundamental techniques and features
- Practice exercises
- Prepares and informs new-to-topic programmers



Developer Reference

- Expert coverage of core topics
- Extensive, pragmatic coding examples
- Builds professional-level proficiency with a Microsoft technology



Focused Topics

- Deep coverage of advanced techniques and capabilities
- Extensive, adaptable coding examples
- Promotes full mastery of a Microsoft technology



See inside cover

Microsoft®
Visual Studio®

Microsoft®

Coding Faster: Getting More Productive with Microsoft® Visual Studio®

Covers Microsoft® Visual Studio® 2005, 2008, and 2010

Zain Naboulsi
Sara Ford

Copyright © 2011 by Zain Naboulsi and Sara Ford

All rights reserved. No part of the contents of this book may be reproduced or transmitted in any form or by any means without the written permission of the publisher.

ISBN: 978-0-7356-4992-7

1 2 3 4 5 6 7 8 9 M 6 5 4 3 2 1

Printed and bound in the United States of America.

Microsoft Press books are available through booksellers and distributors worldwide. If you need support related to this book, email Microsoft Press Book Support at mspinput@microsoft.com. Please tell us what you think of this book at <http://www.microsoft.com/learning/booksurvey>.

Microsoft and the trademarks listed at <http://www.microsoft.com/about/legal/en/us/IntellectualProperty/Trademarks/EN-US.aspx> are trademarks of the Microsoft group of companies. All other marks are property of their respective owners.

The example companies, organizations, products, domain names, email addresses, logos, people, places, and events depicted herein are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

This book expresses the author's views and opinions. The information contained in this book is provided without any express, statutory, or implied warranties. Neither the authors, Microsoft Corporation, nor its resellers, or distributors will be held liable for any damages caused or alleged to be caused either directly or indirectly by this book.

Acquisitions and Developmental Editor: Russell Jones

Production Editor: Adam Zaremba

Technical Reviewer: Kevin Stevens

Copyeditor: Richard Carey

Indexer: Ron Strauss

Proofreader: Nancy Sixsmith

Cover: Karen Montgomery

Compositor: Ron Bilodeau

Illustrator: Robert Romano

*First and foremost to God and Jesus Christ for making this all possible.
To my mom, Helen Naboulsi, for always encouraging me to go above and beyond to reach my goals,
and to Russell Chandler for being the greatest nephew anyone could ever have.*

Zain Naboulsi

Senior Developer Evangelist, Microsoft

*To my parents, Jane and Louie Smolensky, for encouraging me to program on the computer when I was
5 years old, and to Beulah Bourgeois and Annabelle Fayard for being the best babysitters a little girl
could ever hope for.*

Sara Ford

Contents at a Glance

Part I **Productivity Techniques**

1	Getting Started	3
2	Projects and Items	43
3	Getting to Know the Environment.	83
4	Working with Documents	153
5	Finding Things.	171
6	Writing Code	209
7	Debugging.	291

Part II **Extensions for Visual Studio**

8	Visual Studio Extensions	385
----------	------------------------------------	------------



Table of Contents

Foreword	xxiii
Introduction	xxvii

Part I **Productivity Techniques**

1	Getting Started	3
01.01	Running Multiple Versions of Visual Studio Side-By-Side	3
01.02	Getting Table of Contents in Visual Studio 2010 Online Help.....	4
	Online Help	4
	Using Classic View.....	5
01.03	Exporting Your Environment Settings	6
01.04	Remove Projects from the Recent Projects List.....	9
01.05	AutoRecover.....	10
01.06	Improving Performance by Changing the Visual Experience in Visual Studio 2010.....	12
01.07	Change Tool Window Animations	14
01.08	Importing or Changing Your Environment Settings	14
01.09	Change Your Visual Studio Color Scheme.....	17
	Seeing What You Like	18
	Getting the Goods	18
	Changing Your Colors.....	21
	Resetting the Colors.....	24
01.10	Reset All Your Development Settings	25
01.11	Customize Your Toolbars in Visual Studio 2010: Toolbars Tab	27
	Custom Toolbars	28
01.12	Customize Your Toolbars in Visual Studio 2010: Commands Tab	30
	Rearrange	32
	Controls	32
	Buttons	33
	Modify Selection.....	34
01.13	Visual Studio Logging.....	37
01.14	Visual Studio Safe Mode	38
01.15	The ResetSettings Switch.....	39
	Two Different Machines	39
	Same Machine	40

2	Projects and Items	43
02.01	Search for Project Templates in the New Project Dialog Box	43
	Good News	44
	Bad News	44
	More Good News	44
02.02	Recent Project Templates in the New Project Dialog Box	45
02.03	Using Older Frameworks with Multi-Targeting	45
02.04	Create Web Application or Virtual Directory in IIS	46
02.05	Multiple Startup Projects	48
02.06	Change the Default New Project Location	50
02.07	Track Active Item in Solution Explorer	51
02.08	Type-Ahead Selection Support in Solution Explorer	52
02.09	Using Solution Folders	54
	Adding Solution Folders	54
	Removing Solution Folders	55
02.10	Navigating Property Tabs in the Project Properties	55
02.11	Pin a Project to the Recent Projects List	55
02.12	Creating Temporary Projects	56
02.13	Create Your Own Item Template	57
02.14	Roll Your Own Project Template with the Export Template Wizard.	64
02.15	Organizing Your Custom Item Templates	68
02.16	Organizing Your Custom Project Templates	71
02.17	Reorganize the Default Item Templates	74
02.18	Reorganize the Default Project Templates	77
02.19	Change the Templates that Appear in the New Project or Item Dialog Boxes	80
3	Getting to Know the Environment	83
03.01	Rearrange Windows in Visual Studio 2010 Using the Guide Diamond	83
03.02	Dock a Floating Tool Window Back to Its Previous Location	85
03.03	Cycle Through Your Open Tool Windows	86
03.04	Closing Tool Windows	87
03.05	Expand and Collapse All in the Toolbox	87
03.06	Searching in the Toolbox	88
03.07	Navigate Among Tabs in the Toolbox	89

03.08	Window Layouts: The Four Modes.....	90
	Design View	90
	Debugging View	91
	Full Screen	91
	File View	91
03.09	Window Layouts: Design, Debug, and Full Screen	91
	Design Mode	91
	Debug Mode	92
	Full Screen Mode	93
03.10	Working with Tabs in the Toolbox	93
	Creating Tabs.....	93
	Adding Items	94
	Custom Controls	95
	Renaming Tabs.....	95
	Deleting Tabs.....	95
03.11	Using Additional Browsers for Web Development	96
	Adding New Browsers	97
	Changing the Default Browser	97
	Browser Window Size	98
	Removing Browsers	98
03.12	Auto-Hide All Tool Windows.....	99
03.13	Showing Hidden Tool Windows with the Auto Hide Channel.....	100
03.14	Moving Tool Windows Around with Your Keyboard	102
03.15	Keyboard Access to a Tool Window's Toolbar	103
03.16	Command Prompt History.....	104
03.17	Command Prompt Tab Completion.....	105
	Simple Search	105
	Wildcard Search	105
	Finally	107
03.18	Undock and Dock a Single Tool Window in a Group	107
	Undock	107
	Dock	108
03.19	Understanding Commands: Simple Commands	110

What do you think of this book? We want to hear from you!

Microsoft is interested in hearing your feedback so we can continually improve our books and learning resources for you. To participate in a brief online survey, please visit:

microsoft.com/learning/booksurvey

03.20	Understanding Commands: Aliases	113
	Create a New Alias	114
	Viewing Assigned Aliases	114
	Delete an Alias	115
03.21	Understanding Commands: Arguments and Switches	115
	Basic Use	116
	Arguments and Switches	116
	List Current Options	117
	Reset Options	117
	Using the Arguments and Switches	117
	Make an Alias	118
03.22	Testing a Command	119
03.23	Understanding Commands: Running Commands	121
	Shortcuts	121
	Command Window	122
	Immediate Window	122
	Find Combo Box	122
03.24	Find Keyboard Shortcuts	122
03.25	Keyboard Shortcuts: Additional Mapping Schemes	125
03.26	Keyboard Shortcuts: Creating New Shortcuts	127
	Reset	130
03.27	Keyboard Shortcuts: Reset All Your Shortcuts	131
03.28	Understanding Commands: Logging Commands	132
	Arguments	133
	Example	133
03.29	Export Your Window Layouts	134
03.30	Stop the Toolbox from Auto-Populating from the Solution	136
03.31	Using External Tools	136
	Use Output Window	140
	Treat Output As Unicode	140
	Prompt For Arguments	140
	Close On Exit	140
03.32	Create Keyboard Accelerators for External Tools	141
03.33	Exporting Your Command Window Aliases and External Tools List	143
03.34	Creating and Using a Macro	144
03.35	Visual Studio Image Library	147
	Types of Files	147
	Image Library Contents	148
	Using the Images	151

4	Working with Documents	153
04.01	Insert Documents to the Right of Existing Tabs	153
04.02	Recent Files	154
04.03	Working with Documents on Multiple Monitors	155
04.04	Navigate Open Document Windows	157
04.05	Close the Current Document Window	158
04.06	Open a File Location from the File Tab	158
04.07	Open the File Menu Drop-Down List from Your Keyboard	159
04.08	Using the IDE Navigator	160
	Navigator Areas	161
04.09	Multiple Views of the Same Document	163
	Special Note for VB Users in Visual Studio 2010	163
	Multiple Views	164
04.10	Closing Just the Selected Files You Want	164
04.11	Understanding the File Open Location	165
04.12	Show Previous Versions	166
04.13	Using Custom File Extension Associations	168
5	Finding Things	171
05.01	Repeat Your Last Search	171
05.02	Using Quick Find	172
	Find What	173
	Look In	173
	Find Options	174
	Buttons	176
05.03	Using a Simple Quick Replace	176
05.04	Hide the Quick Find and Quick Replace Tool Window After the First Match	178
05.05	Undo Quick Replace and Replace in Files	179
	Quick Replace (Ctrl+H)	179
	Replace in Files (Ctrl+Shift+H)	180
05.06	Using the Find Combo Box Keyboard Shortcuts	182
	Find (Ctrl+D)	182
	Run Command (Ctrl+/.)	182
	Go To Line (Ctrl+G)	183
	Go To File (Ctrl+Shift+G)	183
	Set a Breakpoint (F9)	183

05.07	Using Incremental Search	184
05.08	Search the Currently Selected String Without the Find Window	185
05.09	Find In Files: Find Options	186
	Find What	187
05.10	Find In Files: Result Options	190
	Find Results [1,2] Window	190
	Navigation	190
	Clear All	191
	Display File Names Only	191
	Keep Modified Files Open After Replace All	191
05.11	Replace In Files: Basic Options	192
	Find Options	192
	Replace With	192
	Result Options	193
	Execution	193
05.12	Go To Definition for Cascading Style Sheets	194
05.13	How to Use Navigate To	195
05.14	Understanding Find Symbol	196
	Find What	197
	Look In	197
	Find Options	199
	Search Results	199
05.15	Find Symbol Results Shortcuts	200
	Go To Definition (F12)	200
	Go To Declaration (Ctrl+F12)	201
	Go To Reference (Shift+F12)	201
	Browse Definition	202
	Copy (Ctrl+C)	203
	Clear All	203
05.16	Replace in Files: Tagged Expressions	203
	Example	204
05.17	Customize Results in Find In Files Searches	206
	Variables	207

What do you think of this book? We want to hear from you!

Microsoft is interested in hearing your feedback so we can continually improve our books and learning resources for you. To participate in a brief online survey, please visit:

microsoft.com/learning/booksurvey

6	Writing Code	209
06.01	Zoom In or Out of Text in the Editor Using the Mouse Wheel	209
06.02	Zoom In or Out of Text in the Editor	210
	Combo Box	210
	Keyboard	211
	Universal Zoom	211
06.03	How to Keep from Accidentally Copying a Blank Line	211
06.04	Make IntelliSense Transparent	212
06.05	Cut or Delete the Current Line	213
	Cut	214
	Delete	214
06.06	Using the New IntelliSense: Keywords	214
06.07	Using the New IntelliSense: Pascal Case	216
06.08	Comment and Uncomment in Web Pages	217
	Comment	217
	Uncomment	218
06.09	Insert a Blank Line Above or Below the Current Line	218
06.10	Transpose Lines, Words, and Characters	219
06.11	How to Cycle Through the Clipboard Ring	220
06.12	Using the Undo and Redo Stack	221
06.13	Undo and Redo Global Actions	222
06.14	How to Use Reference Highlighting	223
	Navigation	223
	Turning it Off	224
06.15	Moving or Selecting Between Matching Braces (C++, C# Only)	224
	Moving	225
	Selecting	225
06.16	Invoke Statement Completion	225
06.17	Move Between the Common Tab and All Tab in Statement Completion (VB)	226
06.18	Using Parameter Information	227
06.19	Using Quick Info	228
06.20	Word Completion	228
06.21	Drag and Drop Code into the Toolbox	229
06.22	Using Smart Tags from the Keyboard	231

06.23	Organize Using Statements (C# Only)	232
	Remove Unused Usings	232
	Sort Usings	234
	Remove and Sort	234
06.24	Switch Between Design and Source in Web Projects	235
	Split View	235
06.25	Toggle Designer	236
06.26	Change the Default View in the HTML Editor	236
06.27	Jump Back to the Editor from Just About Anywhere	237
06.28	Replacing Text with a Box Selection	237
06.29	Pasting the Contents of One Box Selection into Another	238
06.30	Pasting a Single Selection into a Box Selection	239
06.31	Using Zero-Length Box Selection	240
06.32	View White Space	241
06.33	Collapsing Your Code with Outlining	242
	Minus Sign	243
	Vertical Line	243
	Click Anywhere in Area (Keyboard Shortcut)	243
	Click Anywhere in Area (Menu Item)	243
06.34	Using Hide Selection	244
06.35	Collapse to Definitions with Outlining	246
06.36	Cut, Copy, and Paste Collapsed Code with Outlining	247
06.37	Understanding Word Wrap	248
06.38	Properties Window Keyboard Shortcuts	249
	Working with the Tool Window	250
	Working with Categories	250
	Property Items	251
06.39	Document Outline: Web Projects	251
06.40	Inserting Code Snippets	253
	Tab	253
	Keyboard Shortcut and Context Menu	254
	Result	256
06.41	Surround with a Code Snippet	256
06.42	Using Code Snippets	258
06.43	HTML Code Snippets	259
06.44	JavaScript Code Snippets	260
06.45	Using the Code Snippets Manager	261
06.46	Insert Quotes When Typing Attribute Values	264

06.47	Format the Current Document or Selection (Web)	265
06.48	Using the Navigation Bar	266
06.49	HTML Editor Tag Navigation	267
06.50	Format HTML on Paste	267
06.51	Display HTML/CSS Warnings as Errors	268
06.52	Updating JScript IntelliSense	269
06.53	Using JScript Libraries in Other JScript Files	270
06.54	Create New Code Snippets from Existing Ones	271
06.55	Understanding the Navigation Stack	275
06.56	Navigate Backward and Navigate Forward Using Go Back Markers	277
06.57	Select from the Current Cursor Location to the Last Go Back Marker	278
06.58	Track Changes in the Editor	280
06.59	Edit Read-Only Files	281
	Edit In-Memory	283
	Make Writable	283
06.60	Choosing CSS Versions	283
	Dedicated Style Sheets	283
	Embedded Styles	284
	Finally	284
06.61	Understanding Tag Specific Options	285
	Exploring the Tag Specific Options Dialog Box	285
	Finally	289
7	Debugging	291
07.01	Setting a Breakpoint with Code	291
	Compiler Directive	292
07.02	Using Ctrl+Alt+B to Open the Breakpoints Window	293
07.03	Adding Labels to Breakpoints	293
07.04	Enable or Disable All Breakpoints	295
07.05	TODO Comments in the Task List	296
	VB	296
	C#	296
	C++	296
07.06	Create Custom Tokens for the Task List	297
	Sharing Tokens	299
07.07	Create Code Shortcuts in the Task List	300
07.08	Code Definition Window	301

07.09	Save Changes Before Building	302
	Save Changes To Open Documents Only	302
	Prompt To Save All Changes	303
	Don't Save Any Changes	303
07.10	Navigate Errors in the Error List	303
07.11	Ordering and Multicolumn Sorting in Tool Windows	304
	Column Ordering	304
	Multicolumn Sorting	305
07.12	Pin a DataTip to Source Code	305
07.13	Create a Floating DataTip	306
07.14	Adding Comments to a DataTip	307
07.15	Use a DataTip to Edit a Value	308
07.16	DataTip Value from the Last Debug Session	309
07.17	Import and Export DataTips	309
07.18	Using the Call Hierarchy	310
07.19	Searching Breakpoints	312
07.20	Breakpoint Hit Count	314
	Break When The Hit Count Is Equal To	315
	Break When The Hit Count Is A Multiple Of	315
	Break When The Hit Count Is Greater Than Or Equal To	316
07.21	Set a Breakpoint on a Function	316
	Breakpoints Window	317
07.22	Set a Simple Breakpoint Condition	318
	Is True	319
	Has Changed	320
	Special Notes	320
07.23	Set a Complex Breakpoint Condition	321
07.24	Setting a Breakpoint Filter	324
07.25	Setting a Tracepoint in Source Code	325
	Setting Tracepoints	326
	Change Default Message	329
07.26	Import and Export Breakpoints	329
07.27	Run to Cursor	330

07.28	Using the Exception Assistant.....	331
	Exception Object and Description	331
	Troubleshooting Tips	332
	Help Online	332
	Actions	333
	Turning Off the Exception Assistant	334
	Unwind The Call Stack On Unhandled Exceptions	334
07.29	Use a Specific Port for the Development Server (Web Applications) ..	335
07.30	Application and Page Level Tracing.....	336
	Application Level Tracing	336
	Attributes	337
	Trace Details	338
	Page Level Tracing	338
	Combined Tracing	339
	Finally	339
07.31	The Watch Window: Watching and Changing Values.....	340
	Watch Expressions	340
	Watch Window	340
	Creating a Watch Expression.....	341
	Changing Values	342
07.32	Understanding QuickWatch	343
	What Does It Do?	343
	Other Options	344
07.33	The Watch Window: Visualizers	345
07.34	The Watch Window: Refreshing Data	346
	Refresh Icons	347
	Turning It Off	348
07.35	The Watch Window: Adding Watches from Variable Windows	348
	Locals Window	348
	Autos Window	348
	QuickWatch	349
	Watch [1, 2, 3, 4] Window	349
	Keyboard Mapping.....	349

07.36	Create Folders in Class View	350
	Create a New Folder	351
	Putting Items into Your Folder	352
	Removing Items from Folders.....	352
	Creating Subfolders	352
	Deleting Folders	352
07.37	Search in Class View	353
	View.ClassViewSearch Command.....	354
	Use a Previous Search.....	354
	Clear Your Search	355
07.38	Synchronize Your Class View.....	355
07.39	The Misnamed and Misunderstood Object Browser.....	356
07.40	The Object Browser: Setting the Browsing Scope.....	358
	Browse	359
	Edit Custom Component Set	360
07.41	The Object Browser: Navigation and References	362
	Navigation	363
	References	363
07.42	The Exceptions Dialog Box	365
07.43	Setting a Breakpoint in the Call Stack Window	367
07.44	Setting a Tracepoint in the Call Stack Window	369
07.45	Using the WPF Tree Visualizer.....	371
07.46	Understanding Break All Processes When One Process Breaks	374
07.47	Changing Context in the Locals Window	376
	Debug Location Toolbar	377
	Process	377
	Thread.....	377
	Stack Frame	377
07.48	Understanding the Autos Window.....	378
	Changing Values	379
	Current and Previous Statement	379
	VB Shows Three Statements on Either Side	381

Part II Extensions for Visual Studio

8	Visual Studio Extensions	385
	Introducing Visual Studio Extensions	385
	Installing an Extension	385
	Installing from the Extension Manager	386
	Installing from the Visual Studio Gallery	386
	Installing Through Xcopy	387
	Inside a .vsix File	388
	Disabling an Extension	388
	Uninstalling an Extension	389
	Resources for Developing Extensions	389
08.01	Create Themes Using All Visual Studio Elements	389
	Visual Studio Color Theme Editor	389
	To Use	390
	To Customize	390
	More Information	391
08.02	Insert Images into Your Code	391
	Image Insertion	392
	To Use	393
	To Save	393
	To Customize	393
08.03	Add Visual Guidelines to Your Code	394
	Editor Guidelines	394
	To Install	394
	To Use	395
	To Customize	395
08.04	Get More IntelliSense in Your XAML Editor	395
	XAML IntelliSense Presenter	395
	To Use	396
	For More Information	396
08.05	Sync the Solution Explorer to the Current File	396
	Solution Explorer Tools	397
	To Use	397
08.06	Add PowerCommands Options to the IDE	398
	PowerCommands for Visual Studio 2010	398
	To Use	398
	For More Information	400

08.07	Use Emacs Commands in the Editor	400
	Emacs Emulation	400
	To Use	400
	To Uninstall	401
	More Information	401
08.08	Submit to “The Daily WTF”	401
	Share Bad Code with the World	401
	To Install	402
	To Use	402
	More Information	402
08.09	Diff Files Using the Editor	402
	CodeCompare	402
	To Install	403
	To Use	403
	Features	403
	To Uninstall	404
	More Information	404
08.10	Run Windows PowerShell Within the IDE	404
	To Use	404
	More Information	405
08.11	Visualize OData in a Graphical View	405
	Open Data Protocol Visualizer	405
	To Use	405
	More Information	406
08.12	Run VIM Commands in the Editor	406
	To Use	406
	More Information	406
08.13	Check Spelling in Your Code	407
	Spell Checker	407
	To Use	407
08.14	Zoom Across All Files	407
	Presentation Zoom	408
	To Use	408
	Control Zooming with a Slider Using the ZoomEditorMargin Extension	408

08.15	View Code Blocks Using Vertical Lines	409
	StructureAdornment	409
	To Use	409
	To Customize	410
	To Uninstall	410
08.16	Get a Bird's-Eye View of Your Code in an Editor Margin	411
	AllMargins	411
	To Use	411
	To Uninstall	412
08.17	Build Projects from the Windows 7 Taskbar	413
	Win7 Taskbar Extension	413
	To Install	413
	To Use	413
	To Uninstall	414
08.18	Triple-Click to Select an Entire Line	414
	Triple Click	414
	More Information	414
08.19	Create Regular Expressions Within Your Code	415
	Regex Editor	415
	To Use	415
	More Information	415
08.20	Get More Productivity Tools in the IDE	416
	To Use	416
	To Customize the Document Tab Well User Interface	417
08.21	Create and Find Code Snippets	418
	Snippet Designer	418
	To Use	419
	More Information	419
08.22	Document Your Code with Three Keystrokes	419
	GhostDoc	419
	To Use	419
	To Customize	421
	More Information	421

- 08.23 Customize Visual Studio Using Windows PowerShell..... 421
 - StudioShell 422
 - To Use 422
 - To Get Help 423
 - To Customize 423
 - More Information 423
- A Visual Studio Keyboard Shortcut Posters 425**
- Index 427

Foreword

Visual Studio is quite possibly the most powerful and comprehensive software development suite available. No matter your discipline—developer, test, architect, etc.—Visual Studio provides the tools you need to help get your job done.

However, Visual Studio handles such a large number of development tasks for so many platforms that learning it can be an overwhelming process. Many learn just enough to solve the problem at hand but don't delve deep enough to unearth the gems that enable real productivity.

Visual Studio contains numerous features and options that can help you perform your tasks more effectively. Some are prominently advertised, but many of the real time-savers are buried in obscure dialog boxes and triggered with arcane keyboard shortcuts. While most of this information can be learned by wading through hundreds of pages of documentation, many of the more powerful features are yet undocumented. How can we possibly navigate the vast forest that is Visual Studio? A guide is needed.

Fortunately, we have two.

For several years, Sara Ford has championed productivity with her highly successful Visual Studio "Tip of the Day" blog. Between July 2007 and December 2008, Sara blogged nearly 400 Visual Studio tips and tricks that were essential for many of us (myself included) to get closer to attaining Visual Studio mastery. After Sara completed her journey with Visual Studio, Zain Naboulsi picked up the torch. Zain continues to blog Visual Studio tips and tricks, digging into features in the latest releases and covering some of the popular Visual Studio extensions that are available.

I vividly remember my first encounter with Zain. I had just joined Microsoft as a Program Manager on the Visual Basic and C# IDE experiences and received an email from Zain saying that he was taking the mantle from Sara and starting a Visual Studio tips and tricks blog. My first thought was, who is this guy? Sara was an alumnus of the Visual Studio team and had a great deal of "inside knowledge" to share. How could Zain go to the same level of depth that she had? I wasn't prepared for my initial impression to be shattered so thoroughly.

It didn't take long for me to realize that Zain really knows his stuff. After that first email, Zain kept in touch regularly with me and other members of the Visual Studio team. As he systematically pulled away the layers of Visual Studio to find the golden nuggets of productivity beneath, he would ask questions or confirm the tips that he found. Often, Zain would find features that I didn't even know existed. In some cases, he even found bugs where something had been unintentionally left in the product (e.g., the infamous `Debug.cleartextonfoo` command).

What you hold in your hands is the crème de la crème of the sum of Zain and Sara's Visual Studio knowledge. In these pages, you will find a sure compass to help navigate the treacherous peaks and vast oceans of Visual Studio. By putting these tips, tricks, and techniques into practice, you'll grow closer to attaining Visual Studio mastery and learn to travel in style.

*Dustin Campbell
Program Manager, Visual Studio*

It's hard to imagine but if Sara Ford had her way, there never would have been a "Visual Studio Tip of the Day" blog. Back in 2005, we were colleagues on the Visual Studio Editor team who shared an office and a passion for making developers more productive. We both became intrigued by an email with customer feedback, which was remarkable because all of its suggested features were already in Visual Studio but the customer hadn't discovered them. We realized that many of the great productivity features that we developed in Visual Studio 2005 such as Code Snippets and Smart Tags would go unnoticed by many developers who weren't looking for them. We brainstormed several different ways that we could help customers discover all of the hidden functionality in Visual Studio until we arrived at the ultimate solution: Putting a Visual Studio Tip of the Day on the Start Page.

At the time, the Start Page was being completely rewritten to include an RSS feed which would be the perfect mechanism for users to learn how to use Visual Studio better one tip at a time. We lobbied hard to have the Start Page point to a feed of Tips & Tricks for Visual Studio. Unfortunately, there was too much skepticism that there were enough tips to generate new content every day and so the idea was rejected.

Defeated but undeterred, Sara was determined to demonstrate that not only were there enough hidden gems but that there was a huge audience for a "Tip of the Day." She had recently started blogging and challenged herself to blog every workday until she ran out of tips. Initially, we wrote a list of about 50 different tips before she set out on her challenge. Over the following months and years, Sara has worked tirelessly to find hundreds of useful tips, created an engaging blog and helped thousands of developers become better users of Visual Studio.

As we were putting the finishing touches on Visual Studio 2010 last year, I was reassured to find that Zain Naboulsi had stepped in to create the next generation of the Visual Studio Tips and Tricks blogs. He's engaged directly with the Visual Studio team to highlight each of the new features that were introduced in the latest release and the greatest from previous versions.

Based on their years of experience evangelizing Visual Studio Tips and Tricks, “Coding Faster” distills this knowledge into one easy-to-read book which will make you a better user of Visual Studio. Using the described techniques, you’ll learn to write code with fewer keystrokes, manage projects and documents with ease and powerful debugging techniques. It also introduces Visual Studio Extensibility which allows you to create your own extensions or find those provided by the community.

In retrospect, it’s fortunate that the tip of the day didn’t make it onto the Start Page as it led Sara and Zain to write great blogs and this brilliant book to help everyone code faster.

Sean Laberee
Lead Program Manager, Visual Studio

Introduction

Visual Studio is sexy. In the world of Integrated Development Environments (IDEs), it stands as a beautiful example of how environments should work. Yet many of the features created to improve productivity, I believe, are largely neglected. Most developers use only a small percentage of the capabilities in this wonderful product—not because they don't want to use them, but because developers don't know they exist.

In most books that address the various .NET languages or technologies, Visual Studio seems to be mentioned almost as an afterthought; to be fair, its focus is primarily on the language or technology that's the subject of the book, not the IDE—which is as it should be. On the other side of the coin, books written about Visual Studio do focus on the product, but tend to be broad in scope, describing features, but without saying much about their actual use.

The goal of this book is to arm you with techniques that you can apply immediately to improve productivity. Use the content in this book anywhere, anytime, to dramatically reduce the time required to perform just about any task in Visual Studio. You won't find an exhaustive treatment of every feature in Visual Studio in this book, but it contains sufficient coverage that we're sure you'll find something useful, regardless of how you use the product.

This is much more than just a tips and tricks book. Within these pages are—for the first time ever—the keyboard mapping shortcuts, commands, and menu paths for features, along with detailed descriptions of how to use them. We worked very hard to present the information in a way that makes the book easy to read cover-to-cover or as a quick reference.

Who Should Read This Book

If you use Visual Studio 2005, 2008, or 2010, you should read this book. There are over 365 tips in this book (including the additional online Appendix), all selected with the single goal of helping you be more productive by showing you how to use Visual Studio features. The contents in this work are great on their own or as a perfect complement to any course, book, or other learning tool as you explore Visual Studio.

Assumptions

This book assumes you have, at a minimum, Visual Studio Professional 2005, 2008, or 2010 installed. Specifically, it covers techniques that can be used in Visual Studio as well as examples in C++, C#, and/or VB where appropriate.

With a heavy focus on helping you get work done faster, we assume that you have a basic understanding of how to use Visual Studio, and have had exposure to one of the many languages supported in the product.

Organization of This Book

This book is divided into two sections. Part I, written by me, Zain Naboulsi, called “Productivity Techniques,” provides information that can be used in your daily work with Visual Studio; these techniques range from very easy to quite advanced methods of using the product. The chapters are organized to take you through the most common daily tasks you perform. Within each chapter the information is arranged, essentially in order, from beginning to more advanced optimizations:

- Chapter 1, “Getting Started,” shows key skills to have when starting up and using Visual Studio.
- Chapter 2, “Projects and Items,” shows ways to create and use projects and items more effectively.
- Chapter 3, “Getting to Know the Environment,” gives guidance on how to organize the environment to best advantage.
- Chapter 4, “Working with Documents,” illustrates how best to navigate and manipulate document windows.
- Chapter 5, “Finding Things,” is a collection of tips on how to find just about anything in your code or code related in Visual Studio.
- Chapter 6, “Writing Code,” the largest chapter, shows a host of techniques for using the editor more efficiently.
- Chapter 7, “Debugging,” the second largest chapter, shows you great techniques to improve your debugging experience within the IDE.

Part II, “Extensions for Visual Studio,” written by Sara Ford and other authors, contains an examination of selected extensions from the Visual Studio Gallery that you can install to further accelerate the Visual Studio experience.

Finally, we have included an entire second book of tips in Appendix B (downloadable at <http://www.microsoftpressstore.com/title/9780735649927>) that were cut from the main book so that we could keep the print size manageable. You are literally getting two books for the price of one.

Finding Your Best Starting Point in This Book

Each chapter—and in fact, almost every item—in Part I of this book stands on its own, so feel free to begin reading wherever you like. However, I suggest that you start by choosing the tasks that will have the greatest impact in your daily work. If most of your day is spent debugging, then start with Chapter 7, “Debugging,” first. After you have a good handle on using the features built-in to Visual Studio from Part I of this book, look in Part II to see if there’s an extension that can help you advance your goals even further. Visual Studio has many extensions that can improve your overall experience.

Conventions and Features in This Book

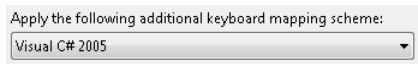
This book presents information using conventions designed to make the information readable and easy to use.

Keyboard Settings

Throughout this book I refer to the keyboard settings often, so it is important to know the connection between development settings and keyboard mapping schemes. When first installed, Visual Studio asks you to choose a collection of settings, as shown here:



The settings chosen are directly related to Tools | Options | Environment | Keyboard within the “Apply the Following Additional Keyboard Mapping Scheme” dropdown list. For example, when you choose Visual C# Development Settings, you will see Visual C# 2005 as the keyboard mapping, as shown here:



The following table lists the setting collection and its corresponding keyboard mapping scheme:

SETTINGS	ADDITIONAL KEYBOARD SCHEME
GENERAL DEVELOPMENT SETTINGS	(Default)
PROJECT MANAGEMENT SETTINGS	(Default)
VISUAL BASIC DEVELOPMENT SETTINGS	Visual Basic 6
VISUAL C# DEVELOPMENT SETTINGS	Visual C# 2005
VISUAL C++ DEVELOPMENT SETTINGS	Visual C++ 6
VISUAL F# DEVELOPMENT SETTINGS	(Default)
WEB DEVELOPMENT	(Default)
WEB DEVELOPMENT (CODE ONLY)	(Default)

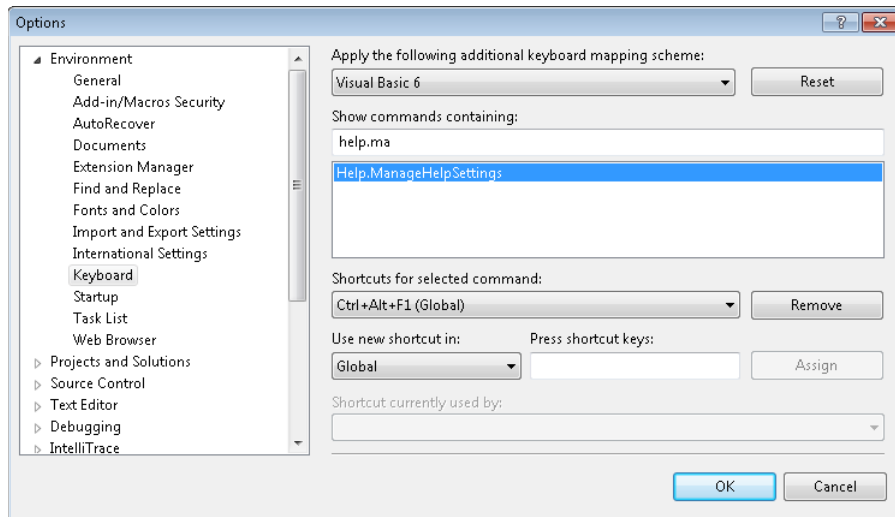
Summary Information

One of the unique features of this book is the summary information at the top of every item. All tips will contain a table with one or more pieces of summary information that looks like this example:

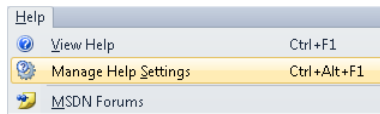
DEFAULT	Ctrl+Alt+F1 (help settings); F1 (view help)
VISUAL BASIC 6	Ctrl+Alt+F1 (help settings); F1 (view help)
VISUAL C# 2005	Ctrl+F1, M (help settings); Ctrl+F1, Ctrl+M (help settings); F1 (view help)
VISUAL C++ 2	Ctrl+Alt+F1 (help settings); F1 (view help)
VISUAL C++ 6	Ctrl+Alt+F1 (help settings); F1 (view help)
VISUAL STUDIO 6	Ctrl+Alt+F1 (help settings); F1 (view help)
WINDOWS	Alt, H, S (help settings); Alt, H, V (view help)
MENU	Help Manage Help Settings; Help View Help
COMMAND	Help.ManageHelpSettings; Help.F1Help
VERSIONS	2010
LANGUAGES	All
CODE	vstipTool0120

Here is what each piece of information means:

- Default to Visual Studio 6**—Keyboard shortcuts assigned that are mapped to the choice made for development settings. These settings can be found at Tools | Options | Environment | Keyboard. If absent, implies there are no keyboard shortcuts that apply.



- **Windows**—Keyboard shortcuts that navigate the Menu Bar for commands. If absent, implies there are no Menu Bar shortcuts that apply.



- **Menu**—Menu Bar path for using a command. Help | Manage Help Settings means click on the Help menu and choose Manage Help Settings item underneath it. If absent, implies the item can not be accessed from the Menu Bar.
- **Command**—Visual Studio command used to assign keyboard shortcuts, aliases, run macros, etc. Found at Tools | Options | Environment | Keyboard. If absent, implies there is no command available for this activity.
- **Versions**—Versions of Visual Studio that support the information given in the tip. If absent, implies all versions are supported.
- **Languages**—Languages supported (C++, C#, and/or VB). If absent, implies all languages are supported.
- **Code**—Unique identifier for each tip for looking up references to the tip in the book, online, etc.

Additional Information

- Boxed elements with labels such as “Warning” are used to tell you about items that may impact you negatively. Be aware that these activities are done at your own risk.
- Boxed elements with labels such as “Note” provide additional information or alternative methods for completing a step successfully.
- Text that you type (apart from code blocks) appears in bold.
- A plus sign (+) between two key names means that you must press those keys at the same time. For example, “Ctrl+Alt+L” means that you hold down the Ctrl key while you press the Alt key and the L key.
- A comma (,) between key names means you press each key separately. For example, “Alt+T, O” means you hold down Alt while you press T then let up on the keys and finally press O by itself.
- A vertical bar between two or more menu items (e.g., File | Close), means that you should select the first menu or menu item, then the next, and so on.

System Requirements

You will need the following hardware and software to complete the practice exercises in this book.

Software Requirements

- Windows XP (x86) with Service Pack 3—all editions except Starter Edition
- Windows Vista (x86 & x64) with Service Pack 2—all editions except Starter Edition
- Windows 7 (x86 & x64)
- Windows Server 2003 (x86 & x64) with Service Pack 2
- Windows Server 2003 R2 (x86 & x64)
- Windows Server 2008 (x86 & x64) with Service Pack 2
- Windows Server 2008 R2 (x64)

Supported Architectures:

- 32-Bit (x86)
- 64-Bit (x64)

Hardware Requirements

- Computer that has a 1.6GHz or faster processor
- 1 GB (32-Bit) or 2 GB (64-Bit) RAM (Add 512 MB if running in a virtual machine)
- 3 GB of available hard disk space
- 5400 RPM hard disk drive
- DirectX 9 capable video card running at 1024 x 768 or higher-resolution display
- DVD-ROM Drive

Depending on your Windows configuration, you might require Local Administrator rights to install or configure Visual Studio.

Acknowledgments

From day one this book has been a community-driven effort. The readers of my (Zain) and Sara's blogs have been a constant source of content, comments, and ideas. Our heartfelt thanks for all our readers have done to make this book a reality.

I used to make fun of those people who win awards on TV because they always have a huge list of people to thank and never seem to get through them. It looks like it's my turn now and I *know* that I will forget someone, so let me just say that behind every effort like this you will always have a great deal of people helping you in one way or another. Below is just a partial list of people Sara and I want to thank for contributing, directly or indirectly, to the effort:

Russell Jones and Adam Zaremba—Editors at O'Reilly Media, who herded the cats to make this book happen.

Kevin Stevens—Who came up with the name of the book and was instrumental in the technical review process.

Paul Millsaps, Bill Needels—For doing some of the technical review for the book.

Sean Laberee—Senior Program Manager Lead at Microsoft who helped both Sara and me get started with Tips and Tricks.

Dustin Campbell—Program Manager at Microsoft who continues to be a constant source of information when I get stuck on a feature or concept.

Brittany Behrens—Program Manager at Microsoft who helped me during those first tenuous days after I took over Sara's work.

Matt Manela—for writing the content for the Snippet Designer extension.

Andrew Steele—for writing the content for the Productivity Power Tools extension.

Jim Christopher—for writing the content for both the GhostDoc and the StudioShell extensions.

Terry Leeper—Principal Architect, Windows C++ Team, my main contact with the C++ folks and a good friend that has helped me resolve questions about features since I started doing the tips.

Lisa Feigenbaum and Beth Massi—Program Managers at Microsoft who constantly provided guidance and support as the content of the book evolved.

Brian Moore—Director, DPE Central Region, for providing support and being a great manager.

Clint Edmonson—Senior Architect Evangelist at Microsoft who I have toured with throughout the country delivering Visual Studio talks to thousands of people.

Phil Wheat—My best friend at Microsoft and a constant source of information. Phil is easily the smartest person I know.

Jared Bienz, Mike Azocar—Very good friends who have been great to bounce ideas around as I worked on the book.

Rob Bagby, Mike Benkovich, John Weston, Keith Combs—My old Microsoft Across America buddies.

Finally, Sara Ford wishes to thank Dr. Terrance Delaney and Dr. Michael McMurray for fixing her chronic shin splints during the course of writing this book.

Errata & Book Support

We've made every effort to ensure the accuracy of this book and its companion content. Any errors that have been reported since this book was published are listed on our Microsoft Press site:

<http://www.microsoftpressstore.com/title/9780735649927>

If you find an error that is not already listed, you can report it to us through the same page.

If you need additional support, email Microsoft Press Book Support at *mspinput@microsoft.com*.

Please note that product support for Microsoft software is not offered through the addresses above.

We Want to Hear from You

At Microsoft Press, your satisfaction is our top priority, and your feedback our most valuable asset. Please tell us what you think of this book at:

<http://www.microsoft.com/learning/booksurvey>

The survey is short, and we read every one of your comments and ideas. Thanks in advance for your input!

Stay in Touch

Let's keep the conversation going! We're on Twitter: *<http://twitter.com/MicrosoftPress>*

Chapter 1

Getting Started

"A beginning is the time for taking the most delicate care [...]"

—Frank Herbert, "Dune"

This chapter addresses tasks that would be immediately beneficial as you work in Visual Studio. The main themes here are exporting your development settings, learning the Start Page, adjusting your performance, and other key tasks.

This chapter is arguably the most important one you will read in this book—and yet, I suspect, the one people will think they need the least. If you have been using Visual Studio for any length of time, you might easily feel that the tasks in in this chapter have little application to your situation. But whether you have been using Visual Studio for ten days or ten years, these tips will help ensure that all your other efforts go smoothly, so taking time to absorb the contents is definitely worthwhile.

01.01 Running Multiple Versions of Visual Studio Side-By-Side

VERSIONS	2005, 2008, 2010
CODE	vstipEnv0054

People often ask whether you can run multiple versions of Visual Studio side-by-side on the same machine. The answer is yes you can!

You can find documentation on MSDN, in the topic "Installing Visual Studio Versions Side-by-Side," at <http://msdn.microsoft.com/en-us/library/ms246609.aspx>.

The recommendation is that you install multiple versions from oldest to newest. So you would install Visual Studio 2005, 2008, and then 2010—in that order.

01.02 Getting Table of Contents in Visual Studio 2010 Online Help

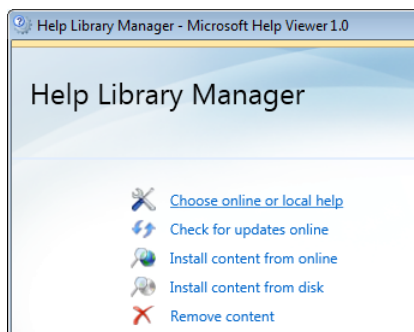
DEFAULT	Ctrl+Alt+F1 (help settings); F1 (view help)
VISUAL BASIC 6	Ctrl+Alt+F1 (help settings); F1 (view help)
VISUAL C# 2005	Ctrl+F1, M (help settings); Ctrl+F1, Ctrl+M (help settings); F1 (view help)
VISUAL C++ 2	Ctrl+Alt+F1 (help settings); F1 (view help)
VISUAL C++ 6	Ctrl+Alt+F1 (help settings); F1 (view help)
VISUAL STUDIO 6	Ctrl+Alt+F1 (help settings); F1 (view help)
WINDOWS	Alt, H, S (help settings); Alt, H, V (view help)
WINDOWS KEYBOARD	Alt, H, S (help settings); Alt, H, V (view help)
MENU	Help Manage Help Settings; Help View Help
COMMAND	Help.ManageHelpSettings; Help.F1Help
VERSIONS	2010
CODE	vstipTool0120

I have to admit I don't like the new online help in Visual Studio 2010. Not that I think it's bad per se, but I was just used to the old help system's look and feel—particularly the table of contents list.

If you are like me and want to get that classic help look-and-feel back for online help, you need to do two things.

Online Help

First, you need to set your default help to online help (you need Internet connectivity to use this feature) by selecting Help | Manage Help Settings. Then click Choose Online Or Local Help.

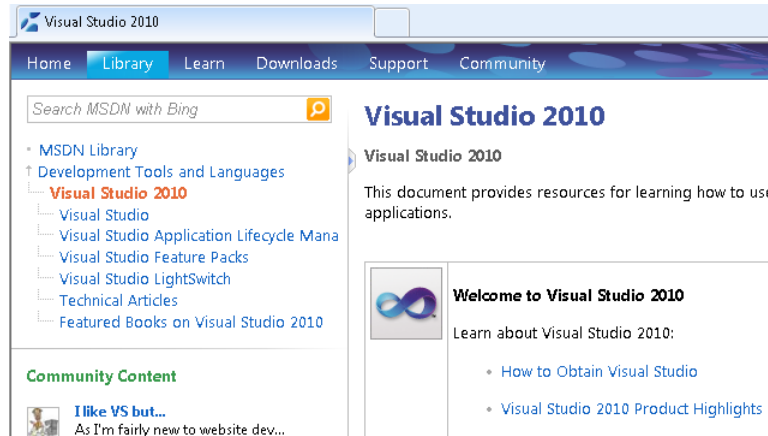


Select I Want To Use Online Help, and click OK.

- I want to use online help
- I want to use local help

Using Classic View

Now that you are using online help, Go to Help | View Help to see a page similar to the following:



In the upper-right corner of the page, if you see the Preferences link, click it.



Note You may not see the Preferences link but instead just three links to Lightweight, ScriptFree, and Classic. In this case, just click Classic and skip the next step.

Choose Classic and click OK.

Choose View

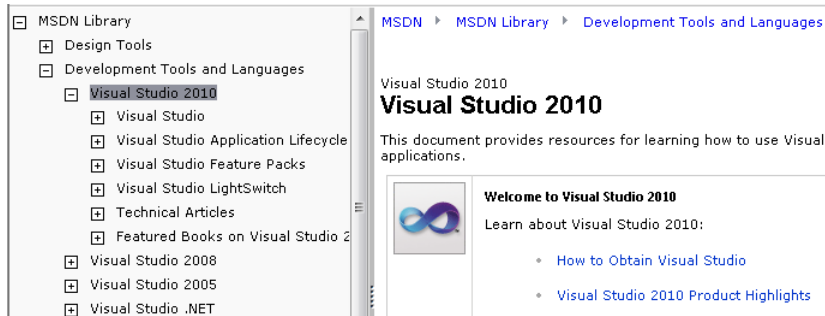


Choose between a Lightweight experience that balances both richness and speed, a fast ScriptFree experience, or a feature-rich Classic experience.

- Lightweight**
A balance between a fast and rich experience.
- ScriptFree**
Super-fast with no script.
- Classic**
Classic. Feature rich view. It includes a side-by-side view of translated and English for some content.



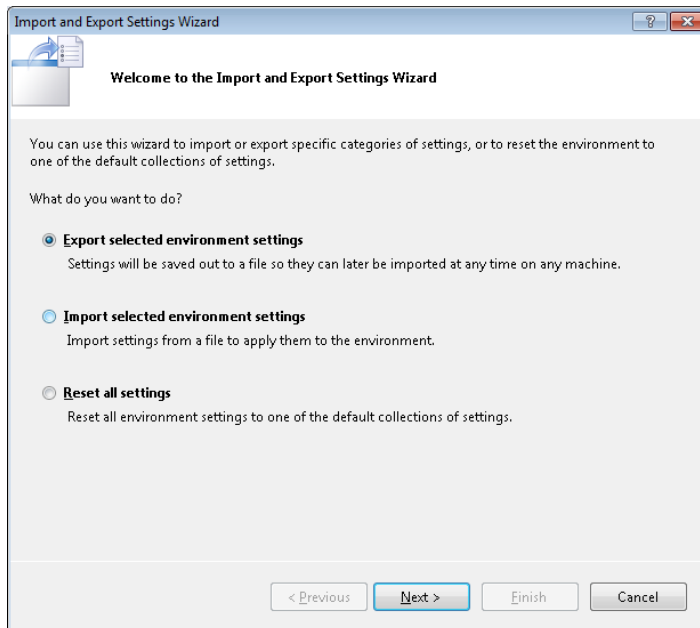
Now your help will use the old-style contents list.



01.03 Exporting Your Environment Settings

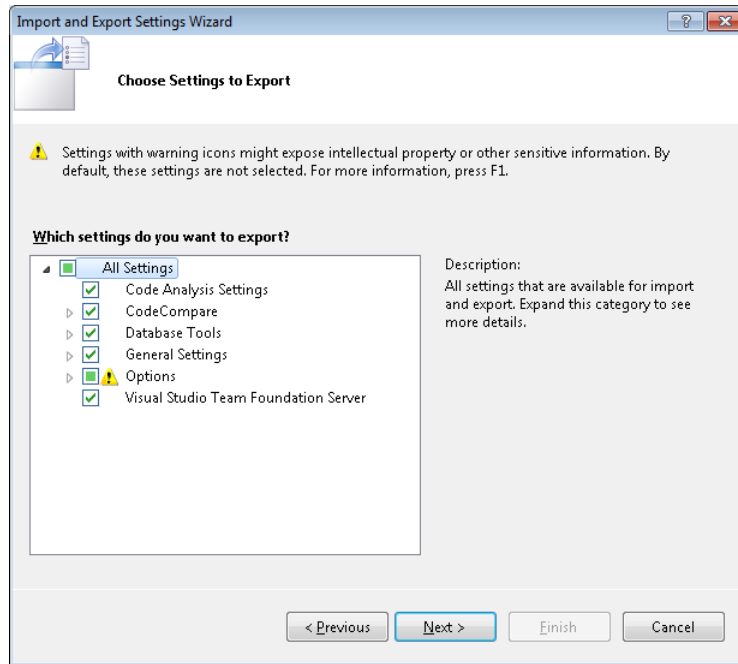
WINDOWS	Alt,T, I
MENU	Tools Import and Export Settings
COMMAND	Tools.ImportandExportSettings
VERSIONS	2005, 2008, 2010
CODE	vstipEnv0021

Exporting your environment settings is a great way to back them up. You can export your settings by selecting Tools | Import And Export Settings Wizard.

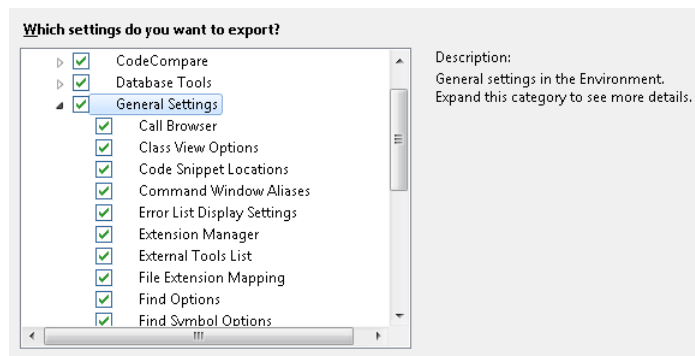


The Export Selected Environment Settings option lets you save your settings to a .vssettings file.

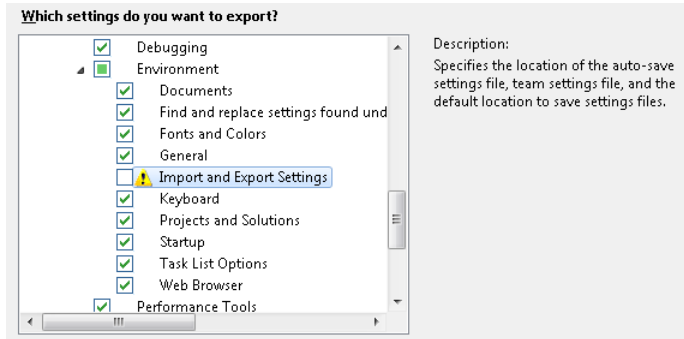
Click Next to see the Choose Settings To Export dialog box.



You can expand the areas to choose the items you want to include or exclude.



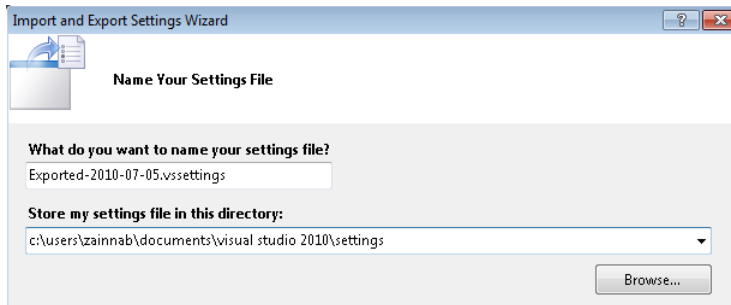
By default, almost everything is selected except for items that could expose sensitive information. You can tell which options these are by the yellow warning symbol icon next to the item.



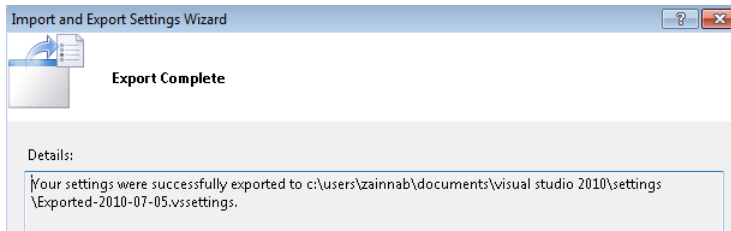
After you make your choices and click Next, you can pick the location (C:\users\\documents\visual studio <version>\settings, by default) and the filename (the current date, by default) where you want to save the exported information.



Warning If you don't give your exported settings good names it will be hard to figure out what they are for later. For example, if you are just exporting your favorite black theme fonts and colors, a name like "Fonts and Colors (Black Theme) 2010-07-05" would make sense.



When you click Finish, Visual Studio exports your settings, and the following dialog box appears.



If you are curious, the exported file is just an XML file. You can open it in Notepad and see the contents, as shown in the following illustration.

```

<UserSettings>
  <ApplicationIdentity version="10.0"/>
  <ToolOptions>
    <ToolOptionsCategory name="Environment"
RegisteredName="Environment">
      <ToolOptionsSubCategory name="documents"
RegisteredName="documents" PackageName="Visual Studio Environment Package">
        <Propertyvalue
name="ShowMiscFilesProject">false</Propertyvalue>
        <Propertyvalue
name="AutoLoadExternalChanges">false</Propertyvalue>
        <Propertyvalue
name="CheckForConsistentLineEndings">true</Propertyvalue>
        <Propertyvalue
name="SaveDocsAsUnicodeWhenDataLoss">false</Propertyvalue>
        <Propertyvalue
name="InitializeOpenFileFromCurrentDocument">true</Propertyvalue>
        <Propertyvalue
name="ReuseSavedActiveDocWindow">false</Propertyvalue>
        <Propertyvalue
name="DetectFileChangesOutsideIDE">true</Propertyvalue>
        <Propertyvalue
name="DontShowGlobalUndoChangeLossDialog">true</Propertyvalue>
        <Propertyvalue
name="AllowEditingReadOnlyFiles">true</Propertyvalue>
        <Propertyvalue
name="DocumentDockPreference">1</Propertyvalue>
        <Propertyvalue
name="MiscFilesProjectsSavesLastNItems">0</Propertyvalue>
      </ToolOptionsSubCategory>
    <ToolOptionsSubCategory name="FindAndReplace"
RegisteredName="FindAndReplace" PackageName="Visual Studio Environment Package">
      <Propertyvalue
name="ShowWarningMessages">true</Propertyvalue>
    </ToolOptionsSubCategory>
  </ToolOptions>
</UserSettings>

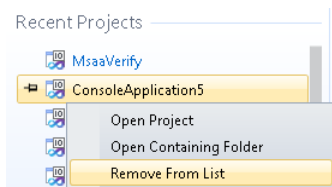
```

10.04 Remove Projects from the Recent Projects List

WINDOWS	Alt,F, J, [Number]
MENU	File Recent Projects and Solutions
VERSIONS	2010
CODE	vstipTool0017

In Visual Studio 2010, you can now remove projects from the Recent Projects list on your Start Page.

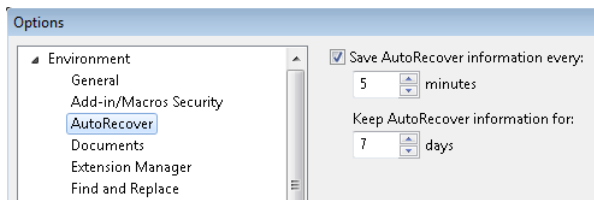
Just right-click the project, and select Remove From List, as shown in the following illustration. That's it. The project is removed from the list but not deleted. If you want to permanently delete the project, you need to do that yourself from the filesystem.



01.05 AutoRecover

WINDOWS	Alt,T, O
MENU	Tools Options Environment AutoRecover
COMMAND	Tools.Options
VERSIONS	2005, 2008, 2010
CODE	vstipEnv0019

AutoRecover can be a real life saver if the development environment crashes or if a power outage occurs. It's simple to use: Just go to Tools | Options | Environment | AutoRecover.

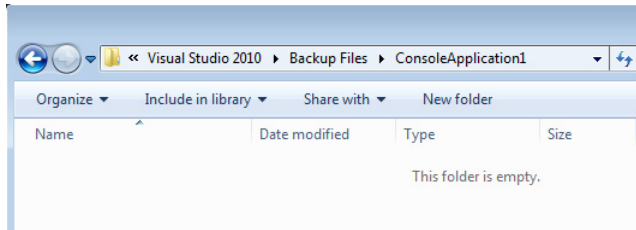


To turn this feature off (not recommended), you can clear the Save AutoRecover Information Every check box. Here's an explanation of what the other options do:

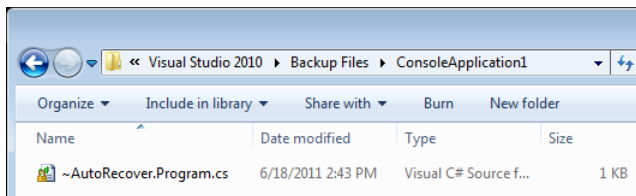
- n minutes** Determines how often Visual Studio saves AutoRecover information for files. The default is to save every five minutes, but you can adjust that interval up or down depending on your needs. There is an inverse relationship between this value and the frequency of your updates to code. If you make frequent code updates, you should set a smaller save interval. Conversely, if you make relatively infrequent code updates, you can increase this interval. It's better to err by using an interval that's too short rather than too long; in other words, it's better to take a performance hit from file I/O than to lose a ton of work.
- n days** Determines how long Visual Studio keeps AutoRecover files in the Backup Files directory. The default is seven days, which is adequate for most situations. If you work with a lot of projects over a short period of time, you might want to decrease this number to keep the Backup Files directory from getting too cluttered. If you're not sure about what you need for this value, it's better to guess high and later reduce the number as needed.

I want to be clear about what exactly gets saved and where it gets saved. First, recovered files are stored at My Documents\Visual Studio <version>\Backup Files\<projectname>. But not every file is saved here. The backup folder is empty when you first create a solution in Visual Studio.

When I make a change to a file and save the change, I wait five minutes to see the result.

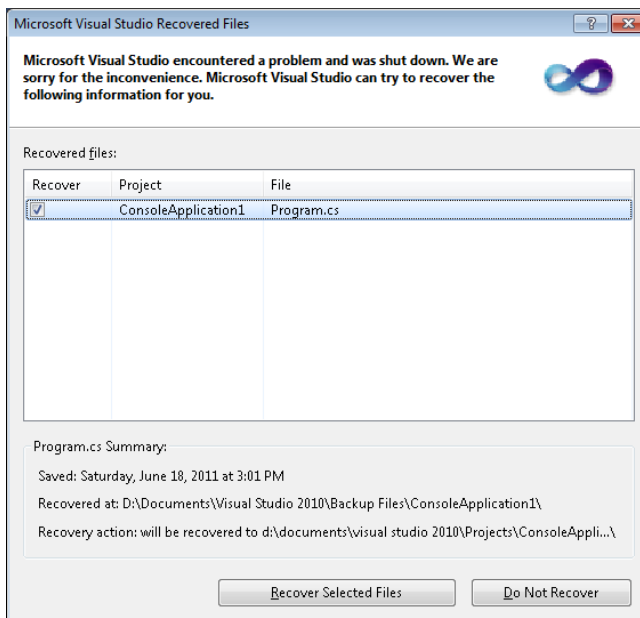


There's still nothing there, because Visual Studio knows there is no need to recover a saved file. But if I make a change to a file without saving it and wait another five minutes, here is what you see:



At this point, the AutoRecover information appears because there are unsaved changes. If Visual Studio crashes now, you would need to make a decision about whether to recover the unsaved changes or keep the last saved version. Giving you that choice is the essential function of the AutoRecover feature.

When you do finally have to recover a file, you will see the following dialog box.



To explain the terminology in the preceding dialog box:

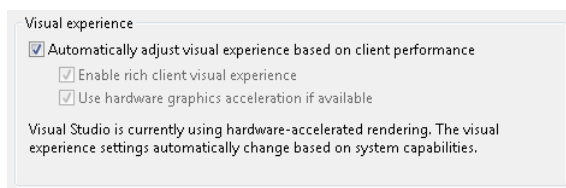
- **Recovered files**
Lists the file(s) that can be recovered. Use a check box to select or clear the files you want to keep as well as to see some basic information.
- **<File Name> Summary**
Shows detailed information about the currently selected file, including date/time information, location of the backup file, and the destination location where the recovered file will be saved.
- **Recover Selected Files**
Performs a recovery action on the selected file(s), copying the recovered source file to the previously indicated destination.
- **Do Not Recover**
Closes the dialog box without recovering any listed files.

01.06 Improving Performance by Changing the Visual Experience in Visual Studio 2010

WINDOWS	Alt,T, O
MENU	Tools Options Environment General Visual Experience
COMMAND	Tools.Options
VERSIONS	2010
CODE	vstipEnv0017

Visual Studio 2010 automatically adjusts the visual experience depending on the situation. For example, it might eliminate or reduce the use of gradients and animations when running in Remote Desktop or virtual machine environments. It also makes use of hardware graphics acceleration when that's available.

In some situations, you can improve Visual Studio's performance by changing its Visual Experience manually. To change these settings, select Tools | Options | Environment | General | Visual Experience to see the following dialog box.



Clear the Automatically Adjust Visual Experience Based On Client Performance check box.



Note As you work with the preceding options, the message at the bottom of the dialog box does not change until you click OK to commit the changes you have made.

Following is a brief explanation of what each option does:

- **Enable Rich Client Visual Experience** This option gives you gradients and animations (also known as “eye candy”) for elements such as sliding tool windows and so on. If you leave this option selected, Visual Studio uses these rich animations in all scenarios—including remote sessions. You should usually turn this option off in such situations to get a bump in performance.
- **Use Hardware Graphics Acceleration If Available** This option lets you decide whether Windows Presentation Foundation (WPF) hardware acceleration is something you want. If this can benefit you, you’ll notice a clear change in performance when you enable or disable this option. Make sure to test both scenarios.



Note If you have a system whose performance doesn’t suffer when animations and gradients are turned on, a little eye candy can be a good thing, so this tip is really for those folks who are having performance issues in their Visual Studio experience, either locally or remotely.

Now that you have played with the preceding options a bit, you might be wondering whether you can actually see how much they can improve (or hurt) performance. The Windows SDK includes a tool called WPFPerf that enables you to measure WPF performance. You can find a great article on how to use it at the Microsoft WindowsClient.NET site, at <http://windowsclient.net/wpf/perf/wpf-perf-tool.aspx>.

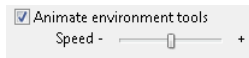
Also, when using Visual Studio 2010 over remote sessions, you should definitely read the article titled “Optimizing Visual Studio 2010 and WPF Applications for Remote Desktop,” at <http://blogs.msdn.com/b/jgoldb/archive/2010/02/27/optimizing-visual-studio-2010-and-wpf-applications-for-remote-desktop.aspx>. This article provides important information about how to dramatically improve performance over Remote Desktop.

01.07 Change Tool Window Animations

WINDOWS	Alt,T, O
MENU	Tools Options Environment General
COMMAND	Tools.Options
VERSIONS	2005, 2008
CODE	vstipEnv0018

In the tip vstipEnv0017 (“Improving Performance by Changing the Visual Experience”, page 14), you saw a discussion of how you can improve the visual performance of Visual Studio 2010. Now let’s look at Visual Studio 2008 and 2005. You can change the animation speed of tool windows in Visual Studio 2008 and 2005, but why would you want to do this?

The answer is that you can get a performance boost by speeding up or completely turning off the animation. Select Tools | Options | Environment | General, and locate the Animate Environment Tools option.

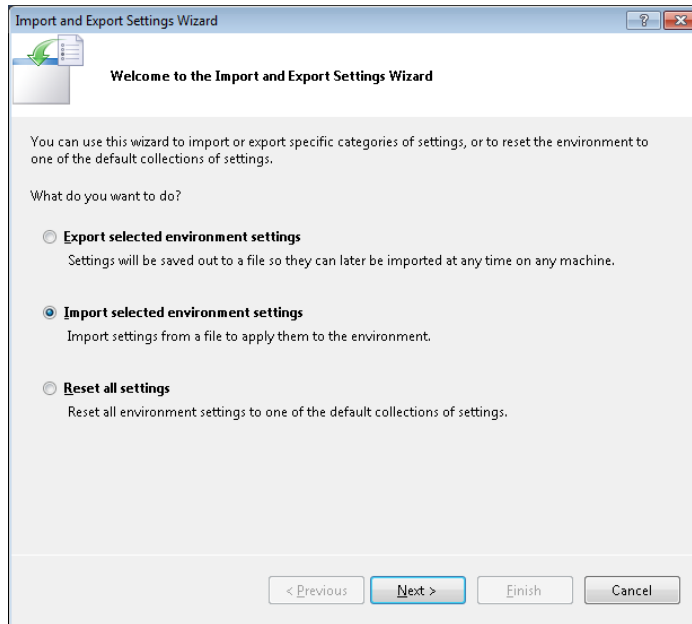


I suggest you turn off this feature to begin with, to see whether you notice any performance improvements. Later, if you want your animations back, turn on the option and set the slider to the far-right side (the fastest speed). As you test the performance, you can adjust it back to the left to determine the best setting for you.

01.08 Importing or Changing Your Environment Settings

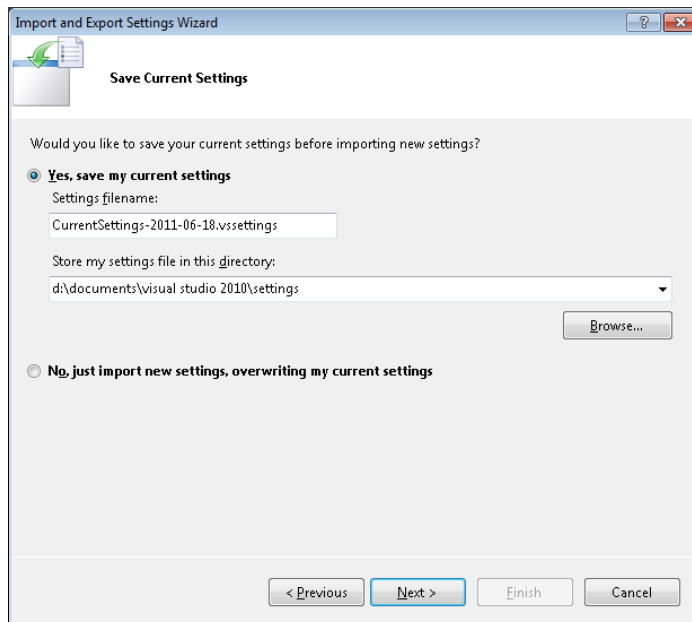
WINDOWS	Alt,T, I
MENU	Tools Import and Export Settings
COMMAND	Tools.ImportandExportSettings
VERSIONS	2005, 2008, 2010
CODE	vstipEnv0022

Assuming you have exported your settings (vstipEnv0021, “Exporting Your Environment Settings”, page 6), you can import your settings by going to Tools | Import And Export Settings Wizard and selecting Import Selected Environment Settings:

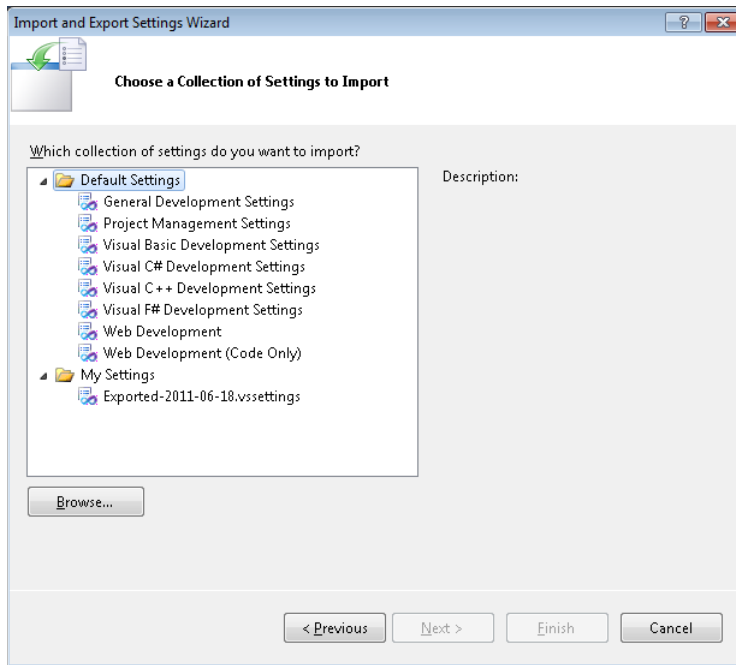


This setting enables you to import a previously exported .vssettings file.

After you click Next on the Welcome page shown in the preceding illustration, you have the option to save your existing settings (recommended) before overwriting or to just overwrite them:



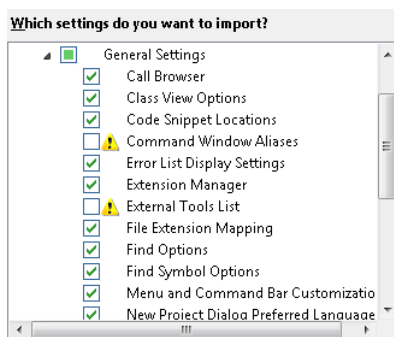
Click Next, and you can choose from the default settings, settings that have been saved previously, or you can browse for your own .vssettings file:



Now click Next again to choose what settings you want to import. All the previously exported settings are selected by default except for Command Window Aliases, External Tools List, and Import and Export Settings, which are considered potentially dangerous.



Warning You have to determine the potentially dangerous areas for yourself, but if doing a full export or import, you would most likely check all the items in this dialog. It is not recommended that you share full exports with team members as there may be information in the file you don't want to share. Instead, just export the items you want to share with team members in a separate file.



After you have checked (or unchecked) the items you want, click Finish to import the settings and to see the final page of the wizard:



Now, just click Finish and you are done.

01.09 Change Your Visual Studio Color Scheme

WINDOWS	Alt,T, I
MENU	Tools Import and Export Settings
COMMAND	Tools.ImportandExportSettings
VERSIONS	2005, 2008, 2010
CODE	vstipEnv0034

Ever see a set of colors your friend or coworker has and wish you could get it too? Ever go to <http://studiostyles.info> and want some of those cool color schemes?

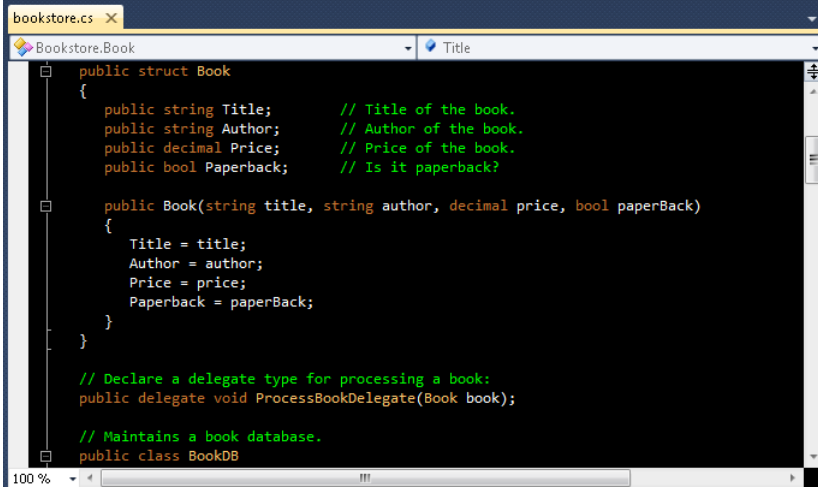
Top-rated schemes

rated 165	58545 dls	95933 views	<pre>public class Settings : IThemeable { static void Main() { if (Theme.Length < (43 A 2)) new Uri("Circles/obsidian.com"); } } // Son of Obsidian</pre>	Son of Obsidian advanced razor 1 year ago, updated 7 months ago einaros
rated 134	62639 dls	112573 views	<pre>public class Settings : IThemeable { static void Main() { if (Theme.Length < (43 A 2)) new Uri("WekeRoad Ink"); } } // WekeRoad Ink</pre>	WekeRoad Ink advanced razor 1 year ago, updated 7 months ago Luke Sampson
rated 74	13450 dls	30879 views	<pre>public class Settings : IThemeable { static void Main() { if (Theme.Length < (43 A 2)) new Uri("Circles/obsidian.com"); } } // Selenitic</pre>	Selenitic advanced razor 1 year ago, updated 4 months ago Tim G. Thomas

Well you can get the colors you want! Let's walk through how it's done.

Seeing What You Like

First, you see a seriously cool color scheme on someone's screen or at the Studio Styles site:



```
public struct Book
{
    public string Title;           // Title of the book.
    public string Author;         // Author of the book.
    public decimal Price;         // Price of the book.
    public bool Paperback;         // Is it paperback?

    public Book(string title, string author, decimal price, bool paperBack)
    {
        Title = title;
        Author = author;
        Price = price;
        Paperback = paperBack;
    }
}

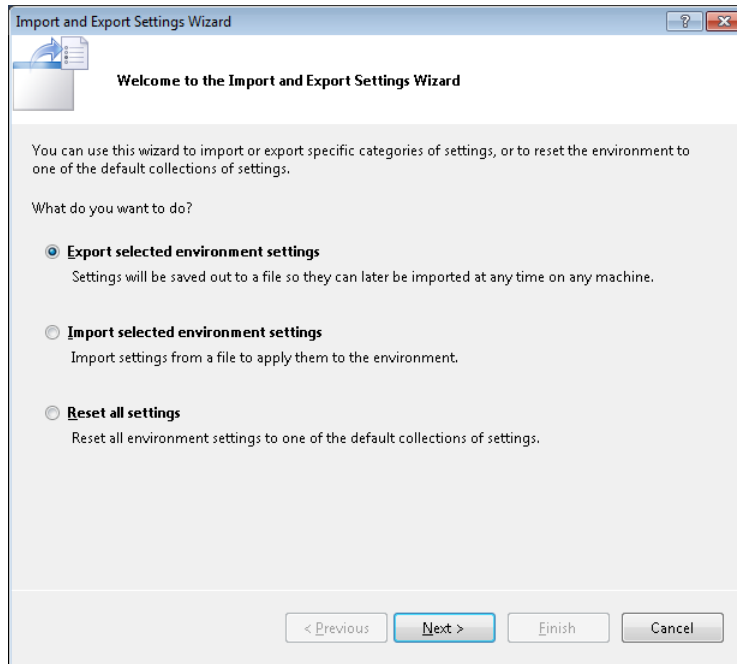
// Declare a delegate type for processing a book:
public delegate void ProcessBookDelegate(Book book);

// Maintains a book database.
public class BookDB
```

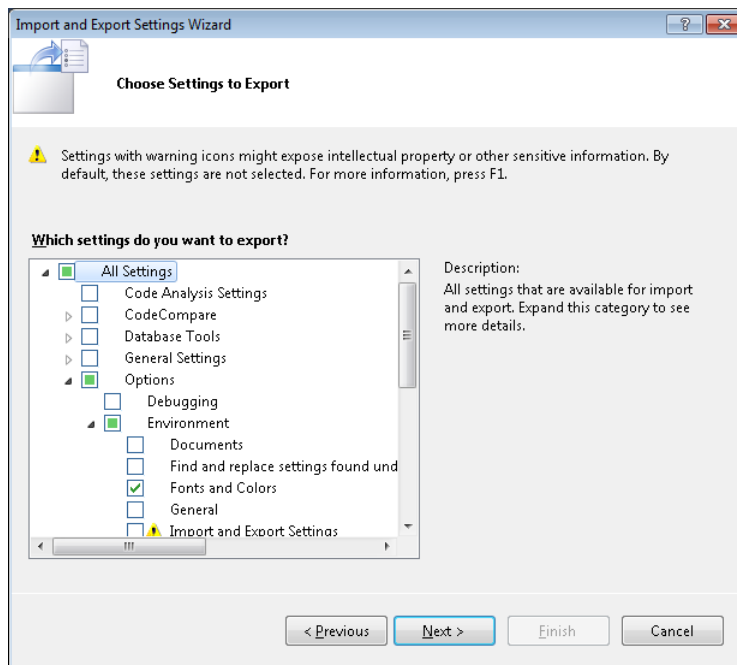
Getting the Goods

On someone's computer

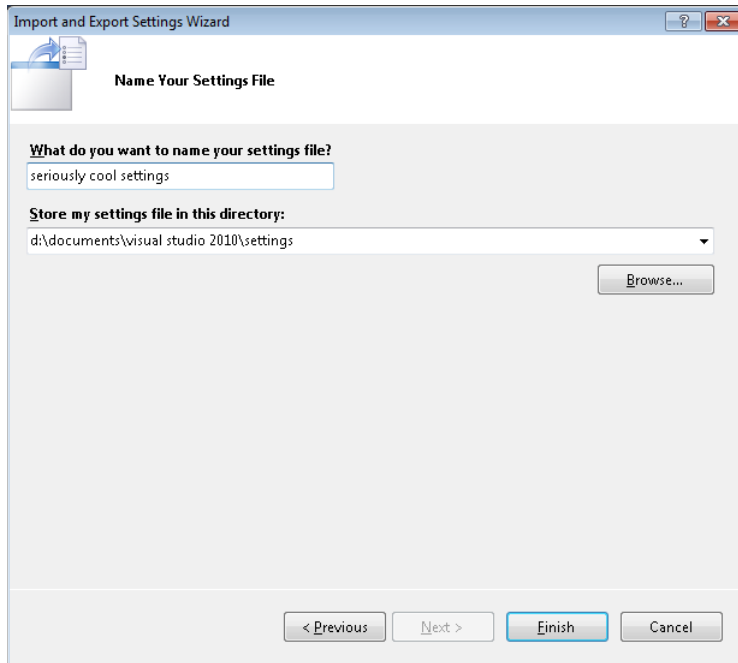
Now that you see what you like, get them to export their fonts and colors. Go to Tools | Import And Export Settings:



Click Next, and export *only* the fonts and colors—*nothing else*:



Click Next, give the settings a cool name, and click Finish:



On the Studio Styles site

Click the style you want:

Top-rated schemes

rated 165	58548 dls	95939 views		Son of Obsidian advanced razor 1 year ago, updated 7 months ago einaros
rated 134	62645 dls	112581 views		WekeRoad Ink advanced razor 1 year ago, updated 7 months ago Luke Sampson

Choose your Visual Studio version, click Download This Scheme, and follow the instructions in the next section:

WekeRoad Ink

Submitted by [Luke Sampson](#)

rated
134

This is a refresh of Rob Conery's take on the Vibrant Ink theme WekeRoad (<http://blog.wekeroad.com/2007/10/17/textmate-th>) advantage the expanded syntax settings in VS2010.

Do you like it? [Hot](#) or [not](#)

Download this scheme for

- Visual Studio 2010
- Visual Studio 2008
- Visual Studio 2005

C# code

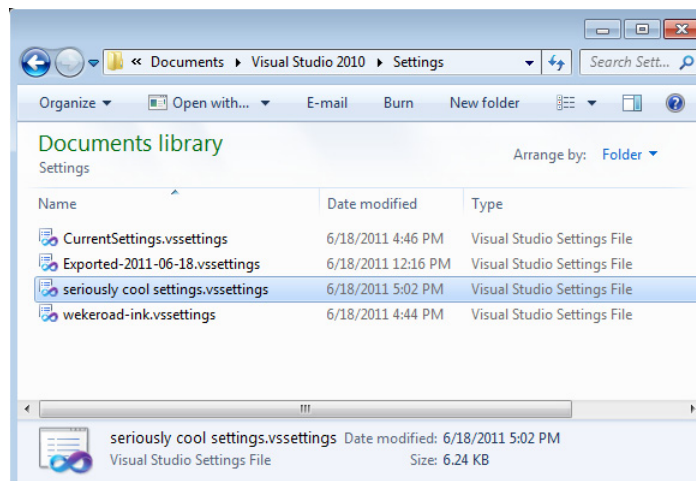
```

1 #region Studio Style
2 class Program : IThemeable
3 {
4     static int I = 1;
5     delegate void DoSomething();
6

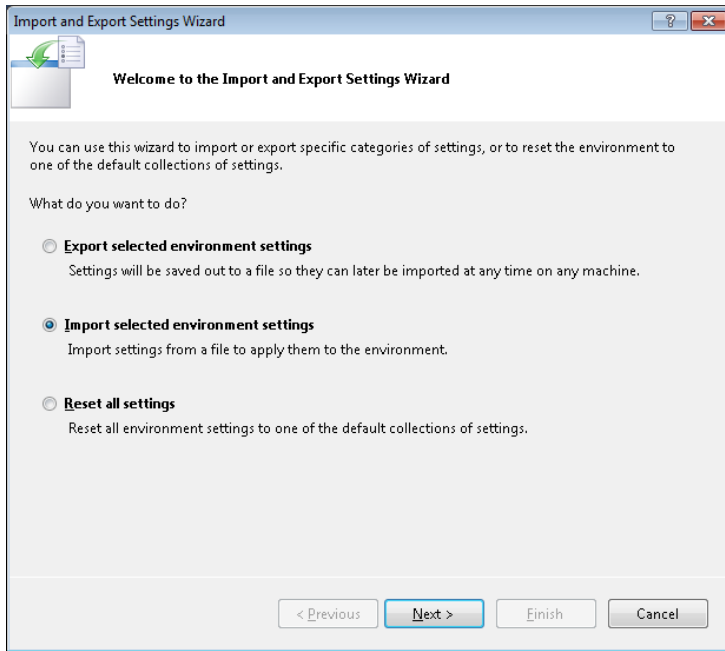
```

Changing Your Colors

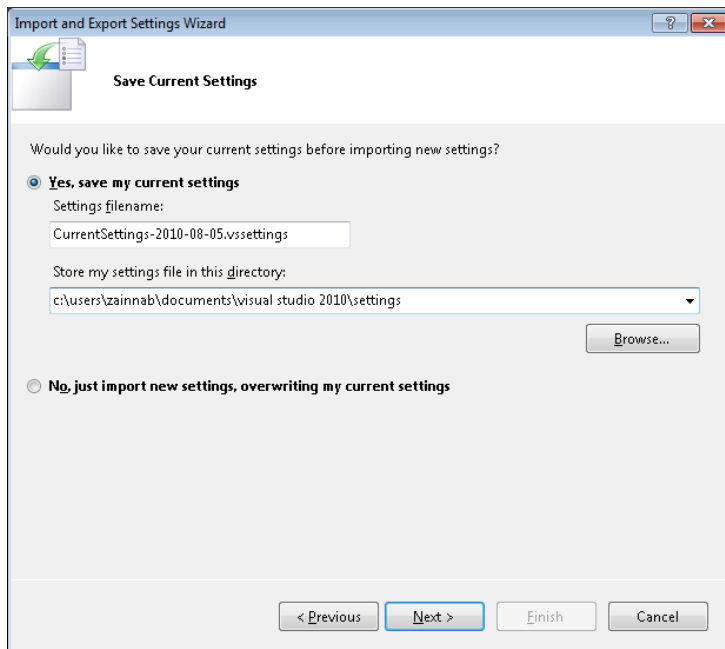
When you have a .vssettings file you want to import, copy or move the file to your computer. While you can put the file anywhere you want on your system, I prefer to put it with the other settings files located at C:\Users\



Now just go to Tools | Import And Export Settings on your computer:



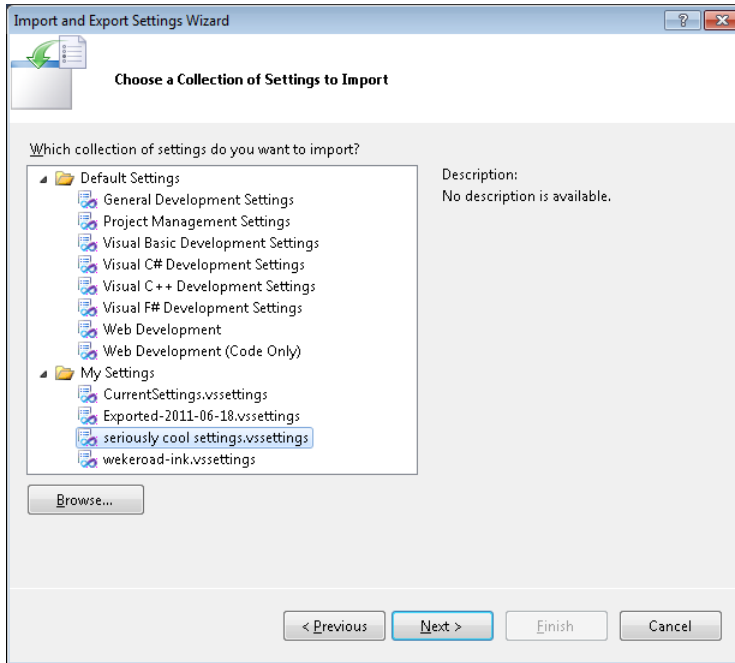
Make sure that Import Selected Environment Settings is selected, as shown in the preceding illustration, and click Next. If you haven't backed up your settings in a while, feel free to do so. Check out [vstipEnv0034](#) ("Change Your Visual Studio Color Scheme", page 17) if you want more information on exporting your settings:



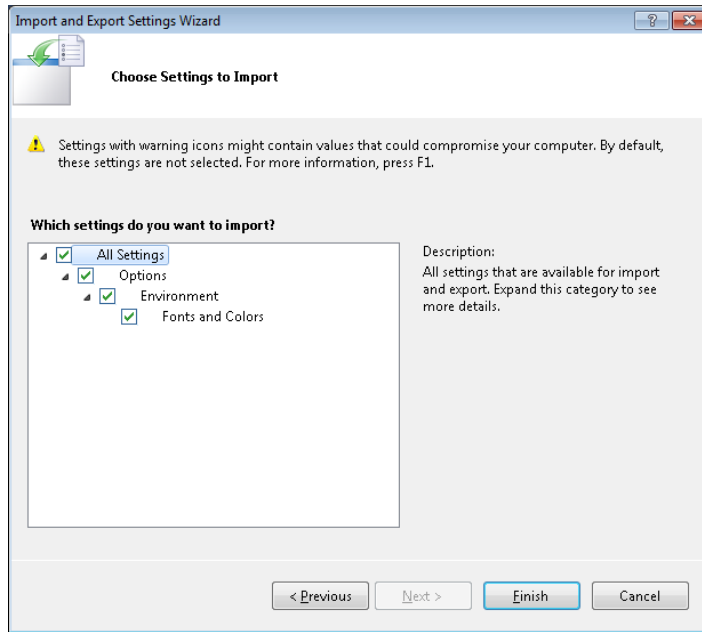
Choose the settings file that has the color scheme you want:



Note Click Browse to find your file if you didn't put it in your Settings folder.



Click Next. Verify that the file is importing *only* fonts and colors, and then click Finish:

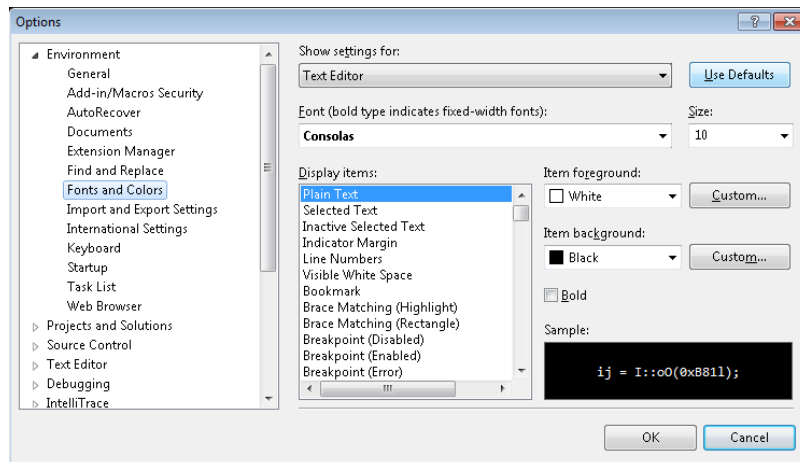


Resetting the Colors

You should have your new colors. If things get bad (for example, you get colors you don't like and didn't make a backup of your old colors) and you need to get the default colors back, all you have to do is go to Tools | Options | Fonts And Colors and click Use Defaults.



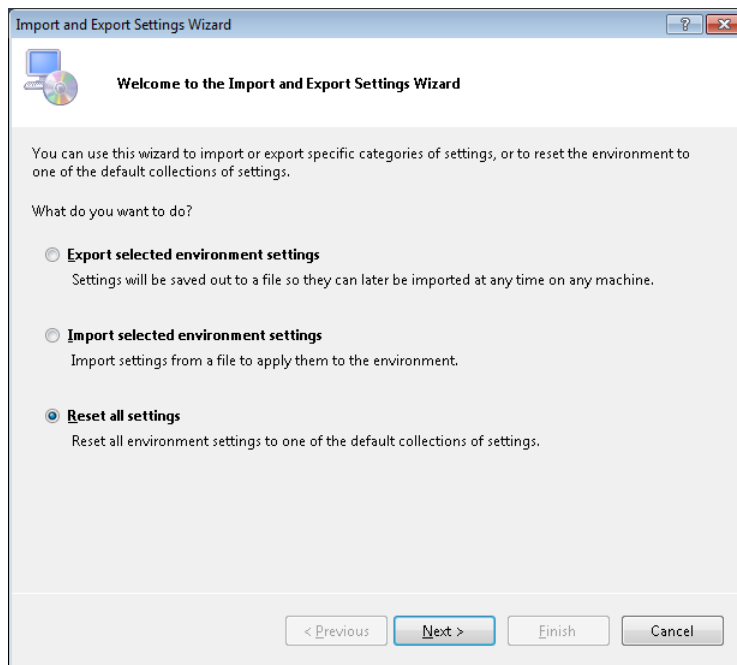
Warning Clicking Use Defaults is an option that wipes out any custom colors used previously.



01.10 Reset All Your Development Settings

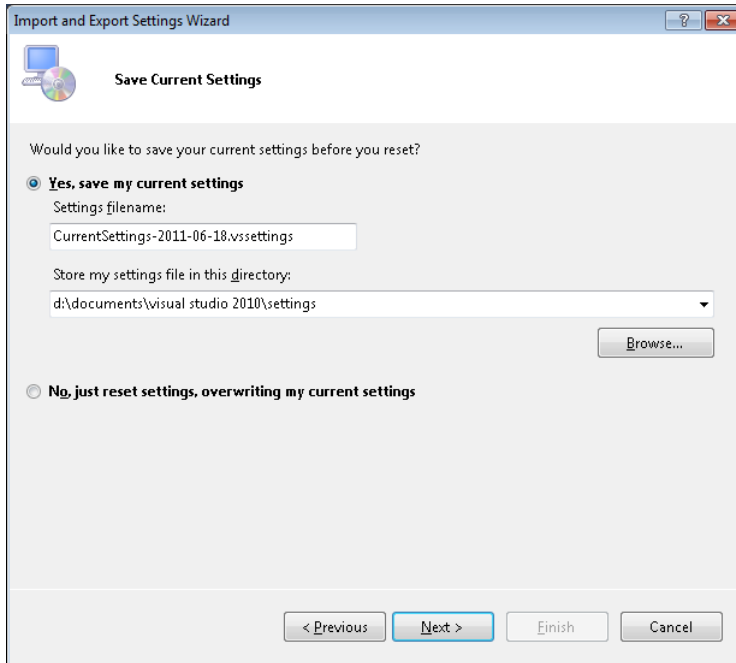
WINDOWS	Alt,T, I
MENU	Tools Import and ExportSettings
COMMAND	Tools.ImportandExport Settings
VERSIONS	2005, 2008, 2010
CODE	vstipEnv0023

Sometimes you need to get all your settings back to their original state. You can do this with the Reset All Settings option found under Tools | Import And Export Settings:

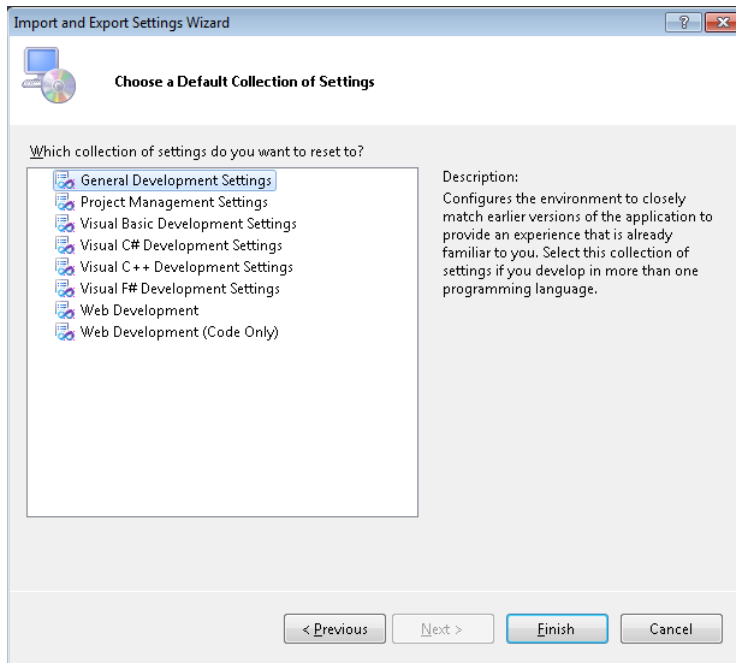


Warning Use the Reset All Settings option at your own risk. It *will* reset your settings, including a reset of your Toolbox, getting rid of any custom items you have put in there.

After you click Next on the Welcome page shown in the preceding illustration, you see the option to save your current settings. *You should absolutely do this.*



The next screen lets you choose from the list of default settings:

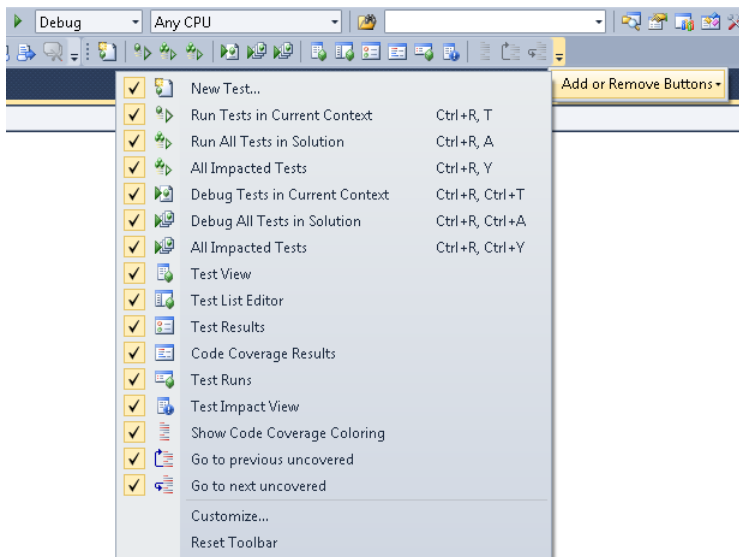


Choose your default settings, and click Finish. After the reset operation runs, it resets all your settings. This is definitely something you would do as a last resort, and remember, you can always bring back your old settings by importing settings you saved earlier (see vstipEnv0021, “Exporting Your Environment Settings”, page 6).

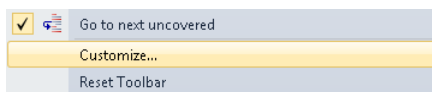
01.11 Customize Your Toolbars in Visual Studio 2010: Toolbars Tab

WINDOWS	Alt,T, C
MENU	Tools Customize
COMMAND	Tools.Customize
VERSIONS	2010
CODE	vstipEnv0030

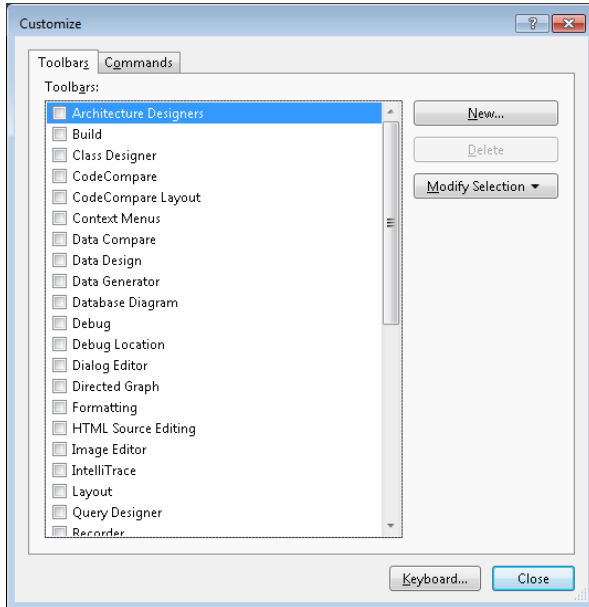
You can customize any toolbar in Visual Studio 2010. Just click the drop-down arrow to the right of any toolbar, and then click Add Or Remove Buttons:



Then click Customize:

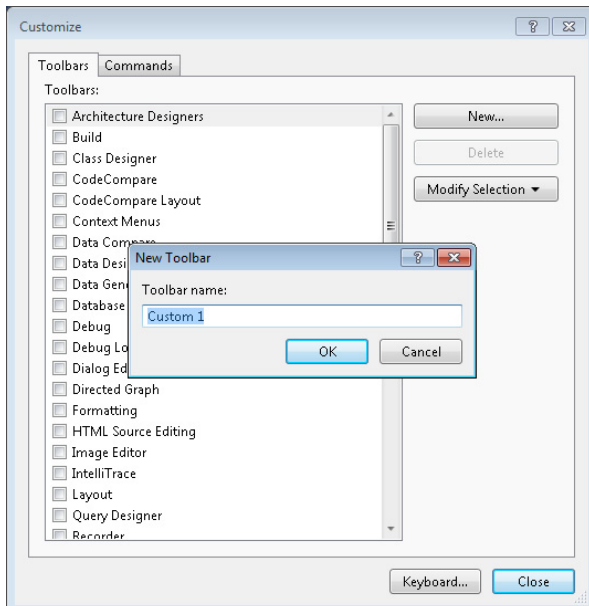


Alternatively, you can go to Tools | Customize on the menu bar. Whichever option you choose opens the Customize dialog box:

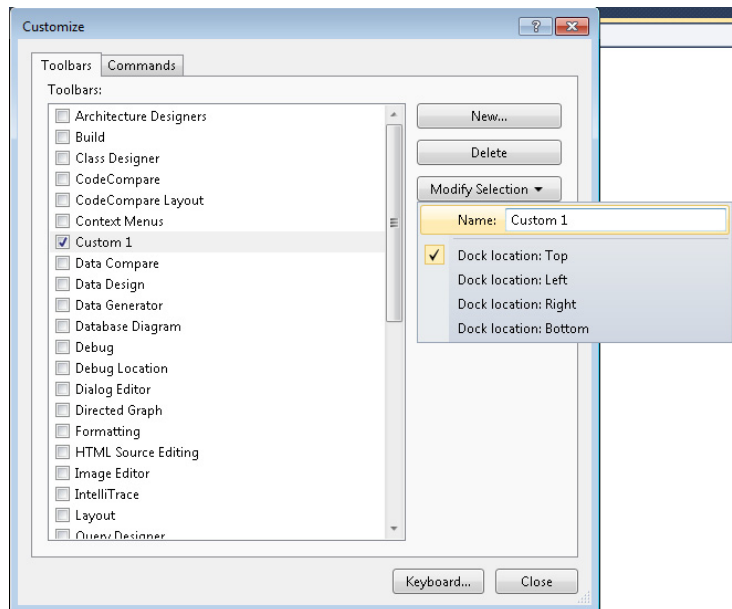


Custom Toolbars

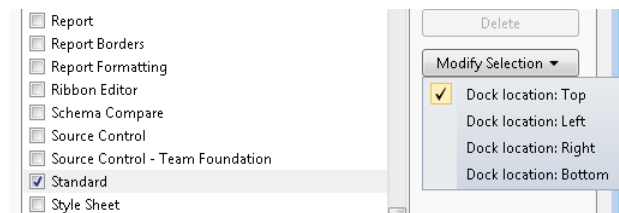
As shown in the preceding illustration, the Toolbars tab lists all the available toolbars. After you click New to create a customized toolbar, you are prompted to give the new toolbar a name:



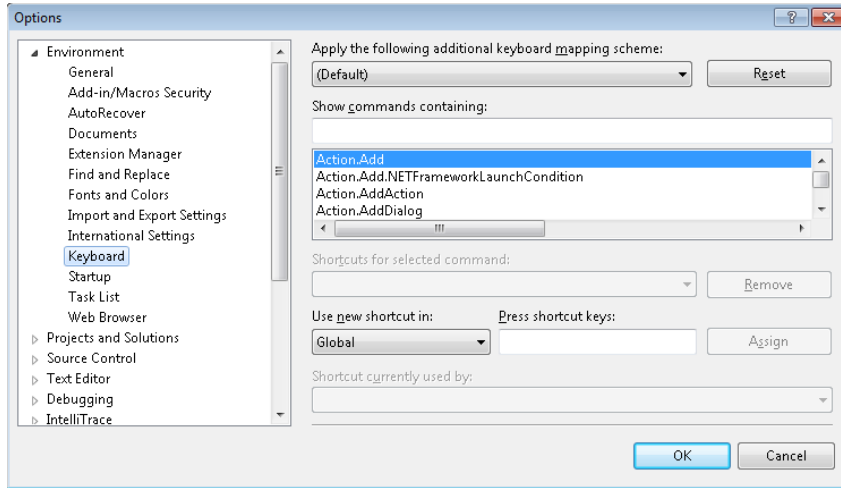
After you name it, you can delete the custom toolbar by clicking Delete, or you can change it by clicking Modify Selection to rename or relocate the toolbar:



Although you can rename custom toolbars by clicking Modify Selection, default toolbars can't be changed in this way:



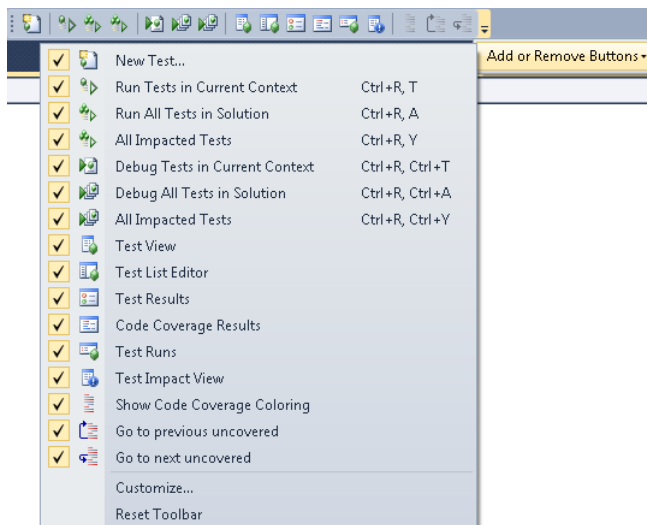
Clicking Keyboard at the bottom of the Customize dialog box takes you to the Tools | Options | Keyboard area, where you can add keyboard shortcuts for selected commands. (See vstipTool0063, "Keyboard Shortcuts: Creating New Shortcuts", page 127, for details.)



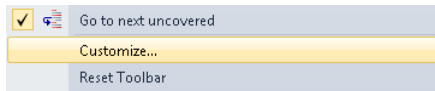
01.12 Customize Your Toolbars in Visual Studio 2010: Commands Tab

WINDOWS	Alt,T, C
MENU	Tools Customize
COMMAND	Tools.Customize
VERSIONS	2010
CODE	vstipEnv0031

You can customize any toolbar in Visual Studio 2010. Just click the drop-down arrow to the right of any toolbar, and then click Add Or Remove Buttons:

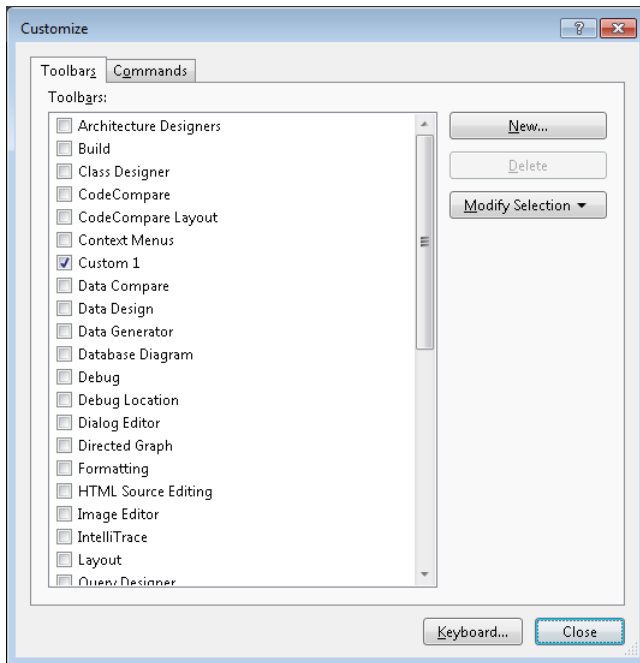


Then click Customize:



Alternatively, you can go to Tools | Customize on the menu bar.

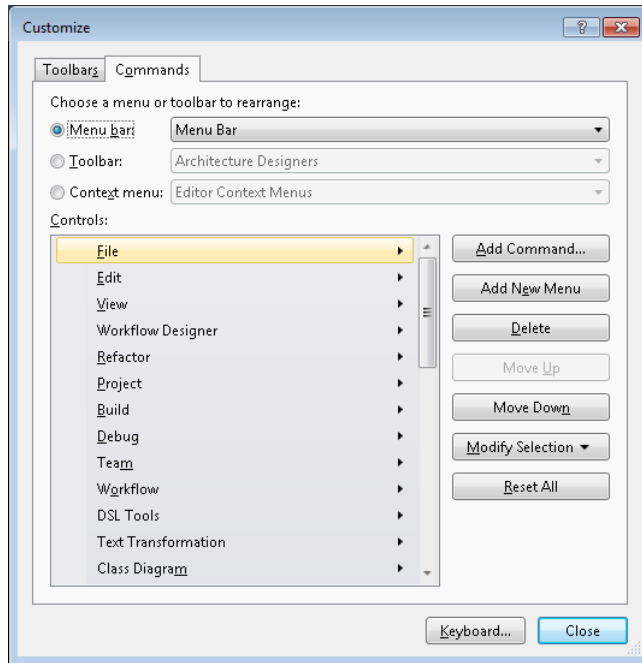
Either option you choose opens the Customize dialog box:



Click the Commands tab:



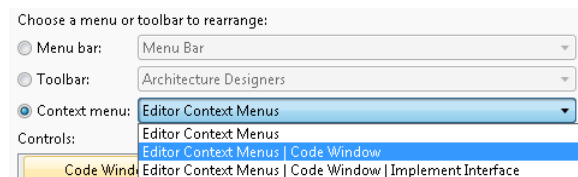
Note For information about the Toolbars tab, see vstipEnv0030, “Customize Your Toolbars in Visual Studio 2010 Toolbars Tab”, page 27.



As you can see, the Customize dialog box is fairly complex, so let's break it down into its parts as we look at an example.

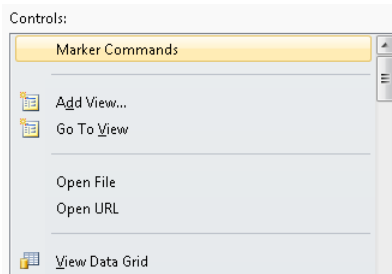
Rearrange

First is the choice of menu or toolbar to modify. In this case, choose the Editor Context Menus | Code Window option, which is what you see when you right-click while writing code:

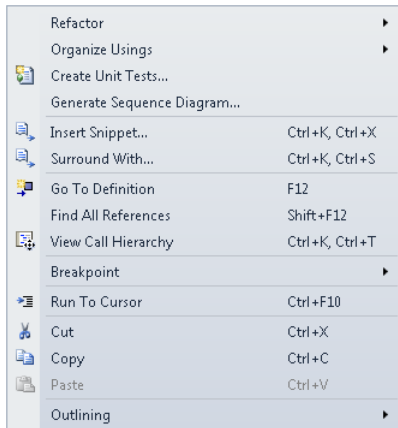


Controls

Next is the Controls area that shows the items on the menu or toolbar you have chosen to modify. For this example, it shows the items available when you right-click in a code window:



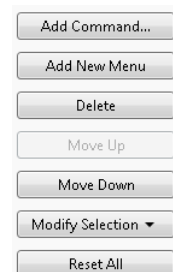
Remember that not all the items you see are available all the time because these items show up only in the proper context. So while it seems you have a large number of buttons currently available, when you right-click in your code window, this is an example of what you will currently see:



Buttons

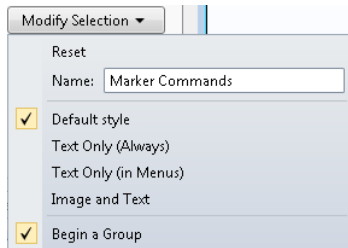
Finally, let's look at the area of the dialog box that has all the buttons that actually *perform* actions:

- **Add Command**
Lets you add a new item to the existing menu or toolbar.
- **Add New Menu**
Creates a new menu in the existing menu or toolbar.
- **Delete**
Removes the current item from the Controls area.
- **Move [Up or Down]**
Changes the location of the item in the Controls area.



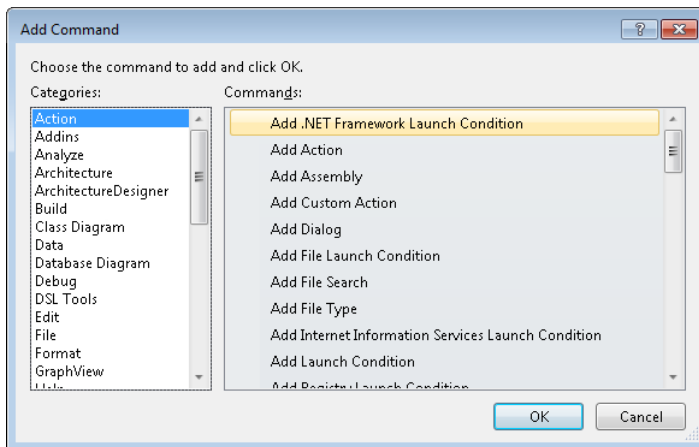
Modify Selection

Choosing Modify Selection enables you to make changes to the existing item in the Controls area, such as resetting it to the default settings, changing the name, and modifying text visibility options. Modify Selection also enables you to make a new group on the menu or toolbar:

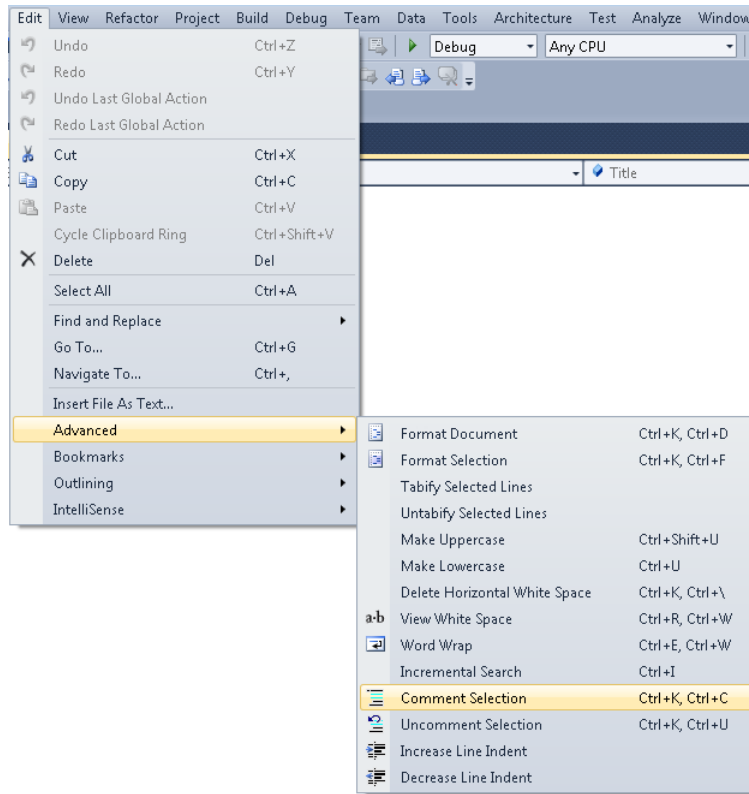


Finally, the Reset All option resets every item in the Controls area to its default settings. This capability is particularly useful if you have made a lot of changes.

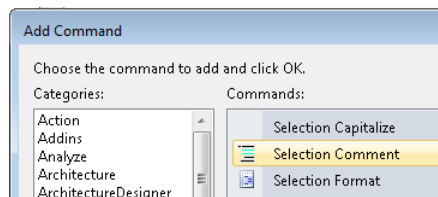
Getting back to our example: Let's assume you want to add the comment and uncomment items to the context menu so that you can use them when you select some code. First, click Add Command to bring up the Add Command dialog box:



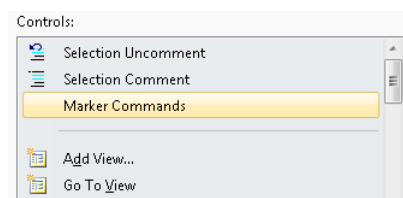
Now you need to figure out where the comment and uncomment items are located. How would you do this? Well, the best path is usually to see whether the item can be found on a menu somewhere and then use that as a clue:



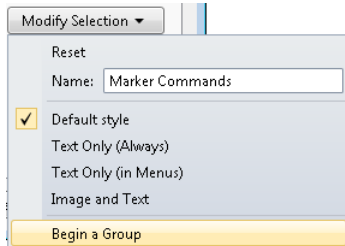
Because the items you want are off the Edit menu, you can search there first. It turns out the items you want are called Selection Comment and Selection Uncomment:



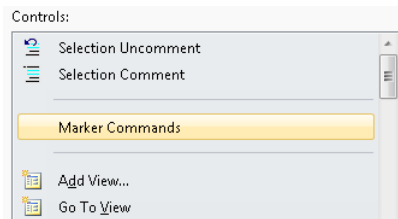
Find each one of these items, and click OK to add them to the Controls list:



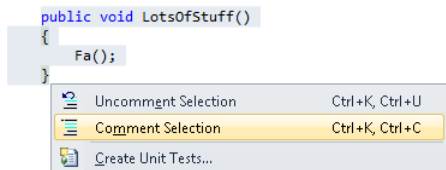
It would be nice to have these buttons in their own group, so select the item in the Controls dialog box, shown in the preceding illustration, where you would like the group line to be (Marker Commands, in this case), and then click Modify Selection and choose Begin A Group:



This creates a new group line, and your commands are in their own group:



Click Close and go to any code area. Select some code, right-click, and select Comment Selection:



It works perfectly, and you are all set to begin making your own modifications to your environment:

```

//public void LotsOfStuff()
//{
//    Fa();
//}

```

01.13 Visual Studio Logging

VERSIONS	2005, 2008, 2010
CODE	vstipEnv0048

There's no doubt Visual Studio is an awesome piece of software, but occasionally you might run into a problem loading it. Did you know that it comes with a logging switch? While the documentation (<http://msdn.microsoft.com/en-us/library/ms241272.aspx>) is lacking, the community comment contributed by Paul Harrington on the Visual Studio team helps a great deal.

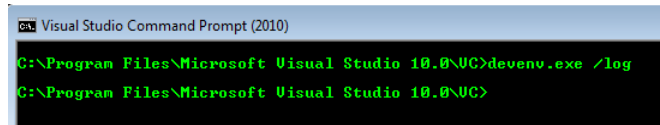
Essentially, the syntax is as follows:

```
devenv.exe /log [filename]
```

The [filename] is optional and, if not specified, the ActivityLog.xml file is called by default. The path is to the log file is:

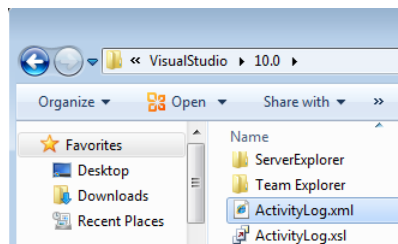
```
%APPDATA%\Microsoft\VisualStudio\<version>\ActivityLog.xml
```

Give it a try. Go to the Visual Studio command prompt, and enter **devenv.exe /log**:



```
Visual Studio Command Prompt (2010)
C:\Program Files\Microsoft Visual Studio 10.0\UC>devenv.exe /log
C:\Program Files\Microsoft Visual Studio 10.0\UC>
```

You can then navigate to the file location:



When you open the log file, the following illustration provides an example of what you might see:

```

ActivityLog.xml - Notepad
File Edit Format View Help
<?xml-stYLESHEET type="text/xsl" href="ActivityLog.xsl"?>
<activity>
  <entry>
    <record>1</record>
    <time>2011/06/19 23:23:26.606</time>
    <type>Information</type>
    <source>VisualStudio</source>
    <description>Microsoft Visual Studio 2010 version:
10.0.30319.1</description>
  </entry>
  <entry>
    <record>2</record>
    <time>2011/06/19 23:23:26.628</time>
    <type>Information</type>
    <source>VisualStudio</source>
    <description>Running in User Groups: Users</description>
  </entry>

```

Fortunately, an XML style sheet (XSL) comes with the data, so if you view the XML file in your browser, you can see a much cleaner view:

Activity Monitor Log		
infos	276	
warnings	1	
errors	0	
#	Type	Description
1		Microsoft Visual Studio 2010 version: 10.0.30319.1
2		Running in User Groups: Users
3		Available Drive Space: C:\ drive has 83539099648 bytes; D:\ drive has 51970506752 bytes
4		Internet Explorer Version: 8.0.7600.16821
5		.NET Framework Version: 4.0.31106.0

Now you can easily see the logging information and look for any issues.

01.14 Visual Studio Safe Mode

VERSIONS	2005, 2008, 2010
CODE	vstipEnv0050

Occasionally you have a situation where Visual Studio might not start up correctly or at all. Using Visual Studio in safe mode, you can load only the default environment, services, and shipped versions of third-party packages to see whether the problem is caused by one or more third-party add-ins. Just go to the Visual Studio command prompt, and type **devenv.exe /safemode**:

```

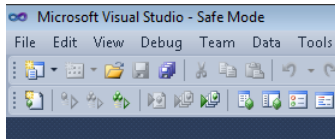
Visual Studio Command Prompt (2010)
Setting environment for using Microsoft Visual Studio 2010 x86 tools.
C:\Program Files\Microsoft Visual Studio 10.0\VC>devenv.exe /safemode
C:\Program Files\Microsoft Visual Studio 10.0\VC>

```




Note Although I don't show it here, I suggest using the Visual Studio logging feature before running safe mode to see whether it can help you determine the source of the problem. For more information, see [vstipEnv0048](#), "Visual Studio Logging", page 37.

When Visual Studio starts up, it indicates it is running in safe mode in the title bar:



From here, you can start determining what might have caused Visual Studio to fail and remedy the issue.

01.15 The ResetSettings Switch

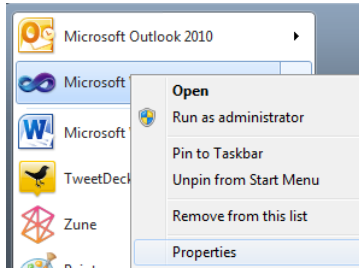
VERSIONS	2005, 2008, 2010
CODE	vstipEnv0047

Visual Studio supports several switches. One of these is the `/ResetSettings` switch. When used by itself, it resets Visual Studio to the default settings you initially chose during install. That's nice, but an even better option is available that can be particularly useful for people in other scenarios.

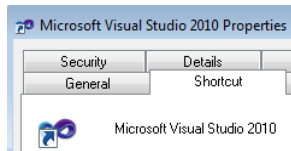
Let's take a classic example: You might have two (or more) monitors at work, but when you get home, you work with just one monitor. Your window layouts (among other things) could be very different in each place. In [vstipEnv0040](#), "Export Your Window Layouts" (page 134), I showed you how to export just your window layouts. Using the exported information, you could create and use two different window layouts: one for work and one for home. This tip shows you how to do this when using two different machines or the same machine at work and home.

Two Different Machines

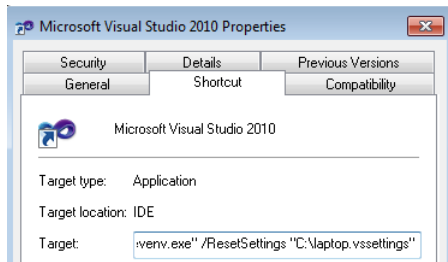
The question is: What do you do once you have exported the window layouts? Well, now you put the `.vssettings` files where you can easily get to them on your machines and then you go to the Properties dialog box of the Visual Studio program icon:



Click the Shortcut tab:



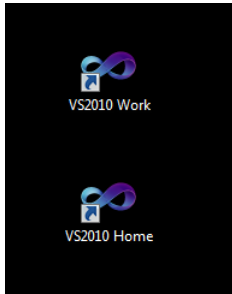
In the Target area, type **/ResetSettings [settings file]**, where [settings file] is the path to the settings file for one of your layouts:



Now Visual Studio loads up with the settings appropriate for your machine.

Same Machine

What if you use the same machine for home and work, like a laptop? Just make two copies of the Visual Studio program icon, put them somewhere (on your Desktop, most likely), and give them names:



Now just follow the steps for the different machines for each icon, and you can use one icon when you are at work to get the work window layouts and the other for home with the home window layout.

Chapter 4

Working with Documents

“He turned off the word processor, realizing just a second after he’d flicked the switch that he’d forgotten to save the document. Well, that was all right. Maybe it had even been the critic in his subconscious, telling him the document wasn’t worth saving.”

—Stephen King, “Secret Window, Secret Garden” in “Four Past Midnight”

Documents serve as the cornerstone of your activities in Visual Studio. Writing code, debugging code, creating interfaces, or just about anything else you do is done with documents. Yet we still seem to take our documents for granted. This chapter focuses on working with documents in the File Tab Channel as well as ways to navigate better. Several advanced topics, such as creating custom file extensions and working with previous versions are covered as well.

04.01 Insert Documents to the Right of Existing Tabs

WINDOWS	Alt,T, O
MENU	Tools Options Environment Documents
VERSIONS	2010
CODE	vstipEnv0001

By default, Visual Studio 2010 opens up new tabs to the left of existing ones, as shown in the following illustration.



You now have an option to put newly opened documents in the file channel to the right of existing tabs.

Just go to Tools | Options | Environment | Documents, and select the Insert Documents To The Right Of Existing Tabs option.

You should see new tabs show up to the right of existing tabs.

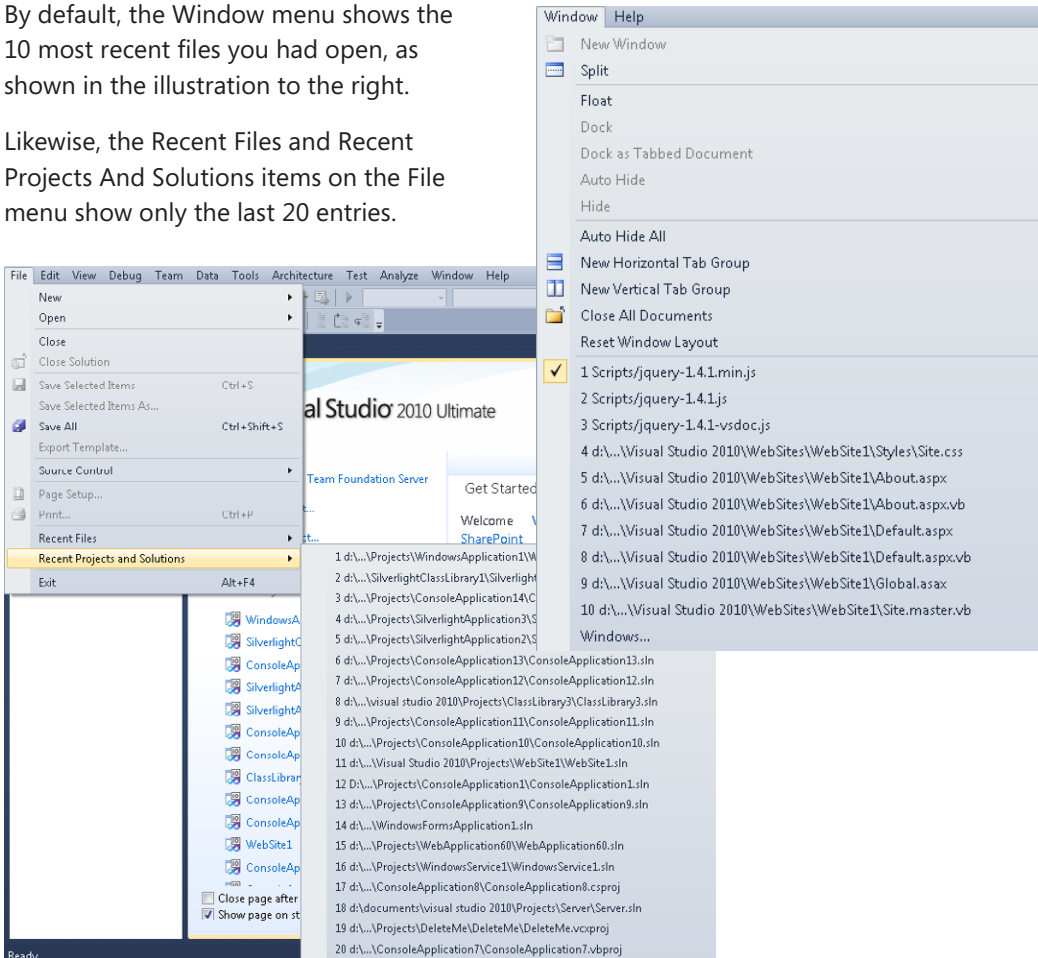


04.02 Recent Files

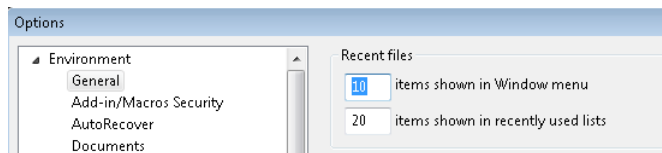
WINDOWS	Alt,W, [1,2,3, etc] (windows); Alt,F, F, [1,2,3, etc] (files); Alt,F, J, [1,2,3, etc] (projects and solutions)
MENU	Tools Options General Recent files; Window [1,2,3, etc]; File Recent Files [1,2,3, etc]; File Recent Projects and Solutions [1,2,3, etc]
COMMAND	Tools.Options
VERSIONS	2005, 2008, 2010
CODE	vstipEnv0013

By default, the Window menu shows the 10 most recent files you had open, as shown in the illustration to the right.

Likewise, the Recent Files and Recent Projects And Solutions items on the File menu show only the last 20 entries.



You can easily modify these numbers (up or down) by going to Tools | Options | General | Recent files.

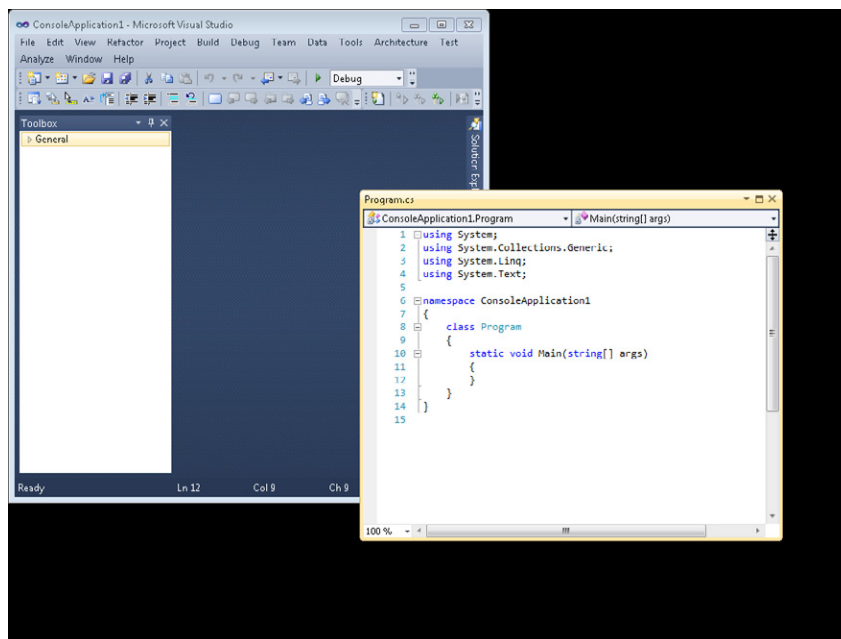


For each of these options, the minimum value is 1 and the maximum value is 24. Experiment with numbers that suit your taste.

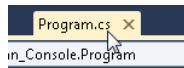
04.03 Working with Documents on Multiple Monitors

WINDOWS	Alt,W, F (float); Alt,W,T (dock)
MENU	Window Float; Window Dock as Tabbed Document
COMMAND	Window.Float; Window.DockasTabbedDocument
VERSIONS	2010
CODE	vstipTool0004

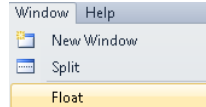
This is one we have all been wanting for a long time: detachable document windows. You can now detach document windows and put them on another monitor! You have a couple of ways to do this.



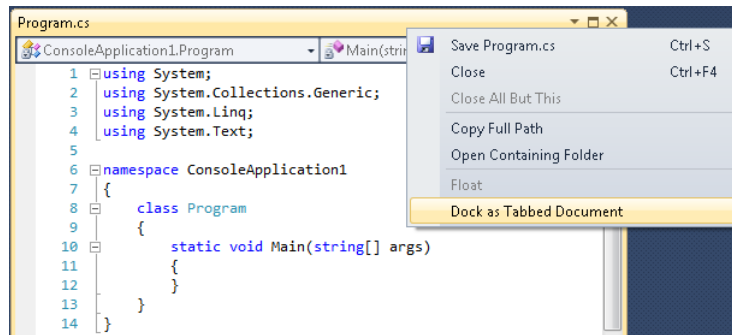
The first way is to simply click and drag the tab for the document window out of the IDE.



The second way is to go to Window | Float on the menu bar, as shown in the following illustration.

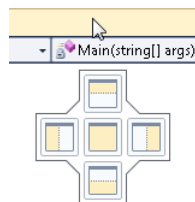


What if you want to put the window back? No worries; just right-click the title bar of the document and choose Dock As Tabbed Document.

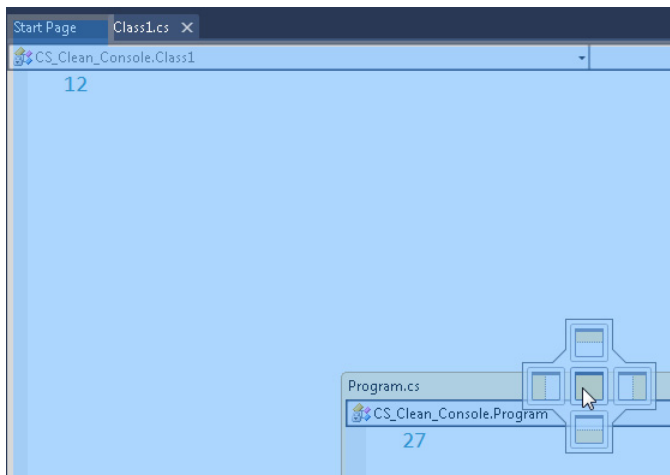


If you want an alternative method, you can go the following route: Click and drag the document window by its title bar into the IDE.

The guide diamond appears in the IDE, as shown in the following illustration.



Hold down the left mouse button, and move your cursor over the middle item in the guide diamond. You should see an outline of where the window will be docked.



Release the mouse button, and it should dock where you want it to go.

04.04 Navigate Open Document Windows

DEFAULT	Ctrl+F6 (next); Ctrl+Shift+F6 (previous)
VISUAL C++ 2	Ctrl+F6; Ctrl+Tab (next) Ctrl+Shift+F6; Ctrl+Shift+Tab (previous)
VISUAL STUDIO 6	Ctrl+F6; Ctrl+Tab (next) Ctrl+Shift+F6; Ctrl+Shift+Tab (previous)
COMMAND	Window.NextDocumentWindow; Window.PreviousDocumentWindow
VERSIONS	2005, 2008, 2010
CODE	vstipTool0013

OK, so you have a lot of files open in the file channel:



And you don't want to use your mouse to switch between tabs. Just press Ctrl+F6 to go forward.

Or press Ctrl+Shift+F6 to go backward.

04.05 Close the Current Document Window

DEFAULT	Ctrl+F4
VISUAL BASIC 6	Ctrl+F4
VISUAL C# 2005	Ctrl+F4
VISUAL C++ 2	Ctrl+F4
VISUAL C++ 6	Ctrl+F4
VISUAL STUDIO 6	Ctrl+F4
WINDOWS	Alt,F, C
MENU	File Close
COMMAND	Window.CloseDocumentWindow; File.Close
VERSIONS	2005, 2008, 2010
CODE	vstipTool0014

You can close the current document window from the keyboard. Just make sure you are in the document you want to close.

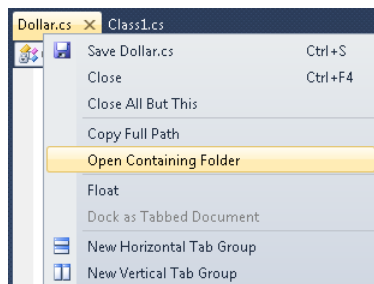


Then press Ctrl+F4. The current document closes, and it prompts you to save changes if you haven't already.

04.06 Open a File Location from the File Tab

DEFAULT	Alt+- (minus sign), O (VS2010 Only)
VISUAL BASIC 6	[no shortcut]
WINDOWS	Alt+- (minus sign), O (VS2010 Only)
COMMAND	File.OpenContainingFolder; Window.ShowDockMenu
VERSIONS	2005, 2008, 2010
CODE	vstipEdit0014

Do you often find yourself needing to go to your project location in Windows Explorer? Just right-click the file's tab, and choose Open Containing Folder.

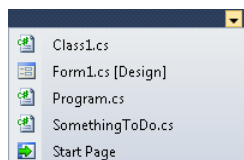


The file location opens in Windows Explorer, and you can manipulate the files from there.

04.07 Open the File Menu Drop-Down List from Your Keyboard

DEFAULT	Ctrl+Alt+Down Arrow
VISUAL BASIC 6	Ctrl+Alt+Down Arrow
VISUAL C# 2005	Ctrl+Alt+Down Arrow
VISUAL C++ 2	Ctrl+Alt+Down Arrow
VISUAL C++ 6	Ctrl+Alt+Down Arrow
VISUAL STUDIO 6	Ctrl+Alt+Down Arrow
WINDOWS	[no shortcut]
COMMAND	Window.ShowEzMDIFileList
VERSIONS	2005, 2008, 2010
CODE	vstipEnv0003

When you have a lot of files open, it is sometimes easier to view them as a list instead of tabs. The File menu drop-down list does that for you. You can click the drop-down button to the far right on the file tab, or you can simply use Ctrl+Alt+Down Arrow to activate it.

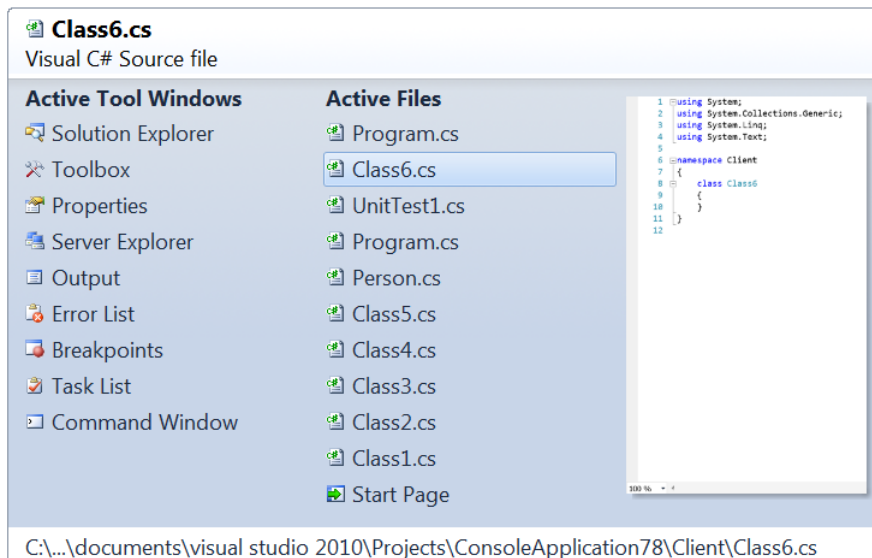


An interesting thing about this list is that it has type-ahead functionality. So, in this example, if you type the letter **S**, it automatically selects SomethingToDo.cs. Hitting **S** again results in Start Page being selected. If you have a lot of files, you can type more characters to narrow down the selection. For example, typing **ST** jumps straight to Start Page.

04.08 Using the IDE Navigator

DEFAULT	Ctrl+Tab (forward in Active Files); Ctrl+Shift+Tab (backward in Active Files); Alt+F7 (forward in Active Tool Windows); Alt+Shift+F7 (backward in Active Tool Windows)
VISUAL BASIC 6	[no shortcuts]
VISUAL C# 2005	Ctrl+Tab (forward in Active Files); Ctrl+Shift+Tab (backward in Active Files); Alt+F7 (forward in Active Tool Windows); Alt+Shift+F7 (backward in Active Tool Windows)
VISUAL C++ 2	[no shortcuts]
VISUAL C++ 6	Ctrl+Tab (forward in Active Files); Ctrl+Shift+Tab (backward in Active Files); Alt+F7 (forward in Active Tool Windows); Alt+Shift+F7 (backward in Active Tool Windows)
VISUAL STUDIO 6	Ctrl+Tab (forward in Active Files); Ctrl+Shift+Tab (backward in Active Files); Alt+F7 (forward in Active Tool Windows); Alt+Shift+F7 (backward in Active Tool Windows)
WINDOWS	[no shortcuts]
COMMAND	Window.NextDocumentWindowNav; Window.PreviousDocumentWindowNav; Window.NextToolWindowNav; Window.PreviousToolWindowNav
VERSIONS	2005, 2008, 2010
CODE	vstipTool0023

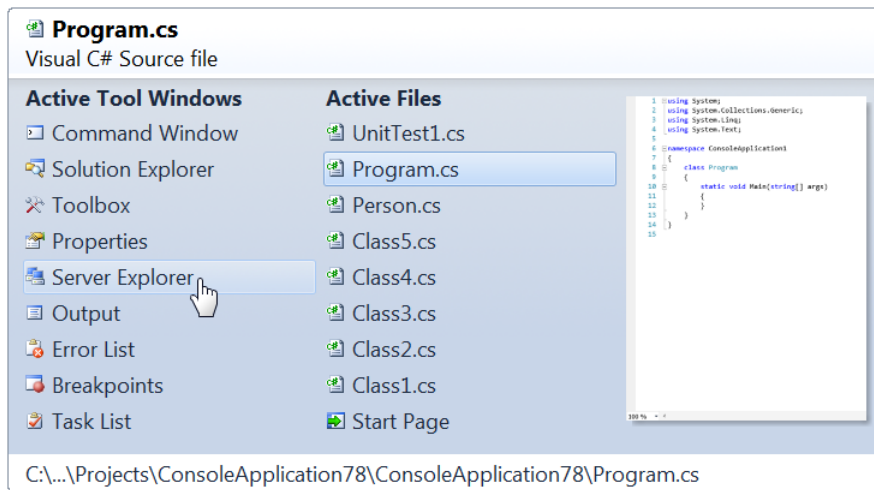
Navigating documents and tool windows in the IDE is a critical part of your development experience. You can easily move among active file and tool windows by pressing Ctrl+Tab.





Note The images in this tip show the IDE Navigator with document preview (image to the right of the lists). This feature is off by default in Visual Studio 2010, but can be turned on as shown in vstipTool0113, “Thumbnail Previews in the IDE Navigator”, in Appendix B (<http://go.microsoft.com/fwlink/?Linkid=223758>).

Some interesting things come with using this feature. For example, holding down the Ctrl key keeps the IDE Navigator showing once it is up. Also, you can select any item in this dialog box, while it is showing, by using your mouse or arrow keys.



Navigator Areas

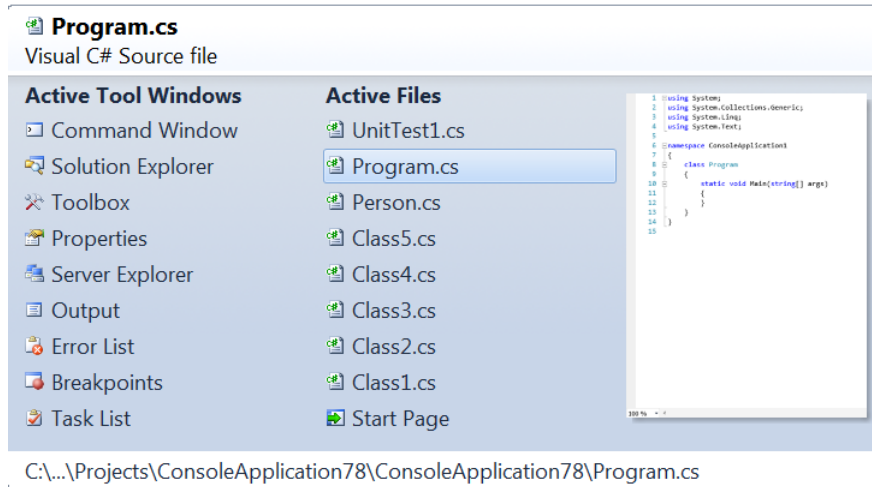
Let's take a look at the two major areas in the navigator: Active Files and Active Tool Windows.

Active files

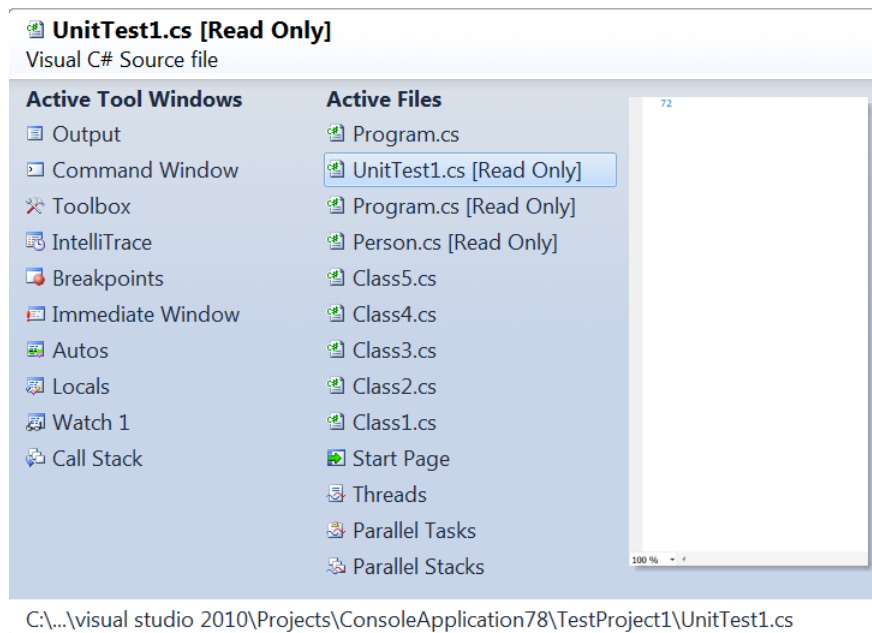
To navigate active files, press Ctrl+Tab to go forward and Ctrl+Shift+Tab to go backward through the list. The currently selected file is highlighted, and its name is displayed at the top of the dialog box. Also, notice that the full file path is shown at the bottom of the IDE navigator.

Active tool windows

This part of the dialog box shows all your tool windows that are currently open. To get to this area, you can use Alt+F7 or Alt+Shift+F7. The interesting part is that this list changes depending on when you use it. The following illustration shows what mine looks like while I am writing code.



And here's what it looks like when I'm debugging:



04.09 Multiple Views of the Same Document

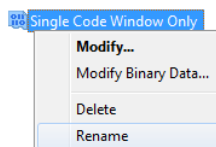
WINDOWS	Alt,W, N
MENU	Window New Window
COMMAND	Window.NewWindow
VERSIONS	2005, 2008, 2010
CODE	vstipEnv0016

Sometimes you might want to look at a particularly large document in several different areas at the same time. For example, you might want to look at the same document on multiple monitors. This tip shows you how to make this happen.

Special Note for VB Users in Visual Studio 2010

This feature is turned off by default in VB. A lot of history and reasoning is behind this, but the long and short of it is that this was fixed for 2010 but time ran out and it wasn't tested. So you can turn this on for VB, but you do so at your own risk. Special thanks to my friend Dustin Campbell for supplying the history and the fix.

To fix this, go to "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\VisualStudio\10.0\Languages\Language Services\Basic\" and rename the Single Code Window Only registry key to something like **[your initials here] Single Code Window Only**. The following illustration shows what I did:



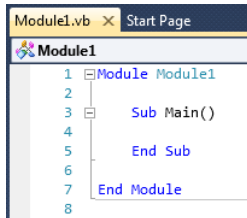
zn Single Code Window Only

Now restart Visual Studio, and you are good to go for the rest of this tip.

Multiple Views

I came across this while I was checking my email one day and noticed a thread started by the legendary Deborah Kurata concerning the Window | New Window menu item. The following example describes how it works.

Open a document window.



```

Module1.vb × Start Page
Module1
1 Module Module1
2
3 Sub Main()
4
5 End Sub
6
7 End Module
8

```

Now go to Window | New Window on the menu bar to open a duplicate window of the current document.



```

Module1.vb:2 × Module1.vb:1

```

Notice that “:1” is added to the existing document tab text and that “:2” is appended to the name on the new document tab. You can apparently do this *forever* (or at least up to 150, which is as high as I have tested this feature).



```

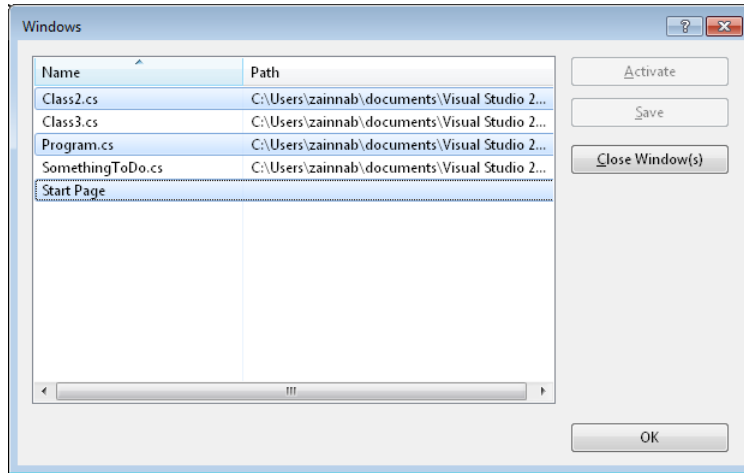
Module1.vb:150 × Module1.vb:149 Module1.vb:148

```

04.10 Closing Just the Selected Files You Want

WINDOWS	Alt,W, W
MENU	Window Windows
COMMAND	Window.Windows
VERSIONS	2005, 2008, 2010
CODE	vstipEnv0010

What do you do when you have a lot of files open and want to close only a few of them? Just go to Window | Windows on the menu bar, as shown in the following illustration.

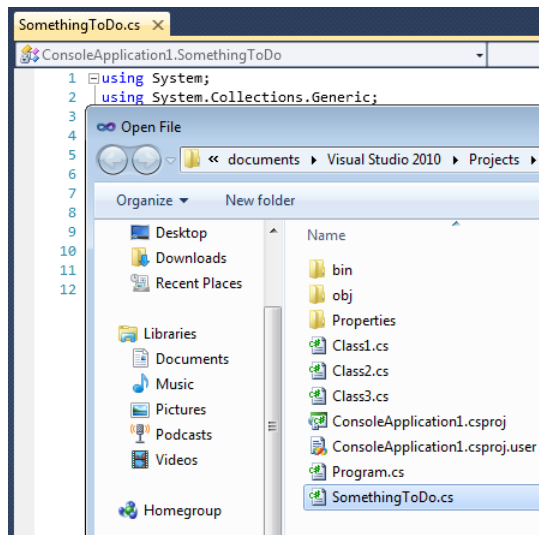


Select the files you want to close (Ctrl+Left-click), and then click Close Window(s). It closes the windows you selected and leaves the rest open.

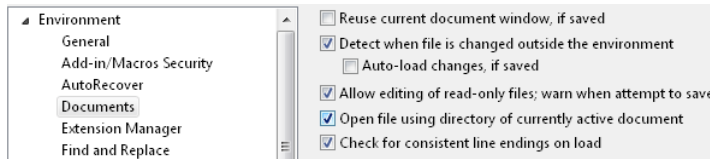
04.11 Understanding the File Open Location

MENU	Tools Options Environment Documents
VERSIONS	2005, 2008, 2010
CODE	vstipEnv0035

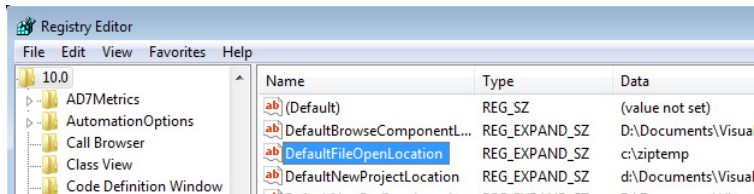
Have you ever noticed that when you go to open a file in Visual Studio (Ctrl+O) it automatically uses the directory of the current active document?



This is controlled by the Open File Using Directory Of Currently Active Document option. You can find this at Tools | Options | Environment | Documents.



You can turn this feature off by clearing its check box, and Visual Studio then uses the DefaultFileOpenLocation from HKEY_CURRENT_USER\Software\Microsoft\VisualStudio\<version> in the registry instead.



Be aware that the DefaultFileOpenLocation changes every time you successfully open a file in the Open File dialog box. However, the update is not written to the registry until you close Visual Studio.

04.12 Show Previous Versions

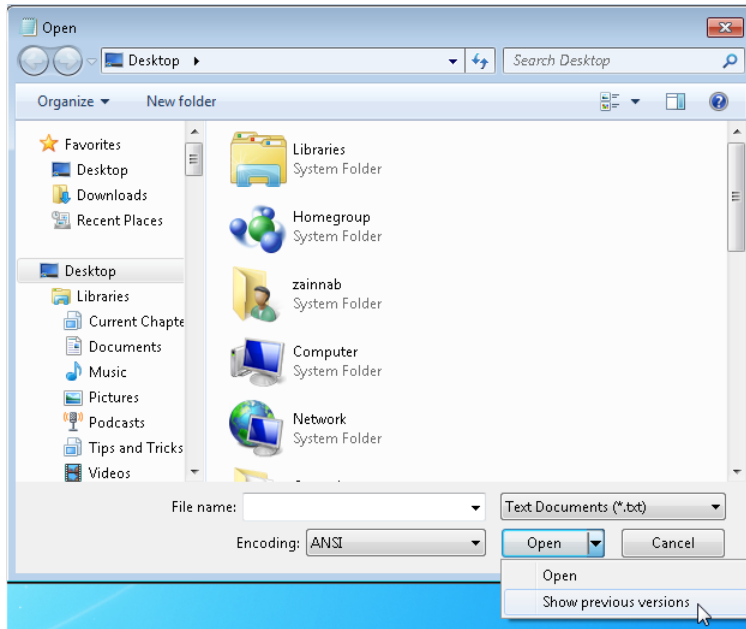
VERSIONS	2008, 2010
CODE	vstipEnv0036



Note For more information about previous versions including how to activate it if you don't currently have it turned on, go to <http://windows.microsoft.com/en-US/windows-vista/Previous-versions-of-files-frequently-asked-questions>.

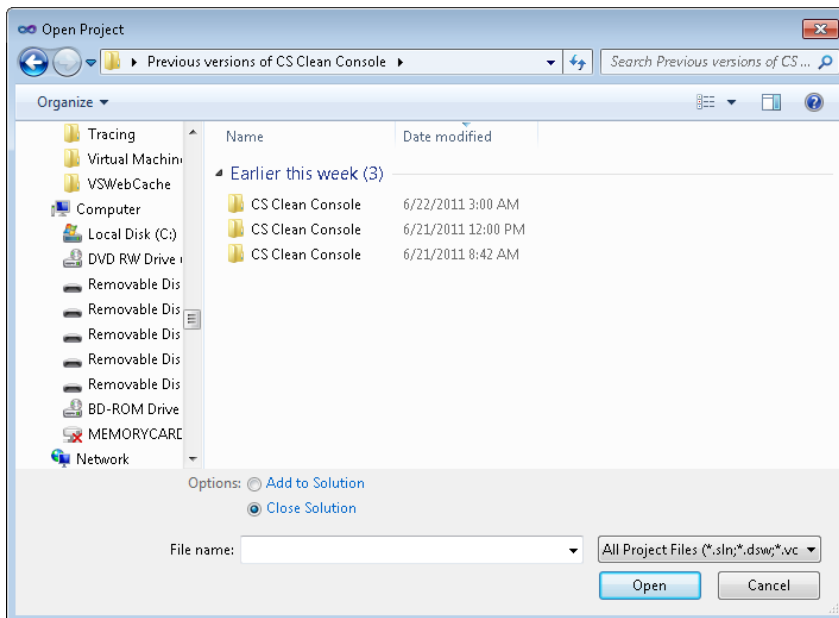
Ever want to go back in time when you save a change to your code that you didn't want saved? If you use source control, you are usually OK, but if you don't, this tip is for you.

If you run a Windows Vista or later operating system (excluding Home Editions), you have an option you might not have noticed before called Show Previous Versions. It shows up in various applications, such as Notepad, as shown in the following illustration.



You can also see this option in the Open Project dialog box as well.

When you click Show Previous Versions, you can see prior versions of the current directory you are in, as shown in the following illustration.

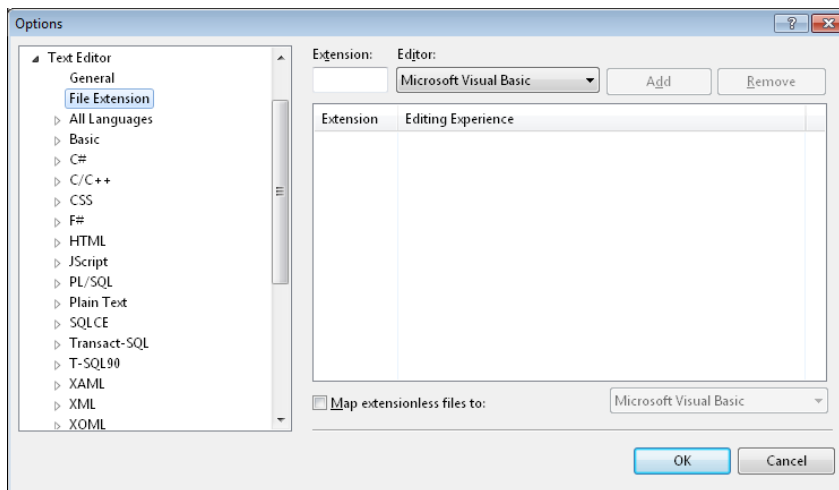


Now you can open previous versions of solutions, projects, files, and so forth, and do what you like.

04.13 Using Custom File Extension Associations

WINDOWS	Alt,T, O
MENU	Tools Options Text Editor File Extension
COMMAND	Tools.Options
VERSIONS	2005, 2008, 2010
CODE	vstipEnv0038

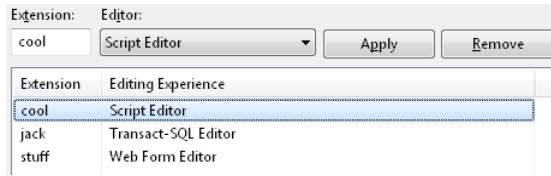
If you have a custom file extension that you would like to associate with an editing experience, just go to Tools | Options | Text Editor | File Extension to see the options shown in the following illustration.



Simply type in your extension and the editing experience you want to have when it is opened. In the following example, I have a .cool extension that is associated with the Script Editor:



When I click Add, as shown in the preceding illustration, the following information is added to the overall list:



Notice that you can select any item in the list, modify the extension and/or editor, and then click Apply to save the changes. Additionally, you can click Remove to take any entry out of the list.

Index

A

- accelerators, keyboard. *See* keyboard accelerators
- board accelerators
- Actions area (Exception Assistant), 333–334
- Actions folder (VS Image Library), 149
- Active Files area (IDE navigator), 161
- active items, tracking in Solution Explorer, 52–53
- Active Tool Windows area (IDE navigator), 161
- Add Command dialog box, 34–35
- Add New Item dialog box, 77
- Add Or Remove Buttons (toolbars), 27
- aliases
 - command aliases, 113–115
 - Command Window
 - Aliases, 143–145
 - creating from commands, 118
- All Components option (Look In dialog), 197
- AllMargins extension (VS), 411–413
- All tab/Common tab in statement completion, 226
- ampersand (&) for accelerator keys, 142
- animations
 - Animations folder (VS Image Library), 150
 - tool window, changing, 14–15
- Annotations_Buttons folder (VS Image Library), 150–151
- application-level tracing, 336–339
- arguments and switches (commands), 115–119
- asterisk (*)
 - to expand Toolbox, 87
 - for undoing Quick Replace, 179
 - for wildcard searches, 105
- attributes, trace, 337–338
- auto-hide all tool windows, 99–100
- Auto Hide Channel, 100–101
- Automatically Adjust Visual Experience Based On Client Performance check box, 13
- auto-populating, stopping Toolbox from, 136

- AutoRecover feature, 10–13, 57
- Autos window, 348, 379–382

B

- Begin A Group option (Modify Selection), 36
- blank lines
 - accidentally copying, 211
 - inserting above/below current line, 218
- Boolean expressions, 319
- Boolean logic, 323
- box selections
 - pasting content
 - between, 238–239
 - pasting single selection into, 239–240
 - replacing text with, 237–238
 - zero-length box selection, 240–241
- braces, moving/selecting between matching, 224–225
- Break All Processes When One Process Breaks, 374–376
- breakpoints
 - adding labels to, 293–294
 - enabling/disabling all, 295–296
 - hidden Breakpoints tool window, 86
 - Hit Count, 314–316
 - importing/exporting, 329
 - searching, 312–313
 - setting complex conditions for, 321–323
 - setting filters for, 324–325
 - setting in Call Stack window, 367–368
 - setting on functions, 316–317
 - setting simple conditions of, 318–320
 - setting with code, 291–292
 - window, opening with Ctrl+Alt+B, 293
- buttons
 - Buttons area (customize dialog box), 33–34
 - Code Snippets Manager, 264
 - Quick Find, 176
 - Replace in Files, 193–194

C

- Call Hierarchy window, 310–312
- call stacks, unwinding on unhandled exceptions, 334
- Call Stack window
 - setting breakpoints in, 367–368
 - setting tracepoints in, 369–371
- characters, transposing, 219–220
- Char variables (customizing search results), 208
- Choose Search Folders option (Find What combo box), 189
- Choose Settings To Export dialog box, 7
- circling arrows icon, 347
- C# language
 - setting breakpoints in, 292
 - TODO comments in, 296
- C++ language
 - setting breakpoints in, 292
 - TODO comments in, 296
- Class Diagrams, 125
- Classic view (Visual Studio 2010 Online Help), 5–6
- Class View
 - “Class View and Object Browser Icons”, 199
 - creating folders in, 350–352
 - searching in, 353–355
 - synchronizing, 355
- Clear All command (Find Symbol), 203
- Clear All option (Find Results window), 191
- click and drag for docking/undocking tool windows, 107–109
- client projects, launching, 50
- Clipboard Ring, cycling through, 220–221
- Close On Exit option (Initial Directory), 140
- closing current document window, 158
- closing tool windows, 87
- code
 - adding visual guidelines to, 394–396
 - CodeCompare extension (VS), 402–404
 - Code Definition window, 301
 - Code view, 236

- code (*continued*)
 - Code Window option (Editor Context Menu), 32
 - collapsing with outlining, 242–243
 - commenting/uncommenting in web pages, 217–218
 - creating Regular Expressions within, 415–416
 - documenting with 3
 - keystrokes, 419–421
 - drag and drop into Toolbox, 229
 - Hide Selection, 244–246
 - inserting images into, 391–393
 - setting breakpoints with, 291–292
 - spell-checking, 407–408
 - Task Lists, creating code shortcuts in, 300–301
 - viewing code blocks with vertical lines, 409–411
- code snippets
 - Code Snippets Manager, 261–263
 - HTML, 259–260
 - inserting, 253–256
 - JavaScript, 260–261
 - new, creating from
 - existing, 271–273
 - Snippet Designer, 418–419
 - surrounding existing code with, 256–257
 - using, 258–259
- codes, vstip
 - arguments and switches (commands), 115
 - auto-hide all tool windows, 99
 - auto-populating, stopping
 - Toolbox from, 136
 - AutoRecover function, 10–13
 - Autos window, 379
 - blank lines, accidentally copying, 211
 - blank lines, inserting above/below current line, 218
 - box selections, pasting content between, 238
 - box selections, zero-length, 240
 - Break All Processes When One Process Breaks, 374
 - breakpoint filters, setting, 324
 - breakpoint Hit Count, 314–316
 - breakpoints, adding labels to, 293–294
 - breakpoints, enabling/disabling all, 295–296
 - breakpoints, importing/exporting, 329
 - breakpoints, setting in Call Stack window, 367
 - breakpoints, setting on functions, 316
 - breakpoints, setting with code, 291–292
 - breakpoints window, opening, 293
 - browsers for web development, 96
 - Call Hierarchy window, 310
 - Call Stack window, setting tracepoints in, 369
 - changing templates in New Project/Items dialogs, 80
 - changing visual experience in Visual Studio 2010, 12
 - changing Visual Studio color schemes, 17
 - Class View, creating folders in, 350
 - Class View, searching in, 353
 - Class View, synchronizing, 355
 - Clipboard Ring, cycling through, 220–221
 - closing current document window, 158
 - closing only specified files, 164
 - closing tool windows, 87
 - Code Definition window, 301
 - code snippets, creating new from existing, 271–273
 - code snippets, inserting, 253
 - Code Snippets Manager, 261
 - code snippets, surrounding existing code with, 256
 - code snippets, using, 258
 - Collapse To Definitions with outlining, 246
 - column ordering in tools window, 304
 - command aliases, 113
 - command prompt history, 104
 - command prompt tab completion, 105–107
 - commands, simple, 110–113
 - Commands tab, customizing (toolbars), 30
 - commenting/uncommenting code in web pages, 217–218
 - comments, adding to DataTips, 307–308
 - Common tab/All tab in statement completion, 226
 - conditional breakpoints, setting, 318
 - conditional breakpoints, setting complex, 321
 - creating Web Applications/Virtual Directories in IIS, 46–47
 - CSS versions, choosing, 283
 - current line, cutting/deleting, 213
 - custom file extension associations, 168
 - custom item templates, organizing, 68
 - custom project templates, organizing, 71–74
 - cut/copy/paste collapsed code with outlining, 247
 - cycling through open tool windows, 86
 - DataPoints, importing/exporting, 309
 - DataTips, editing values with, 308
 - DataTips, pinning to source code, 305–306
 - DataTip values, viewing from last debug session, 309
 - default item templates, reorganizing, 74
 - default New Project location, 50
 - default project templates, reorganizing, 77
 - default view in HTML editor, changing, 236
 - Design and Source views in web projects, 235
 - Design view, toggling, 236
 - development servers, using specific port for, 335
 - displaying HTML/CSS warnings as errors, 268
 - docking floating tool windows to previous location, 85
 - docking/undocking tool windows, 107
 - Document Outline (web projects), 251
 - documents on multiple monitors, 155–156
 - editing read-only files, 282–283
 - Esc for returning to Editor return to, 237
 - Exception Assistant, 331–333
 - Exceptions dialog box, 365
 - exporting environment settings, 6
 - exporting external tools list, 143
 - exporting window layouts, 134
 - Export Template Wizard, 64
 - external tools, running, 136–140
 - file open location, 165
 - Find Combo box keyboard shortcuts, 182
 - Find in Files, 186, 190
 - Find in Files search results, customizing, 206
 - finding keyboard shortcuts in Visual Studio, 122
 - Find Symbol, 196–200

- Find Symbol results shortcuts, 200
- floating DataTips, 306–307
- formatting documents/selections for HTML, 265
- formatting on HTML paste, 268
- Go Back markers, navigating with, 277
- Go To Definition for CSS, 194
- Guide Diamond, rearranging windows in Visual Studio 2010 with, 83
- Hide Selection, 244
- HTML code snippets, 259
- HTML Editor tag navigation, 267–268
- IDE navigator, 160
- Image Library (Visual Studio), 147
- importing/changing environment settings, 14
- incremental search, 184
- inserting documents to right of existing tabs, 153
- inserting quotes when typing attribute values, 264
- Intellisense keywords, 215–216
- Intellisense, making transparent, 212
- invoke statement completion, 225
- item templates, creating, 57
- JavaScript code snippets, 260
- JScript Intellisense, updating, 269
- JScript libraries, using in JScript files, 270
- keyboard accelerators for external tools, 141
- keyboard access to tool window toolbar, 103
- Keyboard Mapping Schemes, 125
- Locals window, changing context in, 376
- logging commands, 132
- macros, creating/using, 144–146
- matching braces, selecting/moving between, 224
- moving tool windows with keyboard, 102
- multiple start-up projects, 48–50
- multiple views of same document, 163
- Navigate To dialog, 195
- navigating among tabs in Toolbox, 89
- navigating errors in Errors List, 303
- navigating open document windows, 157
- navigating property tabs in project properties, 55
- Navigation Bar, using, 266
- navigation stack, 275
- Object Browser navigation/references, 362
- Object Browser overview, 356
- Object Browser scope settings, 359
- opening file location from file tab, 158
- outlining, collapsing code with, 242
- parameter information, 227
- Pascal case (Intellisense), 216
- pasting single selection into box selections, 239
- pinning projects to Recent Projects list, 56
- Properties window, 249
- Quick Find, 172
- Quick Info option, 228
- Quick Replace (searching), 176–178
- QuickWatch, 343–345
- recent files, 154
- Recent Project templates in New Project dialog, 44
- reference highlighting, 223
- removing projects from Recent Projects list, 9
- repeating last search in VS, 171–172
- Replace in Files basic options, 192
- ResetSettings switch, 39–41
- resetting all keyboard shortcuts, 131
- running commands, 121
- running multiple versions of Visual Studio side-by-side, 3
- Run To Cursor, 330
- saving changes before building, 302
- searching breakpoints, 312–313
- searching currently selected string without Find window, 185
- searching for project templates, 43–45
- searching in Toolbox, 88
- selecting from current cursor to last Go Back marker, 278
- shortcuts, creating new, 127
- showing hidden tool windows with Auto Hide Channel, 100–101
- show previous file versions, 166
- Solution Folders, 54
- tagged expressions (Replace in Files), 203–206
- Tag Specific options, 285
- Task Lists, creating code shortcuts in, 300
- Task Lists, creating custom tokens for, 297–300
- temporary projects, creating, 56
- testing commands, 119
- text, replacing text with box selection, 237
- TODD comments in Task List, 296
- Toolbars tab, customizing, 27
- tabs in Toolbox, 93
- Toolbox, drag and drop into, 229–230
- Toolbox, expanding/collapsing all in, 87
- tool window animations, changing, 14
- tracepoints, setting in source code, 325
- tracing, application/page level, 336–339
- tracking active items in Solution Explorer, 52
- tracking changes in Editor, 280–281
- transposing lines/words/characters, 219–220
- type-ahead selection support in Solution Explorer, 52–53
- Undo Quick Replace/Replace in Files, 179
- Undo/Redo global actions, 222
- Undo/Redo stack, 221
- using older frameworks with multi-targeting, 46
- using smart tags from keyboards, 231–232
- using statements, organizing (C#), 232
- visualizers (Watch windows), 345
- Visual Studio logging, 37–38
- Visual Studio Online Help, table of contents in, 4
- watches, adding from variable windows, 348
- Watch window, 340
- white space, viewing, 241
- window layouts, 90, 91
- word completion, 229
- word wrap, 248–249
- WPF Tree Visualizer, 371–373
- zoom in/out of text with Editor, 209
- code, writing
 - blank lines, avoiding copying, 211–212
 - blank lines, inserting above/below current line, 218
 - Clipboard Ring, cycling through, 220–221
 - code snippets. *See* code snippets

code, writing (*continued*)

- Collapse to Definitions with outlining, 246–247
- collapsing code with outlining, 242–243
- commenting/uncommenting in web pages, 217–218
- CSS versions, choosing, 283–285
- current line, cutting/deleting, 213–214
- cut/copy/paste collapsed code with outlining, 247
- default view, changing in HTML editor, 236–237
- Design and Source views, switching between, 235
- Design view, toggling, 236
- Document Outline (web projects), 251–253
- documents/selections, formatting for HTML, 265–266
- drag and drop into Toolbox, 229–230
- editing read-only files, 282–283
- Esc to return to Editor, 237
- formatting HTML on paste, 268
- Hide Selection, using, 244–246
- HTML/CSS warnings, displaying as errors, 268–269
- HTML Editor tag navigation, 267–268
- inserting quotes when typing attribute values, 264
- JScript Intellisense, updating, 269
- JScript libraries, 270–271
- keywords (Intellisense), 215
- matching braces, moving/selecting between, 224–225
- moving between Common tab/All tab in statement completion, 226
- navigating backwards/forwards with Go Back markers, 277–278
- Navigation Bar, using, 266–267
- navigation stack, 275–276
- parameter information, using, 227–228
- Pascal case (Intellisense), 216
- pasting contents between box selections, 238–239
- pasting single selections into box selection, 239–240
- Properties window keyboard shortcuts, 249–250
- Quick Info, using, 228–229
- reference highlighting, 223–225
- replacing text with box selection, 237–238
- selecting from current cursor to last Go Back marker, 278–279
- smart tags, using from keyboard, 231–232
- statement completion, invoking, 225–226
- Tag Specific options. *See* Tag Specific Options dialog box
- text, zooming in/out in Editor, 209–211
- tracking changes in Editor, 280–281
- transposing lines/words/characters, 219–220
- Undo/Redo global actions, 222
- Undo/Redo stack, using, 221–222
- using statements, organizing, 232–235
- white space, viewing, 241–242
- word completion, 229
- word wrap, 248–249
- zero-length box selection, 240–241
- collapsing code
 - collapse to definitions with outlining, 246–247
 - cut/copy/paste with outlining, 247
 - with outlining, 242–243
- colors
 - Color Theme Editor extension (Visual Studio), 389–391
 - importing, 17–21
- color schemes, changing
 - colors, importing, 21–24
 - obtaining fonts and colors, 17–21
 - resetting colors, 24
- column ordering in tools window, 304–305
- combo box, Zoom, 210
- command prompt
 - history, 104–105
 - tab completion, 105–107
- commands
 - aliases, 113–115
 - arguments and switches, 115–119
 - Commands tab, customizing (toolbars), 30–31
 - hidden Command tool window, 86
 - logging, 132–134
 - running, 121–123
 - simple, 110–113
 - testing, 119–121
 - “Understanding Commands: Aliases”, 119
 - “Understanding Commands: Running Commands”, 120
- Command Window
 - aliases, exporting, 143–145
 - aliases, setting, 16
 - defined, 122
- comments
 - adding to DataTips, 307–308
 - commenting/uncommenting code in web pages, 217–218
 - Comment/Uncomment items (Modify Selection), 35
- Common Elements source files (VS Image Library), 148–149
- Common tab/All tab in statement completion, 226
- %comspec% variable, 138
- conditional breakpoints, 318–320, 321–323
- controls
 - Control box (Visual Studio 2010), 108
 - Controls area for modifying menus/toolbars, 32–33
 - Controls dialog box (Commands tab), 36
 - custom (Toolbox), 95
- copying
 - blank lines accidentally, 211
 - Copy (Ctrl+C) command (Find Symbol), 203
 - Copy Exception Detail To The Clipboard link (Exception Assistant), 333
 - cut/copy/paste collapsed code with outlining, 247
- “Create a Shortcut Key for a Macro”, 146
- CSS (Cascading Style Sheets)
 - Go To Definition for, 194
 - versions, choosing, 283–285
 - warnings, displaying as errors, 268–269
- current and previous statements (Autos window), 379
- Current Project action
 - Find What combo box, 189
 - Quick Find, 173
- Custom Component Set
 - editing, 360–362
 - options (Look In dialog), 198
- customizing
 - Commands tab, 30–37
 - custom controls (Toolbox), 95
 - custom file extension associations, 168–169
 - custom item templates, organizing, 68–71

- “Customize How Find In Files Results Are Displayed in the Find Results Window”, 191
- Customize dialog box, 29
- custom project templates, organizing, 71–74
- Document Tab Well user interface, 417–418
- Find in Files search results, 206–208
- GhostDoc extension (VS), 421
- StructureAdornment extension (VS), 410–411
- StudioShell module, 423
- Toolbars tab, 27–29
- Visual Studio with Windows PowerShell, 421–423
- cutting
 - current line, 213–214
 - cut/copy/paste collapsed code with outlining, 247
- cycling through Clipboard Ring, 220–221

D

- data, refreshing (Watch window), 346–347
- DataTips
 - adding comments to, 307–308
 - creating floating, 306–307
 - editing values with, 308
 - importing/exporting, 309
 - pinning to source code, 305–306
 - values, viewing from last debug session, 309
- debugging
 - application-level/page-level tracing, 336–339
 - Autos window, 379–382
 - Break All Processes When One Process Breaks, 374–376
 - breakpoints. *See* breakpoints
 - Call Hierarchy window, 310–312
 - Call Stack window, setting breakpoints in, 367–368
 - changing context in Locals window, 376–379
 - Class View, creating folders in, 350–352
 - Class View, synchronizing, 355
 - Code Definition window, 301
 - column ordering in tools window, 304–305
 - Debug.AddWatch command, 349
 - Debug Location toolbar, 377
 - Debug Mode (window layouts), 90, 92

- development server, using specific port for, 335
- Exception Assistant, 331–333
- Exceptions dialog box, 365–367
- multicolumn sorting, 305
- navigating errors in Errors List, 303–304
- Object Browser. *See* Object Browser
- QuickWatch, 343–345
- refreshing data (Watch window), 346–347
- Run To Cursor, 330
- saving changes before building, 302–303
- searching in Class View, 353–355
- setting breakpoints in source code, 325–328
- Task Lists, creating code shortcuts in, 300–301
- Task Lists, creating custom tokens for, 297–300
- TODO comments in Task List, 296–297
- tracepoints, setting in Call Stack window, 369–371
- visualizers (Watch windows), 345–346
- watches, adding from variable windows, 348–350
- Watch window, 340–343
- WPF Tree Visualizer, 371–373
- dedicated style sheets (CSS), 283
- default browsers, changing, 97
- default development settings, 26
- DefaultFileOpenLocation (registry), 166
- default item templates, reorganizing, 74–78
- default New Project location, changing, 50–51
- default project templates, reorganizing, 77–80
- default toolbars, 29
- DefaultTracepointMessage string value, 329
- deleting
 - aliases (commands), 115
 - browsers, 98
 - current line, 213–214
 - custom toolbars, 29
 - folders in Class View, 352
 - projects from Recent Projects list, 9
 - Solution Folders, 55
 - tabs (Toolbox), 95
 - unused using directives, 232–234
- <deployment retail=true> setting, 336

- design
 - Designer, viewing code in, 124–125
 - Design Mode (window layouts), 91
- Design View
 - toggling, 236
 - window layouts, 90
- detachable document windows, 155
- development servers, using specific port for, 335
- development settings, resetting, 25–28
- devenv.com command, 76, 79
- “Disable Mouse Wheel Zoom” extension, 210
- Display File Names Only (Find Results window), 191
- /doc or /d switch, 117
- docking
 - Dock As Tabbed Document option, 156
 - docking floating tool window to previous location, 85
 - docking/undocking tool windows, 107–110
 - Dock menu, 102
- documents
 - closing current document window, 158
 - closing only specified, 164–165
 - custom document extension associations, 168–169
 - “Document Outline: Web Project”, 267
 - document open location, 165–166
 - Document Outline (web projects), 251–253
 - Document Tab Well user interface, 417–418
 - Document This command, 419
 - editing read-only, 282–283
 - formatting for HTML, 265–266
 - IDE navigator, 160–162
 - inserting to right of existing tabs, 153–154
 - multiple views of same document, 163–164
 - navigating open document windows, 157–158
 - on multiple monitors, 155–156
 - recent, 154–155
 - saving changes to open, 302
 - show previous versions, 166–167
- drag and drop code into Toolbox, 229–230
- Dynamic Tab Wells, 417

E

editing

- Custom Component Set, 360–362
- Edit Custom Component Set dialog box, 198
- Edit.Find command, 116–117
 - read-only files, 282–283
 - values with DataTips, 308
- Editor
 - Editor Context Menu (toolbars), 32
 - Editor Guideline extension (VS), 394–396
 - running VIM commands in, 406–407
 - tracking changes in, 280–281
 - using Emacs commands in, 400–401
 - using Esc to return to, 237
 - zoom in/out of text with, 209–211
- Emacs Commands extension (VS), 400–401
- embedded styles (HTML), 284–285
- Enable Rich Client Visual Experience option, 13
- environment settings
 - exporting, 6–9
 - importing/changing, 14–16
- errors
 - displaying HTML/CSS warnings as, 268–269
 - Error List tool window, 86
 - navigating in Errors List, 303–304
- Esc for returning to Editor, 237
- Exception Assistant, 331–333
- Exceptions dialog box, 365–367
- exporting
 - breakpoints, 329
 - Command Window
 - aliases, 143–145
 - DataPoints, 309
 - environment settings, 6–9
 - “Export Your Window Layouts”, 39
 - “Exporting Your Environment Settings”, 27
 - Export Selected Environment Settings, 134, 143
 - Export Template Wizard, 58–60, 64–67
 - external tools list, 143–145
 - Import and Export Settings Wizard, 25
 - Window Layouts, 134–136
- expressions
 - Expression Builder, 174, 204
 - Regular Expressions. *See* Regular Expressions

- “Side Effects and Expressions”, 347
- tagged (Replace in Files), 203–206
- watch, 340–343
- extensions
 - file, 168–169
 - Visual Studio. *See* Visual Studio extensions
- external tools
 - creating keyboard accelerators for, 141–142
 - exporting list, 143–145
 - External Tools List setting, 16
 - running, 136–140
 - “Using External Tools”, 141

F

files

- closing only specified, 164–165
- current, syncing Solution Explorer to, 396–397
- custom file extension associations, 168–169
- Display File Names Only (Find Results window), 191
- editing read-only, 282–283
- File menu, 154
- filename argument (logging), 133–134
- file open location, 165–166
- File variables (customizing search results), 207
- File View (window layouts), 90
- Find In Files, 190–191
- Go To File (Find/Command box), 183
- opening file menu drop-down list from keyboard, 159
- opening location from file tab, 158
- recent, 154–155
- recovered. *See* AutoRecover feature
- Replace in Files, 180–182, 192–194, 203–206
- show previous versions, 166–167
- Skip File button (Replace in Files), 193
- Visual Studio Image Library, 147–148
- filters, setting for
 - breakpoints, 324–325
- finding. *See also* searching
 - “Find Command” documentation, 116
 - “Find Keyboard Shortcuts”, 121
 - Find All option (Find Symbol), 199
 - Find And Replace dialog box, 116
 - Find Combo box, 120, 122, 171, 182–184
 - Find/Command box, 182
 - Find In Files, 186–190, 190–191
 - Find In Files search results, customizing, 206–208
 - finding keyboard shortcuts (Visual Studio), 122–125
 - Find Next button (Quick Find), 176
 - Find Next button (Quick Replace), 178
 - Find Next button (Replace in Files), 193
 - Find options (Find Symbol), 199
 - Find options (Replace in Files), 192
 - Find Result Format string (registry), 207
 - Find Results windows, 190
 - Find What area (Quick Replace), 177
 - Find What combo box, 187–190
 - Find What field, 197
 - Quick Find. *See* Quick Find
- Find Symbol
 - basics, 196–200
 - results shortcuts, 200–203
- floating DataTips, 306–307
- floating tool window, docking to previous location, 85
- folders
 - creating in Class View, 350–352
 - Solution Folders, 54–55
- fonts, importing, 17–21
- Ford, Sara, 402
- Foreground And Background Colors In A Console Window snippet, 263
- formatting
 - documents/selections for HTML, 265–266
 - HTML on paste, 268
- forward slash (/) to collapse
 - Toolbox, 87
- frameworks
 - Framework X/Silverlight X option (Look In dialog), 198
 - older, using with multi-targeting, 46
- Friendly Name field (browsers), 97
- Full Screen Mode (window layouts), 93
- Full Screen View (window layouts), 90
- functions, setting breakpoints on, 316–317

G

GhostDoc tool (VS), 419–421
 global actions, Undo/Redo, 222
 Go Back markers, 277–278, 278–279
 Gopp, Benjamin, 408
 Go To Declaration (Ctrl+F12), 201
 Go To Definition (F12), 200
 Go To Definition for CSS, 194
 Go To File (Find/Command box), 183
 Go To Line (Find/Command box), 183
 Go To Reference (Shift+F12), 201–202
 Granger, Chris, 407
 Guide Diamond, 83–84
 guidelines, adding to code, 394–396

H

Harrington, Paul, 37, 394
 Has Changed condition (breakpoints), 319–320
 Help Library Manager, 4
 Help Online area (Exception Assistant), 332–333
 Hide External Items option (Navigate To), 196
 Hide Selection (code), 244–246
 highlighting, reference, 223–225
 history, command prompt, 104–105
 Hit Count, breakpoint, 314–316
 Horst, Bill, 334
 “How to Customize What Files to Search with Find in Files”, 189
 HTML (Hypertext Markup Language)
 code snippets, 259–260
 editor, changing default view in, 236–237
 embedded styles, 284–285
 formatting documents/selections for, 265–266
 formatting on paste, 268
 HTML Editor tag navigation, 267–268
 HTML Visualizers, 345
 warnings, displaying as errors, 268–269

I

Icon Image field (Export Template Wizard), 61, 66
 icons, refresh, 347–348
 IDE navigator, 160–162

#If DEBUG compiler option, 292
 IIS (Internet Information Services), 46–47
 illustrations
 4 modes of window layouts, 90
 Add New Item dialog box, 77
 AllMargins extension, 411–412
 animating environment tools, 14
 auto-hide all tool windows, 99–100
 AutoRecover, 10–11
 Autos window, 379–381
 blank lines, accidentally copying, 211
 blank lines, inserting above/below current line, 218
 box selections, pasting content between, 238–239
 box selections, pasting single selection into, 239–240
 box selections, replacing text with, 238
 box selections, zero-length, 240–241
 Break All Processes When One Process Breaks, 374–375
 breakpoint filters, setting, 324
 breakpoint Hit Count, 314–315
 breakpoints, adding labels to, 294
 breakpoints, enabling/disabling all, 295–296
 breakpoints, importing/exporting, 329
 breakpoints, setting in Call Stack window, 368
 breakpoints, setting on functions, 316–317
 breakpoint windows, opening, 293
 browsers for web development, 96–98
 Call Hierarchy, 310–311
 Call Stack window, setting tracepoints in, 369
 Choose Settings To Export dialog box, 7
 Classic view (VS 2010 Online Help), 5–6
 Class View, creating folders in, 350–352
 Class View, searching in, 353–354
 Class View, synchronizing, 355
 Clipboard Ring, cycling through, 220–221
 CodeCompare extension (VS), 403
 Code Definition Window, 301
 code snippets, creating new from existing, 272–273

Code Snippets, inserting, 253–255
 Code Snippets Manager, 262–263
 code snippets, surrounding existing code with, 256–257
 color schemes, resetting, 24
 column ordering in tools window, 304–305
 commands, running, 118–119
 Commands tab, customizing (toolbars), 30–37
 Command Window, 113
 commenting/uncommenting code in web pages, 217–218
 comments, adding to DataTips, 307
 conditional breakpoints, setting, 318–320
 conditional breakpoints, setting complex, 321–323
 custom file extension associations, 168
 custom item templates, organizing, 68–70
 custom project templates, organizing, 71–72
 DataTips, pinning to source code, 305–306
 DataTip values, viewing from last debug session, 309
 Debug Mode (windows layouts), 92
 default item templates, reorganizing, 74–76
 default project templates, reorganizing, 78–80
 default view in HTML editor, changing, 236–237
 Design and Source views in web projects, 235–236
 Design Mode, 91
 development servers, using specific port for, 335
 development settings, resetting, 25–28
 docking floating tool window to previous location, 85
 Document Outline (web projects), 252–253
 documents on multiple monitors, 155–156
 document window, closing current, 158
 editing read-only files, 282–283
 Editor Guidelines extension (VS), 394–395
 Exception Assistant, 331–333
 Exceptions dialog box, 365–366

illustrations (*continued*)

- exporting Command Window
 - Aliases/External Tools List, 143–145
- exporting environment settings, 6–8
- Export Template Wizard, 58, 65–67
- external tools, running, 136–140
- file menu drop-down list, opening
 - from keyboard, 159
- file open location, 165–166
- files, closing only
 - specified, 164–165
- files, recent, 154–155
- Find And Replace dialog box, 116
- Find in Files, 186–190
- Find in Files search results, customizing, 206–208
- finding keyboard shortcuts, 122–125
- Find Symbol, 197
- Find Symbols results shortcuts, 200–202
- floating DataTips, 306–307
- Guide Diamond, 84
- Help Library Manager, 4–6
- Hide Selection, 244–245
- HTML code snippets, 259–260
- HTML/CSS warnings, displaying as errors, 268
- HTML Editor tag
 - navigation, 267–268
- IDE navigator, 160–162
- Image Insertion tool (Visual Studio), 392–393
- Import And Export Settings Wizard, 6–7
- importing/changing environment settings, 14–16
- importing colors, 21–24
- incremental search, 184–185
- inserting documents to right of existing tabs, 153–154
- Intellisense keywords, 215–216
- Intellisense, making
 - transparent, 212
- invoke statement completion, 226
- item templates, 63
- JavaScript code snippets, 260–261
- JScript Intellisense, updating, 269
- JScript libraries, 270–271
- keyboard accelerators for external tools, 141–142
- Keyboard Mapping
 - Schemes, 125–127
- keyboard shortcuts, resetting, 131–133
- Local Internet Information Server dialog, 46
- Locals window, changing context in, 376–378
- logging commands, 133–134
- macros, creating/using, 144–146
- multiple Startup Projects, 48–49
- multiple versions of Visual Studio, running side-by-side, 3
- multiple views of same document, 163–164
- navigating errors in Errors List, 304
- navigating open document windows, 157–158
- Navigation Bar, using, 266–267
- navigation stack, 275–276
- New Item dialog box, 74
- New Project dialog box, 44–46, 72
- New Project locations, changing default, 51
- Object Browser navigation/references, 362–364
- Object Browser
 - overview, 357–358
- Object Browser scope settings, 359–362
- outlining, collapsing code with, 242–243
- parameter information, 227
- Pascal case (Intellisense), 216
- Quick Find, 172–176
- Quick Replace, 176–178
- Quickwatch, 343–345
- quotes, inserting when typing attribute values, 264
- Recent Projects list, 56
- Recent Projects list, removing projects from, 9
- reference highlighting, 223–224
- Regex Editor, 415
- repeating last search in VS, 171–172
- Replace in Files basic options, 192–194
- /ResetSettings switch, 39–40
- Run To Cursor, 330
- safe mode (Visual Studio), 38
- saving changes before building, 302–303
- searching breakpoints, 312–313
- searching currently selected string without Find window, 185
- searching in Toolbox, 88
- selecting from current cursor to last Go Back marker, 278–279
- selecting/moving between matching braces, 224–225
- Settings To Export dialog box, 7
- shortcuts, creating new, 128–130
- Show Previous Versions option (documents), 166–167
- simple commands, 110–113
- smart tags, using from keyboards, 231–232
- Snippet Designer, 418
- Solution Explorer Tools extension, 397
- StructureAdornment extension (VS), 409
- tabs in Toolbox, 93
- tagged expressions (Replace in Files), 203–206
- Task Lists, creating code shortcuts in, 300
- Task Lists, creating custom tokens for, 297–300
- templates in New Project/Item dialogs, changing, 81–82
- testing commands, 120–121
- TODO comments in Task List, 296–297
- Toolbars tab, customizing, 27–29
- Toolbox, 87
- Toolbox, drag and drop code into, 229–230
- tracepoints, setting in source code, 325–328
- tracing, application/page level, 336–338
- tracking active items in Solution Explorer, 52
- tracking changes in Editor, 280–281
- transposing lines/words/characters, 219–220
- type-ahead selection support in Solution Explorer, 53
- Undo Quick Replace/Replace in Files, 179–182
- Undo/Redo global actions, 222
- Undo/Redo stack, 221–222
- using statements, organizing, 232–235
- visualizers, 345–346
- Visual Studio 2010, changing visual experience in, 12
- Visual Studio 2010 Online Help, table of contents in, 4–6
- Visual Studio color schemes, changing, 18–21
- Visual Studio Image Library, 147–152
- Visual Studio logging, 37
- watch expressions, 340
- white space, viewing, 242

- word completion, 229
- word wrap, 248–249
- WPF Tree Visualizer, 371–373
- zoom in/out of text with editor, 210–211
- images. *See also* Images Library (Visual Studio)
 - Image Insertion tool, 391–393
- Immediate Window, 122
- importing
 - breakpoints, 329
 - colors, 17–21, 21–24
 - DataPoints, 309
 - fonts, 17–21
 - Import and Export settings, 16
 - Import and Export Settings Wizard, 6, 14, 25
 - importing or changing environment settings, 14–16, 135
 - Import Selected Environment Settings option, 22
- includes operations, 195
- Include Sub-Folders option (Find What combo box), 189
- incremental search, 184–185
- Initial Directory, 139
- Insert Documents To The Right Of Existing Tabs option, 153
- installing
 - CodeCompare extension (VS), 403
 - Editor Guidelines extension (VS), 394
 - “Installing Visual Studio Versions Side-by-Side”, 3
 - Visual Studio extensions, 385–387
 - Win7 Taskbar Extension, 413
- Intellisense
 - JScript Intellisense, updating, 269
 - keywords, 215–216
 - making transparent, 212
 - Pascal case and, 216
 - Use IntelliSense To Verify The Function Name, 317
 - XAML Intellisense Presenter extension, 395–396
- invoke statement
 - completion, 225–226
- Is True condition (breakpoints), 319–320
- items
 - active, tracking in Solution Explorer, 52–53
 - adding to custom tabs (Toolbox), 94–95
 - New Items dialog, changing templates in, 80–82
 - item templates
 - creating, 57–64
 - custom, organizing, 68–71
 - default, reorganizing, 74–78
 - in Items dialog, changing, 80–82
- J**
 - JavaScript code snippets, 260–261
 - Johnson, Matthew, 389
 - JScript
 - Intellisense, updating, 269
 - libraries, using in JScript files, 270–271
- K**
 - Keep Modified Files Open After Replace All option, 181, 191, 193
 - keyboard accelerators, 141–142
 - keyboard shortcuts
 - Additional Keyboard Mapping Schemes, 125–127
 - arguments and switches (commands), 115
 - auto-populating, stopping
 - Toolbox from, 136
 - AutoRecover function, 10–13
 - Autos window, 379
 - blank lines, accidentally copying, 211
 - blank lines, inserting above/below current line, 218
 - box selections, pasting content between, 238
 - box selections, zero-length, 240
 - breakpoints, adding labels to, 293–294
 - breakpoints, enabling/disabling all, 295–296
 - breakpoints, setting in Call Stack window, 367
 - breakpoints, setting on functions, 316
 - breakpoints window, opening, 293
 - Call Hierarchy window, 310
 - changing visual experience in Visual Studio 2010, 12
 - Class View, searching in, 353
 - Clipboard Ring, cycling through, 220–221
 - Code Definition window, 301
 - code snippets, inserting, 253
 - Code Snippets Manager, 261
 - code snippets, surrounding existing code with, 256
 - code snippets, using, 258
 - Collapse To Definitions with outlining, 246
 - command aliases, 113
 - command prompt history, 104
 - commands, running, 121
 - Commands tab, customizing (toolbars), 30
 - Common tab/All tab in statement completion, 226
 - CSS versions, choosing, 283
 - current document window, closing, 158
 - current line, cutting/deleting, 213
 - DataPoints, importing/exporting, 309
 - default view in HTML editor, changing, 236
 - Design and Source views in web projects, 235
 - Design view, toggling, 236
 - development settings, resetting, 25
 - docking floating tool windows to previous location, 85
 - docking/undocking tool windows, 107
 - Dock menu, 102
 - Document Outline (web projects), 251
 - documents on multiple monitors, 155–156
 - Esc for returning to Editor return to, 237
 - Exceptions dialog box, 365
 - exporting environment settings, 6
 - exporting external tools list, 143
 - exporting window layouts, 134
 - Export Template Wizard, 64
 - external tools, running, 136–140
 - file extension associations, custom, 168
 - files, closing only specified, 164
 - files, recent, 154
 - “Find Keyboard Shortcuts”, 121
 - Find Combo box, 182–184
 - Find In Files, 186, 190
 - Find In Files search results, customizing, 206
 - finding (Visual Studio), 122–125
 - Find Symbol, 196–200
 - Find Symbol results shortcuts, 200
 - Go Back markers, navigating with, 277
 - Go To Definition for CSS, 194
 - Guide Diamond, rearranging windows in Visual Studio 2010 with, 83
 - Hide Selection, 244

keyboard shortcuts (*continued*)
 HTML code snippets, 259
 HTML/CSS warnings, displaying as errors, 268
 HTML, formatting documents/ selections for, 265
 importing/changing environment settings, 14
 incremental search, 184
 inserting documents to right of existing tabs, 153
 inserting quotes when typing attribute values, 264
 Intellisense keywords, 215–216
 Intellisense, making transparent, 212
 invoke statement completion, 225
 item templates, creating, 57
 JavaScript code snippets, 260
 JScript Intellisense, updating, 269
 keyboard accelerators for external tools, 141
 Keyboard Mapping Schemes, 125
 “Keyboard Shortcuts: Creating New Shortcuts”, 112
 “Keyboard Shortcuts Reset All Your Shortcuts”, 131
 Locals window, changing context in, 376
 macros, creating/using, 144–146
 matching braces, selecting/moving between, 224
 moving tool windows with, 102–103
 multiple start-up projects, 48–50
 multiple views of same document, 163
 multi-targeting, using older frameworks with, 46
 Navigate To dialog, 195
 navigating errors in Errors List, 303
 navigating property tabs, 55
 Navigation Bar, using, 266
 navigation stack, 275
 New Project location, changing default, 50
 Object Browser navigation/ references, 362
 Object Browser overview, 356
 Object Browser scope settings, 359
 outlining, collapsing code with, 242
 parameter information, 227
 Pascal case (Intellisense), 216
 pasting single selection into box selections, 239

posters, 425
 Properties window, 249–250
 Quick Find, 172
 Quick Info option, 228
 Quick Replace (searching), 176–178
 QuickWatch, 343–345
 read-only files, editing, 282
 Recent Projects list, removing projects from, 9
 Recent Project templates in New Project dialog, 44
 reference highlighting, 223
 repeating last search in VS, 171–172
 Replace in Files basic options, 192
 Run To Cursor, 330
 saving changes before building, 302
 searching currently selected string without Find window, 185
 searching for project templates, 43–45
 selecting from current cursor to last Go Back marker, 278
 shortcuts, creating new, 127, 127–131
 shortcuts, resetting all, 131–133
 smart tags, using from keyboards, 231–232
 Solution Folders, 54
 System menu, 102
 tagged expressions (Replace in Files), 203–206
 Tag Specific options, 285
 Task Lists, creating code shortcuts in, 300
 Task Lists, creating custom tokens for, 297–300
 temporary projects, creating, 56
 testing commands, 119
 text, replacing with box selection, 237
 Toolbars tab, customizing, 27
 tool window animations, changing, 14
 tool window toolbar, access to, 103
 tracking active items in Solution Explorer, 52
 tracking changes in Editor, 280–281
 transposing lines/words/ characters, 219–220
 type-ahead selection support in Solution Explorer, 52–53
 Undo Quick Replace/Replace in Files, 179

Undo/Redo global actions, 222
 Undo/Redo stack, 221
 for using smart tags, 231–232
 using statements, organizing (C#), 232
 Visual Basic 6. *See* Visual Basic 6 keyboard shortcuts
 visualizers (Watch windows), 345
 Visual Studio 6. *See* Visual Studio 6 keyboard shortcuts
 Visual Studio 2010 Online Help, table of contents in, 4
 Visual Studio color schemes, changing, 17
 Watch window, 340
 Web Applications/Virtual Directories, creating in IIS, 46–47
 white space, viewing, 241
 word completion, 229
 word wrap, 248–249
 for zooming, 211
 zoom in/out of text with editor, 209
 keywords, Intellisense, 215–216
 Kurata, Deborah, 164

L

labels, adding to breakpoints, 293–294
 layouts, window. *See* Window Layouts
 Lightweight view (VS 2010 Online Help), 5
 lines
 blank, accidentally copying, 211
 blank, inserting above/below current line, 218
 cutting/deleting current, 213–214
 transposing, 219–220
 lists, opening from keyboard, 159
 loading Visual Studio, 37–38
 Locals window, 348, 376–379
 Location variables (customizing search results), 207
 logging (Visual Studio), 37–38, 132–134
 Look at These File Types option (Find What combo box), 189
 Look In area (Quick Find), 173–174
 Look In dialog (Find Symbol), 197–199
 Look in drop-down list (Find What combo box), 189

M

- Macro Explorer, 146
- macros, creating/using, 144–146
- Manela, Matt, 418
- Mapping Schemes,
 - Keyboard, 125–127
- /markall or /m switch, 117
- Match Case option
 - Find Symbol, 199
 - Find What combo box, 187
 - Quick Find, 174
- matching braces, moving/selecting between, 224–225
- Match option (Find Symbol), 199
- Match Whole Word option
 - Find What combo box, 187
 - Quick Find, 174
- Members list, 266–267
- menu/command references
 - arguments and switches (commands), 115
 - auto-hide all tool windows, 99
 - auto-populating, stopping
 - Toolbox from, 136
 - AutoRecover function, 10–13
 - Autos window, 379
 - blank lines, accidentally copying, 211
 - blank lines, inserting above/below current line, 218
 - box selections, pasting content between, 238
 - box selections, zero-length, 240
 - Break All Processes When One Process Breaks, 374
 - breakpoint filters, setting, 324
 - breakpoint Hit Count, 314–316
 - breakpoints, adding labels to, 293–294
 - breakpoints, enabling/disabling all, 295–296
 - breakpoints, importing/exporting, 329
 - breakpoints, setting in Call Stack window, 367
 - breakpoints, setting on functions, 316
 - breakpoints window, opening, 293
 - browsers for web development, 96
 - Call Hierarchy window, 310
 - Class View, creating folders in, 350
 - Class View, searching in, 353
 - Class View, synchronizing, 355
 - Clipboard Ring, cycling through, 220–221
 - Code Definition window, 301
 - code snippets, inserting, 253
 - Code Snippets Manager, 261
 - code snippets, surrounding existing code with, 256
 - code snippets, using, 258
 - Collapse To Definitions with outlining, 246
 - command aliases, 113
 - commands, running, 121
 - Commands tab, customizing (toolbars), 30
 - commenting/uncommenting code in web pages, 217–218
 - Common tab/All tab in statement completion, 226
 - conditional breakpoints, setting, 318
 - conditional breakpoints, setting complex, 321
 - CSS versions, choosing, 283
 - current line, cutting/deleting, 213
 - cycling through open tool windows, 86
 - DataPoints, importing/exporting, 309
 - DataTips, pinning to source code, 305–306
 - default New Project location, changing, 50
 - default view in HTML editor, changing, 236
 - Design and Source views in web projects, 235
 - Design view, toggling, 236
 - development settings, resetting, 25
 - docking floating tool windows to previous location, 85
 - docking/undocking tool windows, 107
 - Document Outline (web projects), 251
 - documents on multiple monitors, 155–156
 - document windows, closing current, 158
 - Editor, tracking changes in, 280–281
 - Esc for returning to Editor, 237
 - Exceptions dialog box, 365
 - exporting environment settings, 6
 - exporting external tools list, 143
 - exporting window layouts, 134
 - Export Template Wizard, 64
 - external tools, running, 136–140
 - file extension associations, custom, 168
 - file location, opening from file tab, 158
 - file open location, 165
 - files, closing only specified, 164
 - Find Combo box keyboard shortcuts, 182
 - Find In Files, 186, 190
 - Find In Files search results, customizing, 206
 - finding keyboard shortcuts in Visual Studio, 122
 - Find Symbol, 196–200
 - Find Symbol results shortcuts, 200
 - formatting on HTML paste, 268
 - Go Back markers, navigating with, 277
 - Go To Definition for CSS, 194
 - Guide Diamond, rearranging windows in, 83
 - Hide Selection, 244
 - HTML code snippets, 259
 - HTML/CSS warnings, displaying as errors, 268
 - HTML, formatting documents/selections for, 265
 - IDE navigator, 160
 - importing/changing environment settings, 14
 - incremental search, 184
 - inserting documents to right of existing tabs, 153
 - inserting quotes when typing attribute values, 264
 - Intellisense keywords, 215–216
 - invoke statement completion, 225
 - item templates, creating, 57
 - JavaScript code snippets, 260
 - JScript Intellisense, updating, 269
 - keyboard accelerators for external tools, 141
 - Keyboard Mapping Schemes, 125
 - keyboard shortcuts, resetting all, 131
 - Locals window, changing context in, 376
 - logging commands, 132
 - macros, creating/using, 144–146
 - matching braces, selecting/moving between, 224
 - moving tool windows with keyboard, 102
 - multiple start-up projects, 48–50
 - multiple views of same document, 163
 - multi-targeting, using older frameworks with, 46
 - Navigate To dialog, 195
 - navigating errors in Errors List, 303

menu/command references

(continued)

- navigating open document windows, 157
- Navigation Bar, using, 266
- navigation stack, 275
- Object Browser navigation/ references, 362
- Object Browser overview, 356
- Object Browser scope settings, 359
- outlining, collapsing code with, 242
- parameter information, 227
- Pascal case (Intellisense), 216
- pasting single selection into box selections, 239
- Properties window, 249
- Quick Find, 172
- Quick Info option, 228
- Quick Replace (searching), 176–178
- QuickWatch, 343–345
- read-only files, editing, 282
- recent files, 154
- Recent Projects list, removing projects from, 9
- Recent Project templates in New Project dialog, 44
- reference highlighting, 223
- Replace in Files basic options, 192
- Run To Cursor, 330
- saving changes before building, 302
- searching currently selected string without Find window, 185
- searching for project templates, 43–45
- searching in Toolbox, 88
- selecting from current cursor to last Go Back marker, 278
- shortcuts, creating new, 127
- smart tags, using from keyboards, 231–232
- Solution Explorer, tracking active items in, 52
- Solution Folders, 54
- tagged expressions (Replace in Files), 203–206
- Tag Specific options, 285
- Task Lists, creating code shortcuts in, 300
- Task Lists, creating custom tokens for, 297–300
- temporary projects, creating, 56
- testing commands, 119
- text, replacing text with box selection, 237
- Toolbars tab, customizing, 27
- tabs in Toolbox, 93
- tool window animations, changing, 14
- tool windows, closing, 87
- tracepoints, setting in source code, 325
- transposing lines/words/ characters, 219–220
- type-ahead selection support in Solution Explorer, 52–53
- Undo Quick Replace/Replace in Files, 179
- Undo/Redo global actions, 222
- Undo/Redo stack, 221
- using statements, organizing (C#), 232
- visualizers (Watch windows), 345
- Visual Studio 2010, changing visual experience in, 12
- Visual Studio 2010 Online Help, table of contents in, 4
- Visual Studio color schemes, changing, 17
- Visual Studio, repeating last search in, 171–172
- watches, adding from variable windows, 348
- Watch window, 340
- white space, viewing, 241
- word completion, 229
- word wrap, 248–249
- zoom in/out of text with editor, 209
- menus, rearranging, 32
- Microsoft namespace, 234
- Microsoft WindowsClient.NET site, 13
- minus sign (-) to collapse area of code, 243–244
- Misc category (Properties window), 250
- Modify Selection area (customize dialog box), 34–37
- Modify Selection option (toolbars), 29
- monitors, multiple documents on, 155–156
- mouse wheel, zooming in/out of text with, 209–210
- moving
 - between Common tab and All tab in statement completion, 226
 - and selecting between matching braces, 224–225
- multicolumn sorting, 305
- multiple monitors, documents on, 155–156

- multiple Startup Projects, 48–50
- multiple versions of VS, running side-by-side, 3
- multiple views of same document, 163–164
- multi-targeting, using older frameworks with, 46
- My Solution option (Look In dialog), 198

N

- namespaces, System, 234
- naming
 - custom toolbars, 29
 - tabs (Toolbox), 95
- navigating
 - backwards/forwards with Go Back markers, 277–278
 - errors in Errors List, 303–304
 - Find Results list, 190
 - HTML Editor tag navigation, 267–268
 - IDE navigator, 160–162
 - Navigate To dialog, 195–196
 - Navigation Bar, using, 266–267
 - navigation stack, 275–276
 - Object Browser navigation, 362–364
 - open document windows, 157–158
 - property tabs in project properties, 55
 - reference highlighting and, 223
 - among tabs in Toolbox, 89
 - .NET Framework, older versions of, 46
 - “A New Standard For Packaging Your Data”, 388
 - New Item dialog box, 74–75, 80–82
 - New Project location, changing default of, 50–51
 - New Projects dialog, 43–45, 56, 72, 80–82
 - n minutes/n days options (AutoRecover), 10
- O**
- Object Browser
 - Browse Definition command, 202–203
 - navigation and references, 362–364
 - “The Object Browser: Browsing Scope”, 198
 - overview, 356–357
 - setting browsing scope, 359–362

- using for web
 - development, 96–98
- Object images (VS Image Library), 151
- /on /off argument (logging), 133
- Online Help, table of contents in (Visual Studio 2010), 4–6
- Open Data Protocol Visualizer extension (VS), 405–406
- opening
 - file location from file tab, 158
 - file menu drop-down list from keyboard, 159
 - Open File Using Directory Of Currently Active Document option, 166
- Open Packaging Convention, 388
- “Optimizing Visual Studio 2010 and WPF Applications for Remote Desktop”, 13
- ordering, column (tool windows), 304–305
- Organize Usings menu (editor), 232
- outlining
 - Collapse to Definitions with, 246–247
 - collapsing code with, 242–243
 - cut/copy/paste collapsed code with, 247
- output files (Visual Studio Image Library), 147
- Output Location (Export Template Wizard), 61, 66
- Output window option (Initial Directory), 140
- OverviewMargin extension (VS), 411–413
- /overwrite argument (logging), 133

P

- page-level tracing, 336–339
- Papadimoulis, Alex, 402
- parameter information, 227–228
- Parsons, Jared, 406–407
- Pascal Case, 196, 216
- pasting
 - content between box selections, 238–239
 - cut/copy/paste collapsed code with outlining, 247
 - formatting on HTML paste, 268
 - single selection into box selection, 239–240
- performance, improving by changing visual experience (VS 2010), 12–13

- pinning
 - DataTips to source code, 305–306
 - projects to Recent Projects list, 56 tabs, 417
- posters of keyboard shortcuts, 425
- Power Commands extension (VS), 398–400
- PowerConsole extension (VS), 404–405
- PowerShell (Windows), 421–423
- Prefix option (Find Symbol), 199
- Presentation Zoom extension (VS), 211, 408–409
- Preview Image field (Export Template Wizard), 61, 66
- Process combo box (Locals window), 377
- Productivity Power Tools (VS), 416–418
- projects
 - creating temporary, 56–57
 - default templates,
 - reorganizing, 77–80
 - multiple startup, 48–50
 - navigating property tabs in project properties, 55
 - New Project location, changing default of, 50–51
 - pinning projects to Recent Projects list, 56
 - removing from Recent Projects list, 9
- project templates
 - creating with Export Template Wizard, 64–67
 - custom, organizing, 71–74
 - in New Project dialog,
 - changing, 80–82
 - Recent Project templates in New Project dialog, 44
 - searching for, 43–45
- properties
 - project, navigating property tabs in, 55
 - Properties window keyboard shortcuts, 249–250
- Pugh, David, 409, 411

Q

- question mark (?) for wildcard searches, 105
- Quick Find
 - buttons, 176
 - dialog box, 117–118
 - Find Next button, 176

- Find Options area, 174–176
- Find What field, 173
- Look In area, 173–174
- overview, 172–173
- Quick Info option, 228–229
- Quick Replace, 176–178, 179–180
- QuickWatch
 - fundamentals of, 343–345
 - setting watch expressions with, 341
 - window, 348
- quotes, inserting when typing attribute values, 264

R

- read-only files, editing, 282–283
- rearranging menus/toolbars, 32
- recent files, 154–155
- Recent Projects list
 - pinning projects to, 56
 - removing projects from, 9
- Recent Project templates in New Project dialog, 44
- recovered files. *See* AutoRecover feature
- Redo/Undo global actions, 222
- Redo/Undo stack, 221–222
- reference highlighting, 223–225
- references, Object
 - Browser, 362–364
- refreshing data (Watch window), 346–347
- Regex Editor extension (VS), 415–416
- registry, editing, 207, 329, 395, 410
- Regular Expressions
 - creating within code, 415–416
 - Find What combo box and, 188–189
 - Quick Find and, 175
- Remote Desktop, improving performance over, 13
- renaming code snippets, 230
- repeating last search in VS, 171–172
- Replace All buttons (Quick Replace), 178
- Replace buttons (Replace in Files), 193–194
- Replace in Files operation
 - basic options, 192–194
 - tagged expressions, 203–206
 - undoing, 180–182
- Replace With area (Quick Replace), 177

Replace With field (Replace in Files), 192–193

resetting

- color schemes, 24
- development settings, 25–28
- all keyboard shortcuts, 131–133

Reset All option (Modify Selection), 34

Reset button (tools), 130

ResetSettings switch (Visual Studio), 39–41

Result options

- Find in Files, 190–191
- Replace in Files, 193

results shortcuts (Find Symbol), 200–203

Richards, Noah, 414

Run command (Find/Command box), 182

running commands, 121–123

Run To Cursor, 330

S

safe mode (Visual Studio), 38

saving

- changes before building, 302–303
- color schemes, 23
- current settings, 25
- images in .resx files, 393
- Save AutoRecover Information Every check box, 10

Script Editor, 168

ScriptFree view (VS 2010 Online Help), 5

Scrollable Tab Wells, 417

searching. *See also* finding

- breakpoints, 312–313
- in Class View, 353–355
- currently selected string without Find window, 185–186
- Find Combo box keyboard shortcuts, 182–184
- Find in Files, 186–190, 190–191
- Find Symbol, 196–200
- Find Symbol results shortcuts, 200–203
- for project templates in New Project dialog, 43–45
- Go To Definition for CSS, 194
- incremental search, 184–185
- Navigate To dialog, 195–196
- Quick Find. *See* Quick Find
- Quick Replace, 176–178
- repeating last search, 171–172
- Replace in Files, 180–182, 192–194, 203–206

- search results, customizing (Find In Files), 206–208
- Search Results (Find Symbol), 199–200
- Search Up option (Quick Find), 174
- simple searches for files, 105–106
- in Toolbox, 88
- Undo Quick Replace and Replace in Files, 179–182
- wildcard searches, 105–106

selecting

- from current cursor to last Go Back marker, 278–279
- formatting for HTML selections, 265–266
- Hide Selection, 244–246
- and moving between matching braces, 224–225
- Selection Comment/Selection Uncomment, 35

Server projects, launching, 50

settings

- default development, 26
- development, resetting, 25–28
- environment, exporting, 6–9
- environment, importing/ changing, 14–16

Set a Breakpoint (Find/Command box), 183

“Setting a Breakpoint in the Call Stack Window”, 369

“Setting a Tracepoint in Source Code”, 369

Settings To Export dialog box, 7

sharing tokens, 299–300

Shifflett, Karl, 395

shortcuts

- creating new, 127–131
- keyboard. *See* keyboard shortcuts
- for running commands, 121–122

Show Commands Containing area, 111

Show Previous Versions option (documents), 166–167

“Side Effects and Expressions”, 347

Sitnikov, Dmitry, 413

Skip File button (Replace in Files), 193

smart tags, using from keyboard, 231–232

snippets, code

- inserting, 253–256
- Snippet Designer extension (VS), 418–419
- surrounding existing code with, 256–257
- using, 258–259

Solution Explorer

- Properties button, 48
- Solution Explorer Tools extension (VS), 396–397
- tracking active items in, 52–53
- type-ahead selection support in, 52–53

Solution Folders, 54–55

sorting

- multicolumn sorting, 305
- using statements, 234–235

Source and Design views in web projects, 235–236

source files (Visual Studio Image Library), 147

Spell Checker extension (VS), 407–408

Split view, 235

stack frame (Locals window), 377

Startup Projects, multiple, 48–50

statement completion

- invoking, 225–226
- moving between Common tab/All tab in, 226

statements, using, 232–235

Status Bar, 89

StructureAdornment extension (VS), 409–411

StudioShell extension (VS), 421–423

styles, embedded (HTML), 284–285

style sheets, dedicated (CSS), 283

Substring option (Find Symbol), 199

switches and arguments (commands), 115–119

symbols

- Find Symbols results shortcuts, 200–203
- searching for, 196–200

synchronizing Class View, 355

syntax for logging, 132

System menu, 102

System namespaces, 234

T

table of contents in Visual Studio 2010 Online Help, 4–6

tabs

- command prompt tab completion, 105–107
- inserting documents to right of existing, 153–154
- moving between Common tab and All tab in statement completion, 226
- navigating in Toolbox, 89
- opening file location from file tab, 158

- organizing in Toolbox, 231
 - pinning, 417
 - in Toolbox, 93–95
 - tagged expressions (Replace in Files), 203–206
 - tags
 - HTML Editor tag
 - navigation, 267–268
 - Tag Specific options. *See* Tag Specific Options dialog box
 - using smart tags from keyboards, 231–232
 - Tag Specific Options dialog box
 - ASP.NET Controls, 286
 - Bold setting, 289
 - Client HTML Tags, 286–287
 - closing tag option, 287–288
 - Default Settings, 286
 - Delete option, 286
 - Enable Outlining for Tag, 288
 - Indent Contents, 287
 - Line Breaks, 287
 - Minimum lines setting, 288
 - New Folder option, 286
 - New Tag option, 286
 - Outlining in Code Editor, 288
 - overview, 285–286
 - Per Tag Colorization, 289
 - Per Tag Formatting, 286
 - Preview option, 287
 - Tag Background/Foreground settings, 289
 - Tree view, 286
 - Task Lists
 - creating code shortcuts in, 300–301
 - creating custom tokens for, 297–300
 - TODO comments in, 296–297
 - templates
 - custom item, organizing, 68–71
 - default item, reorganizing, 74–78
 - default project, reorganizing, 77–80
 - item, creating, 57–64
 - in NewProject/Items dialogs, changing, 80–82
 - project, creating with Export Template Wizard, 64–67
 - project, organizing custom, 71–74
 - project, searching for, 43–45
 - Recent Project in New Project dialog, 44
 - temporary projects, creating, 56–57
 - temporing commands, 119–121
 - text
 - replacing with box selection, 237–238
 - Text variables (customizing search results), 207
 - Text Visualizers, 345
 - zoom in/out with Editor, 209–211
 - “The Daily WTF” extension (VS), 401–402
 - themes, creating with VS Color Theme Editor, 389–391
 - Thread combo (Locals window), 377
 - Threads window, 325
 - Title Bar, 85
 - TODO comments in Task List, 296–297
 - tooggling Design view, 236
 - tokens, creating custom for Task Lists, 297–300
 - toolbars, customizing
 - Commands tab, 30–37
 - Toolbars tab, 27–29
 - Toolbox
 - context menu, 136
 - drag and drop code into, 229–230
 - expanding/collapsing all in, 87
 - navigating among tabs in, 89
 - searching in, 88
 - stopping from auto-populating, 136
 - tabs in, 93–95
 - tools
 - exporting external tools list, 143–145
 - running external, 136–140
 - tools window
 - animations, changing, 14–15
 - auto-hide all, 99–100
 - closing, 87
 - column ordering in, 304
 - docking floating to previous location, 85
 - docking/undocking, 107–110
 - keyboard access to toolbar, 103
 - moving with keyboard, 102–103
 - open, cycling through, 86
 - showing hidden with Auto Hide Channel, 100–101
 - To, Quan, 388
 - tracepoints
 - setting in Call Stackwindow, 369–371
 - setting in source code, 325–328
 - tracing
 - application-level/page-level, 336–339
 - “Tracing and Implementing Applications”, 325
 - tracking
 - active items in Solution Explorer, 52–53
 - changes in Editor, 280–281
 - transposing lines/words/characters, 219–220
 - “Trees in WPF”, 371
 - Triple Click extension (VS), 414–415
 - Troubleshooting Tips area (Exception Assistant), 332
 - two threads icon, 347
 - type-ahead functionality, 159
 - type-ahead selection support in Solution Explorer, 52–53
- ## U
- uncommenting/commenting code in web pages, 217–218
 - “Understanding Commands: Running Commands”, 120
 - “Understanding Commands: Aliases”, 119
 - undocking single tool windows, 107–108
 - Undo Quick Replace/Replace in Files, 179–182
 - Undo/Redo global actions, 222
 - Undo/Redo stack, 221–222
 - unhandled exceptions, unwinding call stacks on, 334
 - Unicode, 140
 - universal zoom, 211
 - UnresolvedMergeConflict token, 298
 - updating JScript Intellisense, 269
 - Use Defaults option (color schemes), 24
 - Use Hardware Graphics Acceleration If Available option, 13
 - Use option
 - Find What combo box, 187
 - Quick Find, 174–175
 - “Using External Tools”, 141
 - using statements (C#), 232–235
- ## V
- variables
 - for customizing search results, 207–208
 - Variable windows, adding watches from, 348–350
 - Vertical Tab Well, 417
 - views
 - Classic (Visual Studio 2010 Online Help), 5–6
 - multiple views of same document, 163–164
 - “View Code Blocks Using Vertical Lines”, 411

- views (*continued*)
 - View.ClassViewSearch
 - command, 354–355
 - viewing assigned aliases (commands), 114
 - viewing white space, 241–242
 - window layout, 90
- Virtual Directories, creating in IIS, 46–47
- Visual Basic
 - Autos window and, 381
 - setting breakpoints in, 292
 - TODO comments in, 296
 - users in VS 2010, 163
- Visual Basic 6 keyboard shortcuts
 - commands, running, 121
 - current line, cutting/deleting, 213
 - Design view, toggling, 236
 - file location, opening from file tab, 158
 - Find Combo box keyboard shortcuts, 182
 - Find In Files (Result options), 190
 - Find Symbol results shortcuts, 200
 - Go Back markers, navigating with, 277
 - Go To Definition for CSS, 194
 - IDE navigator, 160
 - incremental search, 184
 - macros, creating/using, 144–146
 - multi-targeting, using older frameworks with, 46
 - navigating errors in Errors List, 303
 - navigation stack, 275
 - Object Browser navigation/ references, 362
 - Object Browser overview, 356
 - Object Browser scope settings, 359
 - parameter information, 227
 - Quick Info option, 228
 - Recent Project templates in New Project dialog, 44
 - Run To Cursor, 330
 - searching for project templates, 43–45
 - transposing lines/words/ characters, 219–220
 - type-ahead selection support in Solution Explorer, 52–53
 - Undo/Redo stack, 221
 - white space, viewing, 241
 - word wrap, 248–249
- Visual C++ 2 keyboard shortcuts
 - breakpoints, setting in Call Stack window, 367
 - breakpoints, setting on functions, 316
 - breakpoints window,
 - opening, 293
 - Call Hierarchy window, 310
 - Class View, searching in, 353
 - code snippets, inserting, 253
 - Code Snippets Manager, 261
 - code snippets, surrounding existing code with, 256
 - code snippets, using, 258
 - Collapse To Definitions with outlining, 246
 - commands, running, 121
 - commenting/uncommenting code in web pages, 217–218
 - current line, cutting/deleting, 213
 - cycling through open tool windows, 86
 - Design view, toggling, 236
 - docking floating tool windows to previous location, 85
 - docking/undocking tool windows, 107
 - Document Outline (web projects), 251
 - Find Combo box keyboard shortcuts, 182
 - Find In Files (Result options), 190
 - Find Symbol, 196–200
 - Find Symbol results shortcuts, 200
 - Go To Definition for CSS, 194
 - Hide Selection, 244
 - HTML code snippets, 259
 - HTML, formatting documents/ selections for, 265
 - IDE navigator, 160
 - Intellisense keywords, 215–216
 - invoke statement completion, 225
 - JavaScript code snippets, 260
 - Locals window, changing context in, 376
 - matching braces, selecting/moving between, 224
 - navigating errors in Errors List, 303
 - navigating open document windows, 157
 - navigation stack, 275
 - Object Browser navigation/ references, 362
 - Object Browser overview, 356
 - Object Browser scope settings, 359
 - outlining, collapsing code with, 242
 - Pascal case, 216
 - Properties window, 249
 - Quick Find, 172
 - Quick Info option, 228
 - rearranging windows in Visual Studio 2010 with Guide Diamond, 83
 - Run To Cursor, 330
 - Task Lists, creating code shortcuts in, 300
 - type-ahead selection support in Solution Explorer, 52–53
 - Undo/Redo stack, 221
 - visualizers (Watch windows), 345
 - white space, viewing, 241
- Visual C++ 6 keyboard shortcuts
 - breakpoints, adding labels to, 293–294
 - breakpoints window,
 - opening, 293
 - Clipboard Ring, cycling through, 220–221
 - Code Definition window, 301
 - commands, running, 121
 - current line, cutting/deleting, 213
 - Design view, toggling, 236
 - Document Outline (web projects), 251
 - Esc for returning to Editor return to, 237
 - Find Combo box keyboard shortcuts, 182
 - Find Symbol, 196–200
 - Find Symbol results shortcuts, 200
 - HTML, formatting documents/ selections for, 265
 - Intellisense keywords, 215–216
 - invoke statement completion, 225
 - Locals window, changing context in, 376
 - multiple start-up projects, 48–50
 - navigating errors in Errors List, 303
 - Navigation Bar, using, 266
 - navigation stack, 275
 - Pascal case, 216
 - Properties window, 249
 - text, replacing with box selection, 237
 - white space, viewing, 241
- Visual C# 2005 keyboard shortcuts
 - arguments and switches (commands), 115
 - Autos window, 379
 - breakpoints, setting on functions, 316
 - breakpoints window,
 - opening, 293
 - Code Definition window, 301
 - code snippets, inserting, 253
 - code snippets, surrounding existing code with, 256

- code snippets, using, 258
- Collapse To Definitions with outlining, 246
- command aliases, 113
- commands, running, 121
- commenting/uncommenting code in web pages, 217–218
- current line, cutting/deleting, 213
- Design view, toggling, 236
- Document Outline (web projects), 251
- Exceptions dialog box, 365
- Find Combo box keyboard shortcuts, 182
- Find In Files (Result options), 190
- Find Symbol results shortcuts, 200
- HTML code snippets, 259
- HTML, formatting documents/selections for, 265
- Intellisense keywords, 215–216
- invoke statement completion, 225
- JavaScript code snippets, 260
- Locals window, changing context in, 376
- Object Browser navigation/ references, 362
- Object Browser overview, 356
- Object Browser scope settings, 359
- outlining, collapsing code with, 242
- parameter information, 227
- Pascal case, 216
- Properties window, 249
- Quick Info option, 228
- QuickWatch, 343–345
- searching in Toolbox, 88
- Solution Explorer, type-ahead selection support in, 52–53
- Task Lists, creating code shortcuts in, 300
- testing commands, 119
- visualizers (Watch windows), 345
- Visual Studio 2010 Online Help, table of contents in, 4
- Watch window, 340
- white space, viewing, 241
- word completion, 229
- word wrap, 248–249
- visual experience, changing in VS 2010, 12–13
- visualizers, 345–346
- Visual Studio
 - changing color schemes in. *See* color schemes, changing
 - changing visual experience in, 12–13
 - customizing with Windows PowerShell, 421–423
 - Guide Diamond, rearranging windows with, 83–84
 - Image Library, 147–152
 - logging, 37–38
 - Online Help, table of contents in, 4–6
 - project templates stored in, 77
 - ResetSettings switch, 39–41
 - running multiple versions side-by-side, 3
 - in safe mode, 38
 - storage of item templates, 75
 - “Visual Studio Commands with Arguments”, 115
 - Visual Studio Gallery, 386–387
- Visual Studio 6 keyboard shortcuts
 - breakpoints, setting on functions, 316
 - breakpoints window, opening, 293
 - Collapse To Definitions with outlining, 246
 - commands, running, 121
 - current line, cutting/deleting, 213
 - Design view, toggling, 236
 - Find Combo box keyboard shortcuts, 182
 - Find In Files, 186
 - Find In Files (Result options), 190
 - Find In Files search results, customizing, 206
 - Find Symbol results shortcuts, 200
 - Go To Definition for CSS, 194
 - Hide Selection, 244
 - HTML, formatting documents/selections for, 265
 - incremental search, 184
 - Intellisense keywords, 215–216
 - Locals window, changing context in, 376
 - macros, creating/using, 144–146
 - multi-targeting, using older frameworks with, 46
 - navigating errors in Errors List, 303
 - navigating open document windows, 157
 - Object Browser navigation/ references, 362
 - Object Browser overview, 356
 - Object Browser scope settings, 359
 - outlining, collapsing code with, 242
 - parameter information, 227
 - Pascal case, 216
- Properties window, 249
- Quick Info option, 228
- Recent Project templates in New Project dialog, 44
- searching for project templates, 43–45
- Solution Explorer, type-ahead selection support in, 52–53
- Visual Studio extensions
 - AllMargins extension, 411–413
 - CodeCompare extension, 402–404
 - Color Theme Editor extension, 389–391
 - disabling, 388
 - Editor Guideline extension, 394–396
 - Emacs Commands extension, 400–401
 - GhostDoc tool, 419–421
 - Image Insertion tool, 391–393
 - installing from Extension Manager, 386
 - installing from Visual Studio Gallery, 386–387
 - installing through Xcopy, 387
 - Open Data Protocol Visualizer extension, 405–406
 - overview, 385
 - OverviewMargin extension, 411–413
 - Power Commands extension, 398–400
 - PowerConsole extension, 404–405
 - Presentation Zoom extension, 408–409
 - Productivity Power Tools, 416–418
 - Regex Editor extension, 415–416
 - resources for developing, 389
 - Snippet Designer extension, 418–419
 - Solution Explorer Tools extension, 396–397
 - Spell Checker extension, 407–408
 - StructureAdornment extension, 409–411
 - StudioShell extension, 421–423
 - “The Daily WTF” extension, 401–402
 - Triple Click extension, 414–415
 - uninstalling, 389–390
 - VsVim extension, 406–407
 - Win7 Taskbar Extension, 413–414
 - XAML Intellisense Presenter extension, 395–396
 - ZoomEditorMargin extension, 408–409

.vsix files, 388
 .vssettings files, 21, 135
 VsVim extension (VS), 406–407

W

Watch windows

column ordering and, 304
 visualizers and, 345–346
 watches, adding from variable windows, 348–350
 watching and changing values, 340–343

Web Applications, creating in IIS, 46–47

web development, different browsers for, 96–98

web pages, commenting/uncommenting code in, 217–218

web projects, switching between Design and Source views in, 235–236

websites, for downloading

CodeCompare extension (VS), 404

“Disable Mouse Wheel Zoom” extension, 210

posters of keyboard

shortcuts, 425

“Presentation Zoom” extension, 211

“The Daily WTF”, 402

websites, for further information

Autos window, 379

Emacs keyboard shortcuts, 401

“Export Template Wizard” documentation, 68

“Find Command”

documentation, 116

GhostDoc extension (VS), 421

“HTML Editor Tag Navigation in Visual Web Developer”, 267

Image Insertion tool,

customizing, 393

“Installing Visual Studio Versions Side-by-Side”, 3

MSDN documentation, 110

Open Data Protocol

Visualizer, 406

Open Packaging Convention, 388

“Optimizing Visual Studio 2010 and WPF Applications for Remote Desktop”, 13

Power Commands extension

(VS), 400

Power Console extension

(VS), 405

Properties Window, 249

reference highlighting, 223

Regex Editor, 415

Regular Expressions, 175

“Side Effects and

Expressions”, 347

Snippet Designer extension

(VS), 419

StudioShell module, 423

“The Daily WTF” extension (VS), 402

trace element settings, 338

tracing, 339

“Tracing and Implementing Applications”, 325

“Trees in WPF”, 371

Triple Click extension (VS), 414

unwinding call stacks on unhandled exceptions, 334

VIM extension (VS), 406

“Visual Studio Commands with Arguments”, 115

Visual Studio, documentation for, 37

Visual Studio extension developer’s blog post, 391

Visual Studio extensions, development resources for, 389

Visual Studio Gallery, 386

.vsix file, 388

Wildcard searches, 175

WPFPerf tool, 13

XAML Intellisense Presenter, 396

When Breakpoint Is Hit dialog box, 327

white space

adding extra in code, 218

viewing, 241–242

Whole Word option (Find Symbol), 199

/wild or /l switch, 117

wildcard searches, 105–106, 175, 188

Win7 Taskbar Extension

(VS), 413–414

Window Layouts

Debug Mode, 92

Design Mode, 91

exporting, 39, 134–136

four modes of, 90–91

Full Screen Mode, 93

windows

auto-hide all tool

windows, 99–100

Autos, 348

Breakpoints, 317

browser window size, 98

Call Hierarchy window, 310–312

closing current document

window, 158

closing tool windows, 87

Code Definition window, 301

Command Window, 122

detachable document

windows, 155

docking/undocking tool

windows, 107–110

floating tool window, docking to previous location, 85

Immediate Window, 122

Locals, 348

moving tool windows with keyboard, 102–103

navigating open document windows, 157–158

QuickWatch window, 348

rearranging with Guide Diamond (VS 2010), 83–84

searching currently selected string without Find window, 185–186

showing hidden tool windows with Auto Hide Channel, 100–101

Threads window, 325

tool windows, cycling through open, 86

Watch. *See* Watch windows

Windows, Microsoft

Windows PowerShell, 421–423

Windows Presentation Foundation (WPF), 13

words

transposing, 219–220

word completion, 229

word wrap, 248–249

WPFPerf tool (Windows SDK), 13

WPF Tree Visualizer, 371–373

X

XAML Intellisense Presenter extension (VS), 395–396

Xcopy, installing VS extensions through, 387

XML, exporting DataTips as, 309

Xu, Jianchun, 404

Z

zero-length box selections, 240–241

ZoomEditorMargin extension (VS), 408–409

zoom in/out of text with Editor, 209–211