Microsoft®

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.*

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

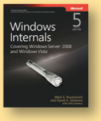
Developer Step by Step

- 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.com/mspress

ISBN: 978-0-7356-4992-7

9 780735 649927 90000

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

Programming/Microsoft Visual Studio

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

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*.

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

Table of Contents

Part II Extensions for Visual Studio

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 prefect 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

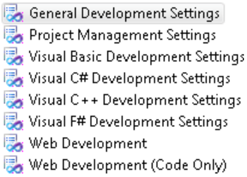
Each chapter—and in fact, almost every item—in Part I of this book stands on its own, so feel free to begin reading wherever you like. However, I suggest that you start by choosing the tasks that will have the greatest impact in your daily work. If most of your day is spent debugging, then start with 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.

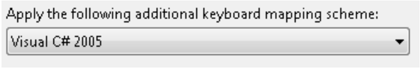
Keyboard Settings

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:



General Development Settings
Project Management Settings
Visual Basic Development Settings
Visual C# Development Settings
Visual C++ Development Settings
Visual F# Development Settings
Web Development
Web Development (Code Only)

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:



Apply the following additional keyboard mapping scheme:
Visual C# 2005

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)

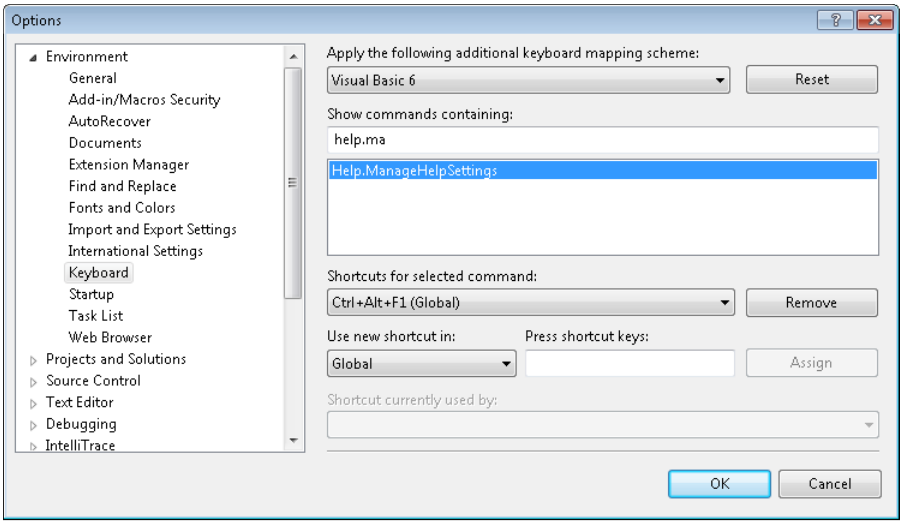
Summary Information

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.



Apply the following additional keyboard mapping scheme: Visual Basic 6

Show commands containing: help.ma

Help.ManageHelpSettings

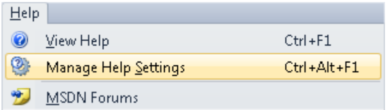
Shortcuts for selected command: Ctrl+Alt+F1 (Global)

Use new shortcut in: Global

Press shortcut keys:

Shortcut currently used by:

- **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

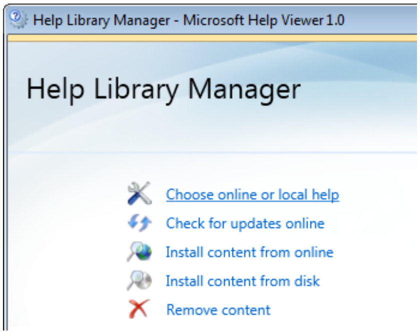
DEFAULT	Ctrl+Alt+F1 (help settings); F1 (view help)
VISUAL BASIC 6	Ctrl+Alt+F1 (help settings); F1 (view help)
VISUAL C# 2005	Ctrl+F1, M (help settings); Ctrl+F1, Ctrl+M (help settings); F1 (view help)
VISUAL C++ 2	Ctrl+Alt+F1 (help settings); F1 (view help)
VISUAL C++ 6	Ctrl+Alt+F1 (help settings); 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.

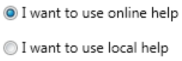
If you are like me and want to get that classic help look-and-feel back for online help, you need to do two things.

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.


Help Library Manager - Microsoft Help Viewer 1.0
Help Library Manager
Choose online or local help
Check for updates online
Install content from online
Install content from disk
Remove content

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

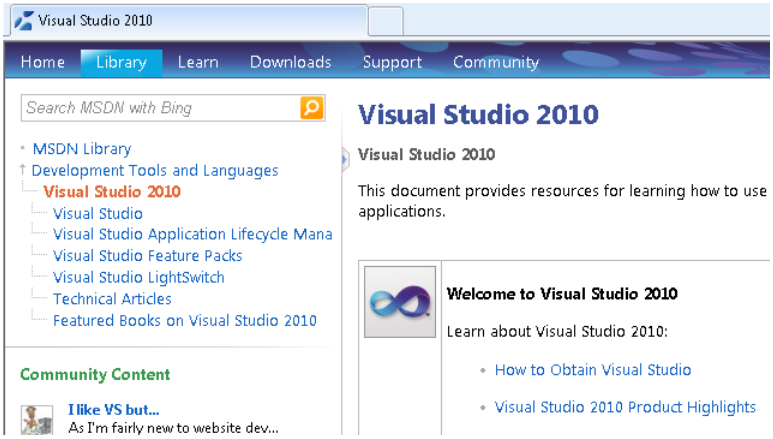


● I want to use online help

○ 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:



- MSDN Library
- Development Tools and Languages
  - Visual Studio 2010
    - Visual Studio
    - Visual Studio Application Lifecycle Mana
    - Visual Studio Feature Packs
    - Visual Studio LightSwitch
    - Technical Articles
    - Featured Books on Visual Studio 2010

Community Content

I like VS but...
As I'm fairly new to website dev...

Visual Studio 2010

Visual Studio 2010

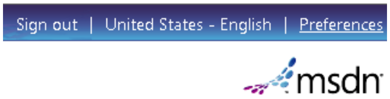
This document provides resources for learning how to use applications.

Welcome to Visual Studio 2010

Learn about Visual Studio 2010:

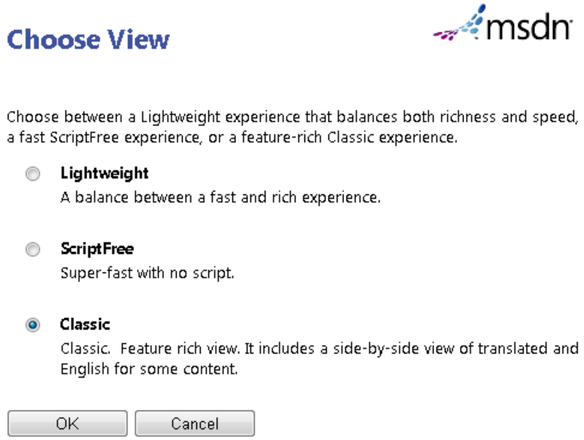
- How to Obtain Visual Studio
- Visual Studio 2010 Product Highlights

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

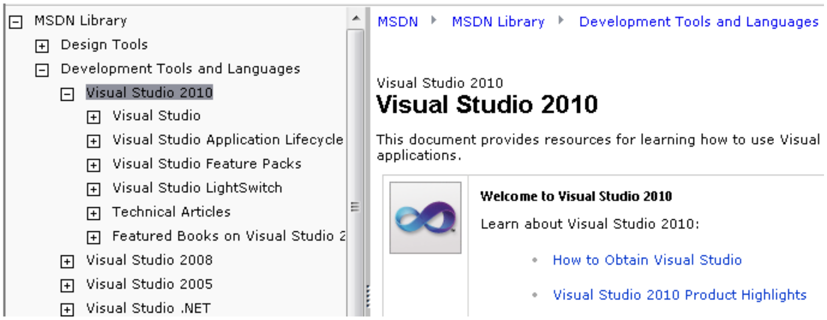
Choose between a Lightweight experience that balances both richness and speed, a fast ScriptFree experience, or a feature-rich Classic experience.

○ **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.



⊟ MSDN Library
⊞ Design Tools
⊟ Development Tools and Languages
⊟ Visual Studio 2010
⊞ Visual Studio
⊞ Visual Studio Application Lifecycle
⊞ Visual Studio Feature Packs
⊞ Visual Studio LightSwitch
⊞ Technical Articles
⊞ Featured Books on Visual Studio 2
⊞ Visual Studio 2008
⊞ Visual Studio 2005
⊞ Visual Studio .NET

MSDN ▸ MSDN Library ▸ Development Tools and Languages

Visual Studio 2010

Visual Studio 2010

This document provides resources for learning how to use Visual applications.

Welcome to Visual Studio 2010

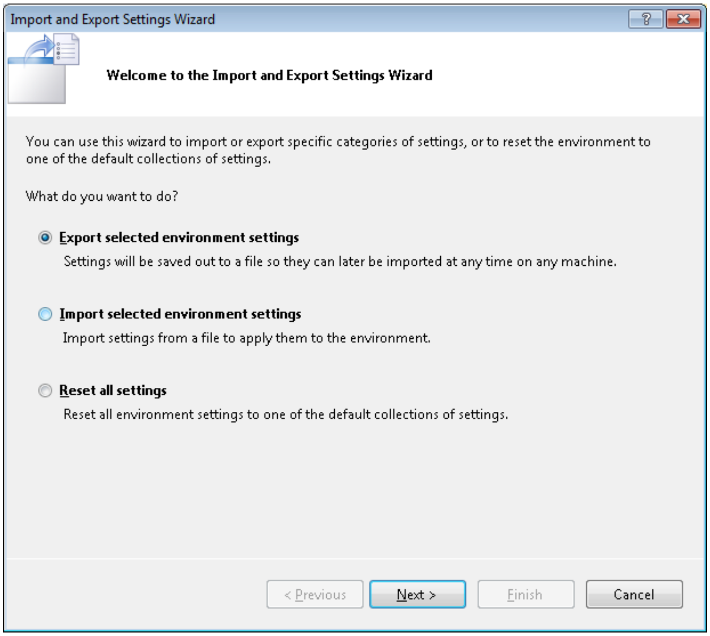
Learn about Visual Studio 2010:

- How to Obtain Visual Studio
- Visual Studio 2010 Product Highlights

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.



Import and Export Settings Wizard

Welcome to the Import and Export Settings Wizard

You can use this wizard to import or export specific categories of settings, or to reset the environment to one of the default collections of settings.

What do you want to do?

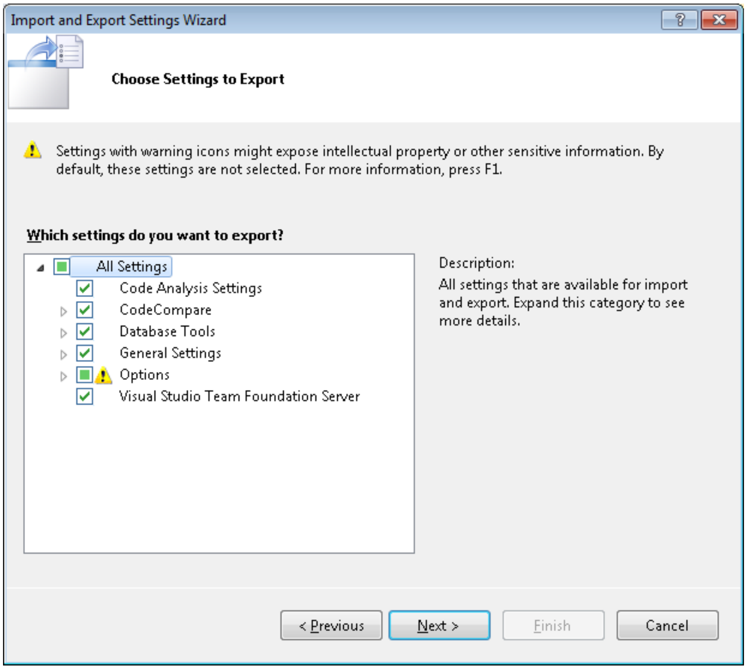
● **Export selected environment settings**
Settings will be saved out to a file so they can later be imported at any time on any machine.

○ **Import selected environment settings**
Import settings from a file to apply them to the environment.

○ **Reset all settings**
Reset all environment settings to one of the default collections of settings.

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.



Choose Settings to Export

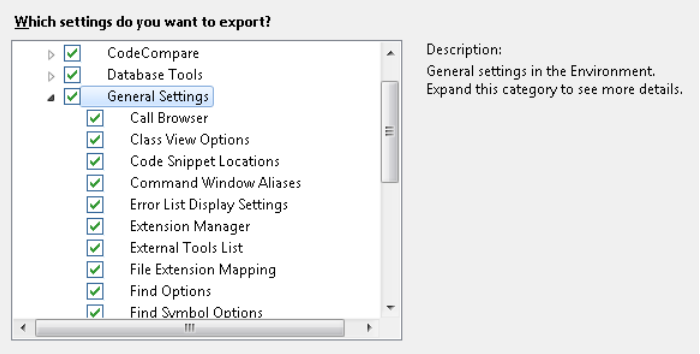
Settings with warning icons might expose intellectual property or other sensitive information. By default, these settings are not selected. For more information, press F1.

Which settings do you want to export?

- ▣ All Settings
  - ☑ Code Analysis Settings
  - ☑ CodeCompare
  - ☑ Database Tools
  - ☑ General Settings
  - ▣ ⚠ Options
  - ☑ Visual Studio Team Foundation Server

Description:
All settings that are available for import and export. Expand this category to see more details.

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

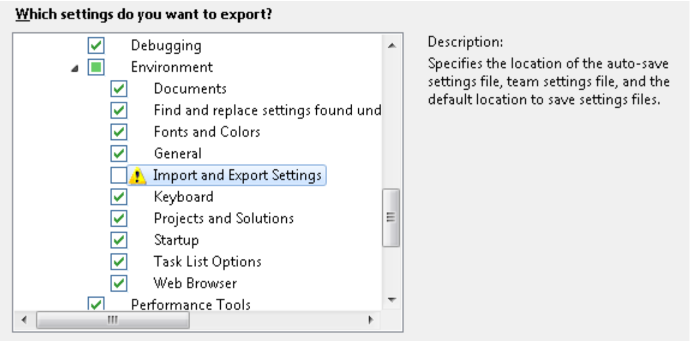


Which settings do you want to export?

- ☑ CodeCompare
- ☑ Database Tools
- ☑ General Settings
  - ☑ Call Browser
  - ☑ Class View Options
  - ☑ Code Snippet Locations
  - ☑ Command Window Aliases
  - ☑ Error List Display Settings
  - ☑ Extension Manager
  - ☑ External Tools List
  - ☑ File Extension Mapping
  - ☑ Find Options
  - ☑ Find Symbol Options

Description:
General settings in the Environment. Expand this category to see more details.

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.



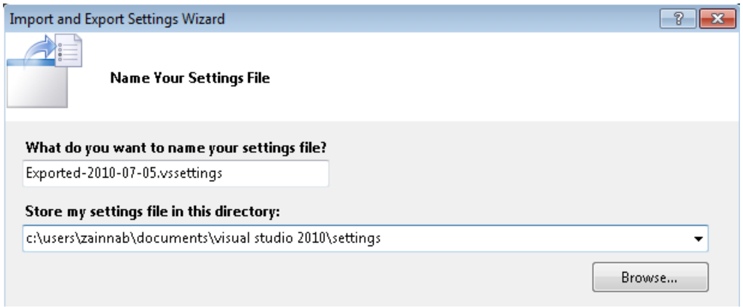
Which settings do you want to export?

- ☑ Debugging
- ■ Environment
  - ☑ Documents
  - ☑ Find and replace settings found und
  - ☑ Fonts and Colors
  - ☑ General
  - □ ⚠ Import and Export Settings
  - ☑ Keyboard
  - ☑ Projects and Solutions
  - ☑ Startup
  - ☑ Task List Options
  - ☑ Web Browser
- ☑ Performance Tools

Description:
Specifies the location of the auto-save settings file, team settings file, and the default location to save settings files.

After you make your choices and click Next, you can pick the location (C:\users\<current user>\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.



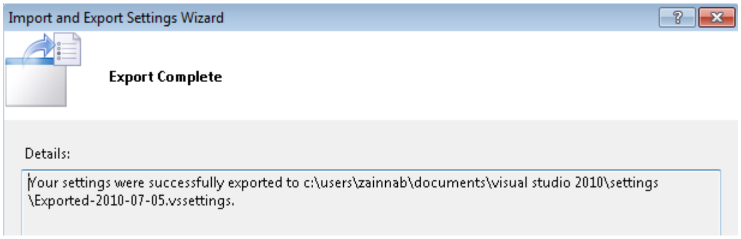
Import and Export Settings Wizard

Name Your Settings File

What do you want to name your settings file?
Exported-2010-07-05.vssettings

Store my settings file in this directory:
c:\users\zainnab\documents\visual studio 2010\settings

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

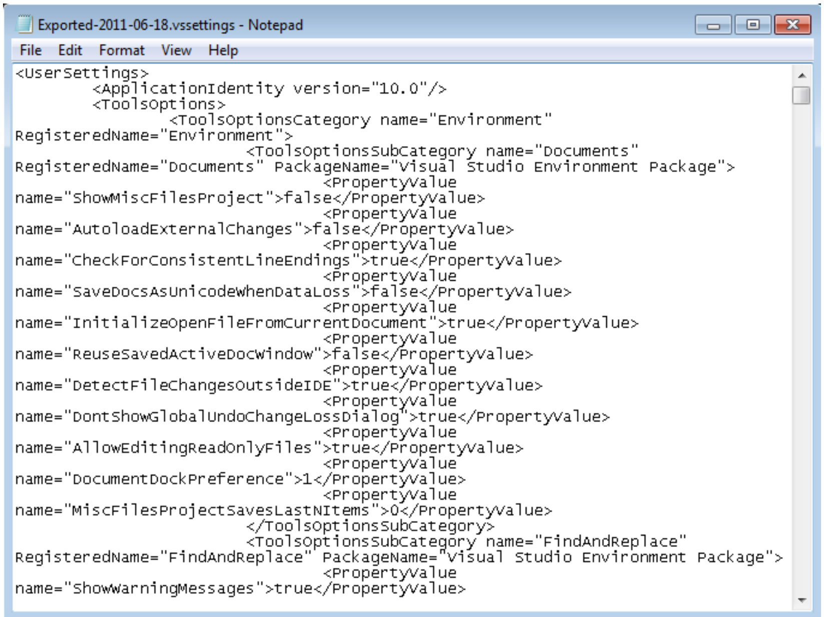


Import and Export Settings Wizard

Export Complete

Details:
Your settings were successfully exported to c:\users\zainnab\documents\visual studio 2010\settings \Exported-2010-07-05.vssettings.

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.



Exported-2011-06-18.vssettings - Notepad
File Edit Format View Help

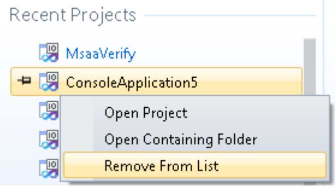
```
<UserSettings>
        <ApplicationIdentity version="10.0"/>
        <ToolsOptions>
                <ToolsOptionsCategory name="Environment"
RegisteredName="Environment">
                        <ToolsOptionsSubCategory name="Documents"
RegisteredName="Documents" PackageName="Visual Studio Environment 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="MiscFilesProjectSavesLastNItems">0</PropertyValue>
                        </ToolsOptionsSubCategory>
                        <ToolsOptionsSubCategory name="FindAndReplace"
RegisteredName="FindAndReplace" PackageName="Visual Studio Environment Package">
                                <PropertyValue
name="ShowWarningMessages">true</PropertyValue>
```

01.04 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.

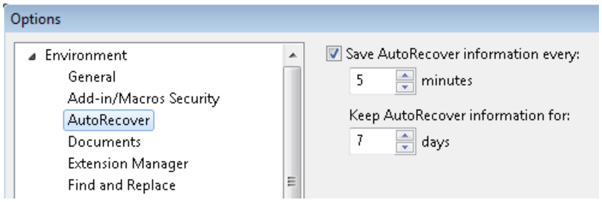
Just right-click the project, and select Remove From List, as shown in the following illustration. That's it. The project is removed from the list but not deleted. If you want to permanently delete the project, you need to do that yourself from the filesystem.



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.



Options

- Environment
  - General
  - Add-in/Macros Security
  - AutoRecover
  - Documents
  - Extension Manager
  - Find and Replace

☑ Save AutoRecover information every:
5 minutes

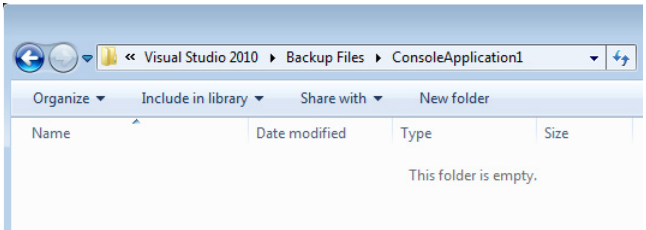
Keep AutoRecover information for:
7 days

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.

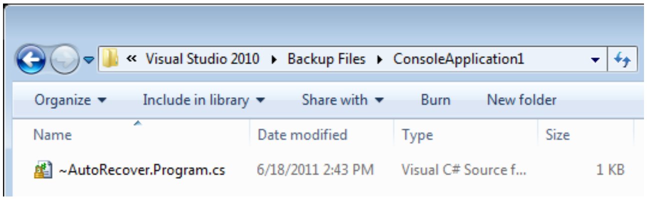
I want to be clear about what exactly gets saved and where it gets saved. First, recovered files are stored at My Documents\Visual Studio <version>\Backup Files\<projectname>. But not every file is saved here. The backup folder is empty when you first create a solution in Visual Studio.

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



This folder is empty.

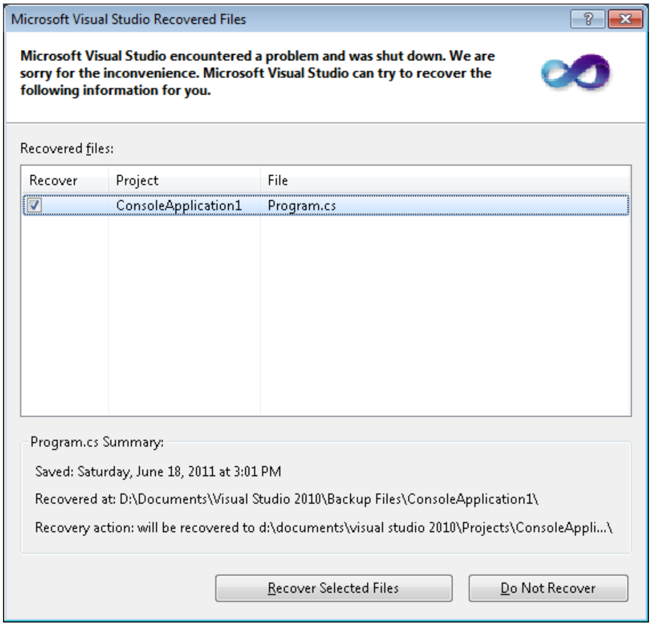
There's still nothing there, because Visual Studio knows there is no need to recover a saved file. But if I make a change to a file without saving it and wait another five minutes, here is what you see:



Name	Date modified	Type	Size
~AutoRecover.Program.cs	6/18/2011 2:43 PM	Visual C# Source f...	1 KB

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.



Microsoft Visual Studio encountered a problem and was shut down. We are sorry for the inconvenience. Microsoft Visual Studio can try to recover the following information for you.

Recovered files:

Recover	Project	File
☑	ConsoleApplication1	Program.cs

Program.cs Summary:

Saved: Saturday, June 18, 2011 at 3:01 PM

Recovered at: D:\Documents\Visual Studio 2010\Backup Files\ConsoleApplication1\

Recovery action: will be recovered to d:\documents\visual studio 2010\Projects\ConsoleAppli...\

To explain the terminology in the preceding dialog box:

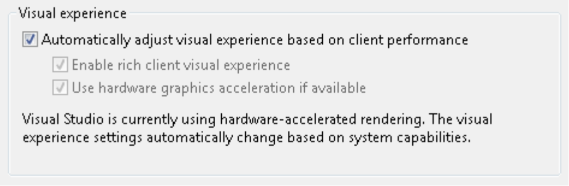
- **Recovered files**
  Lists the file(s) that can be recovered. Use a check box to select or clear the files you want to keep as well as to see some basic information.
- **<File Name> Summary**
  Shows detailed information about the currently selected file, including date/time information, location 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.



Visual experience

☑ Automatically adjust visual experience based on client performance

☑ Enable rich client visual experience

☑ Use hardware graphics acceleration if available

Visual Studio is currently using hardware-accelerated rendering. The visual experience settings automatically change based on system capabilities.

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*.

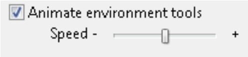
Also, when using Visual Studio 2010 over remote sessions, you should definitely read the article titled "Optimizing Visual Studio 2010 and WPF Applications for Remote Desktop," at *http://blogs.msdn.com/b/jgoldb/archive/2010/02/27/optimizing-visual-studio-2010-and-wpf-applications-for-remote-desktop.aspx*. 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.



☑ Animate environment tools

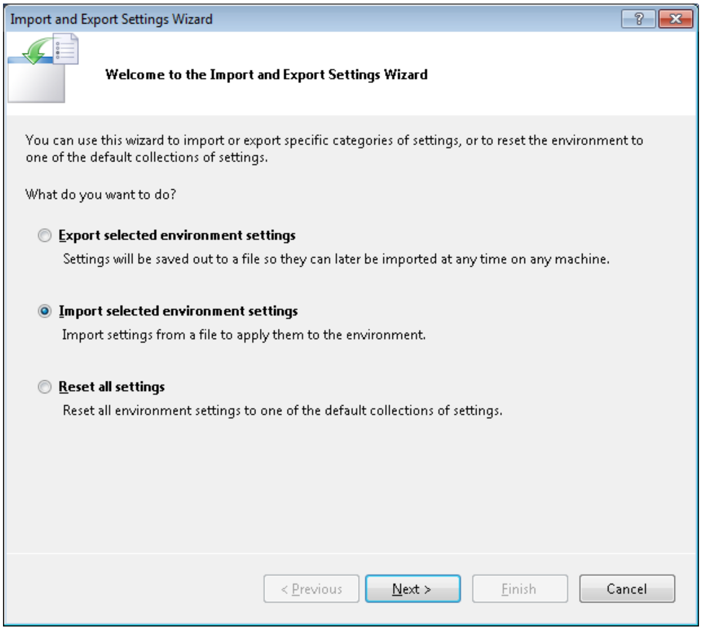
Speed - +

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:



Welcome to the Import and Export Settings Wizard

You can use this wizard to import or export specific categories of settings, or to reset the environment to one of the default collections of settings.

What do you want to do?

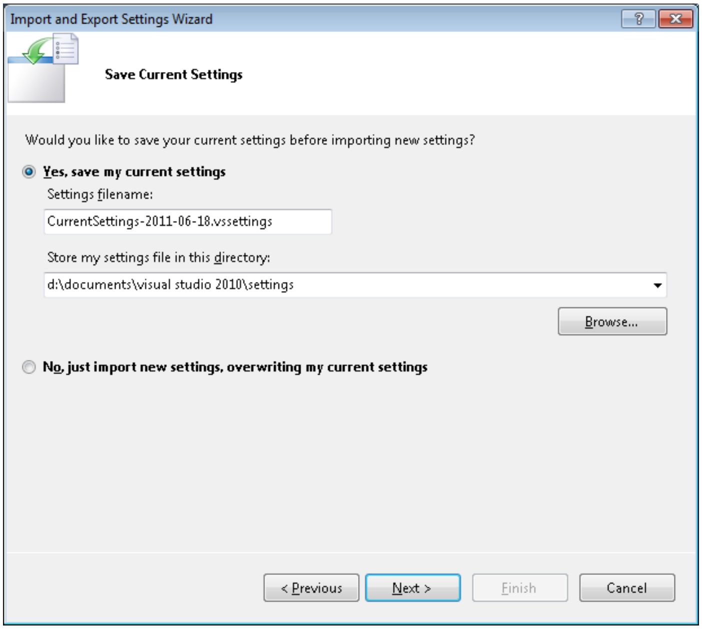
○ **Export selected environment settings**
Settings will be saved out to a file so they can later be imported at any time on any machine.

● **Import selected environment settings**
Import settings from a file to apply them to the environment.

○ **Reset all settings**
Reset all environment settings to one of the default collections of 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:



Save Current Settings

Would you like to save your current settings before importing new settings?

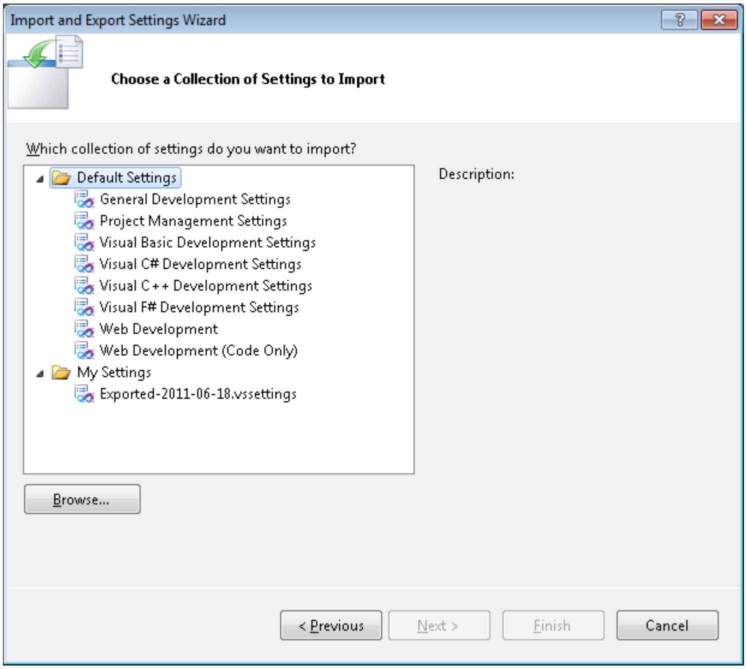
● **Yes, save my current settings**

Settings filename: CurrentSettings-2011-06-18.vssettings

Store my settings file in this directory: d:\documents\visual studio 2010\settings

○ **No, just import new settings, overwriting my current settings**

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:



Choose a Collection of Settings to Import

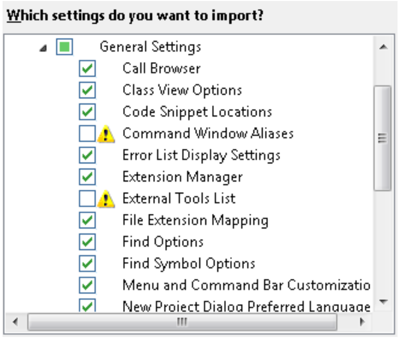
Which collection of settings do you want to import?

- Default Settings
  - General Development Settings
  - Project Management Settings
  - Visual Basic Development Settings
  - Visual C# Development Settings
  - Visual C++ Development Settings
  - Visual F# Development Settings
  - Web Development
  - Web Development (Code Only)
- My Settings
  - Exported-2011-06-18.vssettings

Description:

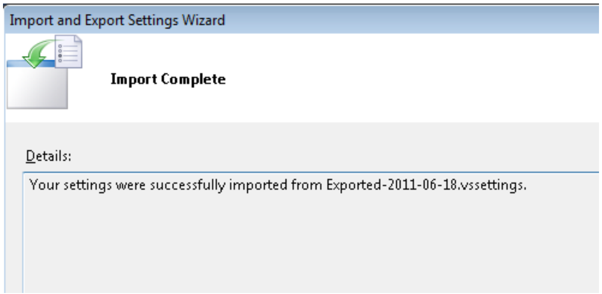
Now click Next again to choose what settings you want to import. All the previously exported settings are selected by default except for Command Window Aliases, External Tools List, and Import and Export Settings, which are considered potentially dangerous.

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.


Which settings do you want to import?

- ▪ General Settings
  - ☑ Call Browser
  - ☑ Class View Options
  - ☑ Code Snippet Locations
  - ☐ ⚠ Command Window Aliases
  - ☑ Error List Display Settings
  - ☑ Extension Manager
  - ☐ ⚠ External Tools List
  - ☑ File Extension Mapping
  - ☑ Find Options
  - ☑ Find Symbol Options
  - ☑ Menu and Command Bar Customizatio
  - ☑ New Project Dialog Preferred Language

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:



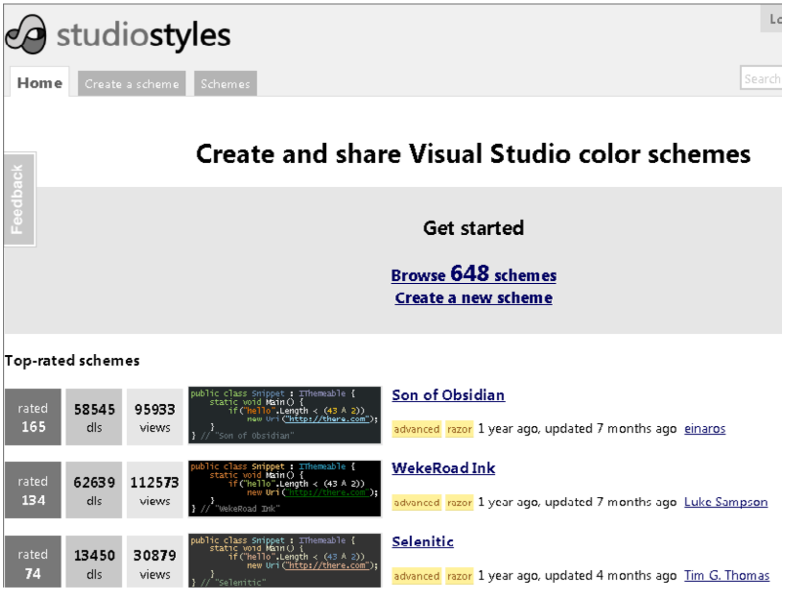
Your settings were successfully imported from Exported-2011-06-18.vssettings.

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?



Create and share Visual Studio color schemes

Get started

Browse 648 schemes
Create a new scheme

Top-rated schemes

rated 165 | 58545 dls | 95933 views

```
public class Snippet : IThemeable {
    static void Main() {
        if("hello".Length < (43 ^ 2))
            new Uri("http://there.com");
    }
} // "Son of Obsidian"
```

Son of Obsidian
advanced razor 1 year ago, updated 7 months ago einaros

rated 134 | 62639 dls | 112573 views

```
public class Snippet : IThemeable {
    static void Main() {
        if("hello".Length < (43 ^ 2))
            new Uri("http://there.com");
    }
} // "WekeRoad Ink"
```

WekeRoad Ink
advanced razor 1 year ago, updated 7 months ago Luke Sampson

rated 74 | 13450 dls | 30879 views

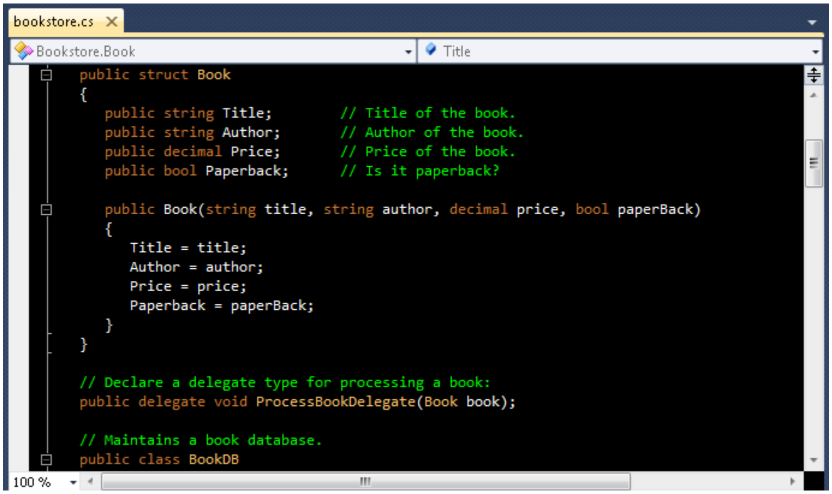
```
public class Snippet : IThemeable {
    static void Main() {
        if("hello".Length < (43 ^ 2))
            new Uri("http://there.com");
    }
} // "Selenitic"
```

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:

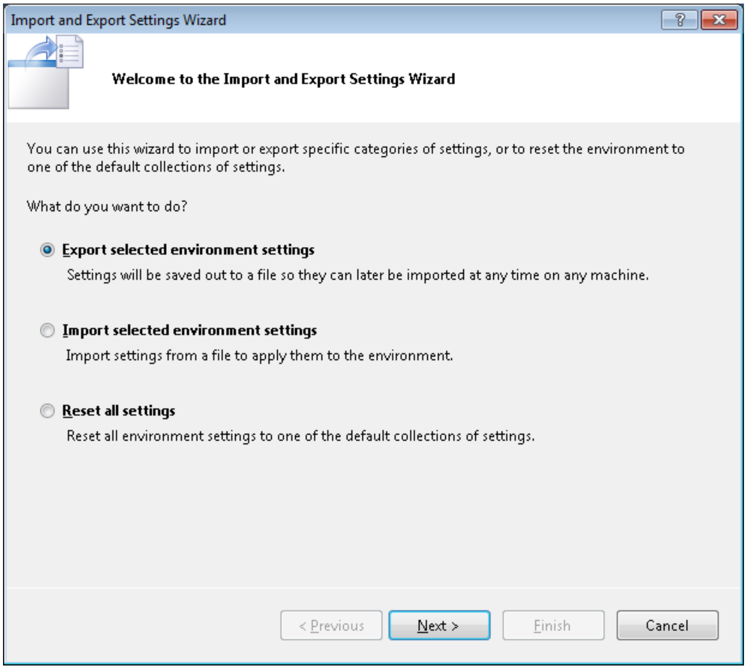


```
public struct Book
{
   public string Title;        // Title of the book.
   public string Author;       // Author of the book.
   public decimal Price;       // Price of the book.
   public bool Paperback;      // Is it paperback?

   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:



Import and Export Settings Wizard

Welcome to the Import and Export Settings Wizard

You can use this wizard to import or export specific categories of settings, or to reset the environment to one of the default collections of settings.

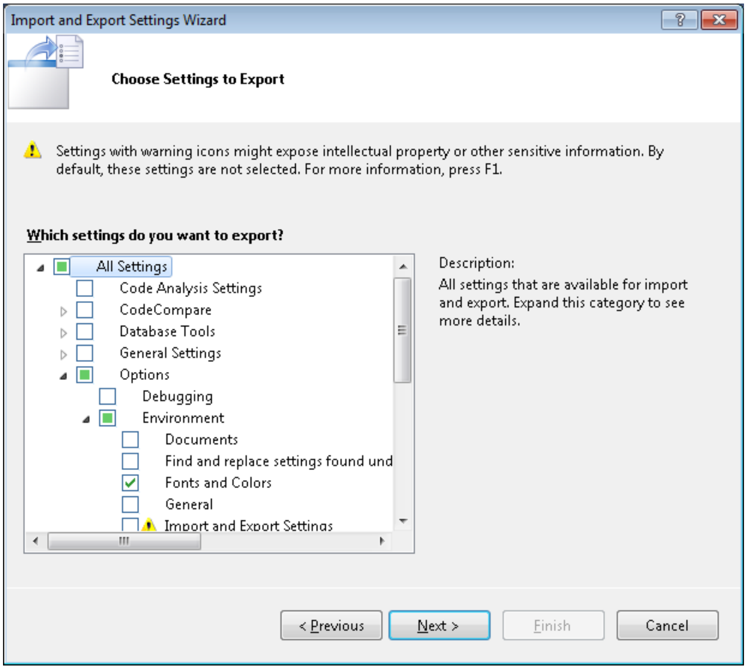
What do you want to do?

● **Export selected environment settings**
Settings will be saved out to a file so they can later be imported at any time on any machine.

○ **Import selected environment settings**
Import settings from a file to apply them to the environment.

○ **Reset all settings**
Reset all environment settings to one of the default collections of settings.

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



Import and Export Settings Wizard

Choose Settings to Export

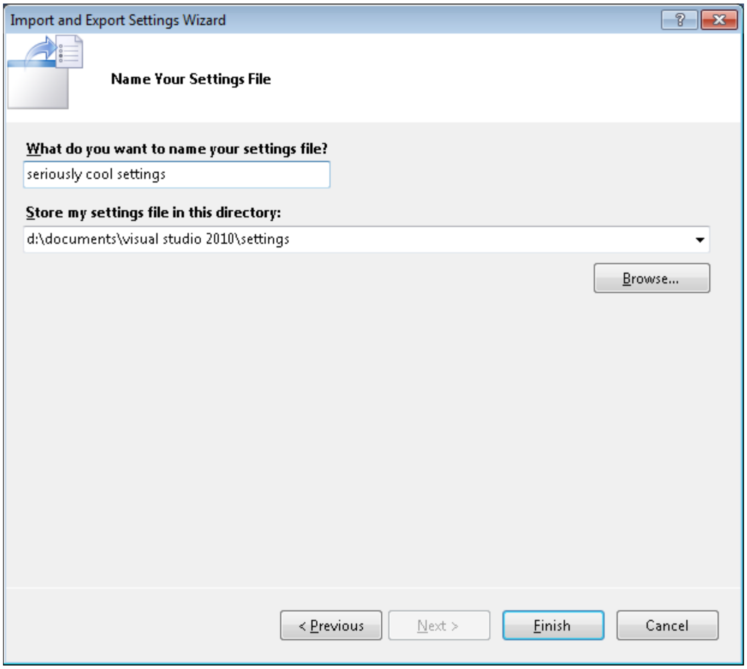
Settings with warning icons might expose intellectual property or other sensitive information. By default, these settings are not selected. For more information, press F1.

Which settings do you want to export?

- ■ All Settings
  - □ Code Analysis Settings
  - □ CodeCompare
  - □ Database Tools
  - □ General Settings
  - ■ Options
    - □ Debugging
    - ■ Environment
      - □ Documents
      - □ Find and replace settings found und
      - ☑ Fonts and Colors
      - □ General
      - □ ⚠ Import and Export Settings

Description:
All settings that are available for import and export. Expand this category to see more details.

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



Import and Export Settings Wizard

Name Your Settings File

What do you want to name your settings file?

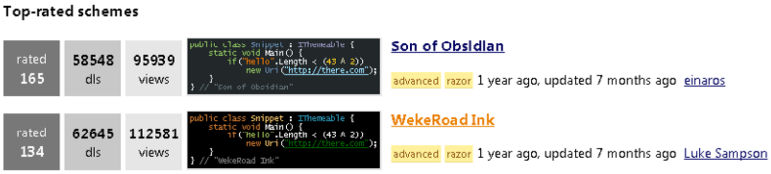
seriously cool settings

Store my settings file in this directory:

d:\documents\visual studio 2010\settings

On the Studio Styles site

Click the style you want:

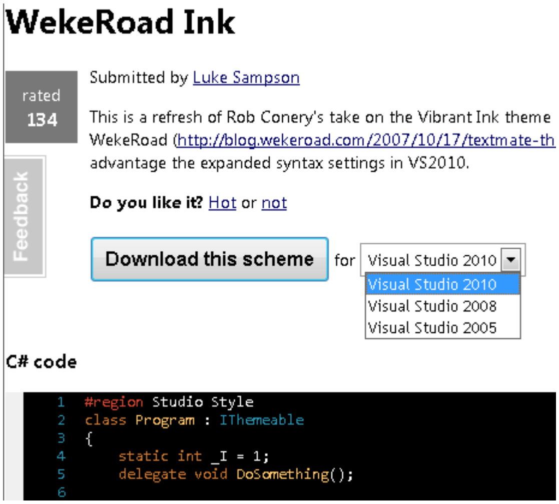


Top-rated schemes

rated
165
	58548
	dls
		95939
		views
			`public class Snippet : IThemeable {`
			`static void Main() {`
			`if("hello".Length < (43 ^ 2))`
			`new Uri("http://there.com");`
			`}`
			`} // "Son of Obsidian"`
				Son of Obsidian
				advanced razor 1 year ago, updated 7 months ago einaros

rated
134
	62645
	dls
		112581
		views
			`public class Snippet : IThemeable {`
			`static void Main() {`
			`if("hello".Length < (43 ^ 2))`
			`new Uri("http://there.com");`
			`}`
			`} // "WekeRoad Ink"`
				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

rated 134

Submitted by Luke Sampson

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.

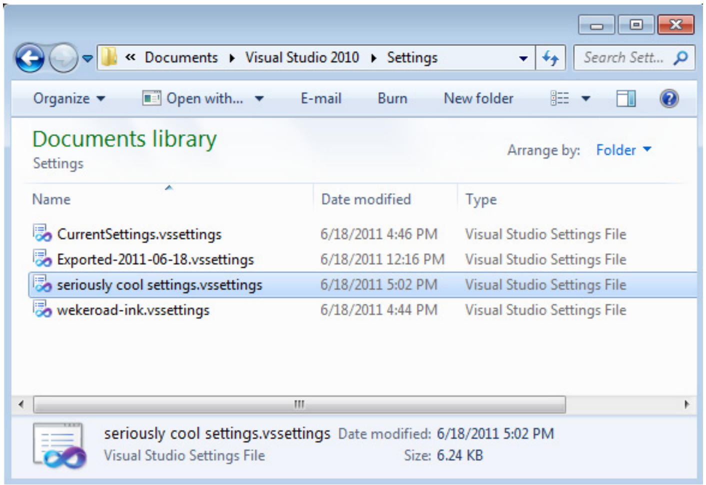
Do you like it? Hot or not

C# code

```
#region Studio Style
class Program : IThemeable
{
    static int _I = 1;
    delegate void DoSomething();
```

Changing Your Colors

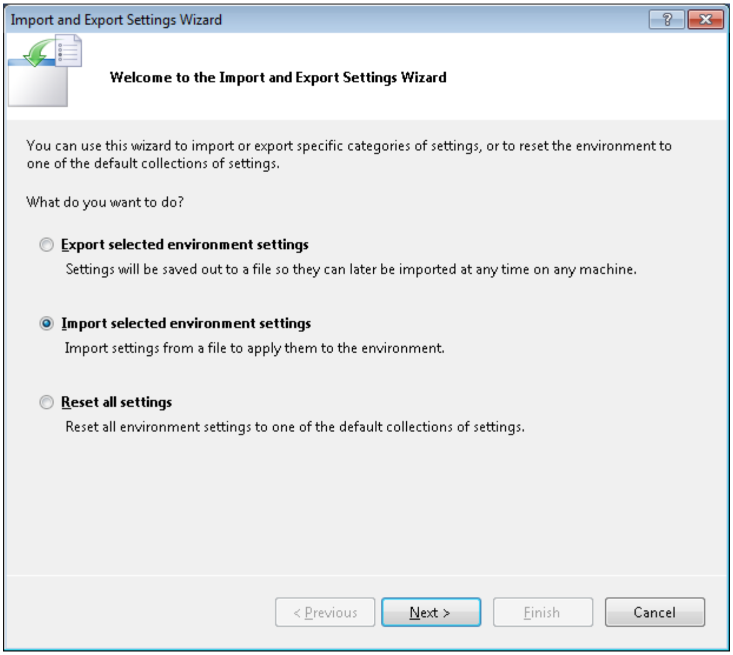
When you have a .vssettings file you want to import, copy or move the file to your computer. While you can put the file anywhere you want on your system, I prefer to put it with the other settings files located at C:\Users\<user>\Documents\Visual Studio <version>\Settings:



Name	Date modified	Type
CurrentSettings.vssettings	6/18/2011 4:46 PM	Visual Studio Settings File
Exported-2011-06-18.vssettings	6/18/2011 12:16 PM	Visual Studio Settings File
seriously cool settings.vssettings	6/18/2011 5:02 PM	Visual Studio Settings File
wekeroad-ink.vssettings	6/18/2011 4:44 PM	Visual Studio Settings File

seriously cool settings.vssettings Date modified: 6/18/2011 5:02 PM
Visual Studio Settings File Size: 6.24 KB

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



Import and Export Settings Wizard

Welcome to the Import and Export Settings Wizard

You can use this wizard to import or export specific categories of settings, or to reset the environment to one of the default collections of settings.

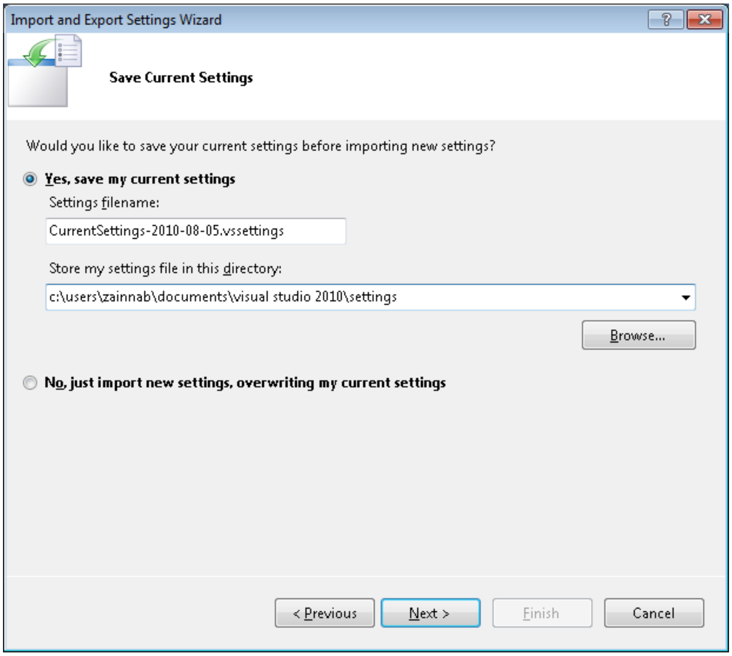
What do you want to do?

○ **Export selected environment settings**
Settings will be saved out to a file so they can later be imported at any time on any machine.

● **Import selected environment settings**
Import settings from a file to apply them to the environment.

○ **Reset all settings**
Reset all environment settings to one of the default collections of settings.

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:



Import and Export Settings Wizard

Save Current Settings

Would you like to save your current settings before importing new settings?

● **Yes, save my current settings**

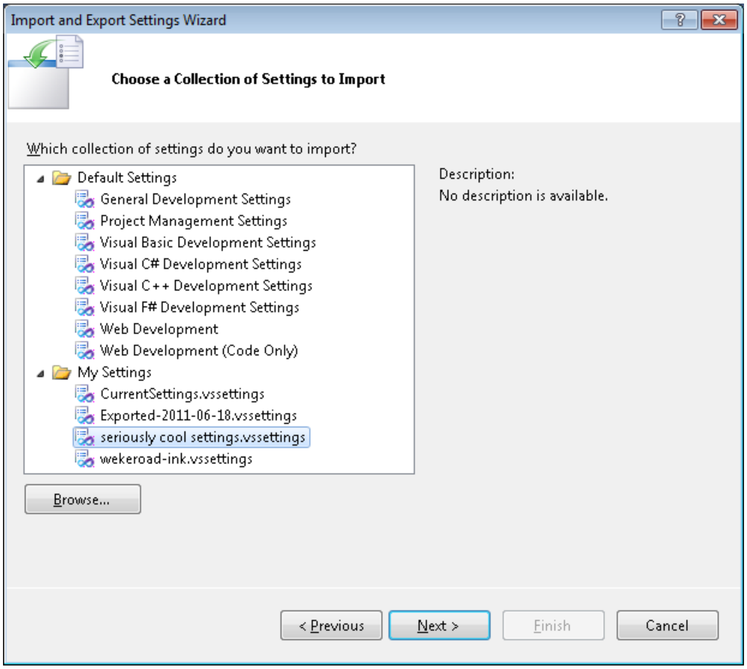
Settings filename:
CurrentSettings-2010-08-05.vssettings

Store my settings file in this directory:
c:\users\zainnab\documents\visual studio 2010\settings

○ **No, just import new settings, overwriting my current 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.



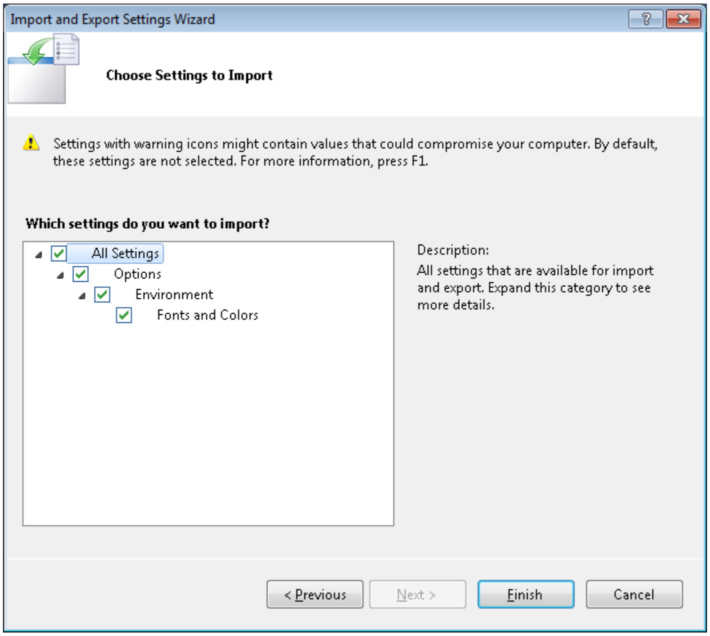
Choose a Collection of Settings to Import

Which collection of settings do you want to import?

- Default Settings
  - General Development Settings
  - Project Management Settings
  - Visual Basic Development Settings
  - Visual C# Development Settings
  - Visual C++ Development Settings
  - Visual F# Development Settings
  - Web Development
  - Web Development (Code Only)
- My Settings
  - CurrentSettings.vssettings
  - Exported-2011-06-18.vssettings
  - seriously cool settings.vssettings
  - wekeroad-ink.vssettings

Description:
No description is available.

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



Import and Export Settings Wizard

Choose Settings to Import

Settings with warning icons might contain values that could compromise your computer. By default, these settings are not selected. For more information, press F1.

Which settings do you want to import?

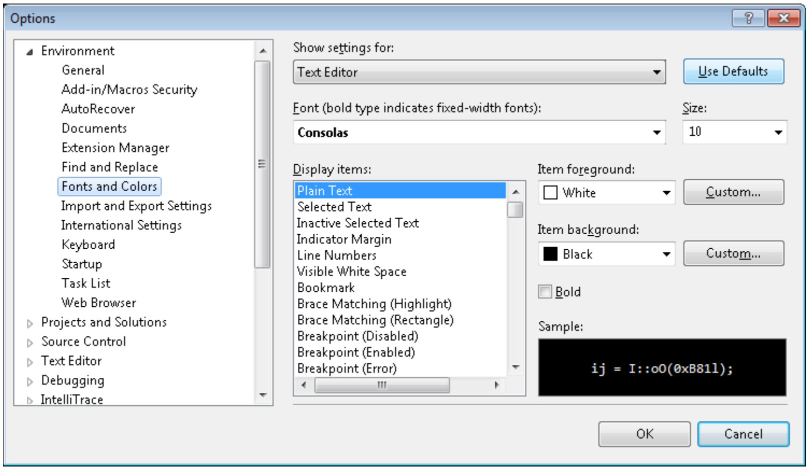
- ☑ All Settings
  - ☑ Options
    - ☑ Environment
      - ☑ Fonts and Colors

Description:
All settings that are available for import and export. Expand this category to see more details.

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.



Options

- Environment
  - General
  - Add-in/Macros Security
  - AutoRecover
  - Documents
  - Extension Manager
  - Find and Replace
  - Fonts and Colors
  - Import and Export Settings
  - International Settings
  - Keyboard
  - Startup
  - Task List
  - Web Browser
- Projects and Solutions
- Source Control
- Text Editor
- Debugging
- IntelliTrace

Show settings for: Text Editor

Font (bold type indicates fixed-width fonts): **Consolas**

Size: 10

Display items:
Plain Text
Selected Text
Inactive Selected Text
Indicator Margin
Line Numbers
Visible White Space
Bookmark
Brace Matching (Highlight)
Brace Matching (Rectangle)
Breakpoint (Disabled)
Breakpoint (Enabled)
Breakpoint (Error)

Item foreground: □ White

Item background: ■ Black

☐ Bold

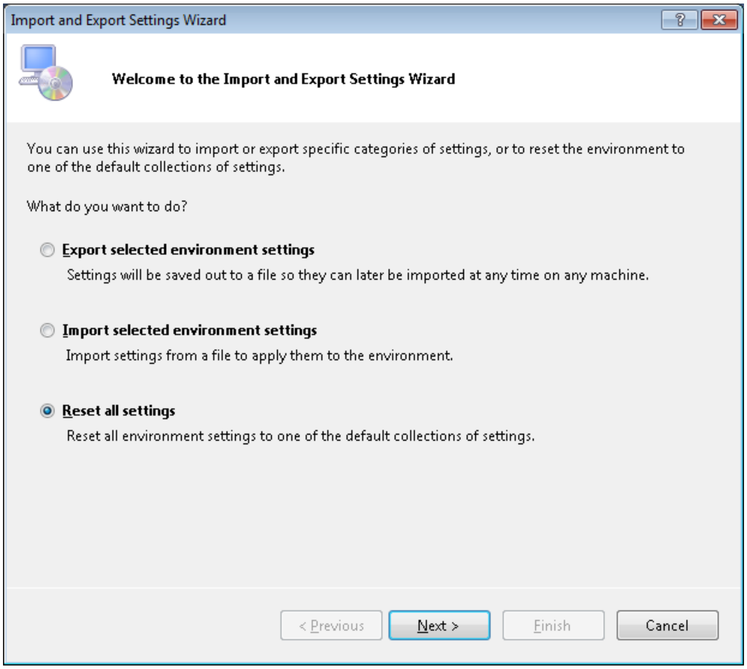
Sample:

```
ij = I::oO(0xB811);
```

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:



Import and Export Settings Wizard

Welcome to the Import and Export Settings Wizard

You can use this wizard to import or export specific categories of settings, or to reset the environment to one of the default collections of settings.

What do you want to do?

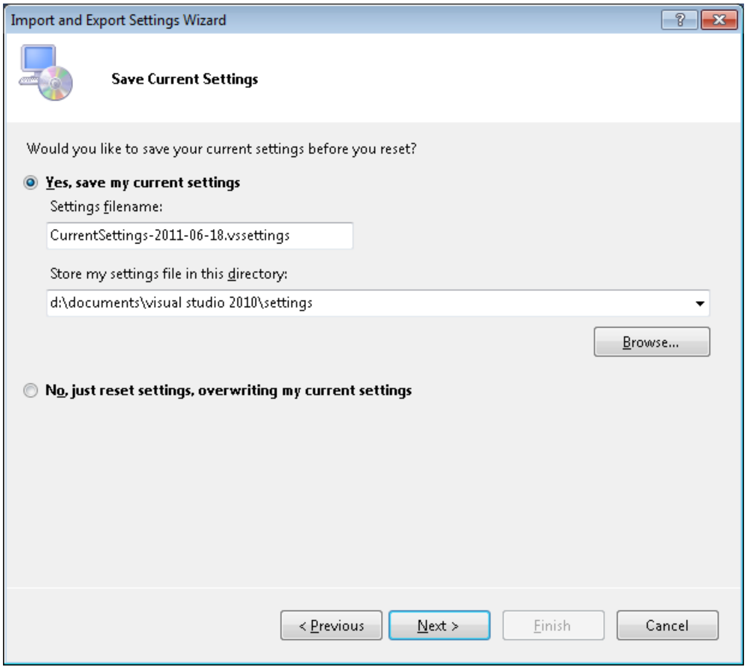
○ Export selected environment settings
Settings will be saved out to a file so they can later be imported at any time on any machine.

○ Import selected environment settings
Import settings from a file to apply them to the environment.

● Reset all settings
Reset all environment settings to one of the default collections of 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.*



Save Current Settings

Would you like to save your current settings before you reset?

● Yes, save my current settings

Settings filename:

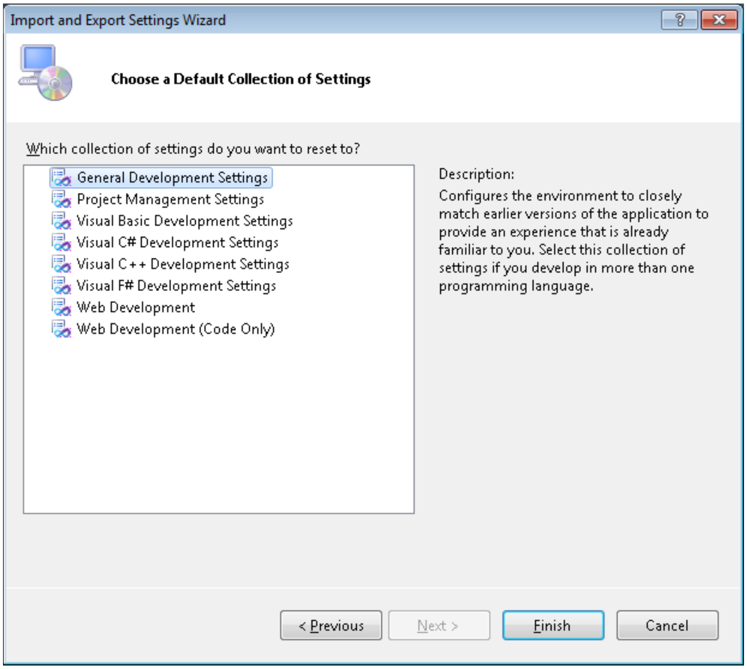
CurrentSettings-2011-06-18.vssettings

Store my settings file in this directory:

d:\documents\visual studio 2010\settings

○ No, just reset settings, overwriting my current settings

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



Choose a Default Collection of Settings

Which collection of settings do you want to reset to?

- General Development Settings
- Project Management Settings
- Visual Basic Development Settings
- Visual C# Development Settings
- Visual C++ Development Settings
- Visual F# Development Settings
- Web Development
- Web Development (Code Only)

Description:

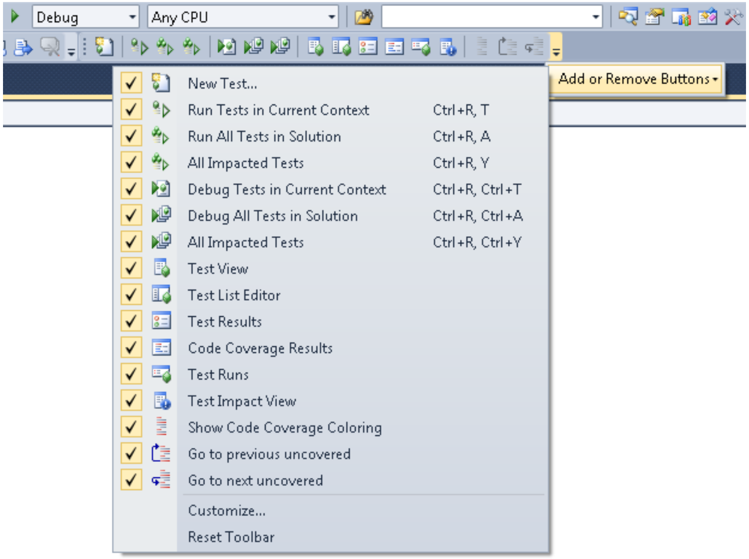
Configures the environment to closely match earlier versions of the application to provide an experience that is already familiar to you. Select this collection of settings if you develop in more than one programming language.

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).

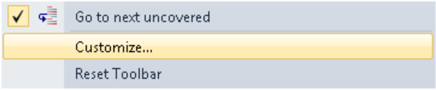
01.11 Customize Your Toolbars in Visual Studio 2010: Toolbars Tab

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

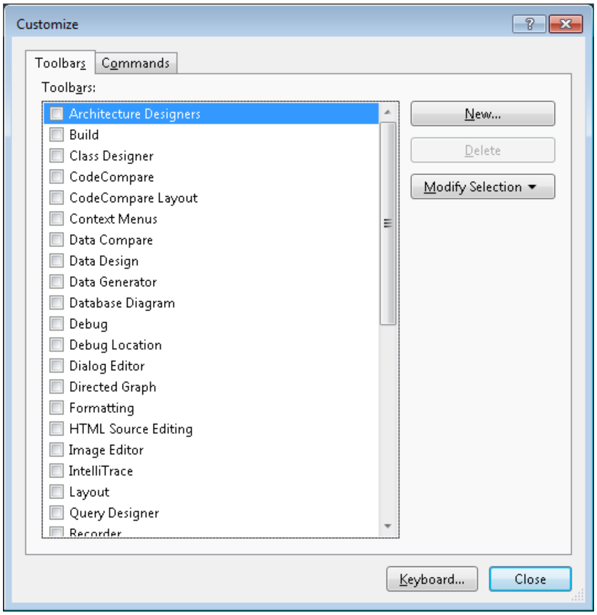
You can customize any toolbar in Visual Studio 2010. Just click the drop-down arrow to the right of any toolbar, and then click Add Or Remove Buttons:


Debug
Any CPU
Add or Remove Buttons
New Test...
Run Tests in Current Context Ctrl+R, T
Run All Tests in Solution Ctrl+R, A
All Impacted Tests Ctrl+R, Y
Debug Tests in Current Context Ctrl+R, Ctrl+T
Debug All Tests in Solution Ctrl+R, Ctrl+A
All Impacted Tests Ctrl+R, Ctrl+Y
Test View
Test List Editor
Test Results
Code Coverage Results
Test Runs
Test Impact View
Show Code Coverage Coloring
Go to previous uncovered
Go to next uncovered
Customize...
Reset Toolbar

Then click Customize:

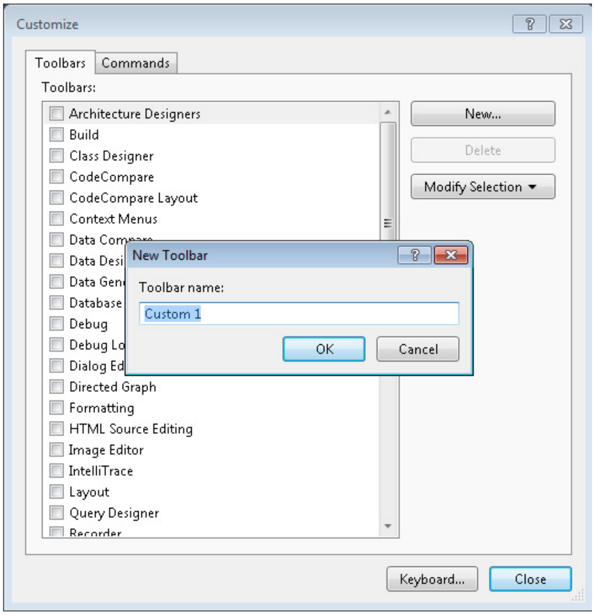


Alternatively, you can go to Tools | Customize on the menu bar. Whichever option you choose opens the Customize dialog box:



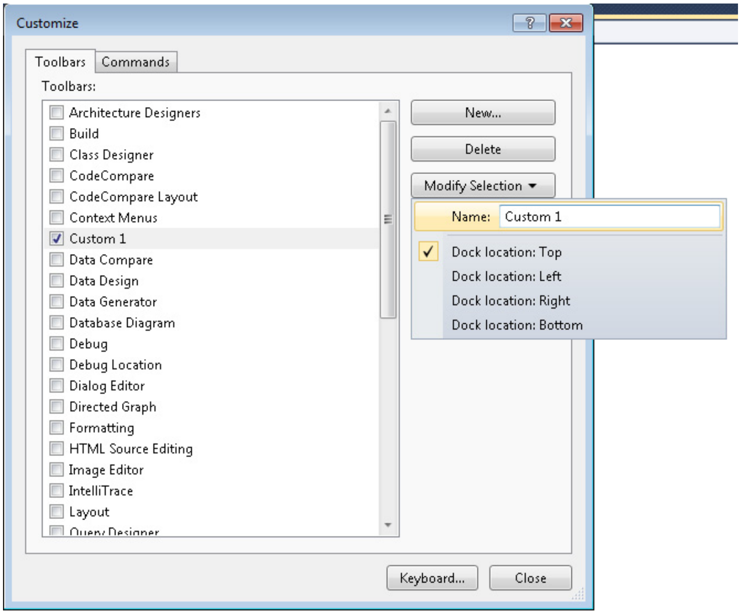
Custom Toolbars

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:



Toolbar name: Custom 1

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:

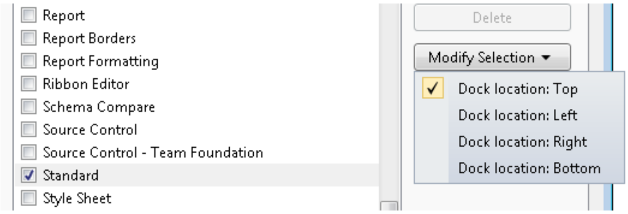


☐ Architecture Designers
☐ Build
☐ Class Designer
☐ CodeCompare
☐ CodeCompare Layout
☐ Context Menus
☑ Custom 1
☐ Data Compare
☐ Data Design
☐ Data Generator
☐ Database Diagram
☐ Debug
☐ Debug Location
☐ Dialog Editor
☐ Directed Graph
☐ Formatting
☐ HTML Source Editing
☐ Image Editor
☐ IntelliTrace
☐ Layout
☐ Query Designer

Name: Custom 1

✓ Dock location: Top
Dock location: Left
Dock location: Right
Dock location: Bottom

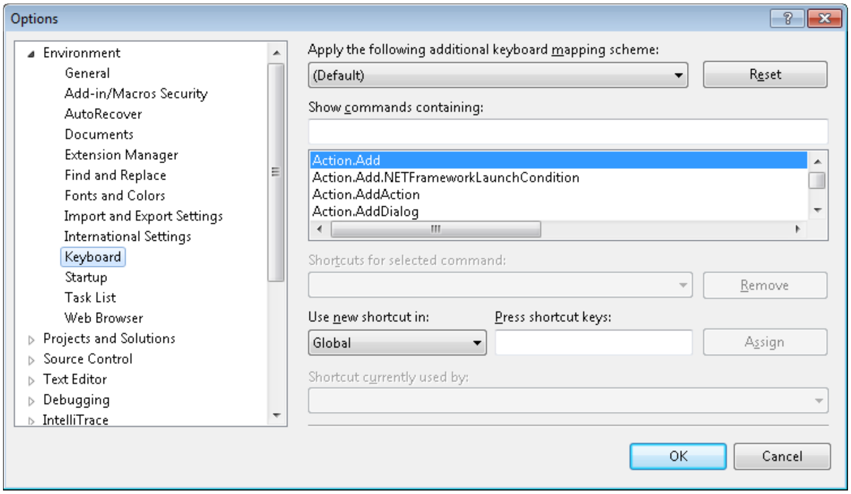
Although you can rename custom toolbars by clicking Modify Selection, default toolbars can't be changed in this way:



☐ Report
☐ Report Borders
☐ Report Formatting
☐ Ribbon Editor
☐ Schema Compare
☐ Source Control
☐ Source Control - Team Foundation
☑ Standard
☐ Style Sheet

✓ Dock location: Top
Dock location: Left
Dock location: Right
Dock location: Bottom

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.)



Apply the following additional keyboard mapping scheme: (Default)

Show commands containing:

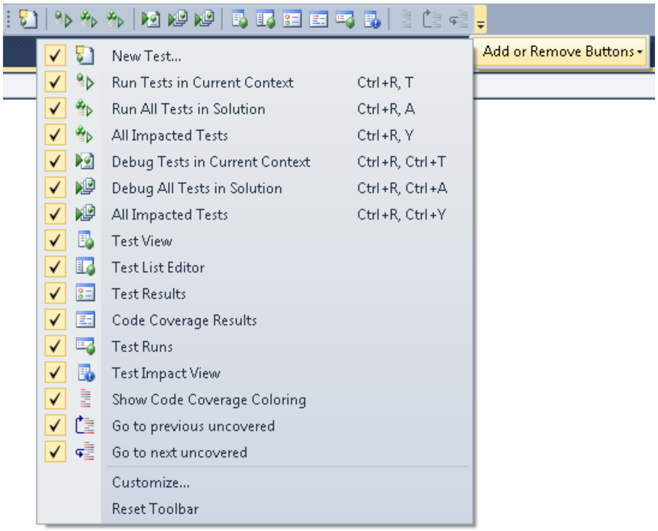
Use new shortcut in: Global

Press shortcut keys:

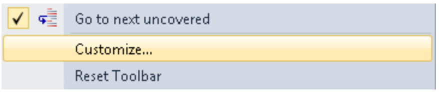
01.12 Customize Your Toolbars in Visual Studio 2010: Commands Tab

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:

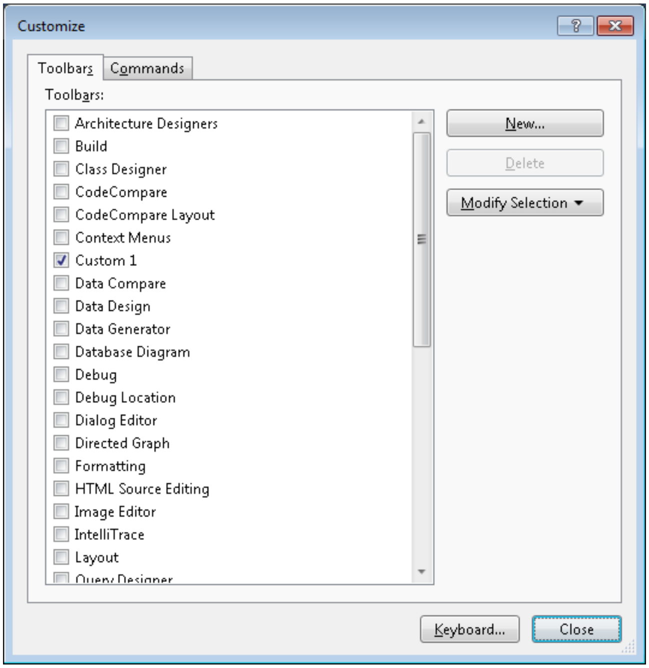

Add or Remove Buttons
New Test...
Run Tests in Current Context Ctrl+R, T
Run All Tests in Solution Ctrl+R, A
All Impacted Tests Ctrl+R, Y
Debug Tests in Current Context Ctrl+R, Ctrl+T
Debug All Tests in Solution Ctrl+R, Ctrl+A
All Impacted Tests Ctrl+R, Ctrl+Y
Test View
Test List Editor
Test Results
Code Coverage Results
Test Runs
Test Impact View
Show Code Coverage Coloring
Go to previous uncovered
Go to next uncovered
Customize...
Reset Toolbar

Then click Customize:



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

Either option you choose opens the Customize dialog box:

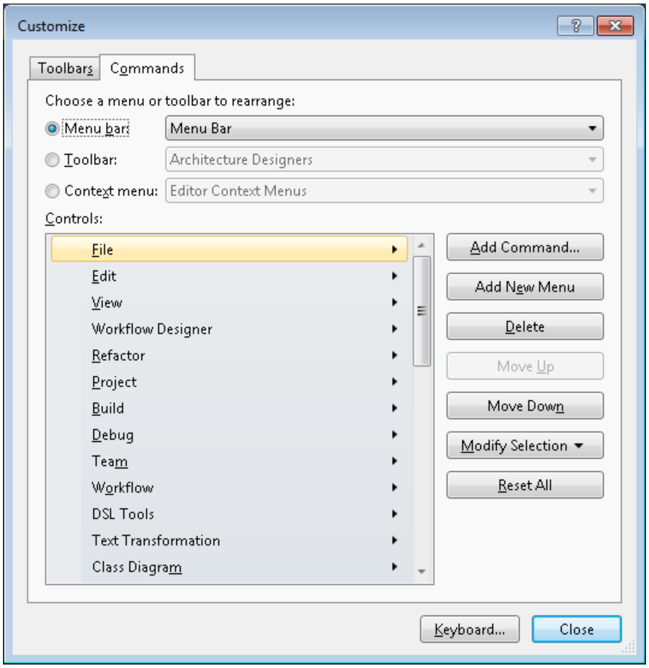


Toolbars:

- ☐ Architecture Designers
- ☐ Build
- ☐ Class Designer
- ☐ CodeCompare
- ☐ CodeCompare Layout
- ☐ Context Menus
- ☑ Custom 1
- ☐ Data Compare
- ☐ Data Design
- ☐ Data Generator
- ☐ Database Diagram
- ☐ Debug
- ☐ Debug Location
- ☐ Dialog Editor
- ☐ Directed Graph
- ☐ Formatting
- ☐ HTML Source Editing
- ☐ Image Editor
- ☐ IntelliTrace
- ☐ Layout
- ☐ Query Designer

Click the Commands tab:

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



Choose a menu or toolbar to rearrange:

● Menu bar: Menu Bar

○ Toolbar: Architecture Designers

○ Context menu: Editor Context Menus

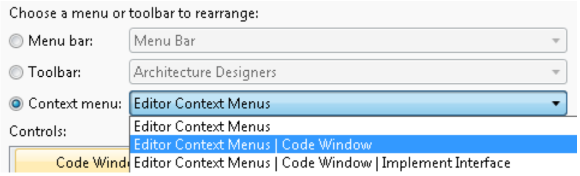
Controls:

File
Edit
View
Workflow Designer
Refactor
Project
Build
Debug
Team
Workflow
DSL Tools
Text Transformation
Class Diagram

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:



Choose a menu or toolbar to rearrange:

○ Menu bar: Menu Bar

○ Toolbar: Architecture Designers

● Context menu: Editor Context Menus

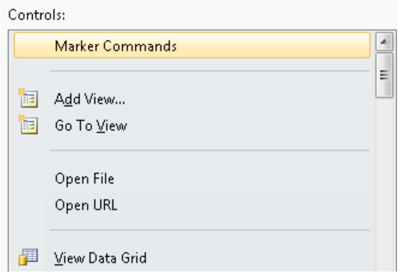
Editor Context Menus
Editor Context Menus | Code Window
Editor Context Menus | Code Window | Implement Interface

Controls:

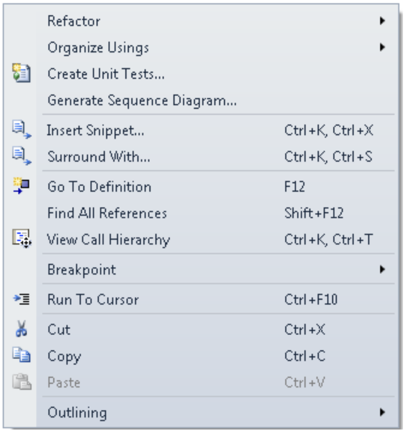
Code Wind

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:



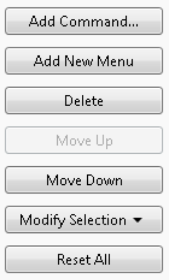
Remember that not all the items you see are available all the time because these items show up only in the proper context. So while it seems you have a large number of buttons currently available, when you right-click in your code window, this is an example of what you will currently see:


Refactor
Organize Usings
Create Unit Tests...
Generate Sequence Diagram...
Insert Snippet... Ctrl+K, Ctrl+X
Surround With... Ctrl+K, Ctrl+S
Go To Definition F12
Find All References Shift+F12
View Call Hierarchy Ctrl+K, Ctrl+T
Breakpoint
Run To Cursor Ctrl+F10
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Outlining

Buttons

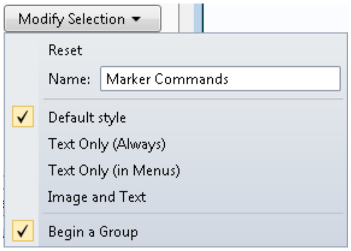
Finally, let's look at the area of the dialog box that has all the buttons that actually *perform* actions:

- **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:



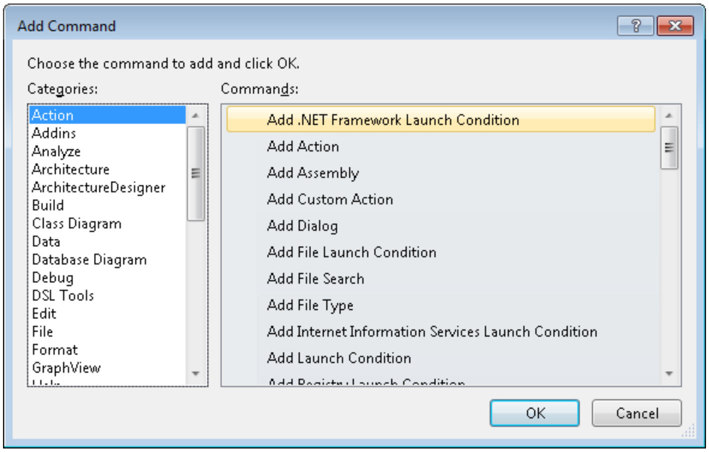
Name: Marker Commands

☑ Default style

☑ Begin a Group

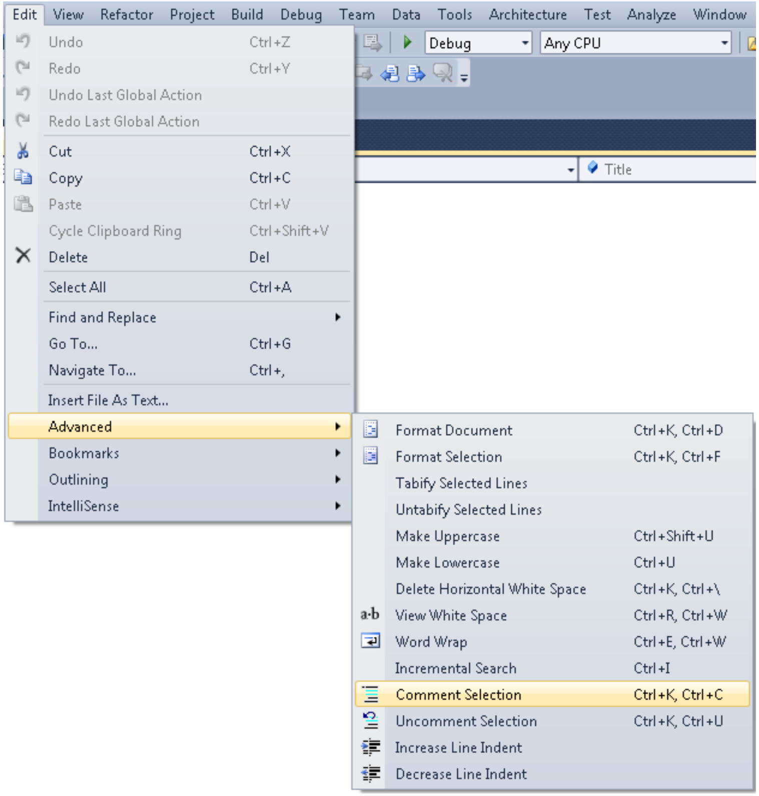
Finally, the Reset All option resets every item in the Controls area to its default settings. This capability is particularly useful if you have made a lot of changes.

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:

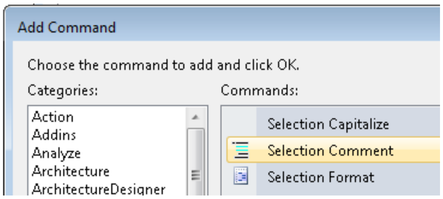


Choose the command to add and click OK.

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:

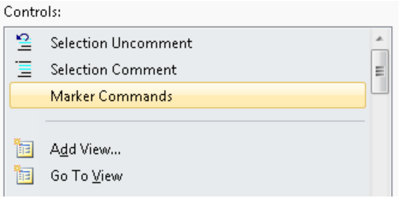


Because the items you want are off the Edit menu, you can search there first. It turns out the items you want are called Selection Comment and Selection Uncomment:

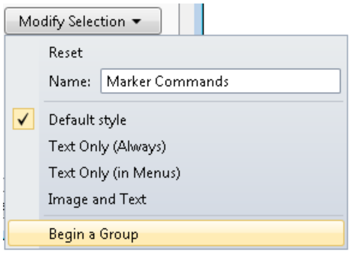

Add Command

Choose the command to add and click OK.

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

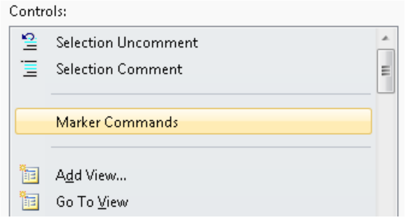


It would be nice to have these buttons in their own group, so select the item in the Controls dialog box, shown in the preceding illustration, where you would like the group line to be (Marker Commands, in this case), and then click Modify Selection and choose Begin A Group:

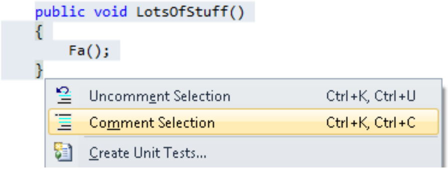


Name: Marker Commands

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



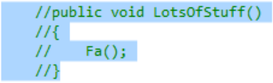
Click Close and go to any code area. Select some code, right-click, and select Comment Selection:



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

Uncomment Selection Ctrl+K, Ctrl+U
Comment Selection Ctrl+K, Ctrl+C
Create Unit Tests...

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.

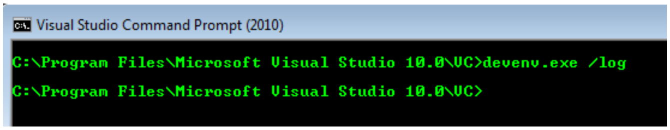
Essentially, the syntax is as follows:

```
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:

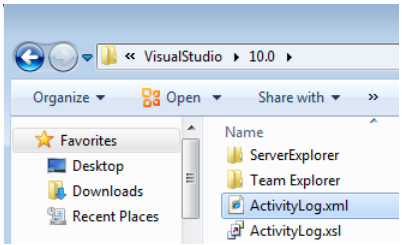
```
%APPDATA%\Microsoft\VisualStudio\<version>\ActivityLog.xml
```

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

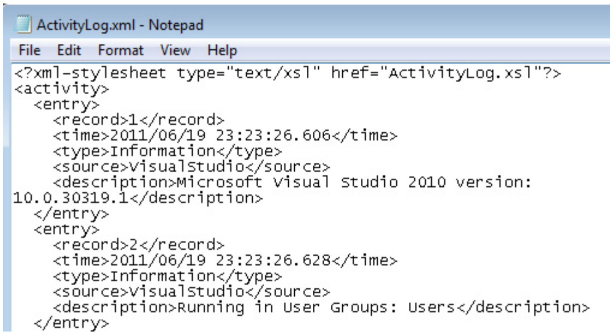


```
C:\Program Files\Microsoft Visual Studio 10.0\VC>devenv.exe /log
C:\Program Files\Microsoft Visual Studio 10.0\VC>
```

You can then navigate to the file location:

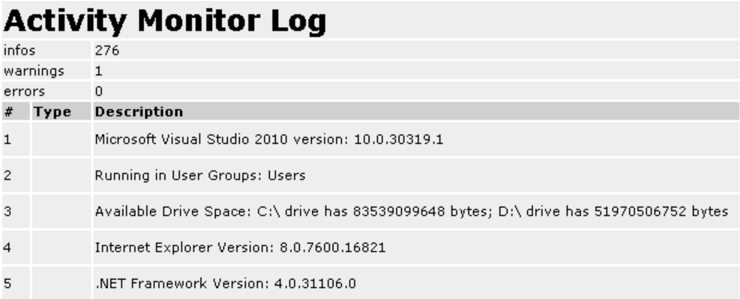


When you open the log file, the following illustration provides an example of what you might see:



```
<?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

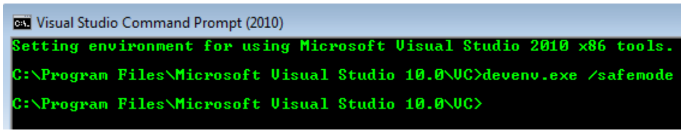
#	Type	Description
1		Microsoft Visual Studio 2010 version: 10.0.30319.1
2		Running in User Groups: Users
3		Available Drive Space: C:\ drive has 83539099648 bytes; D:\ drive has 51970506752 bytes
4		Internet Explorer Version: 8.0.7600.16821
5		.NET Framework Version: 4.0.31106.0

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

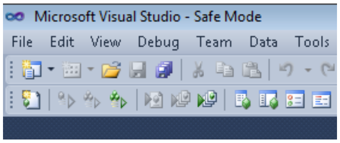
Occasionally you have a situation where Visual Studio might not start up correctly or at all. Using Visual Studio in safe mode, you can load only the default environment, services, and shipped versions of third-party packages to see whether the problem is caused by one or more third-party add-ins. Just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



```
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:


Microsoft Visual Studio - Safe Mode
File Edit View Debug Team Data Tools

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

01.15 The ResetSettings Switch

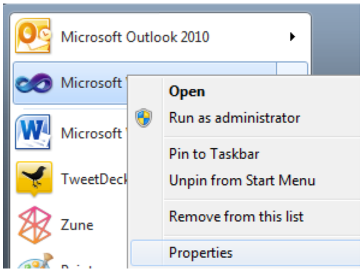
VERSIONS	2005, 2008, 2010
CODE	vstipEnv0047

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.

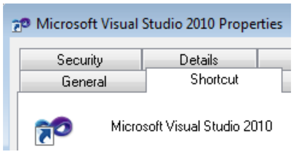
Let's take a classic example: You might have two (or more) monitors at work, but when you get home, you work with just one monitor. Your window layouts (among other things) could be very different in each place. In vstipEnv0040, "Export Your Window Layouts" (page 134), I showed you how to export just 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

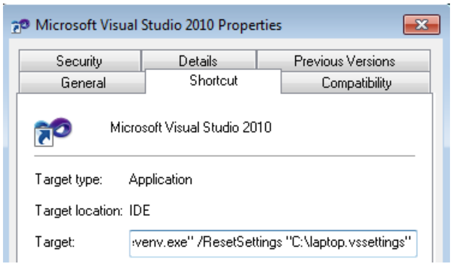
The question is: What do you do once you have exported the window layouts? Well, now you put the .vssettings files where you can easily get to them on your machines and then you go to the Properties dialog box of the Visual Studio program icon:



Click the Shortcut tab:


Microsoft Visual Studio 2010 Properties
Security
Details
General
Shortcut
Microsoft Visual Studio 2010

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


Microsoft Visual Studio 2010 Properties
Security
Details
Previous Versions
General
Shortcut
Compatibility
Microsoft Visual Studio 2010

Target type: Application

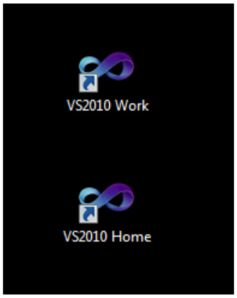
Target location: IDE

Target: :venv.exe" /ResetSettings "C:\laptop.vssettings"

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."

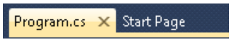
—Stephen King, "Secret Window, Secret Garden" in "Four Past Midnight"

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

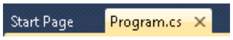
By default, Visual Studio 2010 opens up new tabs to the left of existing ones, as shown in the following illustration.



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

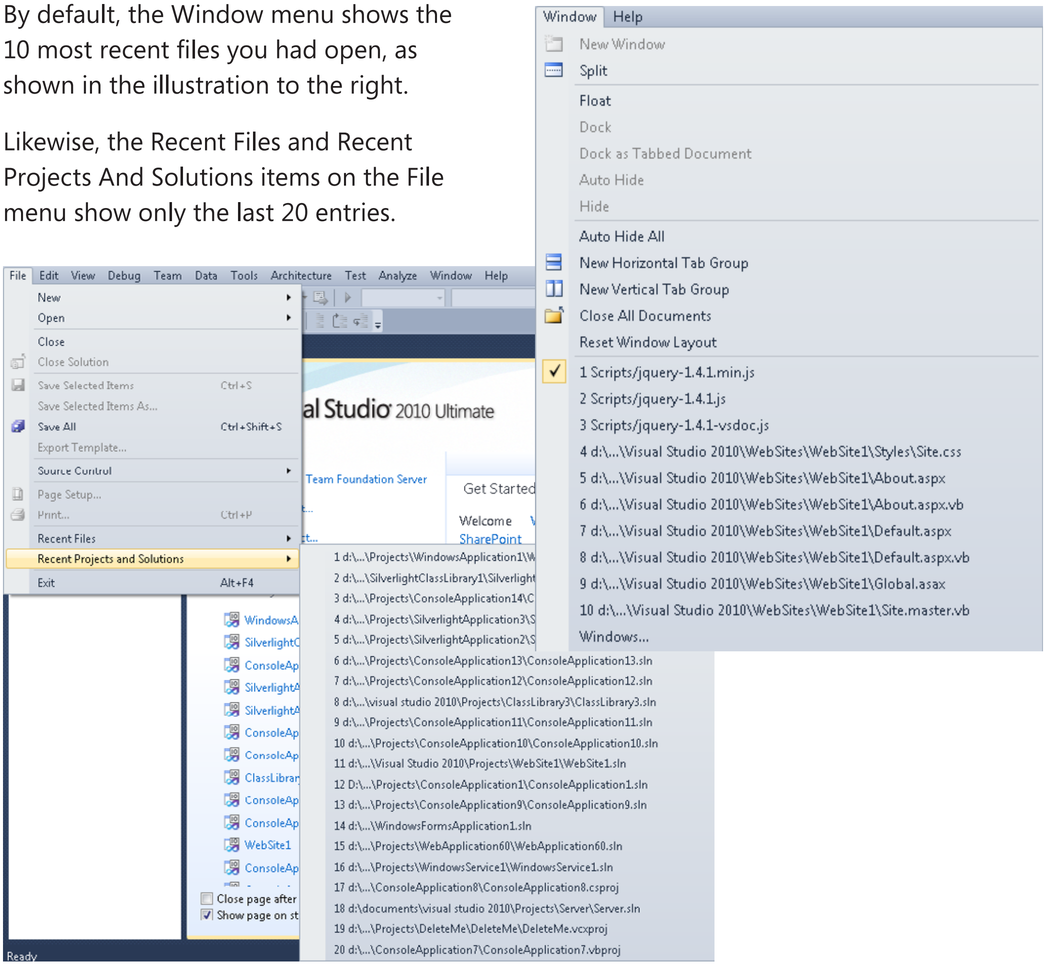
WINDOWS	Alt,W, [1,2,3, etc] (windows);
	Alt,F, F, [1,2,3, etc] (files);
	Alt,F, J, [1,2,3, etc] (projects and solutions)

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]

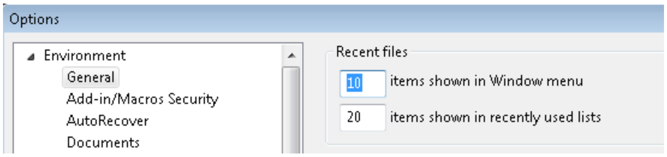
COMMAND	Tools.Options
VERSIONS	2005, 2008, 2010
CODE	vstipEnv0013

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.


Window
Help
New Window
Split
Float
Dock
Dock as Tabbed Document
Auto Hide
Hide
Auto Hide All
New Horizontal Tab Group
New Vertical Tab Group
Close All Documents
Reset Window Layout
✓ 1 Scripts/jquery-1.4.1.min.js
2 Scripts/jquery-1.4.1.js
3 Scripts/jquery-1.4.1-vsdoc.js
4 d:\...\Visual Studio 2010\WebSites\WebSite1\Styles\Site.css
5 d:\...\Visual Studio 2010\WebSites\WebSite1\About.aspx
6 d:\...\Visual Studio 2010\WebSites\WebSite1\About.aspx.vb
7 d:\...\Visual Studio 2010\WebSites\WebSite1\Default.aspx
8 d:\...\Visual Studio 2010\WebSites\WebSite1\Default.aspx.vb
9 d:\...\Visual Studio 2010\WebSites\WebSite1\Global.asax
10 d:\...\Visual Studio 2010\WebSites\WebSite1\Site.master.vb
Windows...
File
Edit
View
Debug
Team
Data
Tools
Architecture
Test
Analyze
Window
Help
New
Open
Close
Close Solution
Save Selected Items Ctrl+S
Save Selected Items As...
Save All Ctrl+Shift+S
Export Template...
Source Control
Page Setup...
Print... Ctrl+P
Recent Files
Recent Projects and Solutions
Exit Alt+F4
al Studio 2010 Ultimate
Team Foundation Server
Get Started
Welcome
SharePoint
1 d:\...\Projects\WindowsApplication1\W
2 d:\...\SilverlightClassLibrary1\Silverlight
3 d:\...\Projects\ConsoleApplication14\C
4 d:\...\Projects\SilverlightApplication3\S
5 d:\...\Projects\SilverlightApplication2\S
6 d:\...\Projects\ConsoleApplication13\ConsoleApplication13.sln
7 d:\...\Projects\ConsoleApplication12\ConsoleApplication12.sln
8 d:\...\visual studio 2010\Projects\ClassLibrary3\ClassLibrary3.sln
9 d:\...\Projects\ConsoleApplication11\ConsoleApplication11.sln
10 d:\...\Projects\ConsoleApplication10\ConsoleApplication10.sln
11 d:\...\Visual Studio 2010\Projects\WebSite1\WebSite1.sln
12 D:\...\Projects\ConsoleApplication1\ConsoleApplication1.sln
13 d:\...\Projects\ConsoleApplication9\ConsoleApplication9.sln
14 d:\...\WindowsFormsApplication1.sln
15 d:\...\Projects\WebApplication60\WebApplication60.sln
16 d:\...\Projects\WindowsService1\WindowsService1.sln
17 d:\...\ConsoleApplication8\ConsoleApplication8.csproj
18 d:\documents\visual studio 2010\Projects\Server\Server.sln
19 d:\...\Projects\DeleteMe\DeleteMe\DeleteMe.vcxproj
20 d:\...\ConsoleApplication7\ConsoleApplication7.vbproj
WindowsA
SilverlightC
ConsoleAp
SilverlightA
SilverlightA
ConsoleAp
ConsoleAp
ClassLibrar
ConsoleAp
ConsoleAp
WebSite1
ConsoleAp
□ Close page after
☑ Show page on st
Ready

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



Recent files

10 items shown in Window menu

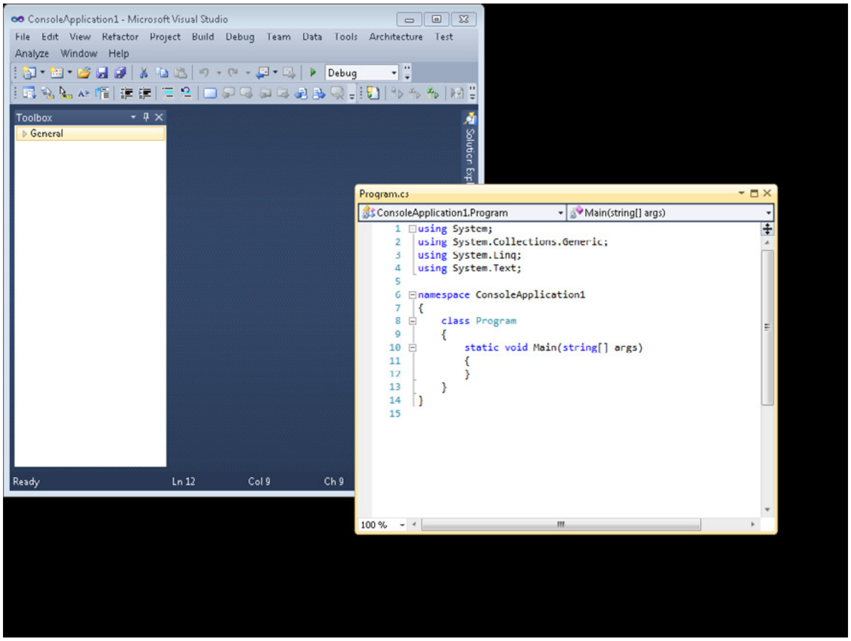
20 items shown in recently used lists

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

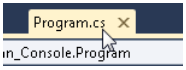
WINDOWS	Alt,W, F (float); Alt,W,T (dock)
MENU	Window | Float; Window | Dock as Tabbed Document
COMMAND	Window.Float; Window.DockasTabbedDocument
VERSIONS	2010
CODE	vstipTool0004

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.

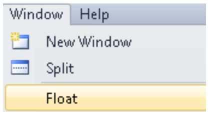

ConsoleApplication1 - Microsoft Visual Studio
File Edit View Refactor Project Build Debug Team Data Tools Architecture Test
Analyze Window Help
Debug
Toolbox
General
Solution Expl
Ready
Ln 12
Col 9
Ch 9
Program.cs
ConsoleApplication1.Program
Main(string[] args)
100 %

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
        }
    }
}

```

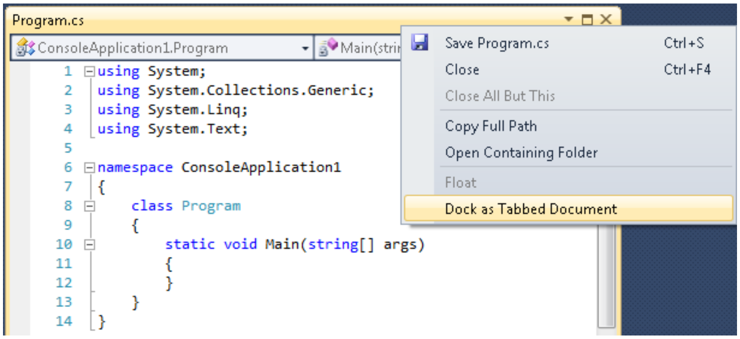
The first way is to simply click and drag the tab for the document window out of the IDE.



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



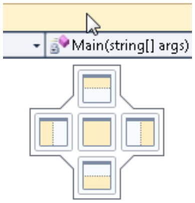
What if you want to put the window back? No worries; just right-click the title bar of the document and choose Dock As Tabbed Document.



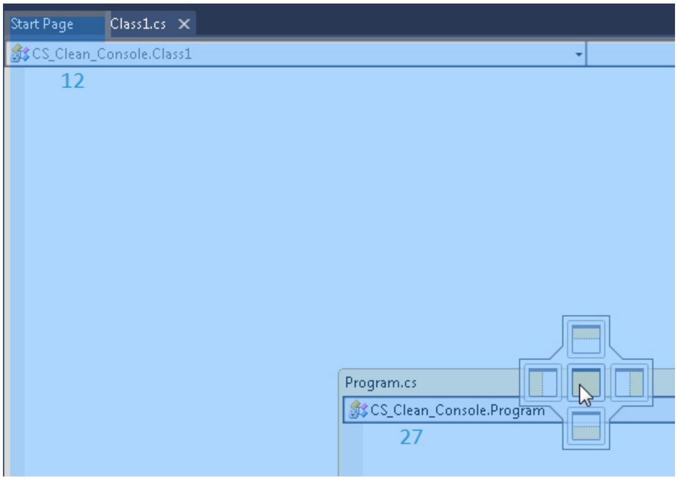
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
        }
    }
}
```

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.

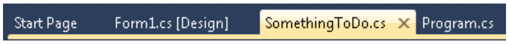


Release the mouse button, and it should dock where you want it to go.

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:



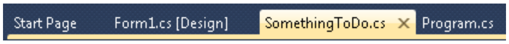
And you don’t want to use your mouse to switch between tabs. Just press Ctrl+F6 to go forward.

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.

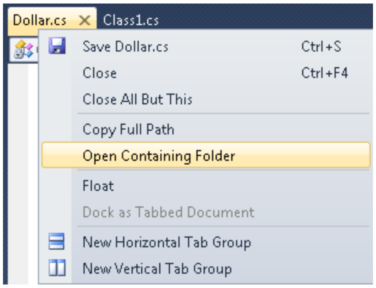


Then press Ctrl+F4. The current document closes, and it prompts you to save changes if you haven't already.

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.

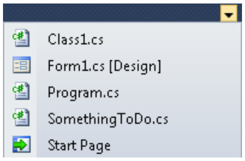


The file location opens in Windows Explorer, and you can manipulate the files from there.

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)

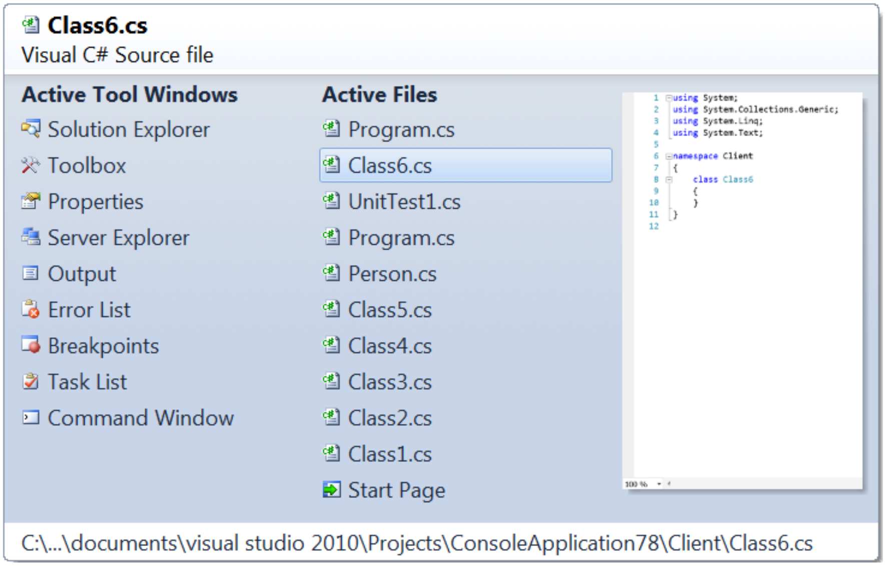
VISUAL STUDIO 6	Ctrl+Tab (forward in Active Files);
	Ctrl+Shift+Tab (backward in Active Files);
	Alt+F7 (forward in Active Tool Windows);
	Alt+Shift+F7 (backward in Active Tool Windows)

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.

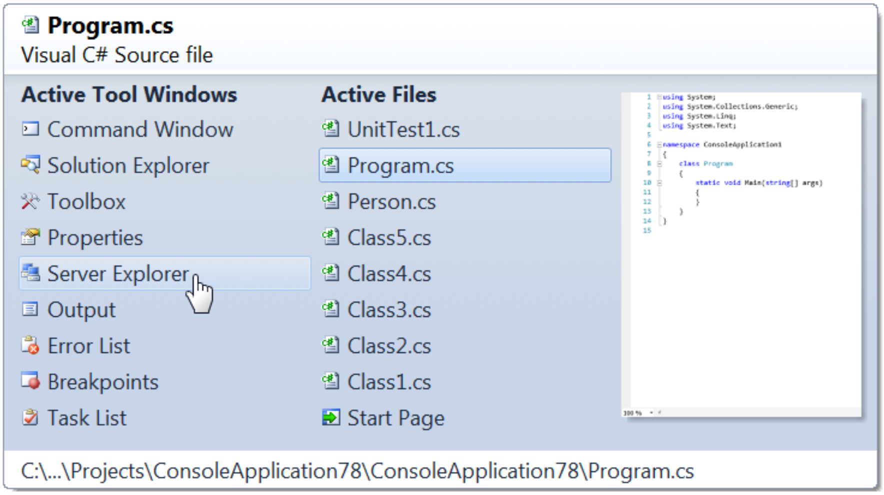


```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Client
{
    class Class6
    {
    }
}

```

C:\...\documents\visual studio 2010\Projects\ConsoleApplication78\Client\Class6.cs

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.



Program.cs
Visual C# Source file

Active Tool Windows	Active Files
Command Window	UnitTest1.cs
Solution Explorer	Program.cs
Toolbox	Person.cs
Properties	Class5.cs
Server Explorer	Class4.cs
Output	Class3.cs
Error List	Class2.cs
Breakpoints	Class1.cs
Task List	Start Page

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
        }
    }
}
```

C:\...\Projects\ConsoleApplication78\ConsoleApplication78\Program.cs

Navigator Areas

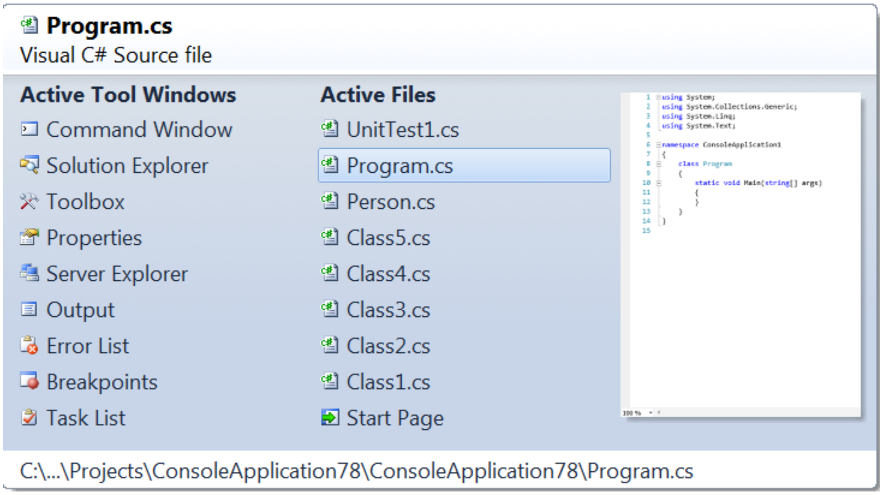
Let's take a look at the two major areas in the navigator: Active Files and Active Tool Windows.

Active files

To navigate active files, press Ctrl+Tab to go forward and Ctrl+Shift+Tab to go backward though 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.



Program.cs

Visual C# Source file

Active Tool Windows

- Command Window
- Solution Explorer
- Toolbox
- Properties
- Server Explorer
- Output
- Error List
- Breakpoints
- Task List

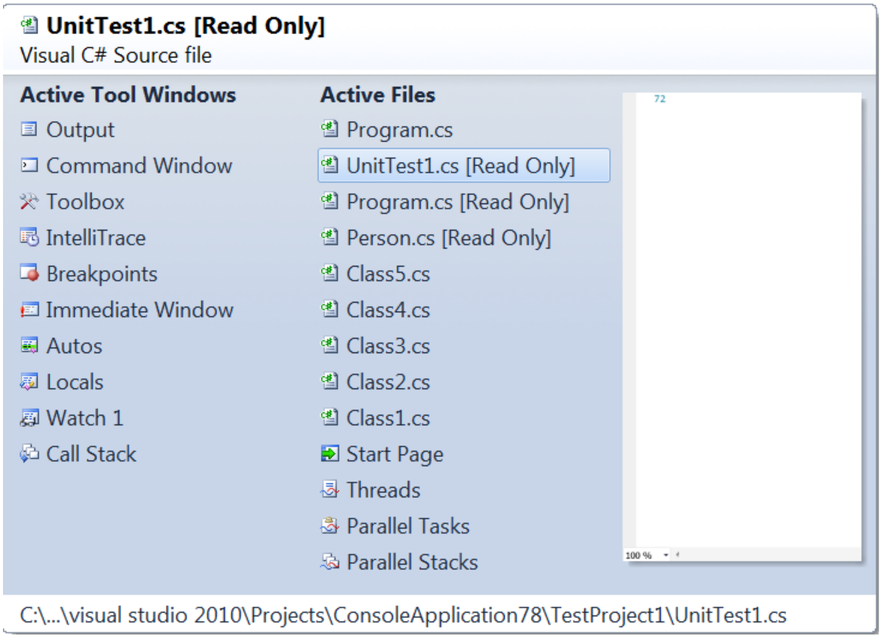
Active Files

- UnitTest1.cs
- Program.cs
- Person.cs
- Class5.cs
- Class4.cs
- Class3.cs
- Class2.cs
- Class1.cs
- Start Page

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
        }
    }
}
```

C:\...\Projects\ConsoleApplication78\ConsoleApplication78\Program.cs

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



UnitTest1.cs [Read Only]

Visual C# Source file

Active Tool Windows

- Output
- Command Window
- Toolbox
- IntelliTrace
- Breakpoints
- Immediate Window
- Autos
- Locals
- Watch 1
- Call Stack

Active Files

- Program.cs
- UnitTest1.cs [Read Only]
- Program.cs [Read Only]
- Person.cs [Read Only]
- Class5.cs
- Class4.cs
- Class3.cs
- Class2.cs
- Class1.cs
- Start Page
- Threads
- Parallel Tasks
- Parallel Stacks

C:\...\visual studio 2010\Projects\ConsoleApplication78\TestProject1\UnitTest1.cs

04.09 Multiple Views of the Same Document

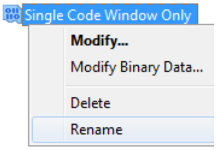
WINDOWS	Alt,W, N
MENU	Window | New Window
COMMAND	Window.NewWindow
VERSIONS	2005, 2008, 2010
CODE	vstipEnv0016

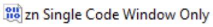
Sometimes you might want to look at a particularly large document in several different areas at the same time. For example, you might want to look at the same document on multiple monitors. This tip shows you how to make this happen.

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 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:



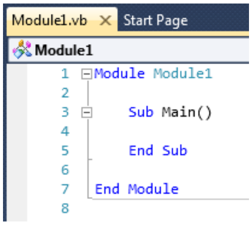


Now restart Visual Studio, and you are good to go for the rest of this tip.

Multiple Views

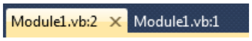
I came across this while I was checking my email one day and noticed a thread started by the legendary Deborah Kurata concerning the Window | New Window menu item. The following example describes how it works.

Open a document window.

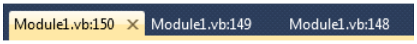


```
Module Module1

    Sub Main()

    End Sub

End Module

```

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



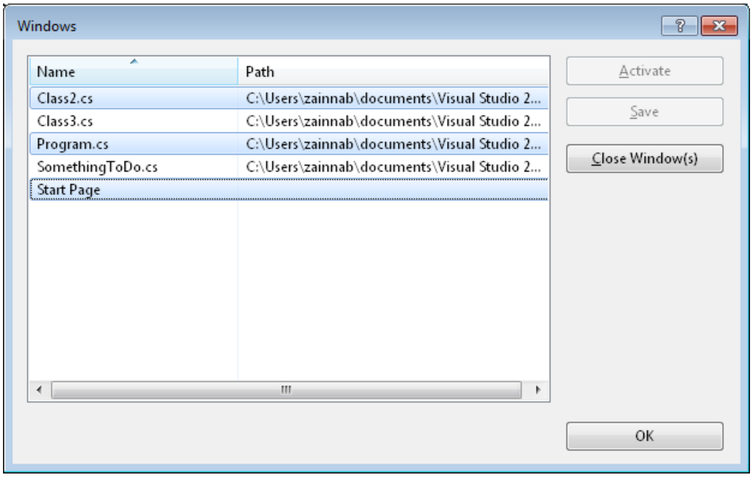
Notice that ":1" is added to the existing document tab text and that ":2" is appended to the name on the new document tab. You can apparently do this *forever* (or at least up to 150, which is as high as I have tested this feature).



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.



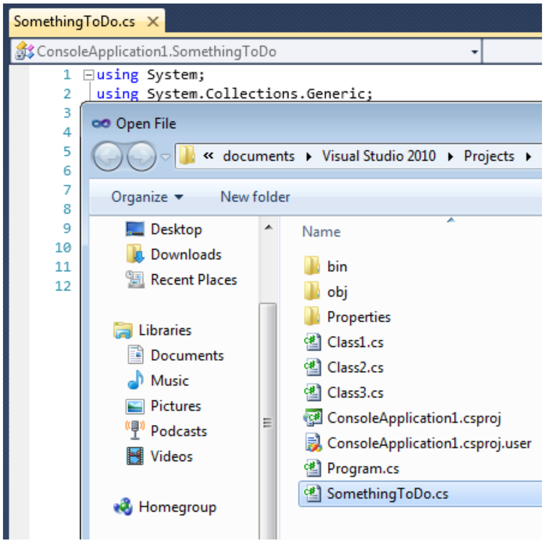
Name	Path
Class2.cs	C:\Users\zainnab\documents\Visual Studio 2...
Class3.cs	C:\Users\zainnab\documents\Visual Studio 2...
Program.cs	C:\Users\zainnab\documents\Visual Studio 2...
SomethingToDo.cs	C:\Users\zainnab\documents\Visual Studio 2...
Start Page	

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

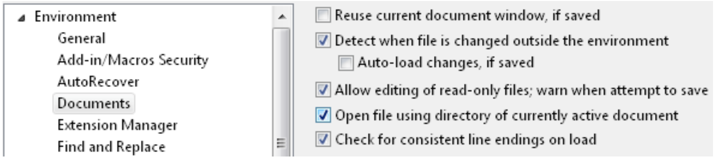
Have you ever noticed that when you go to open a file in Visual Studio (Ctrl+O) it automatically uses the directory of the current active document?



```
using System;
using System.Collections.Generic;
```

bin
obj
Properties
Class1.cs
Class2.cs
Class3.cs
ConsoleApplication1.csproj
ConsoleApplication1.csproj.user
Program.cs
SomethingToDo.cs

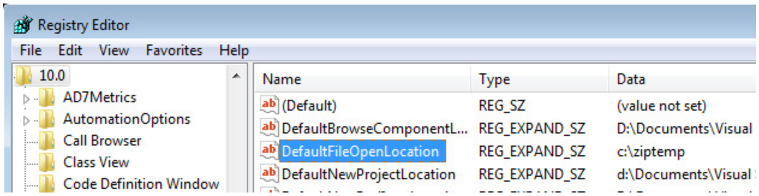
This is controlled by the Open File Using Directory Of Currently Active Document option. You can find this at Tools | Options | Environment | Documents.



- Environment
  - General
  - Add-in/Macros Security
  - AutoRecover
  - Documents
  - Extension Manager
  - Find and Replace

□ Reuse current document window, if saved
☑ Detect when file is changed outside the environment
□ Auto-load changes, if saved
☑ Allow editing of read-only files; warn when attempt to save
☑ Open file using directory of currently active document
☑ Check for consistent line endings on load

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.



- 10.0
  - AD7Metrics
  - AutomationOptions
  - Call Browser
  - Class View
  - Code Definition Window

Name	Type	Data
(Default)	REG_SZ	(value not set)
DefaultBrowseComponentL...	REG_EXPAND_SZ	D:\Documents\Visual
DefaultFileOpenLocation	REG_EXPAND_SZ	c:\ziptemp
DefaultNewProjectLocation	REG_EXPAND_SZ	d:\Documents\Visual

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.

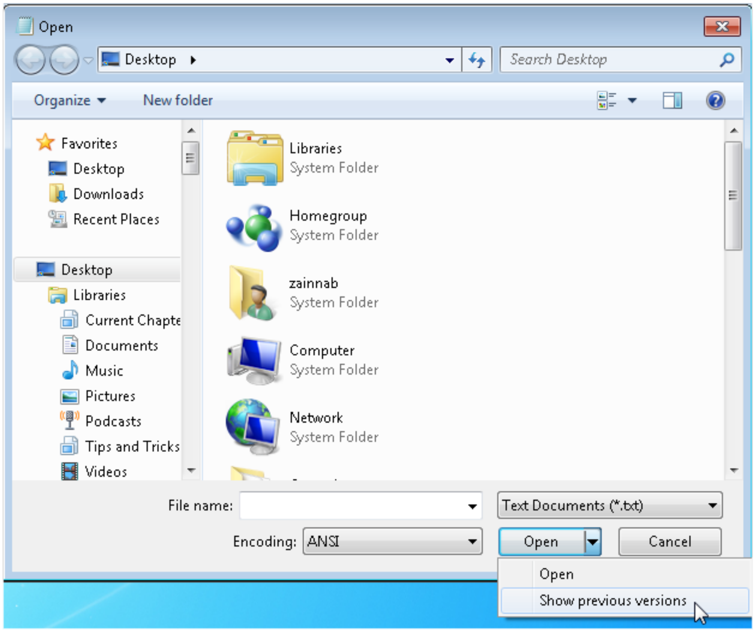
04.12 Show Previous Versions

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*.

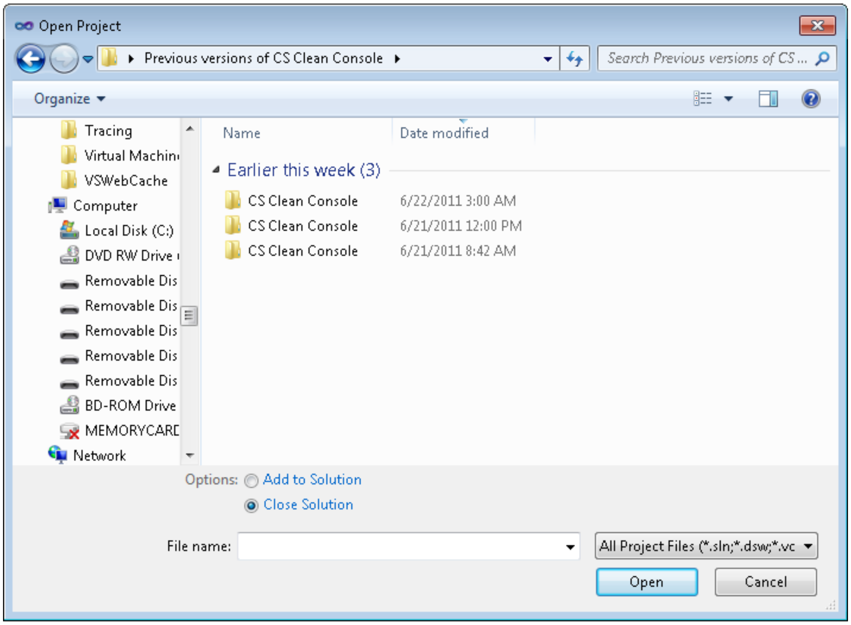
Ever want to go back in time when you save a change to your code that you didn't want saved? If you use source control, you are usually OK, but if you don't, this tip is for you.

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.



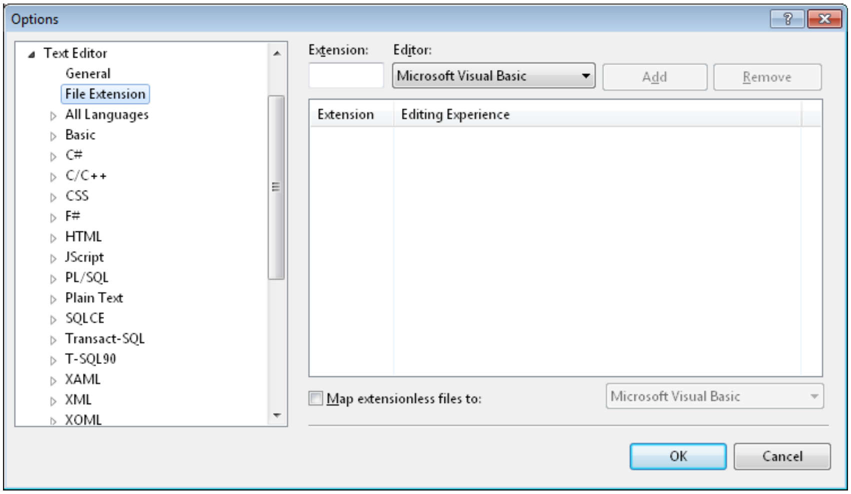
Name	Date modified
Earlier this week (3)	
CS Clean Console	6/22/2011 3:00 AM
CS Clean Console	6/21/2011 12:00 PM
CS Clean Console	6/21/2011 8:42 AM

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.

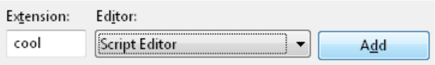


Extension:

Editor: Microsoft Visual Basic

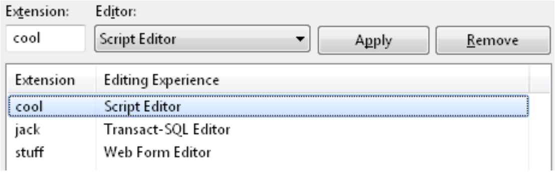
□ Map extensionless files to: Microsoft Visual Basic

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:


Extension: cool

Editor: Script Editor

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



Extension: cool

Editor: Script Editor

Extension	Editing Experience
cool	Script Editor
jack	Transact-SQL Editor
stuff	Web Form Editor

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