



Quick answers to common problems

Visual Studio 2013 Cookbook

Over 50 simple but incredibly effective recipes to get you up and running with the powerful features of Visual Studio 2013

Jeff Martin

Richard Banks

[PACKT] enterprise 
PUBLISHING professional expertise distilled

Visual Studio 2013 Cookbook

Over 50 simple but incredibly effective recipes to get you up and running with the powerful features of Visual Studio 2013

Jeff Martin

Richard Banks

[PACKT] enterprise 
PUBLISHING professional expertise distilled

BIRMINGHAM - MUMBAI

Visual Studio 2013 Cookbook

Copyright © 2014 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either expressed or implied. Neither the authors, nor Packt Publishing, and its dealers and distributors will be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

First published: September 2012

Second Edition: March 2014

Production Reference: 1100314

Published by Packt Publishing Ltd.
Livery Place
35 Livery Street
Birmingham B3 2PB, UK.

ISBN 978-1-78217-196-6

www.packtpub.com

Cover Image by Dan Martin (www.danfm.net)

Credits

Authors

Jeff Martin
Richard Banks

Copy Editors

Janbal Dharmaraj
Alfida Paiva

Reviewers

Hulot
Darren Kopp
Anand Narayanaswamy
Sergiy Suchok
David Thibault
Ken Tucker

Project Coordinator

Swati Kumari

Proofreaders

Simran Bhogal
Mario Cecere
Joel T. Johnson

Acquisition Editor

Rebecca Youé

Indexers

Hemangini Bari
Mariammal Chettiyar

Content Development Editor

Ankita Shashi

Production Coordinator

Shantanu Zagade

Technical Editors

Kunal Anil Gaikwad
Pramod Kumavat
Siddhi Rane

Cover Work

Shantanu Zagade

About the Authors

Jeff Martin is a long-time technology enthusiast and has been a contributing writer for InfoQ (www.infoq.com) for over two years, focusing on .NET and Microsoft-based platforms. Having experience in writing, testing, and designing software, he enjoys learning about new technologies and explaining them to a broader audience.

Readers are encouraged to follow @VSDevTips on Twitter to receive updates on the book as well as news for getting the most out of Visual Studio.

I would like to thank my lovely wife Carolyn for her love and understanding during my frequent late-night writing sessions. Your support for these projects means more to me than I can express.

I would also like to thank my brother Dan Martin for providing a great photo for the cover of this book.

Finally, thank you to my entire production team at Packt Publishing for their effort and assistance in producing this book.

Richard Banks has developed software for the last 20 years for a wide range of industries and development platforms. Over the years he has played many roles including that of a developer, team lead, project manager, and CIO/CTO. He is a Professional Scrum Trainer, runs the Sydney ALT.NET user group and the Talking Shop Down Under podcast. He owns and contributes to a few open source projects, and has spoken at Microsoft TechEd and a number of other events and user groups around Australia. For some strange reason, he gets a real kick out of helping development teams to improve and produce great software. If you want to get in touch, he tweets at @rbanks54 and blogs at <http://www.richard-banks.org/>. He currently works as a Principal Consultant for Readify and is a Microsoft Visual Studio ALM MVP.

It might have my name on the front cover but a book is never the work of just one person.

I would firstly like to thank my fantastic wife, Anne, and my two wonderful children, Hannah and Leisel, for giving me the time and space to work on this book. Their support throughout the process has been invaluable, and without that I would never have undertaken this book in the first place.

I'd also like to thank the staff and Packt Publishing for the opportunity and help in bringing this together, and my tech reviewers who gave up their spare time reading my scribble and checking if what I wrote actually made sense instead of being just a delirium-fuelled pile of nonsense.

Thank you all!

About the Reviewers

Hulot has been working in the IT industry for more than 20 years in different capabilities, from software development, project management to IT marketing product development and management. He has worked for multinational companies such as Royal Philips Electronics, PricewaterhouseCoopers, and Microsoft. Currently, He has been working as an independent IT consultant. He is a Computer Science lecturer at two Brazilian universities. He holds a Ph.D. in Computer Science and Electronics from the University of Southampton, UK and a B.Sc. in Physics from University of São Paulo, Brazil.

I would like to thank my wife Mylene Melly for her continuous support. I would also like to thank my many colleagues over the years who have made it possible to learn what I now know about software development and the computer industry.

Darren Kopp is a father, husband, software engineer, and gamer. Darren started programming when making a website for his clan in the game Tribes using ASP and then moved on to ASP.NET when .NET 1.1 was released.

Darren started professional development work in the golf industry, developing systems ranging from e-commerce solutions for golf shops to systems tracking swing profiles of golfers. He then moved on to the construction industry where he developed software that integrated payroll, human resources, service management, and project management.

Darren currently works for DevResults that provides web-based solutions to the international development and humanitarian community.

When Darren isn't coding or spending time with family, you can find him tweeting jokes and playing Team Fortress 2 and Battlefield 4.

Anand Narayanaswamy works as an editor with InfoQ based in Trivandrum, Kerala, India. He is an ASPInsider and was a Microsoft Most Valuable Professional (MVP) from 2002 to 2011.

Anand also worked as a chief technical editor for `ASPAlliance.com` and contributed several articles and reviews for `DevPro`, `c-sharpcorner.com`, `developer.com`, `codeguru.com`, and various other community sites.

Anand also worked as a technical editor for several popular publishers such as Sams, Addison-Wesley Professional, Wrox, Deitel, Packt Publishing, and Manning.

He blogs at `www.learnxpress.com` and can be reached at `www.facebook.com/anandn`. His twitter handle is `@visualanand`.

Sergiy Suchok graduated in 2004 with honors from the Faculty of Cybernetics, Taras Shevchenko National University of Kyiv (Ukraine) and has since then been keen on information technology. He is currently working in the banking sector and has a Ph.D. in Economics. Sergiy is the co-author of more than 45 articles and participated in more than 20 scientific and practical conferences devoted to the economic and mathematical modeling. He is a member of the *New Atlantis* Youth Public Organization (`newatlantida.org.ua`) and devotes his leisure time to environmental protection issues, historical and patriotic development, and popularization of a grateful attitude towards the Earth. He also writes poetry and short stories and makes macramé.

I would like to thank my wife and my young daughter for their patience and understanding while I reviewed this book.

David Thibault has been a .NET developer since Version 1.1. He has worked on a wide variety of software solutions in various domains such as e-commerce, retailing, social networking, and business management. Besides .NET, he has recently acquired an interest for other technologies such as NodeJS and AngularJS. David currently works for Sigmund, a young agency focusing on bringing businesses to the digital age. He lives in Quebec City, Canada, and he loves Scotch whisky.

Ken Tucker is a Microsoft MVP who enjoys working on Windows Phone and Windows Store apps.

www.PacktPub.com

Support files, eBooks, discount offers, and more

You might want to visit www.PacktPub.com for support files and downloads related to your book.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at www.PacktPub.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at service@packtpub.com for more details.

At www.PacktPub.com, you can also read a collection of free technical articles, sign up for a range of free newsletters, and receive exclusive discounts and offers on Packt books and eBooks.



<http://PacktLib.PacktPub.com>

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can access, read, and search across Packt's entire library of books.

Why Subscribe?

- ▶ Fully searchable across every book published by Packt
- ▶ Copy and paste, print, and bookmark content
- ▶ On demand and accessible via web browser

Free Access for Packt account holders

If you have an account with Packt at www.PacktPub.com, you can use this to access PacktLib today and view nine entirely free books. Simply use your login credentials for immediate access.

Instant Updates on New Packt Books

Get notified! Find out when new books are published by following [@PacktEnterprise](https://twitter.com/PacktEnterprise) on Twitter, or the *Packt Enterprise* Facebook page.

Table of Contents

Preface	1
Chapter 1: Discovering Visual Studio 2013	9
Introduction	9
Synchronizing settings	10
Touring the VS2013 IDE	12
Project round-tripping	19
Managing the editor windows	23
Finding Visual Studio commands	27
Searching and navigating	29
Navigating in depth	32
Chapter 2: Getting Started with Windows Store Applications	37
Introduction	37
Getting a Windows Store app developer license	40
Creating a Windows Store app	44
Adding a Windows Store item template to your app	52
Using the Windows Store app simulator	59
Defining capabilities and contracts	66
Analyzing your App's performance	74
Packaging your Windows Store app	76
Validating your Windows Store app	81
Chapter 3: Web Development – ASP.NET, HTML5, CSS, and JavaScript	85
Introduction	85
Getting started with Bootstrap	86
Previewing changes across multiple browsers	91
Creating HTML5 web pages	94

Taking advantage of the CSS editor improvements	97
Understanding the JavaScript editor improvements	102
Adding bundling and minification to JavaScript and CSS files	107
Verifying pages with the Page Inspector tool	111
Chapter 4: .NET Framework 4.5.1 Development	117
Introduction	117
Adding the Ribbon to a WPF application	118
Creating a state machine in Visual Studio	123
Creating a task-based WCF service	130
Managing packages with NuGet	134
Unit testing .NET applications	139
Sharing class libraries across runtimes	144
Detecting duplicate code	147
Chapter 5: Debugging Your .NET Application	153
Introduction	153
Maximizing everyday debugging	154
Debugging on remote machines and tablets	159
Debugging code in production with IntelliTrace	166
Debugging parallel code	172
Visualizing concurrency	177
Chapter 6: Asynchrony in .NET	183
Introduction	183
Making your code asynchronous	184
Understanding asynchrony and the Windows Runtime	191
Using asynchrony with web applications	196
Working with actors and the TPL Dataflow Library	200
Chapter 7: Unwrapping C++ Development	205
Introduction	205
Using XAML with C++	206
Unit testing C++ applications	210
Analyzing your C++ code	217
Using a custom rule set	219
Working with DirectX in Visual Studio 2013	223
Creating a shader using DGSL	227
Creating and displaying a 3D model	232
Using the Visual Studio Graphics Diagnostics	235

Chapter 8: Working with Team Foundation Server 2013	241
Introduction	241
Creating a new TFS project	242
Managing your work	246
Using local workspaces for source control	253
Performing code reviews	260
Getting feedback from your users	266
Using Git for source control	271
Chapter 9: Languages	281
Introduction	281
Fortifying JavaScript applications with TypeScript	281
Integrating Python into Visual Studio	285
Integrating Python with .NET	289
Appendix: Visual Studio Medley	293
Introduction	293
Creating installer packages	293
Submitting apps to the Windows Store	298
Creating Visual Studio add-ins and extensions	301
Creating your own snippets	303
Index	309

Preface

The *Visual Studio 2013 Cookbook* has been written to provide you with an informative tour around **Visual Studio 2013 (VS2013)**. Topics have been broken down into quick-to-access segments called recipes, allowing you to easily find the material that interests you.

Some recipes will directly apply to your regular tasks while others are intended to shed some light on overlooked corners of Visual Studio. Given its long pedigree, extensive functionality has been added over the years, which is easy to overlook. The goal is that by reading this book, you will become more proficient with the tool that you use most as a Windows developer.

This second edition has been revised and expanded to cover the new features of VS2013 so that regardless of whether you are upgrading from VS2012 or are making a bigger step from an earlier version, there will be helpful tips and discussion of this new version. For those of you who purchased the first edition, welcome back, and to our new readers thank you for joining us. Let's get started!

While you were gone

Since the launch of VS2012, Microsoft has released four updates referred to as 2012.1, 2012.2, and so on. Each have added functionality to the original program, and each installed package is cumulative, meaning that only the newest one has to be installed to provide the benefits of all previous updates.

VS2012 Update 1

First and foremost is the ability VS2012.1 provides for C++ developers to target Windows XP clients while using the VS2012 compiler. This is a key ability if you desire to continue compiling applications that run on Windows XP and Vista as well as Windows 7 and 8. JavaScript programmers will appreciate the inclusion of memory profiling and memory analysis tools. SharePoint developers will gain load testing support and the use of SharePoint emulators. Full details are available at <http://support.microsoft.com/kb/2797915>.

VS2012 Update 2

This update includes ASP.NET and Web Tools 2012.2, unit testing support for Windows Phone, and broader unit testing capabilities for Windows Store apps. 2012.2 includes a host of bug fixes and stability improvements that benefit all areas of the program, right from the debugger to memory leaks in C++ based solutions, and several use cases that would cause the IDE to crash. Full details are available at <http://support.microsoft.com/kb/2797912>.

VS2012 Update 3

2012.3 primarily provides bug patches and performance improvements. Notable for those using VS2012 and VS2013 concurrently is 2012.3's improved compatibility with VS2013 projects. Several stability fixes were made that among other things prevent crashes in web projects, fix a conflict between VS2012 and VS2010, and correct slowdowns when developing mixed-mode C++ applications. Installing Update 3 provides the ability to install the Microsoft Visual Studio Tools for Git, a popular open source version control system. Full details are available at <http://support.microsoft.com/kb/2835600>.

VS2012 Update 4

Similar to 2012.3, this update is focused in bug fixes and stability enhancements. As this release coincides with the production release of VS2013, it provides some improved compatibility for sharing projects/solutions between VS2012 and VS2013. Full details are available at <http://support.microsoft.com/kb/2872520>.



At the time of this writing, there is no reason to avoid installing the latest update available for VS2012 whether using an Express edition or one of the premium editions. Several meaningful improvements have been made to the product since its original release date that benefit all users. Microsoft provides both a web-based installer and a complete standalone ISO at <http://go.microsoft.com/fwlink/?LinkId=301713>.

Choosing a version of Visual Studio 2013

Visual Studio comes in several different versions, each with different capabilities and target audiences. In all cases, the minimum supported operating system is Microsoft Windows 7 SP1 on the desktop or Windows Server 2008 R2 SP1 on the server. The two biggest differences of the Visual Studio line-up are between the Express and non-Express (premium) versions. Express editions are offered by Microsoft free of charge, and target a specific type of application development:

- ▶ **Visual Studio Express 2013 for Web:** As the name implies, this is appropriate for those seeking to build web-based applications using technologies that include HTML5, CSS, and JavaScript.

- ▶ **Visual Studio Express 2013 for Windows:** Probably better titled "...for Windows Store apps", this edition targets Windows Store apps exclusively. HTML5/JavaScript, C#, C++, and Visual Basic are all acceptable choices for app development. An emulator is bundled for testing apps across various devices.
- ▶ **Visual Studio Express 2013 for Windows Desktop:** Traditional Windows desktop applications can be created with this edition using C#, C++, or Visual Basic.



Multiple Express editions can be installed side by side, so you feel free to install any/all of the preceding editions as needed for your work.

The non-Express editions run from Visual Studio Professional 2013 to Visual Studio Ultimate 2013. Ultimate has the entire available functionality, whereas Professional is limited to **Peek Definition**. One advantage all non-Express editions share is that all possible development types (web, apps, and desktop) are available in a single installation. The non-Express editions are as follows:

- ▶ **Visual Studio Professional 2013:** This supports development of all application types and includes the Peek Definition feature.
- ▶ **Visual Studio Premium 2013:** In addition to the preceding edition, notable features include project management functionality, coded UI testing for XAML Windows Store 8.1 apps, and C++ Profile Guided Optimization.
- ▶ **Visual Studio Ultimate 2013:** In addition to the two preceding editions, notable features include CodeLens, IntelliTrace, Memory Dump Analyzer, and cloud-based load testing.
- ▶ **Visual Studio Test Professional 2013:** As its name implies, this edition primarily focuses on testing.

Visual Studio 2013 will make certain options available to you based on the underlying version of Windows that you are running. If you are running Windows 8.1, you will only be able to create new Windows Store applications for Windows 8.1. You will be able to open existing Windows 8 app solutions in VS2013, but not create new ones. You are able to create new Windows 8 apps only with VS2012, but this should be considered a special case given the arrival of Windows 8.1. Windows Store applications cannot be developed with any version of Visual Studio on Windows 7, Windows Server (any version), or previous versions of Windows.



Picking a version

If you are an independent developer, you will most likely want to download the Express edition(s) that apply to the type of program that you are developing. Express editions can be installed side by side, so there is no need to limit yourself to just one. Among the paid versions, choose which one meets the requirements of your work. If you are not price-limited, picking the Ultimate version is the simplest route. A full comparison of the various versions is available at <http://www.visualstudio.com/products/compare-visual-studio-products-vs>.

Visual Studio 2013 Update 1 has been released and primarily consists of stability patches and bug fixes. It should be applied to your system regardless of the edition of VS2013 that you have selected. More information about Update 1 is available at <http://support.microsoft.com/kb/2911573>. As this book goes to print, a preview of Update 2 has been announced. This release will be delivering new features as well bug fixes. Follow this book's Twitter feed for more news as this becomes available (@VSDevTips) or <https://twitter.com/vsdevtips>.

What this book covers

Chapter 1, Discovering Visual Studio 2013, starts us off by taking a tour of the new features found in the editor itself and covers all VS2013's key refinements, from logging in to project navigation.

Chapter 2, Getting Started with Windows Store Applications, examines the development process for Windows Store apps for Windows 8.1. The full process of obtaining a developer license to building, testing, and publishing an app is covered.

Chapter 3, Web Development – ASP.NET, HTML5, CSS, and JavaScript, covers several areas of web development and how VS2013 can assist you. Here, the multi-browser preview is covered as well as editor enhancements that can benefit HTML5, CSS, and JavaScript programmers.

Chapter 4, .NET Framework 4.5.1 Development, focuses on developing applications that run on .NET. Desktop application development is still a key market and shows ways VS2013 can help.

Chapter 5, Debugging Your .NET Application, profiles the various ways to debug your .NET-based code. Code isn't always available on your development environment and this profile shows ways to deal with separate machines, whether they are tables or in production.

Chapter 6, Asynchrony in .NET, deals with the use of asynchronous code to provide more responsive output and how it may benefit your applications.

Chapter 7, Unwrapping C++ Development, tackles the elder statesman of languages severed by VS2013. Several recipes are provided, which will benefit your C++ usage; some of the areas covered include unit testing, XAML, and DirectX.

Chapter 8, Working with Team Foundation Server 2013, describes how Team Foundation Server can benefit your productivity. Whether you are an independent developer or part of a large corporate effort, you'll see how modern source control can help.

Chapter 9, Languages, takes a moment to look at some useful languages that are new to Visual Studio: TypeScript and Python. Python has a long and successful history, and now it is a first-class citizen on Visual Studio. We'll take a look at how Python can help .NET developers.

Appendix, Visual Studio Medley, the assortment of preceding topics doesn't cover everything that VS2013 can do. We'll cover some ways to extend Visual Studio's abilities and how to get your app ready for consumption by end users.

What you need for this book

To follow the recipes in this book, you will need a copy of Visual Studio 2013. Some of the features covered in the recipes may only be available in specific editions of Visual Studio. Whenever possible, any specific version requirements will be noted.

If you wish to follow one of these recipes and you do not have the right edition, trial versions of the premium versions can be downloaded from the Microsoft website enabling you to see if a particular feature would benefit your project.

For any of the recipes that deal with Windows Store applications, you will need to be using Windows 8.1 as your operating system.

Who this book is for

If you already know your way around previous versions of Visual Studio, if you are familiar with Microsoft development, and if you're looking to quickly get up to speed with the latest improvements in the 2013 incarnation of Microsoft's number one development tool, then this book is for you.

If you are an experienced developer who has used Eclipse or Xcode, you should also be able to find this book useful for exploring the differences between your tools and the latest that Microsoft has to offer.

Conventions

In this book, you will find a number of styles of text that distinguish between different kinds of information. Here are some examples of these styles, and an explanation of their meaning.

Code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles are shown as follows: "Open the VS2012_Web solution and run the application."

A block of code is set as follows:

```
<appSettings>
  <add key="vs:enableBrowserLink" value="true"/>
</appSettings>
```

When we wish to draw your attention to a particular part of a code block, the relevant lines or items are set in bold:

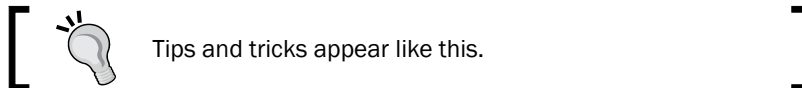
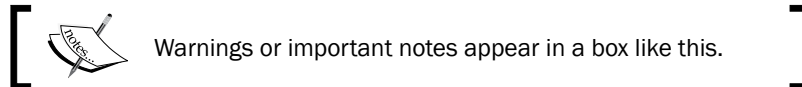
```
<system.webServer>
  <modules runAllManagedModulesForAllRequests="true" />
</system.webServer>
```

Any command-line input or output is written as follows:

```
Get-Command *intelli*
```

New terms and **important words** are shown in bold. Words that you see on the screen, in menus or dialog boxes for example, appear in the text like this: "The **Preview Selected Items** button is a toggle button."

Keyboard shortcuts such as *Ctrl* or *F4* are formatted as shown.



Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book—what you liked or may have disliked. Reader feedback is important for us to develop titles that you really get the most out of.

To send us general feedback, simply send an e-mail to feedback@packtpub.com, and mention the book title through the subject of your message.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide on www.packtpub.com/authors.

Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

Downloading the example code

You can download the example code files for all Packt books you have purchased from your account at <http://www.packtpub.com>. If you purchased this book elsewhere, you can visit <http://www.packtpub.com/support> and register to have the files e-mailed directly to you.

Downloading the color images of this book

We also provide you a PDF file that has color images of the screenshots/diagrams used in this book. The color images will help you better understand the changes in the output. You can download this file from https://www.packtpub.com/sites/default/files/downloads/1966EN_ColoredImages.pdf.

Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books—maybe a mistake in the text or the code—we would be grateful if you would report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting <http://www.packtpub.com/support>, selecting your book, clicking on the errata submission form link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded to our website, or added to any list of existing errata, under the **Errata** section of that title.

Piracy

Piracy of copyright material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works, in any form, on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at copyright@packtpub.com with a link to the suspected pirated material.

We appreciate your help in protecting our authors, and our ability to bring you valuable content.

Questions

You can contact us at questions@packtpub.com if you are having a problem with any aspect of the book, and we will do our best to address it.

1

Discovering Visual Studio 2013

In this chapter, we will cover:

- ▶ Synchronizing settings
- ▶ Touring the VS2013 IDE
- ▶ Project round-tripping
- ▶ Managing the editor windows
- ▶ Finding Visual Studio commands
- ▶ Searching and navigating
- ▶ Navigating in depth

Introduction

The arrival of **Visual Studio 2013 (VS2013)** marks the continuation of Microsoft's desire to produce more frequent updates to the Visual Studio platform. If you are coming to VS2013 from VS2012, you will find a more familiar appearance. Users upgrading to VS2013 from VS2010 will find greater differences and need to spend a bit more time learning the new interface. This chapter will provide a walk-through of the major changes and explain how VS2013's IDE will benefit you, regardless of the languages you are programming with.

VS2013 has focused a great deal on improving the usability and the power of the editor and surrounding windows that you use every day in your work. Before exploring specific features that apply only to certain areas, this chapter examines how the IDE can benefit your work regardless of whether you write C++ console applications or use JavaScript to write Windows Store apps.

The goal of this chapter is to provide you with an overview of the IDE-related features of VS2013. The chapter will begin by describing the integration of Microsoft accounts with the IDE and the value that this can provide. Next comes a review of the IDE, with a focus on the new features found in VS2013. Realizing that in the real world, circumstances can require supporting pre-VS2013 projects, we'll look at how Visual Studio utilizes project round-tripping to enable working with these older project types. The chapter will then wrap up with a couple of recipes on day-to-day editing and navigation tips designed to increase your productivity. Like any set of complex tools, an initial investment in learning the nuances can provide lasting dividends.

Synchronizing settings

One of the first things that you will notice upon opening VS2013 is the request to log in with a Microsoft account. You can use any existing Microsoft account that you have, including Outlook/Hotmail, OneDrive (formerly known as SkyDrive), and Xbox Live. If you have an MSDN account, Microsoft recommends that you use it to log in to Visual Studio. If you don't, or if you would prefer to use a new account, you can create one at <https://login.live.com/>. Microsoft groups the settings by product type, so there is one set of roaming settings linked to all editions of Visual Studio Professional, Premium, and Ultimate. A separate set of settings is synched across the Express editions of Visual Studio. The result is that the settings saved for VS Express 2013 for Windows will not sync when you log in to VS2013 Professional.

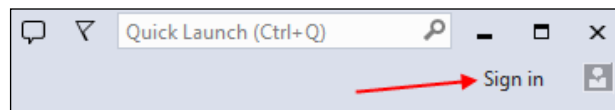
In this recipe, we will look at how this synchronization works and what it will coordinate on your behalf.

Getting ready

To explore, launch your copy of Visual Studio 2013.

How to do it...

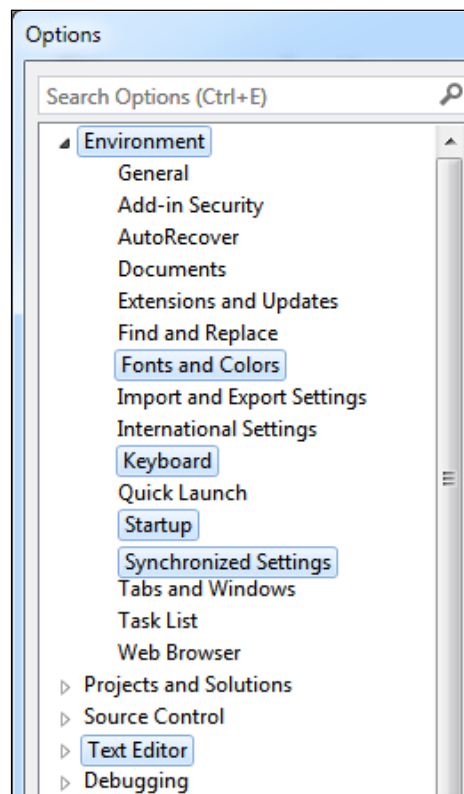
If you are not prompted to sign in at startup, you can always sign in from within Visual Studio. The arrow in the following screenshot indicates where the **Sign in** option is located:



Currently, VS2013 will sync options from the following categories:

- ▶ **Environment** : This section consists of several subitems:
 - ❑ **Fonts and Colors**: This includes preferences for the text used throughout the editor
 - ❑ **Color theme of IDE**: This provides built-in themes including light/dark/blue
 - ❑ **Keyboard**: This includes user-defined keyboard shortcuts and the selected keyboard-mapping scheme
 - ❑ **Startup**: This indicates what should display when VS2013 opens
- ▶ **Text Editor**: A multitude of settings including tabs versus spaces, word wrap, scroll bar placement, and so on
- ▶ **Environment Aliases**: (Not shown, applies to premium versions only) Commands defined in the command window (*Ctrl + Alt + A*)

The following screenshot highlights the synchronized categories:





The **Options** dialog box in VS2013 has a couple of usability enhancements. First, it has its own integrated search box (once **Options** is opened, the hotkey is *Ctrl + E*) for specifically searching within the **Options** dialog box. Second, the dialog is now resizable, making it much more useful for viewing settings that have lengthy configuration options.

How it works...

Microsoft stores a copy of your settings on their servers. A constant Internet connection is not required to use Visual Studio, but your settings will not synchronize until you are reconnected. If a connection is not available, you cannot login until Internet access is restored, but Visual Studio will still be usable. Taking it one step further, VS2013 remembers if you were logged in the last time when you closed the program, so your last known settings will be available as they are stored locally.

There's more...

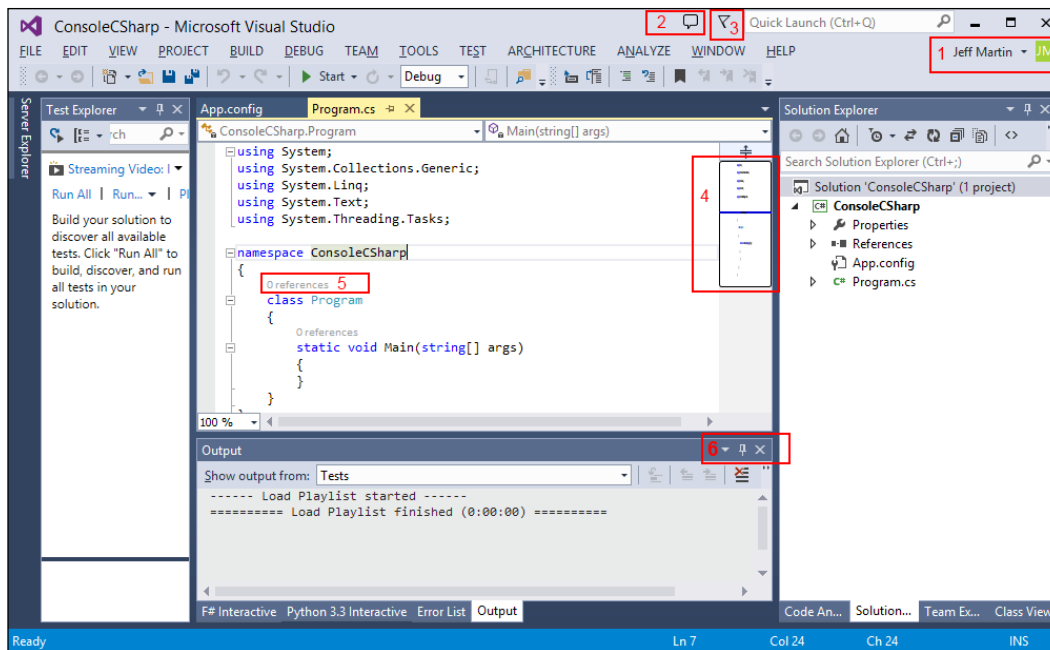
All synchronization options are configurable by navigating to **Options | Environment | Synchronized Settings**, and you may opt to have any combination of the preceding categories synced. This configuration of settings is specific to each machine and does not transfer. By default, VS2013 will attempt to synchronize all settings if you are logged in with a Microsoft account.

Touring the VS2013 IDE

The user interface in VS2013's IDE has several differences from VS2012 and the previous versions of Visual Studio. Let's take a look at what is available in this recipe.

Getting ready

All you will need for this recipe is a copy of VS2013 so that you can follow along with where different options are located. The following screenshot provides an overview of what will be covered:



The legend for the highlighted items in the preceding screenshot is as follows:

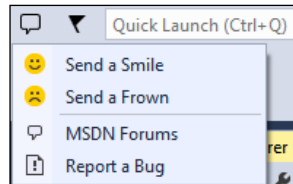
- ▶ The user's account entry/selection is indicated with **1**
- ▶ The notification center is indicated with **2**
- ▶ The **Feedback** menu is indicated with **3**
- ▶ The scrollbar thumbnail is indicated with **4**
- ▶ CodeLens (Ultimate only) is indicated with **5**
- ▶ The editor window controls are indicated with **6**

How to do it...

Over the next few pages, we are going to take a firsthand look at the new areas of the Visual Studio IDE. You may follow along with your own project or use the sample project where indicated.

Feedback

This screenshot shows the choices available when the **Feedback** menu is accessed by clicking on the chat balloon (far left icon):



Indicated by a chat balloon, the feedback icon provides an immediate way to send feedback to Microsoft from within VS2013. When the chat balloon is clicked, a drop-down list appears, allowing you to select from either **Send a Smile** (indicating a positive commentary) or **Send a Frown** (indicating negative commentary). Functionally, there is no difference in the dialog box that appears, but the choice of a smile or frown allows you to provide context for your remarks so that there is no ambiguity in your message. Each option allows you to include an e-mail address so Microsoft has a way to respond, and the frown option has an additional checkbox to indicate whether or not your comments are describing a bug.

The menu also provides the ability to report a bug and access the MSDN forums within Visual Studio. Both options provide a quick way to accomplish these tasks so that you make a report or seek help and get back to coding with minimal distractions.

Notifications

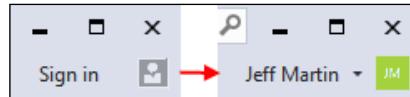
Between the **Feedback** icon and the **Quick Launch** field is the flag icon representing notifications:



In the preceding screenshot, you can see that one update is pending. The notification flag is designed to provide an unobtrusive alert when updates to VS2013 and the installed packages are available. Notifications listed in the sidebar are color-coded with yellow and red, which are used to indicate medium and high priorities respectively. Examples of notifications that could appear include notices about updates to Visual Studio, updates to installed extensions or samples, or notices alerting that the trial period is ending. Notifications may be dismissed en masse, and once they are dismissed they do not reappear.

User account

If you have signed in to VS2013 with a Microsoft account (see the *Synchronizing settings* recipe), this area displays the graphical avatar along with your account's display name. The following screenshot shows the difference in the display that occurs when you log in:

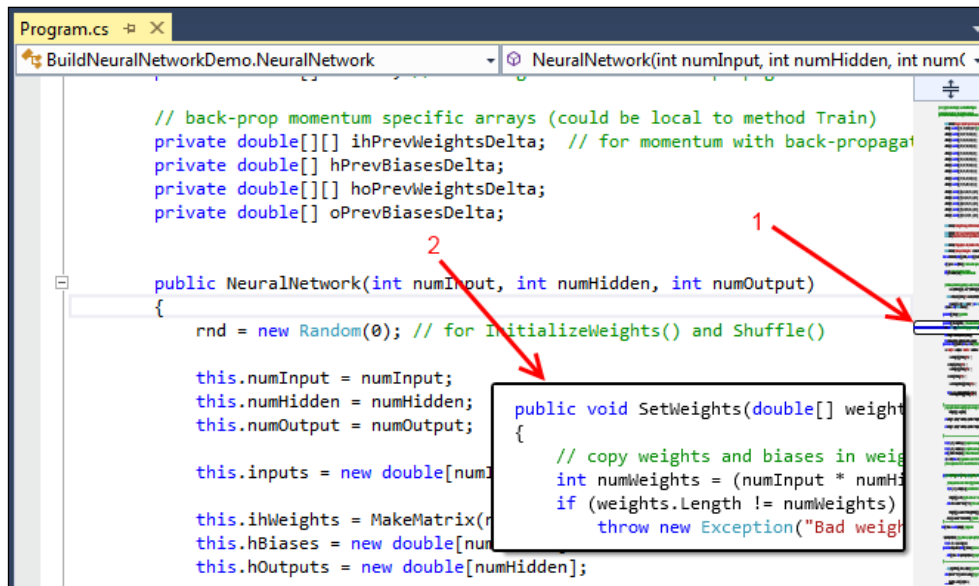


It also provides a way to sign in if you have not done so, or if you wish to change the active account being used. In the preceding screenshot, the left-hand side has the text **Sign in**, and a grey icon indicates that the user has yet to login. On the right-hand side, you can see the difference when the user has logged in, as the username has replaced the **Sign in** text, and the gray icon has changed to reflect the user's avatar (which in this case is a green icon overlaid with the white initials of the user's name).

Scroll bar thumbnail

The scroll bar has expanded its usefulness in VS2013. You can now customize it to show your overall position in a source file, and provide a tool tip that lets you examine code elsewhere in your current file without changing your current location. The new option is called **map mode**, as opposed to the historical behavior that is called **bar mode** (which is the traditional scrollbar appearance and behavior). All aspects of the map mode are customizable, including whether it appears at all, its width (narrow, medium, or wide), and the ability to show the preview tool tip.

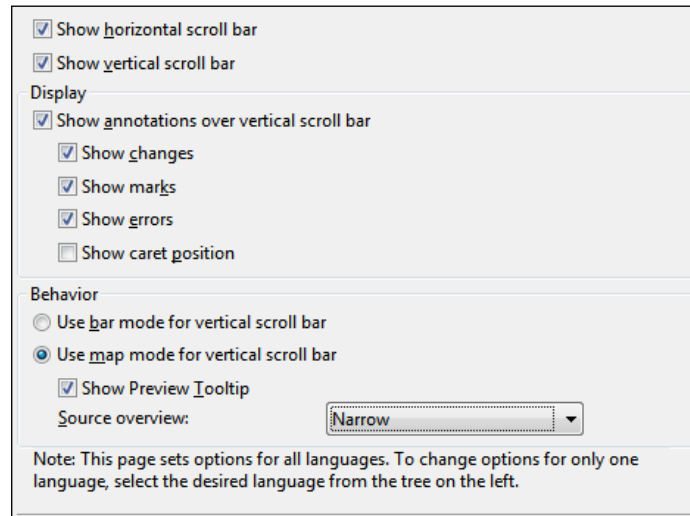
The following screenshot shows these features in action:



The preceding screenshot shows the scroll bar configured to be in map mode. The first arrow (marked as **1**) indicates that what is being displayed in the editor is located relative to the overall source file. The second arrow (marked as **2**) is pointing to the preview tool tip that appears when your mouse cursor hovers over the scroll bar.

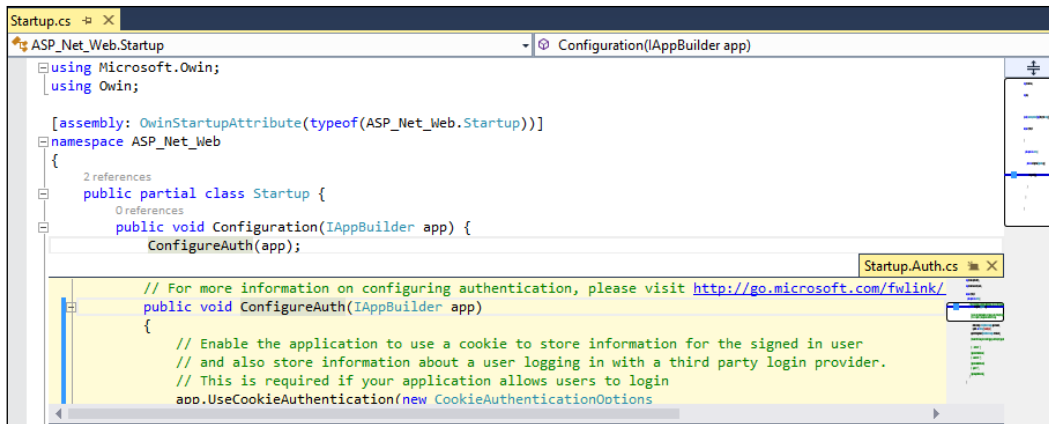
The scroll bar's pull down feature remains. When it's used to split the main window, it allows for two independent viewing panes of the same file, and each can have their own independent vertical scroll bar. In the split view mode, both vertical scroll bars share the same options (width, bar mode versus map mode, and so on).

The following screenshot lists all of the options available for configuring the scroll bar's functionality. It can be accessed directly by right-clicking on the vertical scroll bar and selecting **Scroll Bar Options....** Alternatively, it is accessible in VS2013's main **Options** dialog box by navigating to **Tools | Options**, with the scroll bar settings listed under **Text Editor | All Languages | Scroll Bars**. For additional customizations, you may set the language (C++, C#, and so on) settings for the scroll bar in the **Options** dialog box if desired:



Peek Definition

Visual Studio has had the **Go To Definition** (*F12*) option for several versions now. When we right-click on a method, or move the cursor to it, selecting the **Go To Definition** option will automatically bring you directly to the file with the corresponding definition. While you can easily navigate back and forth with (*Ctrl + -*) and (*Ctrl + Shift + -*), sometimes changing the open file is not what you would like to do. Enter **Peek Definition** (*Alt + F12*). This allows you to select a method and look it up, but instead of switching to the appropriate file, VS2013 will create a mini-window within your current editor. The following screenshot shows the results of using the **Peek Definition** option on the `ConfigureAuth()` method:



Sequential peeks can be performed, each opening via a tabbed interface. Navigation between these tabs (represented by circles) can be done via the mouse or keyboard: peek forward (*Ctrl* + *Alt* + *=*) and peek backward (*Ctrl* + *Alt* + *-*). The **Peek Definition** window uses the same scrollbar behavior as that of the parent editing window, and this behavior can be changed in either window by right-clicking on the scrollbar and selecting **Scrollbar Options...**

CodeLens (Visual Studio Ultimate Only)

CodeLens, also known as code information indicators, is a feature specific to VS2013 Ultimate. Activated by default, CodeLens provides real-time meta-information about the file you open in your main editor window:

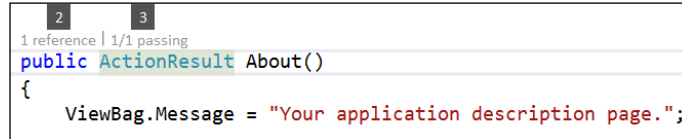
```

6 references
public class HomeController : Controller
{
    1 reference | 1/1 passing
    public ActionResult Index()
    {
        return View();
    }
}

```

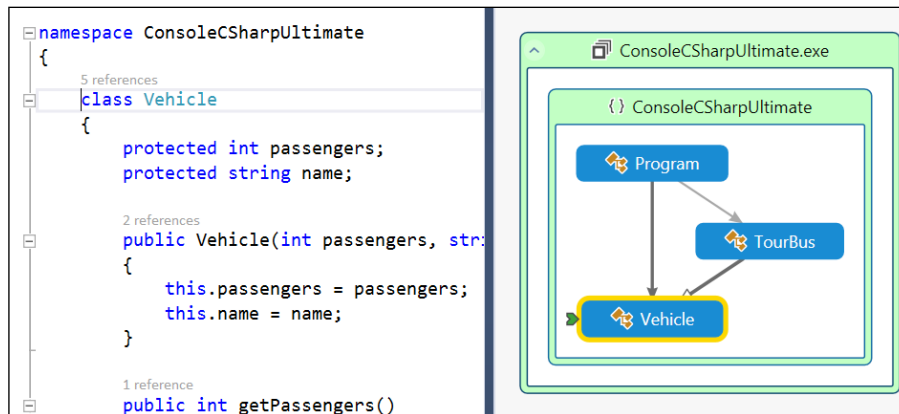
As illustrated in the preceding screenshot, shown inline with your code will be light colored term **references**, and the results of executed tests for each method/function as marked by the **passing** term. The **references** term indicates the number of places a method is used, and can also display a pop-up window on a mouse over that shows where it has been used. Likewise, the passing term relays the unit test results inline. While working with projects checked out from **Team Foundation Server (TFS)**, CodeLens will also display the most recent author of the method in question. Clicking on that name will pop up a details window listing the change history.

If you would prefer to keep your hands on the keyboard, holding down *Alt* will bring up hotkeys that can be used to select among the features discussed earlier. The following screenshot illustrates these options, with *Alt* + 2 opening a pop-up dialog box listing references for the selected method and *Alt* + 3 listing details about unit tests. The display options for CodeLens are labelled as **Code Information Indicators** in the **Options** dialog box and can be found under **Text Editor | All Languages | Code Information Indicators**:



Code Maps (Visual Studio Ultimate Only)

VS2013 Ultimate continues the use of a code visualization tool that Microsoft calls Code Maps to provide a representation of the open project. The following screenshot shows Code Maps in action:



Code Maps can be created and used in VS2013 Ultimate, but VS2013 Premium and VS2013 Professional can only consume them. However, users of Premium and Professional can interact with the maps and add comments/flags as they are reviewed. Code Maps can be activated in an editor window via *Ctrl* + ` , which is *Ctrl* plus the backquote key (typically found sharing a key with tilde). It can also be called by right-clicking in the editor window on a particular method or class that you want to map.

Once generated, the map may be manipulated in several ways, including zooming and the ability to expand to show related elements. Individual elements may be double-clicked so that they are brought up in the code editor for closer analysis. An element may also be right-clicked when in the **Code Map** for further navigation options (**Go To Definition**, **Show Base Types**, and so on) or to be commented on. This is particularly helpful for large or complex code bases where a map can assist in comprehension by visualizing the relationships and adding commentary external to the source code.

The Code Map indicates your position from the active editor window on the map with a green arrow icon. In the preceding example's screenshot, the editor is in the `Vehicle` class, which is pointed to on the Code Map by the green arrow.

See also

- ▶ The *Choosing a version of Visual Studio 2013* section in the *Preface*
- ▶ For an exhaustive list of differences between VS2012 and VS2013, refer to the MSDN article at <http://msdn.microsoft.com/en-us/library/bb386063.aspx>.

Project round-tripping

If you would like your .NET-based projects to be compatible with the previous versions of Visual Studio, be sure to choose a version of the .NET Framework that they support. For example, if you would like your project to support VS2010, be sure to target .NET 4.0, as .NET 4.5 is not compatible. Round-tripping is useful for situations where not all members of a development team have VS2013, as well as for situations when you wish to tread lightly on older projects.



As a friendly reminder, be sure that both versions of Visual Studio are up-to-date while sharing projects between VS2012 and VS2013. As noted in the *While you were gone* section of the *Preface*, this means Update 4 should be applied to VS2012.

In this recipe, we will discuss how projects can be shared across Visual Studio versions and a few details about how projects are handled.

Getting ready

The default .NET Framework in VS2013 is 4.5, so it is very easy to create a project that cannot open in VS2010. If you are not using any features specific to 4.5+, it is merely a matter of changing the project's properties to target 4.0. Keep in mind that some features will not be supported outright, but will be gracefully ignored.

Round-tripping is useful for a number of reasons. While supporting legacy projects, it is usually preferred to alter the existing code as little as possible. The support for older projects means that you can use VS2013 to edit them without keeping a copy of VS2012 installed. It also provides a way for users of VS2013 to work with fellow developers who are yet to upgrade from VS2012.

How to do it...

The best practice is to test upgrading on a backup of your legacy project. This way, you have an easy way to return to the status quo in the event of a failure or complication. Most projects will simply open without any complaint, especially those from VS2012 and, to a lesser extent, VS2010. If Visual Studio doesn't object, it is simply a manner of opening your old project in VS2013 and getting to work. Upon making edits and checking your code (if necessary), fellow developers running VS2010/VS2012 will have no difficulty making their own contributions.

How it works...

Visual Studio uses solution files to store details about the projects and solutions you create. Ending with the `.sln` extension, these files help Visual Studio manage your project. For example, a solution created in VS2012 has the following header at the beginning of its SLN file:

```
Microsoft Visual Studio Solution File, Format Version 12.00
# Visual Studio 2012
```

By comparison, a solution created in VS2013 has this header as follows:

```
Microsoft Visual Studio Solution File, Format Version 12.00
# Visual Studio 2013
VisualStudioVersion = 12.0.20623.1 VSUPREVIEW
MinimumVisualStudioVersion = 10.0.40219.1
```



The initial blank line is intentional for both examples.

The third line (prefaced with #) indicates the "human-readable" version of Visual Studio used to create the project. VS2013 projects add two additional lines to this header as shown in the preceding code snippet. The fourth line shows `VisualStudioVersion`, which specifies the full build version of Visual Studio used, while the fifth line lists the value `MinimumVisualStudioVersion`, which indicates the minimum version of Visual Studio that can be used to open the project.

**Downloading the example code**

You can download the example code files for all Packt books you have purchased from your account at <http://www.packtpub.com>. If you purchased this book elsewhere, you can visit <http://www.packtpub.com/support> and register to have the files e-mailed directly to you.

There's more...

Not every project type supports round-tripping, as some require specific changes to be made in order to run under VS2013. Some notable project types and their exceptions are as follows.

Windows Store applications

Windows Store apps have some special requirements. Apps targeting Windows 8.1 require VS2013 and the underlying OS to be Windows 8.1. VS2013 can work with existing Windows 8 store apps if they were created by VS2012. As previously noted, Windows 8.1 will quickly replace Windows 8, so all new apps should target that platform.

The Model-View-Controller (MVC) framework

Visual Studio 2013 brings support for MVC 5 in addition to supporting MVC 4, while Visual Studio 2012 only supports MVC 3 and MVC 4. Visual Studio 2010 SP1 only supports MVC 2 and MVC 3. These limitations dictate whether or not your project will upgrade. However, there are tools and guidance on how to migrate your application to a newer version of MVC. Upgrading an MVC 2-based application to MVC 3 can be done with the standalone upgrade from CodePlex at <http://aspnet.codeplex.com/releases/view/59008>. Once this is completed, Microsoft provides guidance on upgrading the application from MVC 3 to MVC 4 at http://www.asp.net/whitepapers/mvc4-release-notes#_Toc303253806. Unfortunately, the upgrade from MVC 3 to MVC 4 is a manual process.

MSI setup (.vdproj)

This project type refers to Visual Studio Installer projects, which are not supported in VS2013.