CHAPTER 1

Visual Basic 2013 development opportunities and the Windows Store

After completing this chapter, you will be able to

- Describe the development opportunities provided by Microsoft Visual Basic 2013.
- Understand requirements for distributing applications in the Windows Store.

Are you ready to start working with Microsoft Visual Basic 2013? In this chapter, you’ll get an overview of the features and capabilities of the Microsoft Visual Studio 2013 development system and the different editions of Visual Studio that you can purchase or download for free. You’ll learn about emerging hardware and software platforms and their uses and the impressive range of applications that you can create for these platforms, including Windows Store apps for Windows 8.1; Windows desktop apps for Windows 7, Windows 8, and Windows 8.1; Windows Phone 8 apps; web apps; console apps; and much more.

You’ll also learn about the Windows Store, an exciting new distribution point for apps designed especially for Windows 8.1. You’ll review a checklist of planning tasks to consider before you begin building a Windows Store application, and you’ll learn the procedures for selling and distributing apps through the Windows Store. After you have a clear list of the Windows Store requirements and program features in mind, you’ll be ready to build your own programs, including Windows Store apps that you can distribute to millions of potential customers worldwide.

Before we begin, a word about terminology. This book has been designed and tested using the Windows 8.1 operating system. The Windows Store apps that you create will run under Windows 8.1 and will target the .NET Framework version 4.5.1. You will also learn to create Visual Basic programs using the Windows Forms and console app models, which run on what is now known as the “Windows desktop.” These types of apps will run under Windows 8.1, Windows 8, Windows 7, and earlier versions of Windows, provided that the Windows installation has the proper .NET Framework files installed.

Yet another type of application you will create in this book, using Visual Studio and a technology called ASP.NET, are Web Forms apps. These apps run in a web browser, such as Internet Explorer. Finally, you’ll create mobile phone apps during the course of this book, using Visual Studio and the Windows Phone SDK 8.0. These apps run on the Windows Phone 8 platform.
I’m going to assume that you have purchased this book because you want to learn how to program in Visual Basic. In fact, my underlying assumption is that you might already have some development experience—perhaps even with an earlier version of Visual Basic—and that you are ready to learn about the Visual Studio 2013 product in the context of the Windows Store, Windows Forms, Windows Phone, and Web Forms platforms. Enhancing your Visual Basic development skills is an excellent choice; there are over four million Visual Basic programmers in the world developing innovative solutions, and Microsoft’s newest operating system, Windows 8.1, presents many amazing opportunities for Visual Basic programmers.

“Visual Basic” essentially has two meanings in the software development marketplace. In a narrower engineering sense, Visual Basic is the name of a programming language with specific syntax rules and logical procedures that must be followed when a developer creates code for a compiler with the goal of making an executable program or application. However, Visual Basic is also used in a more comprehensive product-related sense to describe the collection of tools and techniques that developers use to build Windows-based applications with a particular software suite. In the past, developers could purchase a stand-alone version of Visual Basic, such as Microsoft Visual Basic .NET 2003 Professional Edition, but these days Visual Basic is sold only as a component within the Visual Studio software suite, which also includes Microsoft Visual C#, Microsoft Visual C++, and other development tools.

The Visual Studio 2013 development suite is distributed in several different product configurations, including Professional, Premium, and Ultimate, along with a subset of Visual Studio tools designed for test engineers, known as Visual Studio 2013 Test Professional. In addition to these retail products, you can experiment with the Visual Studio 2013 software by downloading free versions of the suite designed for specific development platforms. These limited-feature or “Express” versions of Visual Studio 2013 are called Express for Windows, Express for Windows Desktop, Express for Windows Phone, and Express for Web.

The full retail versions of Visual Studio 2013 have different prices and feature sets, with Ultimate being the most comprehensive (and expensive) development package. The Visual Studio website (http://www.microsoft.com/visualstudio) explains the differences among all of these versions. Typically, the full retail versions of Visual Studio are also available for a 30-day free trial period that can be extended to 90 days. These trial versions are more feature-rich than the Express products. In addition, the faculty, staff, and students of recognized academic institutions can download full editions of Visual Studio 2013 through the Microsoft DreamSpark program, and these free downloads don’t expire.

I wrote this book to highlight the features and development opportunities provided by Visual Studio 2013 Professional and Visual Studio 2013 Premium. If you are using Visual Studio Ultimate, you will also have what you need to complete the exercises in this book—and then some. The extra features included in Visual Studio Ultimate primarily relate to larger team development projects and enterprise-computing scenarios that go beyond the scope of this book.
You can also complete most of the exercises in this book if you install all four of the Express editions of Visual Studio 2013, and then switch among them as directed. (That is, you can complete most of the exercises in this book if you install Visual Studio 2013 Express for Windows, Visual Studio 2013 Express for Windows Desktop, Visual Studio 2013 Express for Web, and Visual Studio 2013 Express for Windows Phone.) I will let you know which Express product is necessary for each chapter and when the individual Express products have limitations that will restrict your ability to complete the exercises. Occasionally, the instructions in this book will apply only to the full retail editions of Visual Studio 2013, such as Chapter 10, “Creating console applications.”

Collectively, the chapters in this volume are designed to open up an exciting new world of technical and business opportunities to Visual Basic 2013 programmers. The book’s extensive collection of step-by-step exercises has a broad focus, and they are written for technical people who understand programming and are not simply hobbyists or absolute beginners. In short, the exercises in this book will give you a taste of real-world programming practices and experiences. If you have no prior knowledge of Visual Basic or Visual Studio, you might want to fill in some of the gaps with my comprehensive introduction to Visual Basic 2012 and Windows 8 development, Start Here! Learn Visual Basic 2012 (Microsoft Press, 2012). From time to time, I will refer to the exercises in that book to give you additional resources for your learning.

An impressive range of development opportunities and platforms

How has Visual Basic programming evolved over time, and what opportunities are available now to Visual Basic 2013 programmers? Before we start writing code, let’s briefly examine some of the recent trends in software development and Windows programming.

Microsoft released Visual Basic 1.0 in 1991. From its initial announcement at Windows World, the product impressed software developers because it innovatively combined an advanced Visual Basic language compiler with an Integrated Development Environment (IDE) that allowed programmers to build Windows applications by visually arranging controls on a Windows form and then customizing the controls with property settings and Visual Basic code. From these modest beginnings, Visual Basic grew into a powerful development tool that was closely aligned with Windows programming, capable of creating fast and efficient Windows-based applications that could run on a variety of hardware platforms.

In the early 2000s, Visual Basic programmers were concerned primarily with creating applications for Windows that helped businesses manage data effectively. Visual Basic’s ability to graphically display information and provide access to it with powerful user interface controls gained many supporters for the product, and the installed base grew into the millions. Over the past decade, the leading Visual Basic applications have been database front-ends, inventory management systems, web applications and utilities, purchasing tools, CAD programs, scientific applications, and games.

However, in the 2010s, the explosion of Internet connectivity and online commerce has dramatically changed the landscape for software developers. In the past, most applications for Windows ran on a server or a desktop PC. Today, laptops, tablet devices, and smartphones are everywhere, and
often the same person owns three or four device types. Consumers need to move applications and
information seamlessly across these devices, and software developers need the tools that will allow
them to create applications that work on multiple platforms or that can be ported easily from one
device to the next.

The Visual Studio 2013 product team took the challenge of coding for diverse platforms seriously,
and they have created a software suite that allows developers to leverage their existing work while
also letting them target a variety of different application models. The following list highlights the
major development platforms and opportunities for Visual Basic programmers (some of which are
supported only by the full retail versions of Visual Studio 2013):

■ **Windows 8.1** Visual Basic developers can create Windows Store apps for Windows 8.1 that
run on a wide range of devices, including desktop PCs, laptops, and Microsoft Surface tablets.
(Note: To create new Windows Store apps for Windows 8, you need to use Microsoft Visual
Studio 2012.)

■ **Windows 8, Windows 7, and Windows Server** Visual Basic developers can create desktop
applications for earlier versions of Windows and distribute them in a variety of ways. You can
create desktop applications using the Windows Forms (“Win forms”) model or the Windows
Presentation Foundation (WPF) model.

■ **Windows Phone 8** Using Visual Studio 2013, Visual Basic programmers can create appli-
cations that run on the Windows Phone 8 platform and take advantage of its unique fea-
tures. You will learn to write mobile phone apps for Windows Phone devices in Chapter
20, “Introduction to Windows Phone 8 development,” and Chapter 21, “Creating your first
Windows Phone 8 application.”

■ **Web development** Developers can use Visual Basic, HTML5, CSS3, or JavaScript to create appli-
cations that will run on the web and look great in a variety of browsers. A technology known as
ASP.NET allows Visual Basic programmers to build websites, web applications, and web services
quickly without knowing all the details about how the information will be stored on the web. The
full list of options is explored in Chapter 19, “Visual Studio web development with ASP.NET.”

■ **Console applications and device drivers** Visual Basic programmers can write applications
that run in command-line mode, which is sometimes called the Windows text console or DOS
window. While console apps primarily handle “behind the scenes” calculations, they can also
use libraries in the .NET Framework. I describe console programming in Chapter 10.

■ **Office applications** Visual Basic programmers can still build macros and other tools that
enhance the functionality of Microsoft Office applications, such as Excel, Word, Access, and
PowerPoint.

■ **Xbox 360** Visual Basic programmers can write games for the Xbox using Visual Studio and
Microsoft XNA Game Studio (version 4.0 and later).

■ **Windows Azure applications for web servers and the cloud** Visual Basic is powerful
evenough to write applications that will be used on sophisticated web servers, distributed data
centers, and a version of Windows designed for cloud computing known as Windows Azure.
This is an amazing list of application types! Although this list might seem daunting at first, the good news is that the fundamental Visual Basic programming skills that you will explore here remain the same from platform to platform, and there are numerous tools and techniques that help you to port work easily between them. This book provides a solid introduction to many of the core skills that you will use, and especially the new tools provided by Visual Studio 2013 to help you develop your solution for Windows 8.1, the Windows desktop, and Windows Phone 8. However, after you master the core Visual Basic programming skills, you can move on to specific platforms by acquiring materials specifically related to those markets.

**Taking a multiplatform approach to learning Visual Basic**

As you have probably discovered by now, applications for Windows 8.1 are often called Windows Store apps. Yes, the connection between Windows 8.1 and the Windows Store is *that* direct. However, Microsoft understands that not all developers are prepared to write applications *only* for Windows 8.1 because developers still need to support earlier versions of Windows, and many developers are designing apps for web browsers, which must be run on a variety of platforms. For this reason, I am describing Visual Basic programming techniques for a wide range of programming platforms in this book. You will learn how to create Windows Store apps, Windows desktop apps, console apps, Web apps, and Windows Phone apps.

In some cases, I will discuss Visual Basic programming techniques related to a specific platform in a chapter, such as Chapter 3, “Creating your first Windows Store application.” In other cases, I move back and forth between the platforms, showing how the Visual Basic language, or Visual Studio features related to different platforms, might be adapted to unique situations. An example of this approach is Chapter 14, “Using arrays, collections, and generics to manage data,” in which I provide data management instruction using examples from both the Windows Store and the Windows desktop (Windows Forms) platforms.

I have taken this comprehensive approach in *Microsoft Visual Basic 2013 Step by Step* because Visual Studio 2013 Professional has been designed to support all of these application types. The current reality is that Visual Basic programming is a multiplatform endeavor, and intermediate Visual Basic programmers need exposure to many environments as they expand and enhance their development skills. At the same time, Windows Store programming is quite new, so I spend a little more time exploring this platform than the others.
Evaluating the Windows Store

Because the Windows Store provides a new and potentially profitable way of selling and distributing apps to a wide audience, I want to begin this book with a description of what the Windows Store is and how you can use it to reach potential customers. In addition to providing a strong business incentive to developing Windows Store apps, I want you to become familiar with the technical requirements of the Windows Store before you begin this type of development so that you know what you will need to do before you get too far along in a big Windows Store project. Microsoft recommends this “up-front education” too, because teams that are creating apps for the Windows Store can be most productive when they know all the certification requirements in advance.

What is the Windows Store?

The Windows Store is an electronic marketplace that allows consumers to search for and acquire applications for Windows. The Windows Store is designed to distribute apps for Windows 8 and Windows 8.1, much like Apple’s Mac App Store allows consumers to download Mac software, and the Windows Phone Store allows consumers to download products for devices running Windows Phone 8.

Note The Windows Phone Store is described in detail in Chapter 20.

The Windows Store allows developers to reach a global marketplace in ways that have been difficult or impossible in the past. Through the Windows Store, Windows-based apps can be monetized, either by charging for an application or by including advertising in the application. Programs downloaded from the Windows Store are certified and ready to run; after you meet the requirements for preparing an app for the marketplace, the details about downloading and deploying the application are handled by the Store.

Throughout this book, you will learn how to create apps to run on Windows 8.1 by using Visual Basic and Visual Studio 2013. At this point, you just need to learn how products are bought and sold in the Windows Store, and to review a Windows Store checklist that identifies which features are necessary for certification and distribution to the global marketplace.
Accessing the Windows Store

If you are running Windows 8.1 on your computer, you will see a Windows Store tile on the Windows Start page, which is the gateway to accessing the Windows Store. If you are not currently running Windows 8.1, you can learn about the Windows Store at http://www.windowsstore.com/, but you won’t be able to access the Windows Store itself, because it is designed for use only within Windows 8.1.

The following illustration shows what the Windows Store looks like when you first access it. Because the list of featured products is always changing, your screen will look different.

![Windows Store Illustration]
If you right-click in the Windows Store, you’ll see a navigation pane that allows you to browse for the top paid and the top new Windows Store apps. In addition, you’ll see useful product categories, such as Games, Social, Entertainment, Photo, Music & Video, Sports, and Books & Reference. When you select a category and an item, you’ll see an app listing page similar to the following screen:

The app listing page is the place where software vendors get a chance to promote their products and describe app benefits. It is tremendously important to present your app in the best possible light here. The application name, description, feature list, age rating, price, and screen shots are all significant factors in making a good impression on your audience. As people purchase or download your app, the rating system (based on five possible stars for the highest level of customer satisfaction) is also an important factor in drawing people to your app.

Installing an app from the Windows Store is extremely simple; you just click the Install button, and within moments, the app will be deployed on your Start page and available for use. A reliable Internet connection is required to download the app and (often) to feed the app data as the program runs.

Sales information and price tiers
Windows-based apps can be distributed free via the Windows Store, or they can be sold for a price. A setting called a price tier sets the fee for the app that you plan to sell. You can set the price tier that you like; tiers start at 1.49 USD and move up in increments of 0.50 USD to 4.99 USD, with higher product prices available.
If you plan to sell apps via the Windows Store, it is important to understand a little about how that process might work, even before you begin development. For the first 25,000 USD of an app’s sales, you will receive 70% of the revenues that Microsoft receives for the product. If and when an application receives more than 25,000 USD in sales, you will receive 80% of the revenues over 25,000 USD. Keep in mind that your product will be sold internationally, and in some countries, the amount that Microsoft receives will be reduced to account for taxes required by local laws.

It is also required that you register to be a Windows Store developer before you can sell products through Microsoft’s new electronic marketplace. The initial annual cost for a developer account in the United States was 49 USD for an individual and 99 USD for a company. You will also need to complete some registration paperwork containing contact information and other details.

Or your application could be free...

Of course, it is not necessary that you sell your application. You can also offer it as a free download to users all over the world. This might be useful if you want to provide general information or a public service or if you want to draw attention to your company or make its products or services more usable. For example, you might want to create a Windows Store app that presents the menu and other services provided by a restaurant, or publish news highlights and photos from an information service.

Within these free applications, you could then decide to use online advertising tools to generate revenue, or you could simply distribute information and know that you had fostered communication about your product throughout the world. The Windows Store has a special marketing category for free apps, as shown in the following illustration:
Whether you sell or distribute your app for free is up to you and the needs of your business and your customers!

Planning ahead for certification
Before you begin serious development on your project, Microsoft recommends that you review the certification requirements carefully for Windows Store apps so that you aren’t surprised by the necessary steps. For the most part, these steps are simply good development practices that will make your programs robust and high quality. Microsoft is enforcing high standards so that customers come to trust the Windows Store and all of the software distributed through it. We all have a lot riding on the success of the Windows Store.

The Visual Studio Professional IDE contains a Store submenu on the Project menu, with eight commands pertaining to the Windows Store, as shown in the following illustration:

| Open Developer Account... |
| Reserve App Name... |
| Acquire Developer License... |
| Edit App Manifest |
| Associate App with the Store... |
| Capture Screenshot... |
| Create App Packages... |
| Upload App Packages... |

Before you begin serious development on a project that you intend to submit to the Windows Store, you should run the first three commands on the Windows Store submenu. The Open Developer Account command will get you signed up with Microsoft as an individual or a company. This enables the submission process and allows you to get more information. The Reserve App Name command lets you reserve a name for your application within the store. You want to do this before you get too far along (and then learn that you need to change the name). The Acquire Developer License command lets you get a temporary developer license, which you might have already done during your work in Visual Studio.

A helpful blog for developers preparing for the Windows Store is available at http://blogs.windows.com/windows/b/appbuilder/. Here you’ll find Microsoft employees and other industry experts explaining key application concepts and answering pertinent questions. For example, in addition to the Windows Store checklist shown in Table 1-1 in this chapter, you’ll need to fill out a complete package manifest for your project and practice other safe programming practices. You can also find useful information in the MSDN article “Take your app to market” at http://msdn.microsoft.com/en-us/library/windows/apps/br230836.aspx.

Windows Store requirements checklist
The formal certification process begins when you upload your app to the Windows Store. Table 1-1 contains a checklist recommended by Microsoft for developers who are creating apps for the Store. Most of these items are required for certification and will be evaluated when you register with Microsoft and fill out the required submission pages online. The certification requirements can be
updated periodically, but this checklist will help you get started. The point is that you need to do some preparation before you get online and submit your app for certification. You should have the necessary information ready, and be sure that it has been proofread carefully.

**TABLE 1-1 Windows Store submission checklist**

<table>
<thead>
<tr>
<th>Submission page</th>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>App Name</td>
<td>Provide a name for your app that is 256 characters or less. Pick a name that will capture your customers' attention. It is best to keep this name short.</td>
</tr>
<tr>
<td>Selling Details</td>
<td>Price Tier</td>
<td>Prepare to specify a selling price for your app (or set the price to “free”).</td>
</tr>
<tr>
<td></td>
<td>Free Trial Period</td>
<td>Allow your customer to download the app for a free trial period. If the customer does not buy it in the set period of time, it will stop working.</td>
</tr>
<tr>
<td></td>
<td>Countries/Regions</td>
<td>Identify the market for your product.</td>
</tr>
<tr>
<td></td>
<td>Release Date</td>
<td>Set the app’s release date.</td>
</tr>
<tr>
<td></td>
<td>Category</td>
<td>Assign a category for your app so that customers can find it in the Windows Store. There is a helpful list of predefined categories to choose from.</td>
</tr>
<tr>
<td></td>
<td>Accessible App</td>
<td>If your app has been designed to meet Microsoft’s accessibility guidelines, indicate that here.</td>
</tr>
<tr>
<td></td>
<td>Minimum DirectX Feature Level</td>
<td>Indicate the video and hardware requirements for your application.</td>
</tr>
<tr>
<td></td>
<td>Minimum System RAM</td>
<td>Indicate how much RAM your app requires. You might want to double-check the basic system requirements for the devices that your app will run on.</td>
</tr>
<tr>
<td>Advanced Features</td>
<td>In-App Offers</td>
<td>Provide information about products that users can purchase from within your app, including what the customer must pay and how long the purchased feature can be used.</td>
</tr>
<tr>
<td>Ratings</td>
<td>Age Rating</td>
<td>Specify an appropriate age rating for your app, using the levels provided.</td>
</tr>
<tr>
<td></td>
<td>Rating Certificates</td>
<td>If you are selling a game, you might need to provide a rating certificate from a ratings board, depending on where you plan to sell your app.</td>
</tr>
<tr>
<td>Cryptography</td>
<td>Question 1</td>
<td>Indicate whether your app makes use of cryptography or encryption.</td>
</tr>
<tr>
<td></td>
<td>Question 2</td>
<td>Verify that any use of cryptography is within the allowable limits imposed by the Bureau of Industry and Security in the United States Department of Commerce.</td>
</tr>
<tr>
<td>Packages</td>
<td>Package Upload Control</td>
<td>Provide the path to your app’s completed package.</td>
</tr>
<tr>
<td>Submission page</td>
<td>Field name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Description</td>
<td>Description</td>
<td>Provide clear and concise marketing copy that describes your application, its features, and its benefits. Review this information carefully before posting. It must be 10,000 characters or less.</td>
</tr>
<tr>
<td>App Features</td>
<td>(Optional) Provide up to 20 features of your app. (Each feature must be 200 characters or fewer.)</td>
<td></td>
</tr>
<tr>
<td>Keywords</td>
<td>(Optional) Provide up to seven concise keywords describing your app.</td>
<td></td>
</tr>
<tr>
<td>Description of Update</td>
<td>Provide a description of how this new version of your app updates the previous version. (Leave blank for the first release of your app.)</td>
<td></td>
</tr>
<tr>
<td>Copyright and Trademark Info</td>
<td>Provide a brief copyright notice, 200 characters or fewer.</td>
<td></td>
</tr>
<tr>
<td>Additional License Items</td>
<td>(Optional) Provide 10,000 characters or fewer.</td>
<td></td>
</tr>
<tr>
<td>Screenshots</td>
<td>Up to 8 quality screen shots of your app as it is running. Each can have a description of up to 200 words. The minimum size of the image must be 1366 x 766 pixels. You can capture these screens using the Store</td>
<td>Capture Screenshots command in Visual Studio.</td>
</tr>
<tr>
<td>Promotional Images</td>
<td>(Optional) Provide other promotional images for your app (up to four).</td>
<td></td>
</tr>
<tr>
<td>Recommended Hardware</td>
<td>(Optional) Provide up to 11 notes about the hardware requirements for your app.</td>
<td></td>
</tr>
<tr>
<td>App Website</td>
<td>Provide the website URL for your product.</td>
<td></td>
</tr>
<tr>
<td>Support Contact</td>
<td>Provide a contact URL for customers so that they can get support or ask additional questions. Prepare to be very responsive to customer questions and feedback.</td>
<td></td>
</tr>
<tr>
<td>Privacy Policy</td>
<td>Prepare an appropriate statement about your privacy policy regarding data collected about users.</td>
<td></td>
</tr>
<tr>
<td>In-App Offer Description</td>
<td>Provide information about products that users can purchase from within your app, including what the customer must pay and how long the purchased feature can be used. (This field was indicated above as well. Use the same information.)</td>
<td></td>
</tr>
<tr>
<td>Notes to Testers</td>
<td>Notes</td>
<td>Give the evaluators at Microsoft additional information about your app so that they can test its functionality. For example, describe hidden features or provide user name and password information if needed.</td>
</tr>
</tbody>
</table>
It’s all in the details

The value of the preceding checklist becomes apparent when you look again at the content for Windows Store apps within the Windows Store. The more you know about your customers and your product’s central features before you get started, the easier it will be to make design and layout decisions as you create your application. In the following screen illustration, notice how important the ratings, description, and features categories are for the featured app, as well as the value of the screen shot that visually describes the product.

The Details page (not shown, but accessible via the Details link) presents additional information, including release notes, supported processors, supported languages, and application permissions. The Reviews page (also not shown) contains comments from actual customers.

Now that you have reviewed the basic marketing and distribution mechanisms for apps in the Windows Store, it is time to get started building Visual Basic apps in Visual Studio. Although many of the apps that you will create in this book will be demonstration programs designed to teach discrete elements of the Visual Basic programming language, you should always keep an eye on the end-goal of your learning—creating software that other people can use.
Each chapter in this book concludes with a Summary section that offers a review of what the chapter has presented. You can use these sections to quickly recap what you have learned in each chapter before you move on to the one that follows.

This chapter has introduced development opportunities for Visual Basic programmers, including the many opportunities available to users of Visual Studio 2013. You’ve learned about the application types that you can create with Visual Studio 2013 and about the specific tools and platforms that are described by this book. You’ve also learned about the Windows Store, an incredible distribution point and marketing opportunity for software developers who want to sell or freely distribute their products. You’ve learned how the Windows Store operates and about some of the requirements you’ll need to satisfy to distribute apps for Windows 8.1 via the Windows Store. Although the process will require some up-front planning, as well as technical and marketing expertise, the upside is significant. The Windows Store has the potential to reach millions of customers worldwide.

In Chapter 2, “The Visual Studio Integrated Development Environment,” you’ll explore the Visual Studio 2013 IDE, including how to run and test Visual Basic programs, how to use the development tools in the IDE, and how to adjust important compiler settings.
Microsoft Press Ebooks — Your bookshelf on your devices!

When you buy an ebook through oreilly.com you get lifetime access to the book, and whenever possible we provide it to you in five, DRM-free file formats—PDF, .epub, Kindle-compatible .mobi, Android .apk, and DAISY—that you can use on the devices of your choice. Our ebook files are fully searchable, and you can cut-and-paste and print them. We also alert you when we’ve updated the files with corrections and additions.

Learn more at ebooks.oreilly.com

You can also purchase O’Reilly ebooks through the iBookstore, the Android Marketplace, and Amazon.com.