Avoid the common pitfalls of Windows Phone programming
with this concise recipe collection written specifically for the busy developer

Windows Phone Recipes

A Problem-Solution Approach

SECOND EDITION

Fabio Claudio Ferracchiati | Emanuele Garofalo

Apress®

Windows Phone Recipes

A Problem-Solution Approach

Fabio Claudio Ferracchiati
Emanuele Garofalo

Apress®

Windows Phone Recipes, Second Edition

ISBN-13 (pbk): 978-1-4302-3371-8

ISBN-13 (electronic): 978-1-4302-3372-5

President and Publisher: Paul Manning
Lead Editor: Ewan Buckingham
Technical Reviewer: Simona Nasetti
Editorial Board: Steve Anglin, Mark Beckner, Ewan Buckingham, Gary Cornell, Morgan Ertel, Jonathan Gennick, Jonathan Hassell, Robert Hutchinson, Michelle Lowman, James Markham, Matthew Moodie, Jeff Olson, Jeffrey Pepper, Douglas Pundick, Ben Renow-Clarke, Dominic Shakeshaft, Gwenan Spearing, Matt Wade, Tom Welsh
Coordinating Editor: Jessica Belanger
Copy Editor: Kimberly Burton
Cover Designer: Anna Ishchenko
Compositor: Bytheway Publishing Services
Indexer: SPI Global
Artist: SPI Global

Distributed to the book trade worldwide by Springer Science+Business Media, LLC., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com.

For information on translations, please e-mail rights@apress.com or visit www.apress.com.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales—eBook Licensing web page at www.apress.com/bulk-sales.

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

The source code for this book is available to readers at www.apress.com. You will need to answer questions pertaining to this book in order to successfully download the code.

To Simona and Claudia, women of mine —Fabio

To my mommy and daddy, my lighthouse in the fog —Emanuele

Contents at a Glance

■ About the Authors xxi
■ About the Technical Reviewer xxii
■ Acknowledgments xxiii
■ About This Book xxiv
■ Chapter 1: Introduction to Windows Phone Application Development 1
■ Chapter 2: Windows Phone Execution Model And Multitasking 43
■ Chapter 3: User Interface 97
■ Chapter 4: User Interface with Expression Blend 179
■ Chapter 5: Gestures 201
■ Chapter 6: Sensors 223
■ Chapter 7: Media Management 265
■ Chapter 8: Isolated Storage 329
■ Chapter 9: In the Cloud 379
■ Chapter 10: Testing Windows Phone Applications 417
■ Index 455

Contents

■ **About the Author .... xxi**

■ **About the Technical Reviewer .... xxii**

■ **Acknowledgments .... xxiii**

■ **About This Book .... xxiv**

■ **Chapter 1: Introduction to Windows Phone Application Development 1**

1-1. Examining Windows Phone 1

Problem 1

Solution 1

1-2. Understanding the Development Tools 4

Problem 4

Solution 4

How It Works 4

1-3. Creating a Simple Silverlight Windows Phone Application 6

Problem 6

Solution 6

How It Works 6

Usage 12

1-4. Creating a Simple XNA Windows Phone Application 13

Problem 13

Solution 13

How It Works 13

Usage 17
1-5. Creating a Simple XNA and Silverlight Mixed Application 18
Problem 18
Solution 18
How It Works 18
Usage 23
1-6. Deploying the Application to the Windows Phone Emulator 24
Problem 24
Solution 24
Usage 25
1-7. Deploying the Windows Phone Application on the Device 28
Problem 28
Solution 28
How It Works 28
Usage 29
1-8. Sending a Windows Phone Application to the Marketplace 31
Problem 31
Solution 31
How It Works 31
1-9. Creating a Trial Windows Phone Application 32
Problem 32
Solution 32
How It Works 33
Usage 39
■ **Chapter 2: Windows Phone Execution Model And Multitasking 43**
2-1. Navigating Between Pages 43
Problem 43
Solution 43

How It Works 43
The Code 45
Usage 47
2-2. Passing Data Through Pages 48
Problem 48
Solution 48
How It Works 48
The Code 49
Usage 50
2-3. Navigating Between Pages Using Global Application Variables 52
Problem 52
Solution 52
How It Works 53
The Code 53
Usage 54
2-4. Navigating Between Pages with State 56
Problem 56
Solution 56
How It Works 56
The Code 57
Usage 61
2-5. Managing Tombstoning in Windows Phone Silverlight Applications 65
Problem 65
Solution 65
How It Works 65
The Code 66
Usage 71
2-6. Managing Tombstoning in Windows Phone XNA Applications 75

Problem 75
Solution 75
How It Works 76
The Code........ 76
Usage........ 78
2-7. Implementing Multitasking 79
Problem 79
Solution 80
How It Works 80
The Code........ 81
Usage........ 86
2-8. Scheduling Alarms and Reminders 88
Problem 88
Solution 88
How It Works 89
The Code........ 89
Usage........ 90
2-9. Managing Obscured and Unobscured Events........ 93
Problem 93
Solution 94
How It Works 94
The Code........ 95
Usage........ 96
■ **Chapter 3: User Interface 97**
3-1. Creating an Animated Splash Screen........ 98
Problem 98
Solution 98
How It Works 98

The Code.. 98

Usage.. 100

3-2. Using the ApplicationBar Control 102

Problem 102

Solution 102

How It Works 103

The Code.. 104

Usage.. 106

3-3. Detecting Changes in the Theme Template 108

Problem 108

Solution 109

How It Works 109

The Code.. 109

Usage.. 110

3-4. Customizing the Soft Input Panel Keyboard to Accept Only Numbers 112

Problem 112

Solution 113

How It Works 113

The Code.. 114

Usage.. 115

3-5. Using the Windows Phone Predefined Styles..116

Problem 116

Solution 117

How It Works 117

The Code.. 118

Usage.. 123

3-6. Localizing Your Application 124

Problem 124

Solution 124
How It Works 124
The Code........ 126
Usage........ 128
3-7. Using Panorama and Pivot Controls 131
Problem 131
Solution 131
How It Works 131
The Code........ 131
Usage........ 137
3-8. Spicing Up the User Interface with the Silverlight Toolkit........138
Problem 138
Solution 138
How It Works 138
The Code........ 140
Usage........ 153
3-9. Using Launchers and Choosers........155
Problem 155
Solution 155
How It Works 155
The Code........ 157
Usage........ 166
3-10. Accessing to the Contacts Library........170
Problem 170
Solution 170
How It Works 170
The Code........ 170
Usage........ 173

3-11. Adding Advertising to Your Application 173

Problem 173

Solution 174

How It Works 174

The Code.. 175

Usage... 176

■ Chapter 4: User Interface with Expression Blend 179

4-1. Getting Ready for Light and Dark 179

Problem 179

Solution 180

How It Works 180

The Code.. 180

Usage... 185

4-2. Changing the Skin of Your App.. 185

Problem 185

Solution 185

How It Works 185

Usage... 190

4-3. Creating Some Cool Animation.. 190

Problem 190

Solution 190

How It Works 190

The Code.. 190

Usage... 194

4-4. Customizing a Control 195

Problem 195

Solution 195

How It Works 195

The Code 195

Usage 199

■ **Chapter 5: Gestures 201**

5-1. Managing Gestures in a Silverlight Windows Phone Application 201

Problem 201

Solution 201

How It Works 202

The Code 204

Usage 205

5-2. Adding Gestures Management to Click-less Silverlight Controls 207

Problem 207

Solution 207

How It Works 208

The Code 208

Usage 211

5-3. Handling Gestures in a Graphical Context, Such as a Game Menu 212

Problem 212

Solution 212

How It Works 213

The Code 214

Usage 216

5-4. Managing Gestures from the Silverlight for Windows Phone Toolkit 217

Problem 217

Solution 217

How It Works 217

The Code 217

Usage 220

■ **Chapter 6: Sensors 223**

6-1. Resetting a Form by Shaking the Phone 224

Problem 224

Solution 224

How It Works 224

The Code... 225

Usage.. 228

6-2. Creating a Seismograph..229

Problem 229

Solution 230

How It Works 230

The Code... 230

Usage.. 235

6-3. Indicating the User’s Position on a Map..235

Problem 235

Solution 235

How It Works 236

The Code... 236

Usage.. 241

6-4. Displaying Sunset and Sunrise..241

Problem 241

Solution 241

How It Works 241

The Code... 242

Usage.. 242

6-5. Indicating the User’s Position via Coordinates 243

Problem 243

Solution 243

How It Works 243
The Code 243
Usage 247
6-6. Using the Radio 247
Problem 247
Solution 247
How It Works 248
The Code 248
Usage 253
6-7. Building an Augmented Reality Application 253
Problem 253
Solution 253
How It Works 254
The Code 254
Usage 262
■ **Chapter 7: Media Management 265**
7-1. Taking a Photo from Your Phone Camera 265
Problem 265
Solution 265
How It Works 266
The Code 266
Usage 270
7-2. Capturing Photo Camera Video Frames 274
Problem 274
Solution 275
How It Works 275
The Code 277
Usage 284

7-3. Picking a Photo from Your Media Library 284

Problem 284

Solution 285

How It Works 285

The Code... 285

Usage.. 287

7-4. Add Integration Between Your Application and Windows Phone 289

Problem 289

Solution 289

How It Works 289

The Code... 289

Usage.. 291

7-5. Using Media Player to Shuffle Songs in Your Media Library 294

Problem 294

Solution 294

How It Works 295

The Code... 295

Usage.. 302

7-6. Playing Music Streamed from SkyDrive 304

Problem 304

Solution 304

How It Works 304

The Code... 305

Usage.. 309

7-7. Using the Microphone in the Funny Repeater Application 312

Problem 312

Solution 312

How It Works 312

The Code.. 313

Usage.. 315

7-8. Using the MediaElement Control to Play Both Music and Video 316

Problem 316

Solution 316

How It Works 316

The Code.. 317

Usage.. 320

7-9. Adding Integration with the Music-Videos Hub..320

Problem 320

Solution 320

How It Works 321

The Code.. 322

Usage.. 325

■ **Chapter 8: Isolated Storage 329**

8-1. Saving and Loading a File in Isolated Storage 330

Problem 330

Solution 330

How It Works 330

The Code.. 331

Usage.. 334

8-2. Saving Serialized Data 337

Problem 337

Solution 338

How It Works 338

The Code.. 338

Usage.. 340

8-3. Implementing the Local Database...340

Problem 340

Solution 340

How It Works 340

The Code........ 342

Usage........ 351

8-4. Modifying the Settings of Your Application 351

Problem 351

Solution 351

How It Works 351

The Code........ 351

Usage........ 355

8-5. Saving a Photo in Isolated Storage 355

Problem 355

Solution 355

How It Works 355

The Code........ 355

Usage........ 361

8-6. Using Isolated Storage Explorer........364

Problem 364

Solution 365

How It Works 365

The Code........ 365

Usage........ 366

8-7. Implementing Background File Transfer........366

Problem 366

Solution 366

How It Works 367

The Code........ 369

Usage 377

Chapter 9: In the Cloud 379

9-1. Interacting with WCF 379

Problem 379

Solution 379

How It Works 379

The Code 380

Usage 385

9-2. Accessing OData Services 385

Problem 385

Solution 385

How It Works 385

The Code 387

Usage 390

9-3. Creating a Feed Reader 391

Problem 391

Solution 391

How It Works 391

The Code 392

Usage 398

9-4. Creating a Google API–based Translator 398

Problem 398

Solution 398

How It Works 398

The Code 398

Usage 402

9-5. Pushing Your Notification 402

Problem 402

Solution 402
How It Works 402
The Code.. 406
Usage.. 414
■ Chapter 10: Testing Windows Phone Applications 417
10-1. Implementing MVVM on Windows Phone Using MVVM Light..418
Problem 418
Solution 418
How It Works 418
The Code.. 420
Usage.. 429
10-2. Implementing MVVM on Windows Phone Using Prism..429
Problem 429
Solution 429
How It Works 430
The Code.. 430
Usage.. 438
10-3. Using MVVM and Performing Unit Testing..440
Problem 440
Solution 440
How It Works 441
The Code.. 441
Usage.. 443
10-4. Using Application Profiling 444
Problem 444
Solution 445
How It Works 445
The Code.. 445

Usage.......... 445

10-5. Following Certification Test Steps.......... 450

Problem.......... 450

Solution.......... 450

How It Works.......... 450

The Code.......... 450

Usage.......... 450

■ **Index.......... 455**

About the Authors

■ **Fabio Claudio Ferracchiati** is a prolific writer and technical reviewer on cutting-edge technologies. He has contributed to many books on .NET, C#, Visual Basic, SQL Server, Silverlight, and ASP.NET. He is a .NET Microsoft Certified Solution Developer (MCSD) and lives in Rome, Italy. He is employed by Brain Force.

■ **Emanuele Garofalo** was born at Torre del Greco (Naples), Italy, and now lives in Rome. He is an active member of the DotNetCampania community, and works with Windows Presentation Foundation (WPF), Silverlight, Windows Communication Foundation (WCF), Asp.Net, and Language Integrated Query (LINQ) as principal technologies. He is employed by BCSoft.

About the Technical Reviewer

■ **Simona Nasetti** is an expert Microsoft Dynamics CRM and Microsoft SQL Server Business Intelligence consultant. She graduated with a mathematics degree and works at Agic Technology (`www.agic.it`) in Rome, Italy, where she creates vertical solutions and reports for the company's clients.

Acknowledgments

First of all, I'd like to thank Ewan Buckingham. During these years we had working experiences together and he has been always patient, kind, and helpful. I'd also like to thank Emanuele for the great job he did working with me to complete this book. Moreover, the Apress guys, each of you have been great. So, thanks to Jessica Belanger, Dominic Shakeshaft, Christine Ricketts, and Sharon Wilkey. Finally, my wife Simona for her efforts especially trying to quiet my daughter's screams while I was writing the book.

—Fabio Claudio Ferracchiati

Thanks to Alessandra for her patience during the writing of this book, to Fabio for the opportunity to write with him, to Ewan, Jessica, Sharon, and all the staff of Apress for their support, and to Michele Aponte and all the DotNetCampania community for the motivation to study new technologies with the spirit of knowledge-sharing.

—Emanuele Garofalo

About This Book

Are you interested in Windows Phone configuration and development? Learn to build, configure, and distribute your applications through a wide variety of step-by-step recipes. This book contains extensive code samples and detailed walk-throughs. Moreover, the book has been updated for the Windows Phone 7.1 SDK version and the Windows Phone 7.5 operating system.

This second edition covers all the new features—multitasking, networking, creating animated Live Tiles, and much more. Also, the recipes have been updated to show what you have to do to update your applications so that they work on this latest operating system.

CHAPTER 1

Introduction to Windows Phone Application Development

This chapter introduces Windows Phone, including its device hardware characteristics and software development tools. After this introduction, you will learn how to create simple applications and how to deploy them. Finally, you are going to look at the application's distribution via Windows Phone Marketplace. The recipes in this chapter describe how to do the following:

- 1-1: Examine Windows Phone hardware
- 1-2: Examine Windows Phone development tools
- 1-3 and 1-4: Create a simple Windows Phone Silverlight and XNA application
- 1-5: Create a simple Silverlight and XNA mixed application
- 1-6 and 1-7: Deploy a Windows Phone application on both the emulator and device
- 1-8: Put a Windows Phone application into Windows Phone Marketplace
- 1-9: Create a Windows Phone trial application

1-1. Examining Windows Phone

Problem

You have just bought your new Windows Phone 7 device and would like to start developing applications. You need to know the device's hardware characteristics, such as screen resolution and memory capability, but also which sensors and services it provides. Finally, you need to understand what the Windows Phone operating system provides in order to integrate your application in the best way.

Solution

If you have Windows Mobile development experience, please erase it! Joking aside, Microsoft has provided a brand new operating system for its new smartphone: Windows Phone. This operating system has been written from scratch in order to reach—and sometime go beyond—other operating systems' functionalities.

To make an operating system that is reliable and fast and has high performance, Microsoft has dictated hardware requirements. So, every Windows Phone–compatible phone on the market right now and in the future has (or will have) at least minimum common hardware characteristics. For us as developers, this is great news, because we can write code having some certainty of what the smartphone provides.

The Windows Phone device provides a screen resolution of 480×800 pixels in portrait orientation. In the future, mobile vendors plan to release smartphones with smaller screens having a resolution of 320×480 pixels. Having this in mind, you can create a game and draw your sprites knowing that your game will be played on a screen with that resolution—so no scale operations, screen text adaptation, and so forth will be necessary. But even for classic applications showing, for example, text boxes and buttons, this resolution is useful for drawing rich user interfaces.

Every phone provides three hardware buttons, usually at the bottom of the screen, as shown in Figure 1-1. They are the Back button, the Start button, and the Search button. The leftmost button is used to go back to the previous application (just like the Back button on an Internet browser). Moreover, holding the Back button for one second, Windows Phone shows the applications that can be resumed (more on this in Chapter 2). The middle button is used to close the current application and to show the Start menu so that the user can open another application. Holding the Start button you can use the Windows Phone voice recognition to retrieve and call a contact, start an application or dictate an SMS message. The rightmost button is used to access the start screen and start a search. You can do simple text searching using Bing site or search song's information letting Windows Phone hearing the song. Moreover, you can use the Windows Phone camera to retrieve information on items either reading their barcode or letting Windows Phone read text and use Microsoft Tag service.

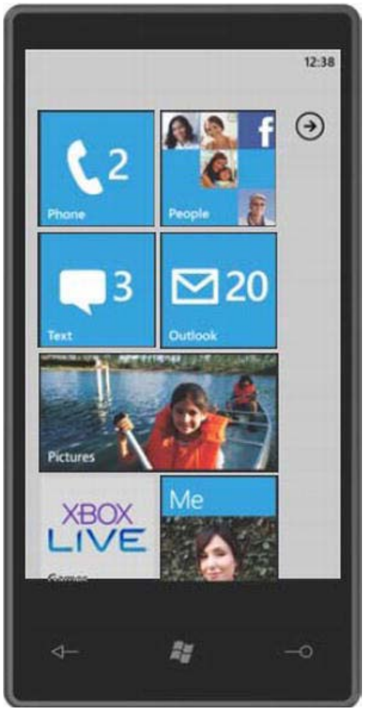

Figure 1-1. *A generic Windows Phone 7 device*

From a developer's point of view, it is important to understand the impact that these buttons have on an application. When each button is pressed, the running application is either deactivated or killed. A developer has to provide code that responds to those events, perhaps saving data in isolated storage (an application's disk-dedicated storage). To redisplay the application, perhaps after the user pushes the Back button, code has to have been written in order to re-create the same situation present before the deactivation. You can see more on this in Chapter 2.

Windows Phone devices have a Soft Input Panel (SIP) that enables users to write text into text boxes. A hardware keyboard is optional. In both cases, the application will receive text input in the same manner. The same is true for key pressure events. The SIP is shown automatically by Windows Phone when text input is required by the user.

In Figure 1-1, you can see the Windows Phone starting page and its *Metro* user interface. Microsoft designers, with users' feedback, have preferred to put the accent on content and information instead of eye-catching graphics. So the screen is populated with something similar to either metro or airport banners. Every square and rectangle is called a *live tile* and gives access to the *hub*. Each live tile is updated in real time with information taken from the hub. The hub is a sort of aggregator to group similar information such as group photos taken from the web, from the phone itself, and from social networks. For example, the Office tile will show counter indicating the number of incoming e-mail when a new e-mail arrives. So the hub contains an aggregation of information that is both local (on the phone) and remote (on the cloud and from the Internet). For example, the *Pictures hub* contains photos taken from the internal camera and from social networks such as Facebook. There are six hubs provided with Windows Phone 7:

- People
- Office
- Pictures
- Music and Videos
- Marketplace
- Games

By the way, the phone is completely customizable, so you can remove live tiles, add your preferred ones, move tiles, and so on. Users can choose between two graphics themes: dark or light. Each presents a different background color (black and white, respectively), which is important to be aware of as you draw your icons, buttons, and images for an application.

The user can interact with Windows Phone by using its multi-touch screen. Using your fingers to perform various gestures such as taps, you can move the tiles, zoom in and zoom out on text and pictures, and so on. Every vendor that produces Windows Phone devices must provide at least a four-point multi-touch capacitive screen so that you can use at least four fingers on the touch screen.

The Windows Phone device ships with 256 MB or more of RAM and with 8 GB or more of flash storage. The CPU depends on vendors but it must provide at least 1 GHz of frequency.

Finally, the Windows Phone device provides sensors and services to bring the user experience to the next level. Here is a list of the most important ones:

> *A-GPS*: This sensor is the Assisted Global Positioning System. It enables users to retrieve their position in the world in terms of longitude and latitude coordinates taken from both satellite services and cell-site triangulation. The latter is less accurate because it represents the nearest radio network from the phone position but it is useful when satellite signals are low or absent.

Accelerometer: This sensor enables programs to understand when the phone has been moved—for example, either because the user has taken it from the desk to respond to a call, or worse, the phone is falling from the user's hands!

Wi-Fi: This sensor enables the phone to connect to a Wi-Fi spot for an Internet connection.

Camera: This sensor enables users to take photos and videos through a 5-megapixel (or more) camera with flash. New Windows Phone models provide a front-camera (usually less powerful) useful to have video calls and video chat.

Office: This service is not so advertised, but every phone has a very usable and powerful version of Microsoft Office with its common applications such as Word, Excel, Outlook, and PowerPoint. The Office Hub is fully integrated with Office 365, and you can also store your documents directly on your SkyDrive cloud hard drive.

Location: Thanks to this service, a user can be located, and that user's position can be represented via Bing Maps.

Push Notifications: This is a great service that prevents phone to polling information from the Internet. The phone waits to receive notifications from programs that live outside the phone avoiding to continually going to search for new information.

Multitasking: Windows Phone implements multitasking allowing developers to create background agents and task scheduling, background file downloading, and background music playing.

Developers can use all these sensors and services together to create innovative applications and sell them on Windows Phone Marketplace. They do not have to worry about hardware differences (for example, whether a certain cell model has the accelerometer) because every Windows Phone has the same minimum features.

1-2. Understanding the Development Tools

Problem

You want to start developing for Windows Phone. You want to know which tools and which languages you have to use to make an application.

Solution

You have to download the Microsoft Windows Phone Developer Tools.

How It Works

We started Recipe 1-1 saying that if you have Windows Mobile development experience, it is better to erase it! This is a joke, of course, but it is not completely false. In Windows Phone development, you don't have the freedom to create low-level applications with C or C++ languages. Using .NET is the only way allowed by Microsoft to develop your applications for Windows Phone. Even if you find a way to go around this limitation—let's say by injecting some Intermediate Language (IL) code at runtime—you still have to remember that every application will be distributed by Windows Phone Marketplace. And, of course, before users can find your application on Marketplace, that application has to go through different approval steps, and you can be sure that any non-.NET application would not pass the certification process.

You can create three kinds of applications: Silverlight for Windows Phone, XNA for Windows Phone, and combining both technologies together. The first uses a custom Silverlight 4 version in which Microsoft has added some specific features. The second uses XNA libraries and is targeted at creating videogames. The third uses a Silverlight and XNA combination. Usually this is great solution for games where the menu and Internet access for scores sharing is accomplished by Silverlight controls and the game engine with animations, music, etc. is done by XNA.

■ **Note** If you plan to release an application for the old Windows Phone 7 operating system you have to know that you can combine Silverlight and XNA technologies in your application, with the only limitation being the user interface; you can't draw controls by using Silverlight and use XNA to draw sprites at the same time. On the other hand, you can use Silverlight for the user interface and XNA libraries to provide full access to media storage on the phone, to capture audio, and more.

You can use either C# or Visual Basic to write your Windows Phone application.

To start developing, you first have to download the Windows Phone Developer Tools from `http://create.msdn.com/en-us/home/getting_started`. This setup includes Visual Studio 2010 Express for Windows Phone, Windows Phone Emulator, Silverlight Tools, XNA 4, WCF Data Services Client for Window Phone, Microsoft Advertising SDK for Windows Phone, and Microsoft Expression Blend for Windows Phone. If you already have Visual Studio 2010 installed on your machine, the setup will install only the necessary files and you will see new project templates the next time you start the development tool.

Let's see the necessary steps to install the Microsoft Windows Phone Developer Tools:

1. Launch the installer (`vm_web.exe`) after having downloaded it.
2. Accept the license agreement.
3. Optionally, choose the Customized installation so you can select a folder in which to install the tools.
4. Wait for the installer to download all the necessary files from the Internet. The number of files downloaded depends on what the installer finds already in your operating system.
5. If you have to install the developer tools on machines not connected to the Internet, you can use the ISO version from `http://go.microsoft.com/fwlink/?LinkID=223971`.