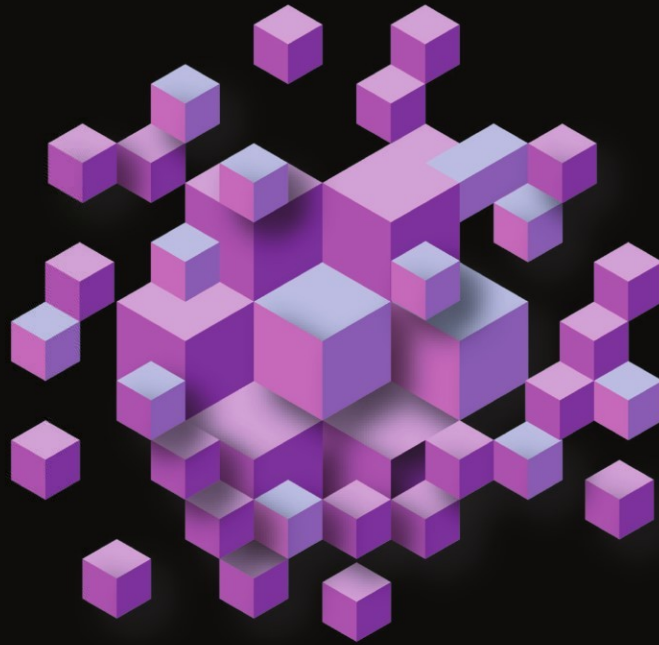


Master the art of creating authentic mobile applications on Microsoft's newest platform



# Pro Windows Phone 7 Development

Rob Cameron

Apress®

# Pro Windows Phone 7 Development



**Rob Cameron**

Apress®

## **Pro Windows Phone 7 Development**

Copyright © 2011 by Rob Cameron

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

ISBN 978-1-4302-3219-3

ISBN 978-1-4302-3220-9 (eBook)

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

President and Publisher: Paul Manning

Lead Editor: Jonathan Hassell

Technical Reviewer: Robert Garrett

Editorial Board: Steve Anglin, Mark Beckner, Ewan Buckingham, Gary Cornell, Jonathan Gennick,

Jonathan Hassell, Michelle Lowman, James Markham, Matthew Moodie, Jeff Olson, Jeffrey

Pepper, Frank Pohlman, Douglas Pundick, Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade,

Tom Welsh

Coordinating Editor: Debra Kelly

Copy Editor: Tracy Brown

Compositor: MacPS, LLC

Indexer: BIM Indexing & Proofreading Services

Artist: April Milne

Cover Designer: Anna Ishchenko

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](mailto:orders-ny@springer-sbm.com), or visit [www.springeronline.com](http://www.springeronline.com).

For information on translations, please e-mail [rights@apress.com](mailto:rights@apress.com), or visit [www.apress.com](http://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](http://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](http://www.apress.com). You will need to answer questions pertaining to this book in order to successfully download the code.

*To my beautiful wife, Ally, whose love and patience is unending,  
and to my daughters Amanda and Anna, who make me so proud every day.*

*—Rob*

# Contents at a Glance

<b>Contents .....</b>	<b>v</b>
<b>About the Author .....</b>	<b>xii</b>
<b>About the Technical Reviewer .....</b>	<b>xiii</b>
<b>Acknowledgments .....</b>	<b>xiv</b>
<b>Introduction .....</b>	<b>xv</b>
<b>■ Chapter 1: Introduction to Windows Phone 7 and Marketplace.....</b>	<b>1</b>
<b>■ Chapter 2: Silverlight User Interface Development .....</b>	<b>47</b>
<b>■ Chapter 3: Handling Input on Windows Phone 7 .....</b>	<b>131</b>
<b>■ Chapter 4: Windows Phone 7 Programming Model .....</b>	<b>183</b>
<b>■ Chapter 5: Windows Phone 7 Advanced UI Development.....</b>	<b>251</b>
<b>■ Chapter 6: Windows Phone 7 Advanced Programming Model .....</b>	<b>309</b>
<b>■ Chapter 7: Building 2D Games with the XNA Framework .....</b>	<b>367</b>
<b>■ Chapter 8: Advanced XNA Framework Development .....</b>	<b>405</b>
<b>Index.....</b>	<b>441</b>

# Contents

<b>Contents at a Glance .....</b>	<b>iv</b>
<b>About the Author .....</b>	<b>xii</b>
<b>About the Technical Reviewer .....</b>	<b>xiii</b>
<b>Acknowledgments .....</b>	<b>xiv</b>
<b>Introduction .....</b>	<b>xv</b>
<b>■ Chapter 1: Introduction to Windows Phone 7 and Marketplace.....</b>	<b>1</b>
<b>Why Should I Care? .....</b>	<b>2</b>
Smartphone Growth .....	2
Is Microsoft Committed? .....	2
<b>Current Microsoft Developers .....</b>	<b>2</b>
<b>Non-Microsoft Developers .....</b>	<b>3</b>
<b>The Mobile Market .....</b>	<b>3</b>
<b>Windows Phone 7 Platform Overview .....</b>	<b>4</b>
Phone Design Philosophy .....	4
Hardware Capabilities .....	6
Software Capabilities .....	7
<b>Windows Phone Developer Tools .....</b>	<b>9</b>
<b>Silverlight Tools and Framework Overview .....</b>	<b>9</b>
Silverlight for Windows Phone .....	9
Hello Silverlight for Windows Phone .....	10
<b>XNA Game Studio and Framework Overview .....</b>	<b>17</b>
Hello XNA for Windows Phone .....	18
Implement the Game.....	24

<b>AppHub and the Windows Marketplace for Mobile .....</b>	<b>35</b>
Windows Phone Marketplace Registration .....	35
Submitting Applications to Marketplace .....	35
Bing Visual Search for Windows Phone 7 .....	37
Windows Phone Marketplace on the PC .....	38
Marketplace on the Device .....	40
Limiting Distribution of Applications .....	44
<b>Conclusion .....</b>	<b>46</b>
<b>■ Chapter 2: Silverlight User Interface Development .....</b>	<b>47</b>
<b>    Designing for Windows Phone 7 .....</b>	<b>47</b>
Design Approach .....	48
Design Resources .....	48
Designer and Developer Mechanics.....	52
<b>    User Interface Development in Visual Studio 2010 .....</b>	<b>54</b>
WAppManifest.xml File .....	56
Anatomy of a WP7 Application .....	57
Anatomy of a PhoneApplicationPage .....	65
Page Navigation .....	67
<b>    .NET Framework for Silverlight .....</b>	<b>71</b>
XAML Overview .....	72
Key Silverlight UI Base Classes.....	78
<b>    User Interface Development with Expression Blend 4 .....</b>	<b>83</b>
Layout System .....	83
Layout Controls Sample and Blend Behaviors .....	85
Data Binding.....	89
Sample Data.....	103
Control Styling.....	105
Control Templates.....	111
<b>    Animation Fundamentals .....</b>	<b>115</b>
Creating an Animation Storyboard .....	115
Creating Animations in Blend.....	118
Creating Animations in Code.....	122

Application Layout.....	123
Panorama.....	123
Pivot.....	128
Conclusion.....	130
<b>■ Chapter 3: Handling Input on Windows Phone 7.....</b>	<b>131</b>
The Keyboard.....	131
Physical Keyboard.....	132
Soft Input Panel (SIP) Keyboard.....	132
Programming with the Keyboard.....	132
Touch Input.....	138
Single-Point Touch.....	138
Raw Touch with Mouse Events.....	141
Multi-Point Touch.....	152
Accelerometer.....	170
Location.....	176
Microphone Input.....	179
Camera Input.....	181
Other Inputs.....	181
Conclusion.....	181
<b>■ Chapter 4: Windows Phone 7 Programming Model.....</b>	<b>183</b>
Device Information.....	183
Identifying Unique Users.....	184
Device Information.....	184
System Environment Information.....	186
Network Status.....	186
System Tray.....	189
Marketplace Status and Trial Mode.....	189
Application Data Persistence.....	190
Basic File IO.....	191
Object Persistence.....	194



Tasks.....	196
<b>Web Browser Control .....</b>	<b>207</b>
Basic WebBrowser Control Sample .....	207
WebBrowser Control Scripting Sample.....	209
<b>Asynchronous Programming.....</b>	<b>213</b>
Background Threads.....	213
<b>Connecting Windows Phone to Services and Feeds .....</b>	<b>219</b>
HttpWebRequest and WebClient .....	220
Windows Communication Foundation Support .....	220
Web Services .....	221
REST+OData.....	226
Plain Old REST+JSON .....	228
<b>Bing Maps Control.....</b>	<b>237</b>
Licensing.....	238
Programmatically Add a Pin.....	238
Data Bind to a Collection.....	240
<b>Application Execution Model.....</b>	<b>243</b>
User Experience .....	244
Event Lifecycle.....	244
Managing State.....	246
Running Under Lock Screen.....	248
<b>Conclusion .....</b>	<b>249</b>
<b>■ Chapter 5: Windows Phone 7 Advanced UI Development.....</b>	<b>251</b>
<b>The Model-View-ViewModel Architecture.....</b>	<b>251</b>
MVVM Overview .....	251
Pick a MVVM Helper SDK .....	260
GalaSoft MVVM Light Toolkit.....	260
<b>Silverlight for Windows Phone 7 Toolkit .....</b>	<b>272</b>
Installation and Overview.....	273
AutoCompleteBox Control .....	275
ContextMenu Control.....	276

DatePicker and TimePicker Controls.....	277
ListPicker Control.....	278
ToggleSwitch Control.....	279
WrapPanel Control.....	281
LongListSelector Control.....	283
<b>Creating Transitions and Interactivity.....</b>	<b>289</b>
Toolkit Page Transitions.....	289
Visual State Manager.....	296
<b>The Microsoft Advertising SDK.....</b>	<b>303</b>
Getting Started.....	303
Adding the Advertising Control.....	303
<b>Conclusion.....</b>	<b>307</b>
<b>■ Chapter 6: Windows Phone 7 Advanced Programming Model.....</b>	<b>309</b>
<b>Advanced Data Binding.....</b>	<b>309</b>
Syndicated Services.....	314
Showing Progress.....	317
Lazy Load Images.....	319
Data Bind to Anything.....	322
<b>Encryption Services.....</b>	<b>324</b>
Secure Sockets Layer.....	324
Securely Encrypting Data.....	324
<b>Working with Images on Windows Phone 7.....</b>	<b>329</b>
Images Classes.....	329
<b>The Windows Phone 7 Media Library.....</b>	<b>330</b>
<b>Building a Photo Extras Application.....</b>	<b>333</b>
Creating a Photos Extra App.....	334
<b>Working with Video and Audio Media.....</b>	<b>343</b>
Supported Codecs and Containers.....	344
Video and Audio in the Emulator.....	344
Progressive Download Video.....	344
Streaming Video.....	348

DRM .....	349
Audio Support .....	350
<b>Building a music+videos Hub Application .....</b>	<b>351</b>
Additional Assets .....	352
Testing music+videos Hub Applications .....	353
Debugging music+videos Hub Applications .....	353
music+videos Hub Sample Application .....	353
<b>Microsoft Push Notification Services .....</b>	<b>358</b>
Raw Push Notification Description .....	359
Toast Push Notification Description .....	359
Tile Notification .....	360
How Microsoft Push Notifications Work .....	361
Updating Application Tile Without Push Notification Services .....	362
Localization .....	364
Conclusion .....	365
<b>■ Chapter 7: Building 2D Games with the XNA Framework .....</b>	<b>367</b>
Xbox LIVE Gamer Services .....	367
Alien Shooter .....	368
From Demo to Game .....	368
Game Management .....	371
Screens and Menus .....	372
AlienShooter Game Structure .....	374
Content Project .....	375
Text and Menus .....	379
AlienShooter Game Play .....	384
Sprite Animation .....	384
Game Object Class .....	388
Enemy Class .....	392
Hero Ship Class .....	393
Missile Class .....	397
Game Status Board Class .....	397

Updated GameplayScreen Class .....	400
Collision Detection and Memory Management .....	403
Conclusion .....	404
<b>■ Chapter 8: Advanced XNA Framework Development .....</b>	<b>405</b>
AlienShooter Enhancements .....	405
Load and Save Game State .....	405
Tombstone Support.....	410
Particle System .....	411
<b>3D Game Development.....</b>	<b>415</b>
3D Game Concepts.....	415
Rendering 3D Primitives .....	418
The Game Class .....	425
Rendering 3D Models.....	431
Conclusion .....	440
<b>Index:.....</b>	<b>441</b>

# About the Author



■ **Rob Cameron** Employed by Microsoft since 2001, Rob Cameron is an Industry Architect Evangelist with Microsoft Corporation based out of Atlanta, Georgia. As part of Microsoft's Communication Sector Developer & Platform Evangelism team, Rob focuses on development tools and technologies targeting mobile devices, gaming, and embedded devices for telecommunications, cable, and media & entertainment companies. Rob has also worked on six books for Apress including authoring the latest Pro Windows Phone 7 Development. He has a master's degree in information technology management and a bachelor's degree in computer science. Visit Rob's blog at <http://blogs.msdn.com/RobCamer>.

# About the Technical Reviewer



■ **Rob Garrett** has worked with SharePoint since the early beta version of MOSS 2007 and has leveraged his talents for SharePoint architecture and design with Portal Solutions—a SharePoint consultant company in Rockville, MD. Rob has extensive background in .NET technologies and has developed software for Microsoft Frameworks since the early days of C++ and MFC. In pursuit of his career dreams, Rob left his birthplace in England for a journey to the United States on Thanksgiving Day 1999. Upon arrival, he enjoyed his first American meal from a gas station.

# Acknowledgments

I would like to acknowledge the incredible support provided by the Apress team who patiently waited as I struggled to balance all the demands in my life the past year. Jonathan Hassell has been there with me through all of my books with Apress, and I sincerely appreciate his support. Debra Kelly has been a trooper prodding me along to keep this book on track through the final push.

# Introduction

Authoring a book on mobile development with a title that starts with “Pro” is a daunting task. Put in too much introductory material and you lose the knowledgeable developer looking to take his application to the next level. Breeze through key concepts and the cross-platform professional mobile developer may struggle learning how to work with targeting a relatively new platform in Windows Phone 7.

Another challenge is culling down what to cover between Silverlight and the XNA Framework. Both are fabulous and interesting technologies so I strived to cover both in a single text, though this book is more geared toward Silverlight developers who are interesting in learning about the XNA Framework as well.

This book provides a comprehensive overview of XAML and Silverlight since it is a critical aspect of WP7 development. It also focuses on many of the more challenging topics such as correctly architecting Silverlight for Windows Phone 7 applications using the Model-View-ViewModel architecture as well as working with services, encryption, and gestures.

There are many tidbits throughout this book extracted from many long hours of working with top media brands to bring their content and applications to Windows Phone 7. I hope that sharing this knowledge on memory and CPU optimization gleaned from these experiences helps to bring your applications to the top of the rankings in the Windows Phone 7 marketplace.

## Who This Book Is For

If you are an existing Silverlight developer, this book will provide details on how to leverage your Silverlight skills for Windows Phone 7. If you are a mobile developer looking to port an existing application to Windows Phone 7, then this book will give you the information you need, covering the core concepts and techniques fundamental for Silverlight for Windows Phone 7 development.

Having an understanding of the .NET Framework will help, but if you are an experienced developer new to the .NET Framework, you will still be able to learn Windows Phone 7 development using this book because much of the development is in the XAML markup language as well as in code.

---

**Note** You still may want to grab a programming in C# text such as Adam Freeman’s *Introducing Visual C# 2010* (Apress, 2010) if you are not confident with C#.

---

The example code in this book is written in C#. However, much of the development is in the XAML, not in C#, so if you are a VB.NET developer, the markup is almost exactly the same. For the examples that do include C# code in the code-behind, the code translates pretty easily, as the .NET Framework for Silverlight for Windows Phone is language agnostic.



## How This Book Is Structured

This book consists of chapters that focus on individual topic areas of Windows Phone 7 and is intended to be read in the order in which it is written; however, I strive to point to related sections in other chapters so if you need to skip ahead to a specific topic feel free to do so.

I spend a bit more time on Silverlight than the XNA Framework, but both are covered throughout the book. Chapter 1 starts with an introduction to the Windows Phone Platform, the AppHub marketplace, Silverlight, and concludes with a detailed introductory sample on the XNA Framework. Chapter 2 focuses exclusively on Silverlight UX development. Chapter 3 covers input, leveraging examples both for Silverlight and the XNA Framework. With the fundamentals for professional development on Windows Phone 7 out of the way, Chapter 4 dives into the programming model with Silverlight as well as how to integrate into the Windows Phone 7 platform via tasks and choosers. Chapter 4 also covers the Bing Maps control and the WebBrowser control, including how to communicate between Silverlight and JavaScript.

Chapter 5 covers Advanced Silverlight UI development starting with a detailed investigation of the Model-View-ViewModel pattern. While this may seem odd in a chapter focused on UI, good architecture focuses on separation of concerns, which leads to better user experience and cleaner XAML. Chapter 6 covers advanced programming model concepts, including Advanced Data Binding, the Silverlight toolkit for WP7, working with the Media Library, creating a Photos Extra and Music plus Video hub application, as well as other integration scenarios.

Chapter 7 dives in deep on XNA Framework development for 2D applications, building on the introductory sample from Chapter 1. It includes coverage on how to leverage the Game Management sample from AppHub to create a professional looking application. Chapter 8 starts off with adding polish to the Chapter 7 2D game development sample, covering saving and load as well supporting tombstoning in XNA. A simple particle system is added to the 2D game to add explosions. Chapter 8 then delves into the world of 3D game development, covering working with primitives and models in 3D.

## Prerequisites

You will need a version of Visual Studio 2010, with Service Pack 1 of Visual Studio 2010 applied. You will also need to install Windows Phone Developer Tools available here:

[http://create.msdn.com/en-us/home/getting\\_started](http://create.msdn.com/en-us/home/getting_started)

You will want to download the Silverlight Toolkit for WP7 available here:

<http://silverlight.codplex.com>

I also highly recommend that you review the Design Resources for Windows Phone 7 available here:

[http://msdn.microsoft.com/en-us/library/ff637515\(VS.92\).aspx](http://msdn.microsoft.com/en-us/library/ff637515(VS.92).aspx)

## Downloading the Code

The code is available in zip file format in the Source Code/Download section of the Apress web site. Please review the `readme.txt` for setup instructions.

## Contacting the Author

To reach the author, please go to his blog and click the Email link to send him an email.

Rob Cameron: <http://blogs.msdn.com/RobCamer>.



# Introduction to Windows Phone 7 and Marketplace

Windows Phone 7 presents an exciting new opportunity for developers to build and monetize mobile applications. It represents a major new investment into mobile computing by Microsoft, and in many ways is a break from the past.

Up until roughly 2007, Windows Mobile was a growing, well-received platform targeting primarily enterprise users. Windows Mobile was taking share from competitors such as Palm and going head-to-head with RIM Blackberry, but overall it was considered fairly successful. The release of the iPhone and its consumer focus and applications store it turned the mobile device market on its head, and was a major wake-up call for Microsoft. Tack on the surge by Google's Android operating system, and the smartphone market became hyper-competitive.

---

■ **Note** Microsoft originally dubbed this new platform Windows Phone 7 Series. After some initial “feedback” from the market, Microsoft dropped the “Series” from the name. It is still a faux pas to call it Windows Mobile 7.

---

Windows Phone 7 represents a sea change in approach by Microsoft toward mobile computing. Although Windows Mobile 6.x and earlier resembles a miniaturized version of desktop Windows, with its iconic Start screen, Windows Phone 7 is very different with a user interface that is more similar to the ZuneHD interface – on steroids.

When the ZuneHD was introduced, it received positive feedback from critics, but it wasn't highly marketed toward consumers. It also did not have a third-party application Marketplace. Still, it was an opportunity for Microsoft to try out new mobile computing concepts on a mass audience of a few million users.

Learning from its own stumbles, taking what works from Zune, and then looking at the existing mobile landscape, Microsoft came up with Windows Phone 7 with a dramatically different user experience from Windows Mobile, a new mobile development paradigm, and a completely different approach in the market.

With Windows Phone 7, Microsoft makes the end-user consumer the top priority, whereas in the past, with Windows Mobile, the original equipment manufacturer (OEM) was Microsoft's primary customer. This does not mean that Microsoft is OEMing the device directly. Instead, Microsoft spec'd out the hardware and software for the platform such that OEMs can take that specification as a basis for further innovation. OEMs can build devices with or without a keyboard, or with slightly different physical sizes, different screen types, and additional features like a camera and more storage, to name a few options.

This may make you nervous as a developer, but it is important to remember that every Windows Phone 7 device at launch will have an 800 × 480 screen resolution, the exact same user interface and APIs, and the same range of sensors no matter what the OEM or Mobile Operator. This means that developers can build an application and be confident that it will run on all available devices. I cover hardware in more detail later in the chapter.

Note Microsoft has stated that it may add a new screen resolution in the future, and will provide guidance and tooling to ease development and ensure applications can run on all screens.

From a software standpoint, the programming model for Windows Phone 7 is very different from Windows Mobile. With Windows Mobile, developers created applications with either C++ or .NET Windows Forms via the .NET Compact Framework. For Windows Phone 7, the available programming models are Silverlight and the XNA Framework. This represents a clean break from the past, which will be painful for existing Windows Mobile ISVs, but in the end it provides a much more powerful programming model for developers. I cover the software and capabilities in more detail in the following sections.

## Why Should I Care?

As a developer, you may wonder whether you should invest in building applications for Windows Phone 7. Is there room in the marketplace? Is Microsoft committed to mobile platforms? These are all reasonable questions.

## Smartphone Growth

The mobile phone has become a major computing platform, experiencing incredible growth. As I write this, Smartphone shipments make up approximately 10% of the market worldwide. It is expected to grow to 70% of the market over the next three or four years. Much of the growth potential is still ahead. A new or remade player in the market, such as Android and Windows Phone 7, can come on to the scene and capture the imagination of both developers and consumers with the right level of marketing and engineering commitment.

## Is Microsoft Committed?

This is a tricky one, because I work for Microsoft and am biased, but what I see in terms of activity and energy around Windows Phone 7 reminds me of what I observed as an IT pro and developer working outside of Microsoft when Windows 95 launched. I was earning my Master's degree and spending a lot of time with other platforms like Unix and IBM OS/2 Warp, generally looking down on Windows for Workgroups 3.11. Microsoft's client OS, WfW 3.11 was looking long in the tooth, very dated, and limited. Out of nowhere Windows 95 exploded on to the scene and changed everything.

Can Windows Phone 7 change the market? Time will only tell, as it is just getting started. However, I see a tremendous level of commitment at Microsoft focused on Windows Phone 7. It is a great platform in version one. Can it improve? Absolutely, but I can honestly say that I imagine this is what it was like at Microsoft when Windows 95 launched, and I am thrilled to be a part of it and to be working with customers and partners on building applications for Windows Phone 7!

## Current Microsoft Developers

If you are an existing .NET Framework, Silverlight, or XNA Game Studio developer, you are well on your way to having the skills needed to create great Windows Phone applications. However, there are many

unique aspects to mobile device development, as well as features that are specific to Windows Phone 7, that require new skills. This book will present to you the information you need to bring your existing skills to bear on this new and exciting platform and to build great mobile application experiences.

---

■ **Note** Windows Phone 7 development initially supported C# only at launch. Since launch, Microsoft has added support for Visual Basic .NET development with Silverlight. XNA Framework game development still requires C#.

---

If you are new to Silverlight and XNA Game Studio, I spend time in this chapter introducing you to the basic concepts in these technologies to get you started. Combine this chapter with the online references available, and you will have the necessary foundation to proceed with the rest of this text.

If you are an existing Windows Mobile developer, you have a great depth of understanding of .NET and deep knowledge of mobile development considerations as well. This book will help you bring your existing knowledge to Silverlight and XNA Game Studio so that you can port your existing applications to Windows Phone 7.

## Non-Microsoft Developers

If you are an existing mobile developer on another platform, this book will help you bring your mobile development skills to bear on Windows Phone 7. Windows Phone 7 applications are written in C#, which is similar to the Java language and has roots in C, so many of the language constructs will be straightforward. For additional information on C# in general, check out the Apress title *Beginning C# 2010: From Novice to Professional*, Second Edition, by Adam Freeman.

## The Mobile Market

The smartphone segment of the mobile device market is expected to continue significant double-digit growth in the foreseeable future. This growth stems from new users purchasing devices, as well as existing function phone and feature phone users moving to smartphones. Despite the hype, it is way too early to suggest that one platform or another has “won” the smartphone war. There is plenty of new growth to be had in the market before competing platforms will need to focus on taking share from one another.

Take Android as an example, which has stormed onto the market and has out-sold iOS devices so far in 2011. Regarding Windows Phone 7, Microsoft CEO Steve Ballmer claims that his company is “all-in” with respect to commitment to the mobile market and Windows Phone 7. The partnership with Nokia announced in early 2011 demonstrates the high-level of commitment and desire needed to bring a third ecosystem to market.

In describing my perspective on what Windows Phone 7 means to Microsoft, I like to use an analogy. When Microsoft first introduced the Xbox game console, the general consensus in the market was that the Sony PlayStation had “won” the market and there was no way the Xbox could catch up. Today, I don’t think anyone would question Microsoft’s investment or commitment with respect to the Xbox 360. Likewise, who would have guessed that Nintendo would have made a comeback like it did with the Wii. Competition is a wonderful thing for consumers. Expect even more competition and dynamism in the mobile device market!

## Windows Phone 7 Platform Overview

In this section, I provide an overview of the design philosophy behind Windows Phone 7. I also cover the available development model to familiarize you with underlying platform architecture and developer capabilities.

### Phone Design Philosophy

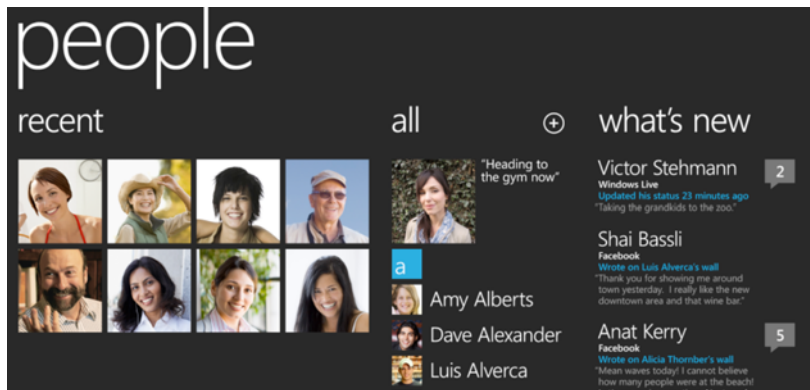
When Microsoft introduced Windows Phone 7, its executives spoke about “integrated experiences” and “smart design.” Without a doubt, Windows Phone 7 presents a unique user interface as compared to other current smartphone operating systems, as shown in Figure 1-1.



*Figure 1-1. Windows Phone 7 Start screen*

The Windows Phone 7 Start screen presents a lot of information in a clear fashion. Information and content is presented in a straightforward, easy to navigate manner without distraction. Controls have a minimalistic feel, allowing the user to focus on the content. This is a high-level overview of what Microsoft calls smart design. I focus more on design in Chapter 3.

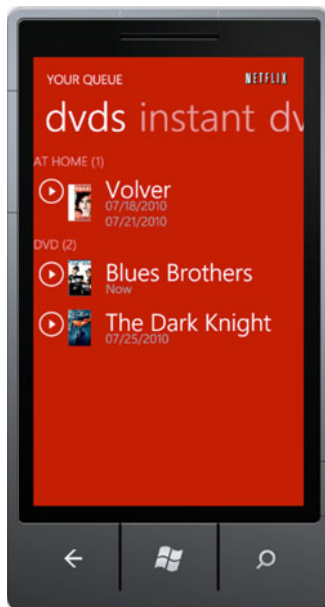
Also shown in Figure 1-1 are several top-level tile icons pointing to the built-in “hubs,” such as the People, Office, Pictures, Zune, Marketplace, and Xbox LIVE hubs. The hubs are the built-in, top-level panoramic experiences, meaning they scroll left and right to present content in a rich fashion. Figure 1-2 shows an early concept of the People hub. Notice that the focus is on the content, not “chrome.”



*Figure 1–2. Windows Phone 7 People hub Panorama experience*

Figure 1–2 shows a wide view of the full user interface width of the Panorama experience. A user swipes left and right to view content in addition to scrolling up and down on each viewable section. Hubs provide an integrated view on data. For example, if you register an Exchange, Facebook, and Windows Live account, the contact data is pulled together into a single integrated view called the People hub. This allows the user to explore contacts as well as see a scrolling view of status updates.

In addition to Panorama, the other major navigation paradigm is called Pivot, with very similar navigation behavior to what you see in the ZuneHD. Figure 1–3 shows an example Pivot interface running in the Emulator.



*Figure 1–3. Windows Phone 7 Pivot experience*

In the application shown in Figure 1–3, the user touches the categories at the top, such as DVDs, instant, and so on, to switch between lists of data. Selecting an item displays the details. The Pivot experience provides a great navigation experience for displaying large amounts of categorized or filtered data.

The Panorama and Pivot experiences are all part of the Metro UI design language, which is the underlying design approach for all of the Windows Phone 7 built-in experiences. The controls available when developing for Windows Phone 7 are skinned for Metro by default. You will see more of this in the “Silverlight Framework” section of this chapter.

Of course, you are not limited to Panorama or Pivot for application design and navigation. Windows Phone 7 supports a page navigation metaphor as well. In addition, many applications may simply consist of a single-page application that takes advantage of animations to render out the user experience.

I cover application user interface design in more detail in Chapter 3, but it is important to have an understanding of the platform philosophy up front in order to help you start to think about your application in terms of what it means to be a Windows Phone 7 application. For more information on Windows Phone 7 design, please check out the design resources available at <http://msdn.microsoft.com/en-us/library/ff637515%28VS.92%29.aspx>, or simply search for “design resources for Windows Phone” in your favorite Web search engine.

## Hardware Capabilities

Unlike the iPhone or Blackberry, Microsoft does not produce the phone directly. Instead, Microsoft has chosen to continue to partner with mobile device OEMs in order to scale production and to promote differentiation. However, unlike with Windows Mobile 6.x and its predecessors, Microsoft has taken a more hands-on approach in specifying the underlying hardware such as chipset, CPU, GPU, Camera, Sensors, minimum memory, and so on, in addition to authoring many of the drivers.

OEMs are still free to innovate in terms of providing additional capabilities such as a physical keyboard, additional memory, and storage, but the underlying consistency ensures that applications work well across device manufactures, which greatly simplifies the efforts required of developers.

Every Windows Phone 7 device has the following capabilities:

- Capacitive 4-point multi-touch screen
- DirectX 9 hardware acceleration
- Digital camera
- Accelerometer
- Assisted GPS
- Compass sensor
- Light sensor
- Proximity Sensor

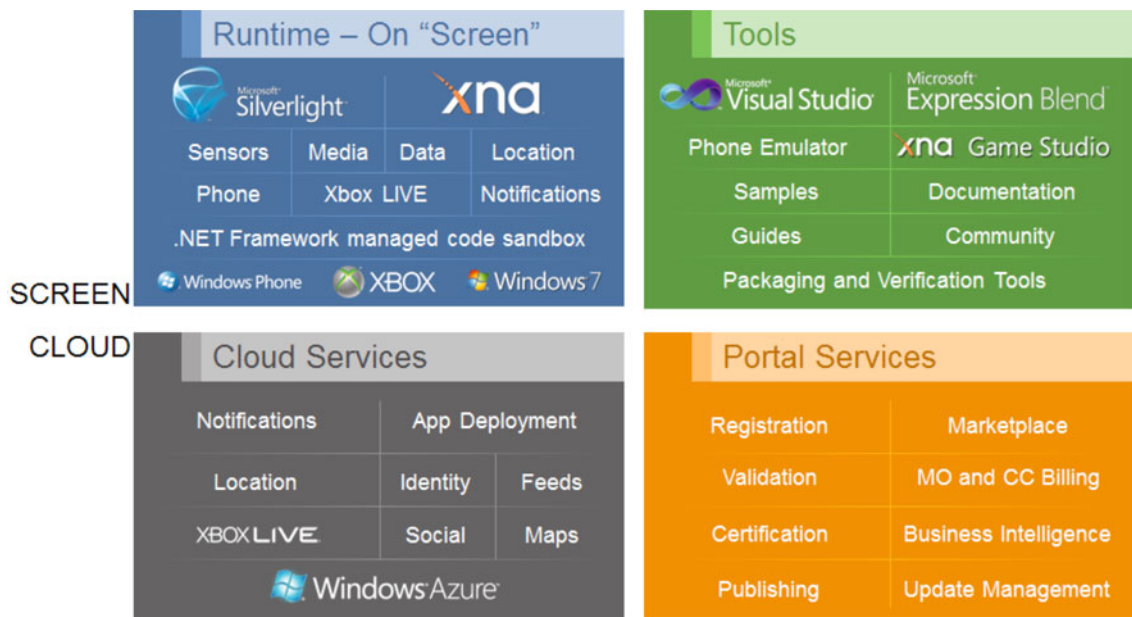
Windows Phone 7 hardware includes a powerful ARM v7 “Cortex/Scorpion” CPU, a powerful NVIDIA Tegra GPU, a minimum of 256 MB of RAM, and 8 GB or more of flash storage. Windows Phone 7 includes a large WVGA (800 × 480) display at launch.

At some point in the future Windows Phone 7 will also be available with an HVGA (480 × 320) display to provide for additional hardware differentiation. Microsoft will provide tools and guidance to assist developers with creating applications that support both resolutions. However, as you learn more about Silverlight, you will be well on your way to correctly writing applications that work well on both resolutions.

For XNA Framework-based games, the Windows Phone 7's powerful hardware scaler can assist with scaling assets as needed, whether to allow sharing of assets with Xbox and PC versions of a game or to assist with scaling assets for the HVGA version of Windows Phone 7 when it becomes available at some point after initial launch.

## Software Capabilities

Windows Phone 7 includes two programming models: the Silverlight and XNA Frameworks. Both Silverlight and XNA Game Studio are in version four of their desktop focused versions, bringing a high level of maturity to mobile development. Figure 1–4 illustrates the iconic breakdown shown by Microsoft when discussing the Windows Phone 7 Application Platform at Mix, TechReady, and pretty much any other setting.



**Figure 1–4.** Windows Phone 7 application platform

The upper left-hand corner of Figure 1–4 lists the two major programming models, Silverlight and XNA Framework, with the various integration points developers can take advantage of. Note that all of the underlying capabilities are accessible from either Silverlight or XNA Framework. All of these capabilities run inside the .NET Framework common language runtime in a sandboxed environment.

The upper right-hand corner of Figure 1–4 lists the two primary tools you use to create Windows Phone 7 applications, Visual Studio 2010 and Expression Blend 4. Visual Studio 2010 is the development tool most developers are familiar with already. XNA Game Studio sits inside of Visual Studio 2010 via a plug-in for game development.

A critically important tool listed in the upper right-hand corner is the Windows Phone Emulator. The Windows Phone Emulator is a virtual machine that allows developers to create applications and games without having a physical device. If you have a touch-enabled monitor, the emulator will pick up touch interactions as if running on a device. You can also simulate location and accelerometer.



---

■ **Tip** Microsoft limits the Phone Emulator to a single processor and injects sleep events to better simulate actual device performance, but generally performance will be better in the emulator than on a device, because there are far more resources available on a desktop PC. It is therefore important to periodically test your application on a real device.

---

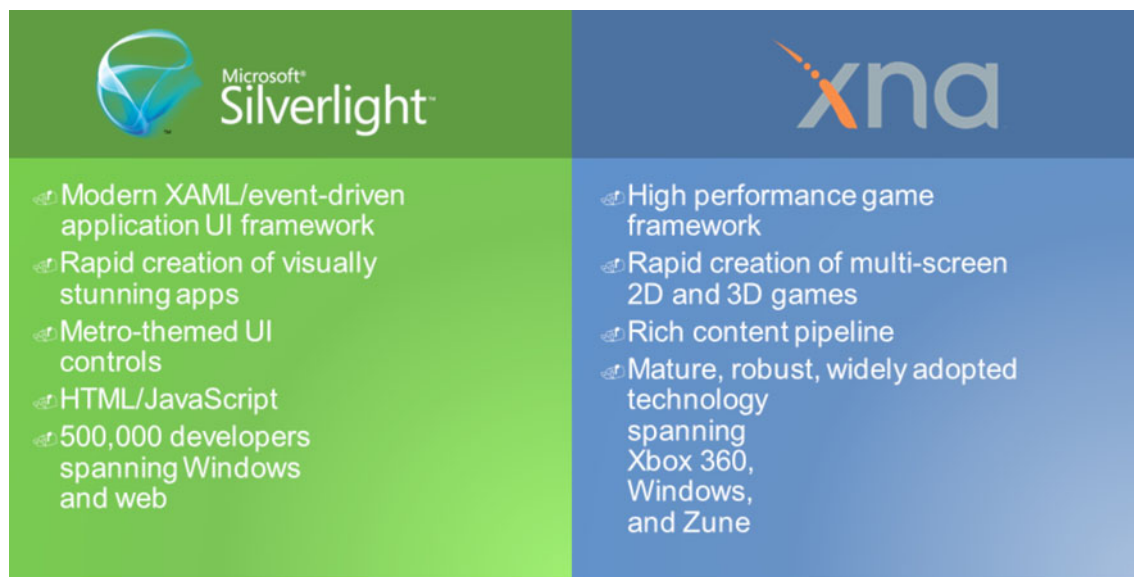
In Figure 1–4, the lower left-hand corner of the diagram represents code that you access in the cloud, such as notification services and location services that Microsoft provides, as well as custom-code you write to support your application such as web services, REST data, and so on.

The lower right-hand corner of in Figure 1–4 contains a reference to the tools that help you ship and promote your applications, primarily the Windows Marketplace for Mobile, or Marketplace for short.

Now that I covered a high-level overview of the application platform, let's dive into the programming models, Silverlight and XNA Framework. You build user experiences in Silverlight via Xml Application Markup Language (XAML) with a C# code-behind model. XAML is the serialization format of the UI elements added to the form. Since XAML is well-formed XML, it provides excellent tooling and validation support. It is also a namespace-based human-editable serialization format, allowing developers to easily modify XAML by hand.

The UI for Silverlight can be created in both Visual Studio and Expression Blend 4 for Windows Phone, though Expression Blend 4 can make some design operations such as animations and data binding much easier to code visually.

XNA Framework applications, on the other hand, are developed purely in C# code in Visual Studio, though there are third-party frameworks that can assist with game development through visually designing levels and the like Figure 1–5 provides a high-level breakdown of Silverlight vs. XNA Framework.



*Figure 1–5. Windows Phone 7 application programming frameworks*

Both Silverlight and XNA Framework share the same application model built on top of the .NET Common Language Runtime (CLR), allowing access to libraries across frameworks. So, a Silverlight application can access the audio APIs available in XNA Framework. Likewise, an XNA Framework application can access remote server access APIs available in Silverlight.

---

■ **Note** 3.7 is the version of the .NET Compact Framework under the covers, but it is more relevant to think in terms of Silverlight 3+ and XNA Game Studio 4.0.

---

## Windows Phone Developer Tools

The Windows Phone Developer Tools are a free download available from AppHub at <http://create.msdn.com>. The tools include Visual Studio 2010 Express for Windows Phone, XNA Game Studio 4, and Expression Blend 4 for Windows Phone – all you need in a single free download.

If you are currently a Visual Studio 2010 Professional or later developer, fear not – you will not lose access to all of your favorite features available in the Professional or later products. When you install the Windows Phone Developer Tools over Visual Studio 2010 Professional or later, the tool simply folds into the existing installation with no loss of functionality.

Download and install the tools to prepare for the next two sections. The first section provides an overview of the Silverlight Tools and Framework. The section after covers XNA Game Studio in detail, because this book does not dive in deeply with XNA Game Studio until Chapter 7. Finally the chapter closes with an overview of AppHub and Marketplace.

## Silverlight Tools and Framework Overview

In general, Silverlight is Microsoft's cross-web browser, cross-platform, and cross-device plug-in for delivering the next generation of .NET Framework-based rich interactive applications for the Web and the Windows desktop. On the Web, Silverlight runs on Windows in Internet Explorer 6 or later, Mozilla Firefox, and Chrome build 1251 and later. Silverlight also runs on the Apple Mac in both Safari and Firefox, as well as on Linux in Firefox as part of the Moonlight project ([www.mono-project.com/Moonlight](http://www.mono-project.com/Moonlight)), a collaboration project between Novell and Microsoft to bring Silverlight to Linux.

## Silverlight for Windows Phone

Windows Phone 7 brings Silverlight to Windows Phone. Silverlight on Windows Phone 7 is based on Silverlight 3, with some Silverlight 4 features that were brought forward to the platform. There are some notable differences between Silverlight for the desktop and Silverlight for Windows Phone 7:

- Silverlight for Windows Phone 7 applications are deployed via Marketplace on the device and via the Zune client. Silverlight applications are not deployed from the Web or side-loaded from a PC to Windows Phone 7.
- Silverlight for Windows Phone 7 applications do not run in IE Mobile. Silverlight applications run directly on the phone via a host process in isolation from one another. If you are a Silverlight developer, you can think of it as a Silverlight out-of-browser application only – applications do not run in a browser.

- Some Silverlight Framework APIs are not applicable to Windows Phone 7. One example is the APIs that enable Silverlight in the web browser to communicate with the host browser. Because Silverlight does not run in the browser on Windows Phone 7, these APIs are not applicable.
- Silverlight for Windows Phone 7 includes additional functionality to make underlying device capabilities available to developers such as the built-in sensors, tasks, and choosers to enable selecting a contact, send an SMS or email, and so on.

Despite these differences, the vast majority of the Silverlight Framework is compatible between Silverlight for the Web/desktop and Silverlight for Windows Phone so learning Silverlight is a great way to also learn how to build cross-platform applications between the Web, desktop, and Windows Phone 7.

---

■ **Note** For more information on Silverlight go to <http://Silverlight.net> and [www.microsoft.com/expression/products/Blend\\_Overview.aspx](http://www.microsoft.com/expression/products/Blend_Overview.aspx).

---

## Hello Silverlight for Windows Phone

Now that you have a high-level overview of the application platform and programming model, let's dive a little bit deeper into Silverlight development with the Windows Phone Developer Tools. We start first with an overview of Visual Studio 2010.

Silverlight development was available in Visual Studio 2005 and 2008; however, support for Silverlight really shines in Visual Studio 2010, making it a first-class citizen development technology with great tooling support. I also provide a very high-level overview of Expression Blend, now in its fourth version and included as a free download for Windows Phone development as part of the Windows Phone Developer Tools.

### Visual Studio 2010 Tour

Once you have the Windows Phone Developer Tools installed, fire up Visual Studio 2010 and select File ► New ► Project... to bring up the New Project dialog. Select Silverlight for Windows Phone on the left of the dialog and then select Windows Phone Application. Enter a name and location and then click OK to create it.

After the project is created, double-click on `MainPage.xaml` in the Solution Explorer tool window on the right to bring up the main application form or page. Figure 1–6 shows the UI with key items highlighted.

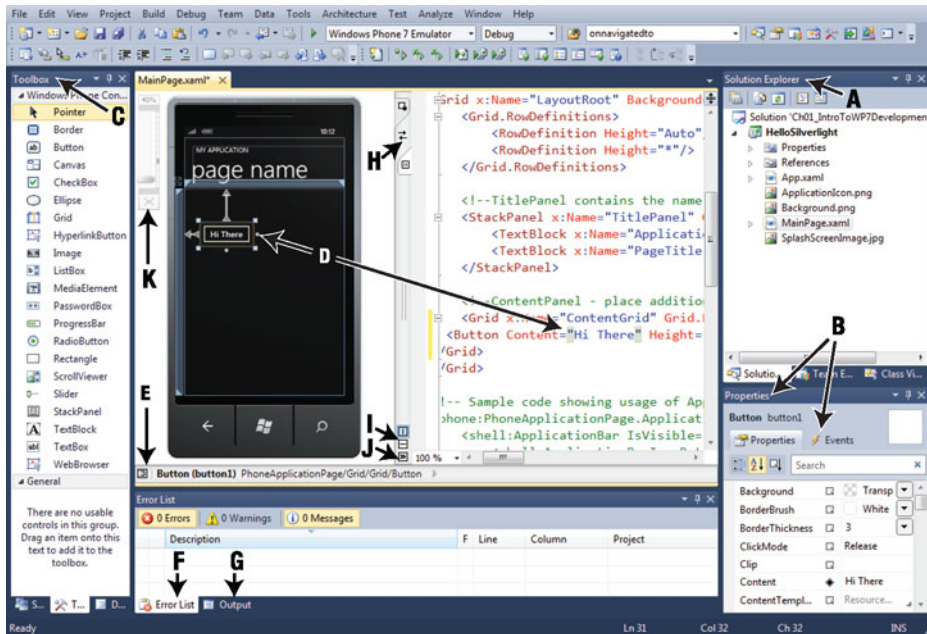


Figure 1–6. Windows Phone 7 Silverlight for Windows Phone design-time

Table 1–1 explains the identified items in the figure. If you are an experienced .NET developer, some of these may be obvious to you, but other items may not be so familiar.

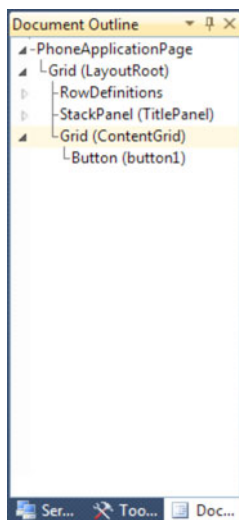
Table 1–1. Visual Studio Silverlight Developer Tools

Letter	Description
--------	-------------

- |   |   |
|---|---|
| A | Solution Explorer tool window: Lists the projects and project files for the entire solution. Right-clicked enabled for those new Visual Studio, so you will want to explore. The toggle buttons at the top show/hide hidden files, refresh file lists, and provide short cuts so show the code or designer for a particular file.   |
| B | Properties tool window: Lists the properties for the selected item in the designer or the item where the cursor is in the XAML markup file. Click the lightning bolt to switch to view the events for the selected object.  |
| C | Toolbox tool window: Lists the controls available for drag-and-drop on to the designer.   |
| D | The main area where designer and code files are viewed and edited. On the left is the Windows Phone designer view of the XAML code on the right. Notice the button that was dropped onto the designer. The areas point to both the design-time visualization of the button, as well as the XAML markup that actually defines the button. You can edit in either view and the changes are reflected in both views. |

Letter	Description
E	Document Outline tool window: Actually the button that displays the Document Outline as shown in Figure 1–7. It may not look like much but it presents the XAML markup, an XML tree, in a hierarchical view. As you design and develop your application, the nesting can be deep. When you select an item in the Document Outline, the focus shifts to that item for the Properties tool window. The Document Outline is an important part of your workflow to quickly find controls, modify properties, and so on.
F	When you build your project, errors show up in this tool window. Many times you can double-click on an error and it takes you to the correct line of code.
G	The Output tool window will show you what’s going on during a build. It is also the window where Debug.WriteLine messages appear if you have them in your code.
H	Click this button to swap the sides where the Windows Phone designer view and the XAML view appear.
I	This arrow points to two buttons that position the Windows Phone design view and XAML view either side-by-side-or vertically.
J	Click this button to expand the Windows Phone designer to full screen.
K	Use this slider to Zoom the designer in size to fit your screen best or to drill in for more exact positioning of controls.

Figure 1–7 shows the Document Outline tool window. Once you start using it as part of your development workflow, you will find it invaluable for anything but the most trivial projects.



*Figure 1–7. Document Outline tool window*

Now that I have provided a quick tour of Visual Studio 2010 for Windows Phone development, let's dive in with some code. Add a button toward the top middle portion of the form. Double-click on the button to create the default event handler for the control, which is the Click event for a Button control. Doing so creates the following event handler that is edited to adjust the content for the button:

```
private void button1_Click(object sender, RoutedEventArgs e)
{
    (sender as Button).Content = "Hi There Reader";
}
```

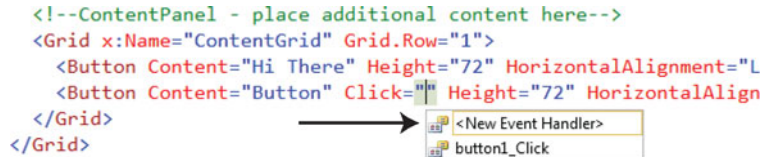
We can reference the button by name, `button1`, created by default. In the above code, since the `button1` is also the sender, we simply cast sender to `button` and set the `Content` property. A question you may be asking is how is the event handler associated with the `Button` object? The answer is via the XAML markup for `button1`, shown here:

```
<Button Content="Hi There" Height="72" HorizontalAlignment="Left" Margin="47,101,0,0"
Name="button1" VerticalAlignment="Top" Width="160" Click="button1_Click" />
```

All of the XAML was automatically generated when the control was dropped onto the designer surface except for the value configured in `Content` property and this line of code that associates the event handler in the `MainPage.xaml.cs` code-behind with the `Click` event for the control:

```
Click="button1_Click"
```

You can also generate event handlers by editing XAML directly. Drag another button on to the design-surface below the first. Change the `Width` property to `Auto` for this button, more on that later. Next go to the markup and type `Click=` inside of the `<Button ...>` markup for the second `Button` control. Figure 1–8 shows the IntelliSense pop-up.



```
<!--ContentPanel - place additional content here-->
<Grid x:Name="ContentGrid" Grid.Row="1">
    <Button Content="Hi There" Height="72" HorizontalAlignment="L
    <Button Content="Button" Click="  Height="72" HorizontalAlign
</Grid>
</Grid>
```

**Figure 1–8.** XAML IntelliSense

You can arrow up or down in the pop-up window shown in Figure 1–8 to select an existing event handler, or simply type `Tab` to generate a new event handler into the code-behind file named `button2_Click` by default. We copy the same code from the first event handler into the second event handler and click `F5` or the green arrow in the toolbar in Visual Studio to run the application the emulator as shown in Figure 1–9.