

Pro Silverlight 2 in VB 2008



Matthew MacDonald

Pro Silverlight 2 in VB 2008

Copyright © 2009 by Matthew MacDonald

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-13 (pbk): 978-1-4302-1602-5

ISBN-13 (electronic): 978-1-4302-1603-2

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

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

Lead Editor: Ewan Buckingham

Technical Reviewer: Fabio Claudio Ferracchiati

Editorial Board: Clay Andres, Steve Anglin, Mark Beckner, Ewan Buckingham, Tony Campbell,

Gary Cornell, Jonathan Gennick, Michelle Lowman, Matthew Moodie, Jeffrey Pepper,

Frank Pohlmann, Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh

Project Manager: Sofia Marchant

Copy Editor: Ami Knox

Associate Production Director: Kari Brooks-Copony

Production Editor: Ellie Fountain

Compositor: Dina Quan

Proofreader: Nancy Bell

Indexer: Broccoli Information Management

Artist: Kinetic Publishing Services, LLC

Cover Designer: Kurt Krames

Manufacturing Director: Tom DeBolski

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 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 <http://www.springeronline.com>.

For information on translations, please contact Apress directly at 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705. Phone 510-549-5930, fax 510-549-5939, e-mail info@apress.com, or visit <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 <http://www.apress.com/info/bulksales>.

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 <http://www.apress.com>.

For my family

Contents at a Glance

About the Author	xvii
About the Technical Reviewer	xix
Acknowledgments	xxi
Introduction	xxiii
CHAPTER 1 Introducing Silverlight	1
CHAPTER 2 XAML	29
CHAPTER 3 Layout	53
CHAPTER 4 Dependency Properties and Routed Events	99
CHAPTER 5 Elements	125
CHAPTER 6 The Application Model	161
CHAPTER 7 Shapes and Geometries	189
CHAPTER 8 Brushes and Transforms	223
CHAPTER 9 Animation	245
CHAPTER 10 Sound, Video, and Deep Zoom	291
CHAPTER 11 Styles, Templates, and Custom Controls	333
CHAPTER 12 Browser Integration	379
CHAPTER 13 ASP.NET and Web Services	409
CHAPTER 14 Data Binding	435
CHAPTER 15 Isolated Storage	491
CHAPTER 16 Multithreading	505
CHAPTER 17 Networking	529
INDEX	577

Contents

About the Author	xvii
About the Technical Reviewer	xix
Acknowledgments	xxi
Introduction	xxiii
CHAPTER 1 Introducing Silverlight	1
Silverlight and Visual Studio	1
Understanding Silverlight Websites	2
Creating a Stand-Alone Silverlight Project	3
Creating a Simple Silverlight Page	5
Adding Event Handling Code	7
Browsing the Silverlight Class Libraries	9
Testing a Silverlight Application	11
Silverlight Compilation and Deployment	13
Compiling a Silverlight Application	13
Deploying a Silverlight Application	14
The HTML Test Page	16
The Application Manifest	22
Creating an ASP.NET-Hosted Silverlight Project	24
The Last Word	27
CHAPTER 2 XAML	29
XAML Basics	29
XAML Namespaces	31
The Code-Behind Class	33
Properties and Events in XAML	34
Simple Properties and Type Converters	36
Complex Properties	37
Attached Properties	39
Nesting Elements	41
Events	44
The Full Eight Ball Example	45

Resources	46
The Resources Collection	46
The Hierarchy of Resources	48
Accessing Resources in Code	50
The Last Word	51

■ CHAPTER 3 **Layout** 53

The Layout Containers	53
The Panel Background	54
Borders	56
Simple Layout with the StackPanel	57
Layout Properties	59
Alignment	60
Margins	61
Minimum, Maximum, and Explicit Sizes	62
The Grid	64
Fine-Tuning Rows and Columns	66
Nesting Layout Containers	67
Spanning Rows and Columns	69
The GridSplitter	70
Coordinate-Based Layout with the Canvas	75
Layering with ZIndex	77
Clipping	77
Custom Layout Containers	79
The Two-Step Layout Process	79
A Wrapping Panel	82
Sizing Pages	86
Scrolling	87
Scaling	89
Full Screen	92
Navigation	93
Loading Child User Controls	94
Hiding Elements	96
The Last Word	97

■ CHAPTER 4 **Dependency Properties and Routed Events** 99

Dependency Properties	99
Defining and Registering a Dependency Property	100
Dynamic Value Resolution	102

Attached Properties	103
The WrapPanel Example	104
Routed Events	107
The Core Element Events	107
Event Bubbling	109
Handled (Suppressed) Events	110
An Event Bubbling Example	111
Mouse Movements	114
Capturing the Mouse	114
A Mouse Event Example	115
Mouse Cursors	118
Key Presses	119
Key Modifiers	121
Focus	122
The Last Word	123

CHAPTER 5 Elements 125

The Silverlight Elements	125
Static Text	127
Font Properties	129
Underlining	132
Runs	132
Wrapping Text	133
Images	134
Image Sizing	135
Image Errors	136
Content Controls	137
The Content Property	138
Aligning Content	140
Buttons	141
The HyperlinkButton	141
The ToggleButton and RepeatButton	141
The CheckBox	142
The RadioButton	143
Tooltips and Pop-Ups	144
Customized ToolTips	144
The Popup	146

Items Controls	148
The ListBox	148
The ComboBox	150
The TabControl	150
Text Controls	152
Text Selection	153
The PasswordBox	153
Range-Based Controls	154
The Slider	155
The ProgressBar	155
Date Controls	156
The Last Word	159

CHAPTER 6 The Application Model

Application Events	161
Application Startup	162
Application Shutdown	162
Unhandled Exceptions	162
XAML Resources	165
Application Tasks	166
Accessing the Current Application	166
Initialization Parameters	167
Changing the Page	170
Splash Screens	172
Resources	176
Placing Resources in the Application Assembly	177
Placing Resources in the Application Package	179
Placing Resources on the Web	180
Class Library Assemblies	184
Using Resources in an Assembly	184
Downloading Assemblies on Demand	185
The Last Word	187

CHAPTER 7 Shapes and Geometries

Basic Shapes	189
The Shape Classes	190
Rectangle and Ellipse	191
Sizing and Placing Shapes	193
Line	196

Polyline	197
Polygon	198
Line Caps and Line Joins	200
Dashes	202
Paths and Geometries	204
Line, Rectangle, and Ellipse Geometries	205
Combining Shapes with GeometryGroup	206
Curves and Lines with PathGeometry	208
The Geometry Mini-Language	213
Clipping with Geometry	215
Exporting Clip Art	217
Expression Design	217
Conversion	219
Save or Print to XPS	219
The Last Word	222

CHAPTER 8 **Brushes and Transforms**

Brushes	223
The LinearGradientBrush	224
The RadialGradientBrush	227
The ImageBrush	229
Transparency	230
Opacity Masks	232
Making the Silverlight Control Transparent	234
Transforms	237
Transforming Shapes	239
Transforms and Layout Containers	241
A Reflection Effect	242
The Last Word	244

CHAPTER 9 **Animation**

Understanding Silverlight Animation	246
The Rules of Animation	246
Creating Simple Animations	247
The Animation Class	247
The Storyboard Class	248
Starting an Animation with an Event Trigger	249
Starting an Animation with Code	250
Configuring Animation Properties	251

Animation Lifetime	254
Simultaneous Animations	256
Controlling Playback	257
Desired Frame Rate	259
Animation Types Revisited	260
Animating Transforms	261
Animating Brushes	263
Key Frame Animation	265
Animations in Code	269
The Main Page	270
The Bomb User Control	272
Dropping the Bombs	273
Intercepting a Bomb	276
Counting Bombs and Cleaning Up	278
Encapsulating Animations	280
Page Transitions	280
The Base Class	281
The Wipe Transition	283
Frame-Based Animation	285
The Last Word	289

CHAPTER 10 Sound, Video, and Deep Zoom 291

Supported File Types	291
The MediaElement	292
Controlling Playback	293
Handling Errors	294
Playing Multiple Sounds	294
Changing Volume, Balance, and Position	295
Playing Video	299
Progressive Downloading and Streaming	299
Client-Side Playlists	303
Server-Side Playlists	303
Advanced Video Playback	304
Video Encoding	304
Markers	310
VideoBrush	316
Video Effects	317

Deep Zoom	323
Creating a Deep Zoom Image Set	325
Using a Deep Zoom Image Set in Silverlight	328
The Last Word	331

CHAPTER 11 Styles, Templates, and Custom Controls 333

Styles	333
Defining a Style	334
Applying a Style	335
Organizing Styles	336
Template Basics	337
Creating a Template	338
The Parts and States Model	345
Understanding States with the Button Control	347
Showing a Focus Cue	350
Transitions	352
Understanding Parts with the Slider Control	357
Creating Templates for Custom Controls	361
Planning the Expander	362
Creating the Solution	363
Starting the Expander Class	363
Adding the Default Style with Generic.xaml	365
Choosing Parts and States	367
Starting the Default Control Template	369
The Expand or Collapse Button	370
Defining the State Animations	371
Wiring Up the Elements in the Template	372
Using the Expander	375
Using a Different Control Template	376
The Last Word	378

CHAPTER 12 Browser Integration 379

Interacting with HTML Elements	380
Getting Browser Information	381
The HTML Window	382
Inspecting the HTML Document	384
Manipulating an HTML Element	387
Handling JavaScript Events	391

Code Interaction	393
Calling Browser Script from Silverlight	394
Calling Silverlight Methods from the Browser	394
Instantiating Silverlight Objects in the Browser	397
A Browser History Example	398
Combining Silverlight and HTML Content	404
Sizing the Silverlight Control to Fit Its Content	404
Placing the Silverlight Control Next to an HTML Element	406
The Last Word	408

CHAPTER 13 ASP.NET and Web Services 409

ASP.NET Controls That Use Silverlight	410
The Silverlight Control	411
The MediaPlayer Control	414
Custom Controls	420
Hybrid Pages	420
Building Web Services for Silverlight	423
Creating a Web Service	423
Consuming a Web Service	425
Data Types	428
Cross-Domain Web Service Calls	429
Using ASP.NET Platform Services	431
The Last Word	433

CHAPTER 14 Data Binding 435

Binding to Data Objects	436
Building a Data Object	436
Displaying a Data Object with DataContext	438
Storing a Data Object As a Resource	440
Editing with Two-Way Bindings	441
Validation	442
Change Notification	444
Building a Data Service	445
Calling the Data Service	448
Binding to a Collection of Objects	450
Displaying and Editing Collection Items	450
Inserting and Removing Collection Items	454
Binding to a LINQ Expression	456
Master-Details Display	458

Data Conversion	461
Formatting Strings with a Value Converter	462
Creating Objects with a Value Converter	465
Applying Conditional Formatting	468
Data Templates	470
Separating and Reusing Templates	472
More Advanced Templates	473
Changing Item Layout	474
The DataGrid	476
Creating a Simple Grid	476
Resizing and Rearranging Columns	478
Defining Columns	479
Formatting and Styling Columns	482
Formatting Rows	484
Row Details	486
Selection	487
Sorting Rows	487
Freezing Columns	488
The Last Word	489

CHAPTER 15 **Isolated Storage**

Understanding Isolated Storage	491
The Scope of Isolated Storage	491
What to Put in Isolated Storage	492
Using Isolated Storage	493
File Management	493
Writing and Reading Data	494
Requesting More Space	495
Storing Objects with the XmlSerializer	498
Storing Application Settings	501
Reading Local Files with OpenFileDialog	502
The Last Word	504

CHAPTER 16 **Multithreading**

Understanding Multithreading	505
The Goals of Multithreading	506
The DispatcherTimer	507

The Thread Class	507
Marshalling Code to the User Interface Thread	509
Creating a Thread Wrapper	511
Creating the Worker Class	513
Using the Thread Wrapper	515
Cancellation Support	517
The BackgroundWorker	519
Creating the BackgroundWorker	519
Running the BackgroundWorker	520
Tracking Progress	523
Supporting Cancellation	525
The Last Word	527

■ CHAPTER 17 Networking

Interacting with the Web	529
Cross-Domain Access	530
HTML Scraping	532
REST and Other Simple Web Services	538
Processing Different Types of Data	543
XML	543
Services That Return XML Data	544
Services That Require XML Data	551
Services That Return SOAP Data	554
Services That Return JSON Data	554
RSS	558
Sockets	560
Understanding Sockets and TCP	561
The Policy Server	563
The Messaging Server	567
The Messenger Client	571
The Last Word	576

■ INDEX

577

About the Author

■ **MATTHEW MACDONALD** is an author, educator, and Microsoft MVP in Windows Client Development. He's a regular contributor to programming journals and the author of more than a dozen books about .NET programming, including *Pro WPF with VB 2008* (Apress, 2008), *Pro .NET 2.0 Windows Forms and Custom Controls in VB 2005* (Apress, 2005), and *Pro Silverlight 2 in C# 2008* (Apress, 2008). He lives in Toronto with his wife and two daughters.

About the Technical Reviewer

■ **FABIO CLAUDIO FERRACCHIATI** is a prolific writer on cutting-edge technologies and has contributed to more than a dozen books on .NET, C#, Visual Basic, and ASP.NET. He is a .NET MCSD and lives in Milan, Italy.

Acknowledgments

No author can complete a book without a small army of helpful individuals. I'm deeply indebted to the whole Apress team, including Sofia Marchant and Ellie Fountain, who shepherded this book through its many stages; Ami Knox, who speedily performed the copy edit; Fabio Ferracchiati, who hunted down errors in tech review; and many other individuals who worked behind the scenes indexing pages, drawing figures, and proofreading the final copy. I also owe a special thanks to Gary Cornell, who always offers invaluable advice about projects and the publishing world.

Finally, I'd never write any book without the support of my wife and these special individuals: Nora, Razia, Paul, and Hamid. Thanks everyone!

Introduction

Silverlight is a framework for building rich, browser-hosted applications that run on a variety of operating systems. Silverlight works its magic through a *browser plug-in*. When you surf to a web page that includes some Silverlight content, this browser plug-in runs, executes the code, and renders that content in a specifically designated region of the page. The important part is that the Silverlight plug-in provides a far richer environment than the traditional blend of HTML and JavaScript that powers ordinary web pages. Used carefully and artfully, you can create Silverlight pages that have interactive graphics, use vector animations, and play video and sound files.

If this all sounds eerily familiar, it's because the same trick has been tried before. Several other technologies use a plug-in to stretch the bounds of the browser, including Java, ActiveX, Shockwave, and (most successfully) Adobe Flash. Although all these alternatives are still in use, none of them has become the single, dominant platform for rich web development. Many of them suffer from a number of problems, including installation headaches, poor development tools, and insufficient compatibility with the full range of browsers and operating systems. The only technology that's been able to avoid these pitfalls is Flash, which boasts excellent cross-platform support and widespread adoption. However, Flash has only recently evolved from a spunky multimedia player into a set of dynamic programming tools. It still offers far less than a modern programming environment like .NET.

That's where Silverlight fits into the picture. Silverlight aims to combine the raw power and cross-platform support of Flash with a first-class programming platform that incorporates the fundamental concepts of .NET. At the moment, Flash has the edge over Silverlight because of its widespread adoption and its maturity. However, Silverlight boasts a few architectural features that Flash can't match—most importantly, the fact that it's based on a scaled-down version of .NET's common language runtime (CLR) and allows developers to write client-side code using pure VB.

Understanding Silverlight

Silverlight uses a familiar technique to go beyond the capabilities of standard web pages: a lightweight browser plug-in.

The advantage of the plug-in model is that the user needs to install just a single component to see content created by a range of different people and companies. Installing the plug-in requires a small download and forces the user to confirm the operation in at least one security dialog box. It takes a short but definite amount of time, and it's an obvious inconvenience. However, once the plug-in is installed, the browser can process any content that uses the plug-in seamlessly, with no further prompting.

Note Silverlight is designed to overcome the limitations of ordinary HTML to allow developers to create more graphical and interactive applications. However, Silverlight isn't a way for developers to break out of the browser's security sandbox. For the most part, Silverlight applications are limited in equivalent ways to ordinary web pages. For example, a Silverlight application is allowed to create and access files, but only those files that are stored in a special walled-off *isolated storage* area (described in Chapter 15). Conceptually, isolated storage works like the cookies in an ordinary web page. Files are separated by website and the current user, and size is limited.

Figure 1 shows two views of a page with Silverlight content. On the top is the page you'll see if you *don't* have the Silverlight plug-in installed. At this point, you can click the Get Microsoft Silverlight picture to be taken to Microsoft's website, where you'll be prompted to install the plug-in and then sent back to the original page. On the bottom is the page you'll see once the Silverlight plug-in is installed.

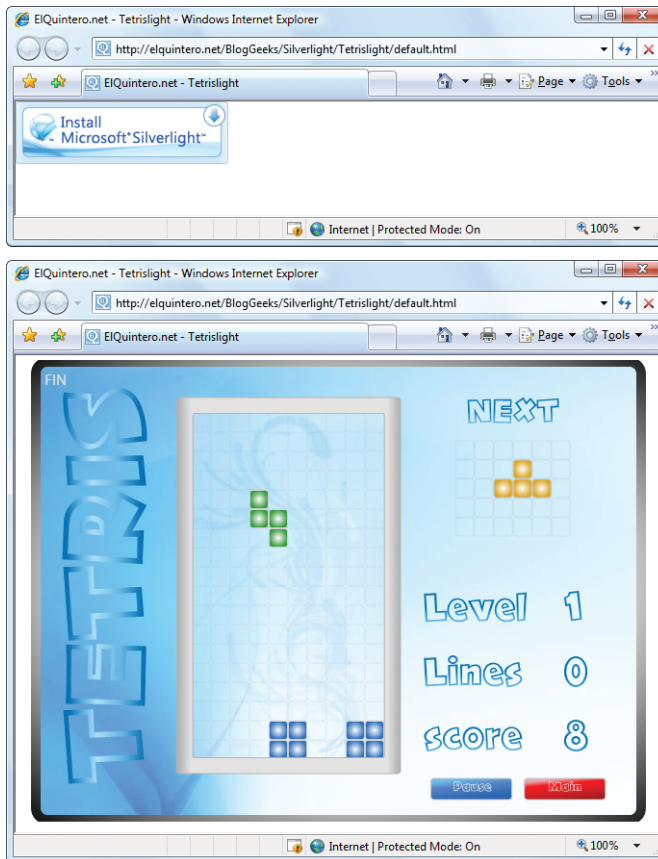


Figure 1. *Installing the Silverlight plug-in*

Silverlight System Requirements

With any Web-centric technology, it's keenly important to have compatibility with the widest possible range of computers and devices. Although Silverlight is still evolving, it already stacks up fairly well in this department:

- **Windows computers.** Silverlight 2 works on PCs with Windows Vista and Windows XP. The minimum browser versions that Silverlight 2 supports are Internet Explorer 6 and Firefox 1.5. Silverlight 2 will also work in Windows 2000, but only with Internet Explorer 6.
- **Mac computers.** Silverlight works on Mac computers with OS X 10.4.8 or later, provided they have Intel hardware (as opposed to the older PowerPC hardware). The minimum browser versions that Silverlight 2 supports are Firefox 1.5 and Safari.
- **Linux computers.** Although Silverlight 2 doesn't currently work on Linux, the Mono team is creating an open-source Linux implementation of Silverlight 1 and Silverlight 2. This project is known as Moonlight, and it's being developed with key support from Microsoft. To learn more, visit <http://www.mono-project.com/Moonlight>.

■ **Note** The system requirements for Silverlight may change as Microsoft releases plug-ins for other browsers. For example, the Opera browser currently works on PCs through an unsupported hack, but better support is planned in the future. To see the latest system requirements, check <http://www.microsoft.com/silverlight/resources/install.aspx>.

Installing Silverlight requires a small-sized setup (less than 5 MB) that's easy to download. That allows it to provide an all-important "frictionless" setup experience, much like Flash (but quite different from Java).

Silverlight vs. Flash

The most successful browser plug-in is Adobe Flash, which is installed on over 90 percent of the world's web browsers. Flash has a long history that spans more than ten years, beginning as a straightforward tool for adding animated graphics and gradually evolving into a platform for developing interactive content.

It's perfectly reasonable for .NET developers to create websites that use Flash content. However, doing so requires a separate design tool, and a completely different programming language (ActionScript) and programming environment (Flex). Furthermore, there's no straightforward way to integrate Flash content with server-side .NET code. For example, creating Flash applications that call .NET components is awkward at best. Using server-side .NET code to render Flash content (for example, a custom ASP.NET control that spits out a Flash content region) is far more difficult.

Note There are some third-party solutions that help break down the barrier between ASP.NET and Flash. One example is the innovative SWFSource.NET (<http://www.activehead.com/SWFSource.aspx>), which provides a set of .NET classes that allow you to dynamically generate Flash (.swf) files. However, these tools work at a relatively low level. They fall far short of a full development platform.

Silverlight aims to give .NET developers a better option for creating rich web content. Silverlight provides a browser plug-in with many similar features to Flash, but one that's designed from the ground up for .NET. Silverlight natively supports the VB language and embraces a range of .NET concepts. As a result, developers can write client-side code for Silverlight in the same language they use for server-side code (such as C# and VB), and use many of the same abstractions (including streams, controls, collections, generics, and LINQ).

The Silverlight plug-in has an impressive list of features, some of which are shared in common with Flash, and a few of which are entirely new and even revolutionary. Here are some highlights:

- **2-D drawing.** Silverlight provides a rich model for 2-D drawing. Best of all, the content you draw is defined as shapes and paths, so you can manipulate this content on the client side. You can even respond to events (like a mouse click on a portion of a graphic), which makes it easy to add interactivity to anything you draw.
- **Controls.** Developers don't want to reinvent the wheel, so Silverlight is stocked with a few essentials, including buttons, text boxes, lists, and a grid. Best of all, these basic building blocks can be restyled with custom visuals if you want all of the functionality but none of the stock look.
- **Animation.** Silverlight has a time-based animation model that lets you define what should happen and how long it should take. The Silverlight plug-in handles the sticky details, like interpolating intermediary values and calculating the frame rate.
- **Media.** Silverlight provides playback of Windows Media Audio (WMA), Windows Media Video (WMV7–9), MP3 audio, and VC-1 (which supports high definition). You aren't tied to the Windows Media Player ActiveX control or browser plug-in—instead, you can create any front-end you want, and you can even show video in full-screen mode. Microsoft also provides a free companion hosting service (at <http://silverlight.live.com>) that gives you space to store media files. Currently, it offers a generous 10 GB.
- **The common language runtime.** Most impressively, Silverlight includes a scaled-down version of the CLR, complete with an essential set of core classes, a garbage collector, a JIT (just-in-time) compiler, support for generics, threading, and so on. In many cases, developers can take code written for the full .NET CLR and use it in a Silverlight application with only moderate changes.

- **Networking.** Silverlight applications can call old-style ASP.NET web services (.asmx) or WCF (Windows Communication Foundation) web services. They can also send manually created XML requests over HTTP and even open direct socket connections for fast two-way communication. This gives developers a great way to combine rich client-side code with secure server-side routines.
- **Data binding.** Although it's not as capable as its big brother, WPF, Silverlight data binding provides a convenient way to display large amounts of data with minimal code. You can pull your data from XML or in-memory objects, giving you the ability to call a web service, receive a collection of objects, and display their data in a web page—often with just a couple of lines of code.

Of course, it's just as important to note what Silverlight *doesn't* include. Silverlight is a new technology that's evolving rapidly, and it's full of stumbling blocks for developers who are used to relying on .NET's rich libraries of prebuilt functionality. Prominent gaps include a lack of database support (there's no ADO.NET), no support for 3-D drawing, no printing, no command model, and few rich controls like trees and menus (although many developers and component companies are building their own). All of these features are available in Windows-centric WPF applications, and they may someday migrate to the Silverlight universe—or not.

Silverlight 1 and Silverlight 2

Silverlight exists in two versions:

- The first version, Silverlight 1, is a relatively modest technology. It includes the 2-D drawing features and the media playback features. However, it doesn't include the CLR engine or support for .NET languages, so any code you write must use JavaScript.
- The second version, Silverlight 2, adds the .NET-powered features that have generated the most developer excitement. It includes the CLR, a subset of .NET Framework classes, and a user interface model based on WPF (as described in the next section, "Silverlight and WPF").

Many developers consider Silverlight 2 to be the *real* first release of the Silverlight platform. It's the only version you'll consider in this book.

Note At present, Silverlight is only on a fraction of computers. However, Microsoft is convinced that if compelling content exists for Silverlight, users will download the plug-in. There are a number of factors that support this argument. Flash grew dramatically in a short space of time, and Microsoft has obvious experience with other web-based applications that have started small and eventually gained wide adoption. (Windows Messenger comes to mind, along with numerous ActiveX plug-ins for tasks ranging from multiuser coordination on MSN Games to Windows verification on MSDN.)

Silverlight and WPF

One of the most interesting aspects of Silverlight is the fact that it borrows the model WPF uses for rich, client-side user interfaces.

WPF is a next-generation technology for creating Windows applications. It was introduced in .NET 3.0 as the successor to Windows Forms. WPF is notable because it not only simplifies development with a powerful set of high-level features, but it also increases performance by rendering everything through the DirectX pipeline. To learn about WPF, you can refer to *Pro WPF in VB 2008* (Apress, 2008).

Silverlight obviously can't duplicate the features of WPF, because many of them rely deeply on the capabilities of the operating system, including Windows-specific display drivers and DirectX technology. However, rather than invent an entirely new set of controls and classes for client-side development, Silverlight uses a subset of the WPF model. If you've had any experience with WPF, you'll be surprised to see how closely Silverlight resembles its big brother. Here are a few common details:

- To define a Silverlight user interface (the collection of elements that makes up a Silverlight content region), you use XAML markup, just as you do with WPF. You can even map data to your display using the same data-binding syntax.
- Silverlight borrows many of the same basic controls from WPF, along with the same styling system (for standardizing and reusing formatting), and a similar templating mechanism (for changing the appearance of standard controls).
- To draw 2-D graphics in Silverlight, you use shapes, paths, transforms, geometries, and brushes, all of which closely match their WPF equivalents.
- Silverlight provides a declarative animation model that's based on storyboards, and works in the same way as WPF's animation system.
- To show video or play audio files, you use the `MediaElement` class, as you do in WPF.

Microsoft has made no secret about its intention to continue to expand the capabilities of Silverlight by drawing from the full WPF model. In future Silverlight releases, you're likely to find that Silverlight borrows more and more features from WPF. This trend is already on display with the shift from Silverlight 1 to Silverlight 2.

Note WPF is not completely cut off from the easy deployment world of the Web. WPF allows developers to create browser-hosted applications called XBAPs (XAML Browser Applications). These applications are downloaded seamlessly, cached locally, and run directly inside the browser window, all without security prompts. However, although XBAPs run in Internet Explorer and Firefox, they are still a Windows-only technology, unlike Silverlight.

THE LIMITATIONS OF SILVERLIGHT

Silverlight compares well to any browser-based technology, with a full suite of modern features and some remarkable innovations. However, Silverlight can't offer all the power of a dedicated rich client technology like WPF, which is designed explicitly for the Windows platform and the DirectX libraries.

Here are some of the WPF features that you *won't* get in Silverlight—at least not now:

- **3-D graphics.** You can draw 3-D shapes using the 2-D drawing primitives that Silverlight offers. However, that leaves you with a lot of custom code to write and a huge amount of math to crunch. True 3-D drawing support, like that offered in WPF, takes care of issues like rotation, lighting, occlusion, and hit testing.
- **Hardware acceleration.** Silverlight will never reach the blistering speed of WPF, because it's designed for widespread compatibility, not native hardware. However, its performance is still impressive, and it offers a serious challenge to other browser-based technologies, like Flash.
- **Documents.** WPF has a rich flow model for showing large amounts of text content, with intelligent line breaking and justification algorithms. Silverlight doesn't.
- **Printing.** Silverlight doesn't provide any way for you to print with the client's printer.
- **Commands.** WPF uses a command model that allows you to define higher-level tasks that can be wired to different user interface controls. Silverlight doesn't include this abstraction—although you could build your own.
- **Triggers.** Silverlight control templates are vastly different than WPF control templates, because they don't support *triggers*, a tool for declaratively mapping events to state changes and animations. The solution is something called the Visual State Manager, which you'll study in Chapter 11.
- **Styles.** *Styles* are a way of reusing formatting on multiple elements. Silverlight supports styles, but in a limited fashion. Notably, it doesn't let you change styles after applying them (which limits some potential designs for skinnable applications), and it doesn't include a mechanism for applying styles to certain types of elements automatically.
- **Custom routed events.** Silverlight supports the concept of routed events—events that occur in one element and then bubble up the element hierarchy, giving you the opportunity to handle them in a containing element (as you'll see in Chapter 4). However, Silverlight imposes severe restrictions, including preventing you from using routed events in your own custom controls.
- **Offline mode.** Silverlight applications are downloaded to the client and executed in the browser. Although this model lends itself to the possibility of caching applications on the client's hard drive and executing them later, perhaps even outside the browser, Silverlight doesn't include this feature.

Expect to see at least some of these features appear in future versions of Silverlight.

About This Book

This book is an in-depth exploration of Silverlight for professional developers who know the .NET platform, the VB language, and the Visual Studio development environment.

Tip Previous experience with WPF—the Windows-based big brother of Silverlight—isn't required. However, if you've programmed with WPF before, you'll breeze through many of Silverlight basics. When useful, this book points out the key differences between Silverlight and the WPF platform.

What You Need to Use This Book

In order to *run* Silverlight applications, you simply need the Silverlight browser plug-in, which is available at <http://silverlight.net>. In order to *create* Silverlight applications (and open the sample projects included with this book), you need Visual Studio 2008. You'll also need the Visual Studio extensions that allow you to create Silverlight projects (known as the Silverlight Tools for Visual Studio), which are available at <http://silverlight.net/GetStarted>. The Silverlight Tools for Visual Studio include both the Silverlight 2 runtime and the Silverlight 2 SDK, so a single download is all you need.

There's one other option. Instead of using any version of Visual Studio, you can use Expression Blend 2.5—a graphically oriented design tool—to build and test Silverlight applications. Overall, Expression Blend is intended for graphic designers who spend their time creating serious eye candy, while Visual Studio is ideal for code-heavy application programmers. This book assumes you're using Visual Studio. If you'd like to learn more about Expression Blend, you can consult one of many dedicated books on the subject.

Code Samples

It's a good idea to check the Apress website or <http://www.prosetech.com> to download the up-to-date code samples. You'll need to do this to test most of the more sophisticated code examples described in this book because the less significant details are usually left out. This book focuses on the most important sections so that you don't need to wade through needless extra pages to understand a concept.

To download the source code, surf to <http://www.prosetech.com> and look for the page for this book.

Feedback

This book has the ambitious goal of being the best tutorial and reference for programming Silverlight. Toward that end, your comments and suggestions are extremely helpful. You can send complaints, adulation, and everything in between directly to apress@prosetech.com. I can't solve your Silverlight problems or critique your code, but I will benefit from information about what this book did right and wrong (or what it may have done in an utterly confusing way).

The Last Word

As you've seen, Silverlight is a .NET-based Flash competitor. It aims to compete with Flash today, but provide a path to far more features in the future. Unlike the Flash development model, which is limited in several ways due to how it's evolved over the years, Silverlight is a starting-from-scratch attempt that's thoroughly based on .NET and WPF, and will therefore allow .NET developers to be far more productive. In many ways, Silverlight is the culmination of two trends: the drive to extend web pages to incorporate more and more rich-client features, and the drive to give the .NET Framework a broader reach. It's also a new direction that will only get more interesting in the months ahead.

