

Laurent Bugnion



**Full Color**

Code samples  
appear as they do  
in Visual Studio  
and Expression  
Blend!

# Silverlight 2

## UNLEASHED

**SAMS**

## **Silverlight 2 Unleashed**

Copyright © 2009 by Pearson Education, Inc.

All rights reserved. No part of this book shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission from the publisher. No patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained herein.

This material may be distributed only subject to the terms and conditions set forth in the Open Publication License, v1.0 or later (the latest version is presently available at <http://www.opencontent.org/openpub/>).

ISBN-13: 978-0-672-33014-8

ISBN-10: 0-672-330148

Library of Congress Cataloging-in-Publication Data

Bugnion, Laurent.

Silverlight 2.0 unleashed / Laurent Bugnion. – 1st ed.

p. cm.

ISBN 978-0-672-33014-8

1. Silverlight (Electronic resource) 2. Multimedia systems. 3. User interfaces (Computer systems) 4. Application software—Development. I. Title.

QA76.575.B84 2008

006.7—dc22

2008040836

Printed in the United States of America

First Printing October 2008

### **Trademarks**

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Sams Publishing cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

### **Warning and Disclaimer**

Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied. The information provided is on an "as is" basis. The author and the publisher shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book.

### **Bulk Sales**

Sams Publishing offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales. For more information, please contact

#### **U.S. Corporate and Government Sales**

**1-800-382-3419**

**[corpsales@pearsontechgroup.com](mailto:corpsales@pearsontechgroup.com)**

For sales outside of the U.S., please contact

#### **International Sales**

**[international@pearson.com](mailto:international@pearson.com)**

### **Editor-in-Chief**

Karen Gettman

### **Executive Editor**

Neil Rowe

### **Development Editor**

Mark Renfrow

### **Managing Editor**

Patrick Kanouse

### **Project Editor**

Mandie Frank

### **Copy Editor**

Geneil Breeze

### **Indexers**

Tim Wright

Brad Herriman

### **Proofreader**

Deborah Williams

### **Technical Editors**

J. Boyd Nolan

Martha Rotter

### **Publishing Coordinator**

Cindy Teeters

### **Designer**

Gary Adair

### **Compositor**

Bronkella Publishing, LLC

# Foreword

There are two undeniable trends in the software development world. The first is toward web style deployment of applications. The web is quickly maturing as an application delivery platform, and increasingly, web applications are seen as the right solution for many styles of applications. The second is toward richness of applications. User experience is becoming increasingly important; consumer-facing applications attracting and retaining users has never been harder and even small usability improvements often make the difference. In business applications, CIOs are realizing that if they can save even a few seconds on every transaction by making employees more effective they can save millions of dollars.

Not surprisingly, Silverlight 2 sits squarely in this sweetspot.

Silverlight 2 is a web application technology. The Silverlight runtime itself is smaller than the average Britney Spears song and is seamless to install on *any* browser (Internet Explorer, Firefox, Safari, and so on) and on *any* platform (Windows, Mac, Linux). Silverlight applications are small and fast to deploy and have zero impact on the client machine. There is no install or uninstall step for Silverlight applications, and all these applications run in a sandbox that ensure they cannot harm user machines or other applications.

Silverlight 2 enables building great user experiences. Through the power of the convergence around video, vector graphics and a powerful developer framework, Silverlight is the ideal platform for UX. Developers have full expressiveness with the power of the CLR under the hood in Silverlight. The C# and VB languages allow developers to be productive today without learning new languages and development tools. The XAML markup allows designers to use Expression and other design tools to build great transitions and animations so that applications look and feel great. All of this together enables Silverlight applications to deliver more productive experiences to end users helping them get their jobs done faster.

This book unleashes Silverlight for you. Laurent not only has a deep understanding of Silverlight, where it is now and where it is going, but he also deeply understands real world programming concerns. This book will make learning Silverlight fun.

I can't wait to see what you build with it.

*Brad Abrams*  
*Product Unit Manager*  
*Microsoft Corporation*  
<http://blogs.msdn.com;brada>

*September 2008*

# Introduction

With the release of Windows Presentation Foundation (a new graphical user interface framework for Windows desktop applications) in 2006 and of Silverlight in 2008, client application development took a turn for the best. Microsoft boldly decided to abandon some concepts and technologies that had been used since the first release of Windows and to do something new and better. While it sometimes seems difficult to keep up with the pace of change imposed on software developers, this one is really worth it. Microsoft's bet on Silverlight and WPF is huge, and it cannot fail. These technologies represent the future of client application development.

Because it runs on multiple platforms in a web browser plug-in that will soon be available on most of the rich clients accessing the Internet, because it can be deployed as easily as any web content and be served from any web server without additional infrastructure, and because of the rich graphic interfaces it allows to be built and the amazingly easy connectivity to remote services that it offers, Silverlight will be a major player in the world of *rich interactive applications (RIA)*. Silverlight is also a gateway to Windows Presentation Foundation, the client application technology that represents the future of Microsoft Windows programming for desktop computers.

In a World Wide Web where Adobe Flash currently has a leading edge, Silverlight represents much more than just an alternative: It is the .NET way! Every .NET programmer will feel at home with Silverlight, because the libraries, the programming languages (C#, VB.NET, Ruby, Python), and the development environment (Visual Studio, Expression Studio) are the same. In addition, new concepts developed and refined in Windows Presentation Foundation are made available to Silverlight programmers, such as data binding, separation of behavior and looks, lookless controls that can be styled and templated at will in powerful design tools such as Expression Blend, a rich animation system, media integration, and so on. XAML, the new XML-based Application Markup Language developed by Microsoft, can be leveraged as a bridge between developers and designers to enable new workflows.

This book is not and was never intended to be a complete reference of the Silverlight platform. Honestly, I am not even sure that you need a book for this: The Internet is at your disposal and has a better, more complete, and more actual reference base than any book can ever offer. No, this book is here to help you discover why programming is fun and why Silverlight is even more fun, and to contaminate you with the Silverlight virus. Complex concepts are explained in simple terms, with many hands-on demos and figures so that beginners as well as advanced developers quickly will feel at home.

## About Code in This Book

We tried to keep formatting as consistent as possible throughout the book and to make the code look like it does in Visual Studio. The code is color coded to help you work faster

and recognize key concepts in XAML, C#, JavaScript, and HTML in Studio and in Expression Blend.

The source code lines are only numbered where it is relevant, for example, when the text makes explicit reference to a line number.

The whole source code for this book is available online at [www.galasoft.ch/SL2U/Code](http://www.galasoft.ch/SL2U/Code). For C# code, a translation in VB.NET is also available, courtesy of this book's technical editor, J. Boyd Nolan.

## One Year Older

I started working on this book in September 2007, and I am now exactly one year older. Professionally speaking, it has been the most interesting year of my life. Since I started working as a developer in 1996, I have worked with many client technologies and programming languages, including C, VB, Java, HTML, CSS, JavaScript, ASP.NET, Windows Forms and finally Windows Presentation Foundation and Silverlight. In all these years, I have never been as excited about a new programming platform. Writing a book is hard, and it's a lot of work. But it was also fun and so interesting that I always felt right doing it. If I had to do it again, I would sign without hesitation. And now that it's going to be published, I can't wait to see what you, the reader, will create in Silverlight. Software has much to do with art, and Silverlight is the richest palette you can imagine. So grab the book and your computer, start coding and designing, and show the world what you can do. I will be waiting.

Happy coding!

Laurent

# CHAPTER 1

## Introducing Silverlight

It all started when Microsoft presented its revolutionary user interface (UI) framework, *Windows Presentation Foundation*, to an enthusiastic crowd of graphics designers, software developers, and businessmen in March 2006 at the new MIX conference in Las Vegas. Microsoft also added one session about a lesser-known technology with the rather barbarian name *Windows Presentation Foundation Everywhere, or WPF/E*. There was nothing much to see yet, but the abstract was enticing: “With WPF/E you’ll be able to build rich, interactive experiences that run in major Web browsers on major platforms as well as on mobile devices.”

A little more than a year later, at the second edition of the same MIX conference, Scott Guthrie (general manager at Microsoft, responsible for most of the .NET teams) climbed on stage and gave the crowd an amazing software demonstration. The barbarian WPF/E was gone; in its place was Silverlight (see Figure 1.1).

A bright new logo revolved on the screens. Gradients and animations were all over the place. Planes flew over the web browser’s window, connecting US cities while Scott was planning his next trips; a chess application let the browser’s JavaScript engine play against .NET, demonstrating without any doubt the superior power of the compiled .NET application over JavaScript’s interpreted code. Yes, the browser was running .NET!—not only in Internet Explorer but also on Firefox! And yes, even on a Macintosh computer.

### IN THIS CHAPTER

- ▶ **Where Does Silverlight Come From?**
- ▶ **Using Third-Party Plug-Ins**
- ▶ **Running on Multiple Platforms**
- ▶ **Making the Web Application Secure**
- ▶ **Introducing Silverlight.net**
- ▶ **What Do You Need to Run Silverlight?**
- ▶ **Updating Your Runtime—Automatically**
- ▶ **Trying Silverlight Demos**
- ▶ **What Do You Need to Develop Silverlight?**
- ▶ **Reading the Documentation**
- ▶ **Looking into Silverlight’s Future**

It was an intense hour! Later that day, more demos of the new Expression Encoder were presented, which allows users to produce movies and run them in the web browser, after having copied them to a streaming (or to a nonstreaming) web server. Even better, with the Expression Encoder you can now change the look and feel of the video player completely, basing your work on predefined templates, but also changing everything from colors to shapes in Expression Blend.

These first demos were only the start of an amazing three days packed with Silverlight content. In a little less than a year, Silverlight had made it from little-known side attraction to major technology with a huge potential.



FIGURE 1.1 Microsoft Silverlight logo

### NOTE

All the sessions presented at MIX 2006, 2007 and 2008 can be viewed online at <http://sessions.visitmix.com>.

## Where Does Silverlight Come From?

A little history can be interesting to understand where Silverlight came from, and how it is positioned among the many UI technologies available today. Depending on your past experience, you may be well aware of the facts presented in the next sections. In that case, feel free to skip ahead (the next interesting section for you will probably be “Running on Multiple Platforms”). If you don’t want to skip ahead, or if you feel like refreshing your memory, keep reading as this section and the next few sections fast forward from the antiquity of the World Wide Web to the magic of today.

Web pages have been around for a long time. In fact, HTML was presented to the world as early as 1992. (You know the old line about dog years being equal to seven human years? Well, 1992 in computer years is, like, Middle Age for humans!) Of course, HTML was limited in the beginning. Web pages were simple, composed mostly of text and hypertext (the most revolutionary idea of HTML, and the origin of the name *Hypertext Markup Language*).

Very quickly, however, the inventors of HTML saw the need for richer content to be made available in web pages. The first step in that direction was the invention of the IMG tag, allowing images to be placed inline. The basis for richer content was there, and the first popular web browser, Mosaic, displayed these pages.

### Automating Web Pages with JavaScript

The next big step forward in the quest for rich content was the addition of a JavaScript engine running in the web browser.

This programming language (added in 1995 to the Netscape web browser, and then in 1996 in Microsoft Internet Explorer) enabled a much richer interaction with the user. Earlier, the only possible user action was through a *post-back* to the web server, meaning that any data entered by the user had to be sent back to the server for handling, and the result was returned as a *response*. Obviously, this model was slow, especially at the time of dial-up connections.

With JavaScript, it was possible to provide a basic treatment of data on the client directly, without postback. A first obvious use for this client-side technology was a prevalidation of data, making sure that the user was not sending useless information to the server, wasting bandwidth and valuable server time. With JavaScript enabled, errors could be caught early, and faulty data would never leave the client. Of course, as soon as a web server interaction was needed, for example, to retrieve data, for complex calculations, to authenticate users, and so on, a postback was needed. This model still prevails in many websites even today.

## Cascading Style Sheets

Another big step forward in the creation of rich web content has been the creation of *Cascading Style Sheets* (CSS). Finally a separation of content and layout was made possible. The look and feel of a web page could be extracted from the page itself and governed by the use of rules placed in external files. This model has many advantages:

- ▶ The responsibilities can easily be shared by various teams, depending on their skills. Graphics designers can be in charge of the look, developers of the functionality, while marketing is in charge of the content.
- ▶ In smaller projects, the developer can concentrate on the page's structure and functionality first, and create the look later, after the content is fully known. This way of working is much more efficient than doing everything at the same time.
- ▶ The page's look can be changed easily without touching the page itself.
- ▶ Same styles can be reused over and over again without having to rewrite the code. Classes can be located in a central location and *referenced* anywhere. In fact, it's even possible to have styles on a central server and be used on other web servers, related or not.

As CSS became more and more sophisticated, the pages became more and more beautiful, to a point where the aspect of the pages, in some cases, becomes as important as the

### NOTE

There are many names for this scripting language, from Netscape's JavaScript to Microsoft's JScript to the standard ECMAScript. In this book, the term "JavaScript" will be used without distinction.

### WARNING

Client-side validation can easily be side tracked, and it is *always* necessary to validate data on the server too!

content itself. One can argue endlessly whether that's a good thing (after all, content is what users are really looking for, isn't it?), but if you can choose between boring content and beautiful content, surely the beauty will make the experience more pleasurable.

## WARNING

Although CSS allows creating beautiful pages, it also allows creating ugly looks. Unfortunately, the technology has no taste; only the developer/designer has some (or hasn't any). That's true for Silverlight (and any UI technology) as well.

## Progressing to DHTML

As the JavaScript and CSS engines became more advanced, some interaction was made possible between them. *DHTML (Dynamic HTML)* was born. Actually, DHTML is not a technology; it's a combination of JavaScript and CSS used in the context of HTML.

Using JavaScript as a programming language, it is possible to access elements of the page itself (content and style), and to modify them. Since this is client-side technology, no interaction with the server is required for this kind of effect.

Because CSS allows the setting of an element's opacity, it is possible to create fade-in/fade-out effects. With the possibility to set an element's position in an absolute manner, JavaScript allows you to move an element across the screen. In addition to "just" adding some kind of intelligence and good looks to the web pages, the interaction between CSS and JavaScript enabled the creation of richer, smoother web pages.

Although this all sounds nice, DHTML is rather limited:

- ▶ Opacity is not supported in all browsers, and the ones that support it don't use a standard syntax, making it kind of a maintenance nightmare.
- ▶ Moving elements across the screen can only be done at limited speed and is not as smooth as you would want.
- ▶ It is impossible to rotate elements. If you want to show a rotated button, you need to simulate this using multiple images, and the effect will not be smooth.
- ▶ Text (such as a title, in a logo, etc...) can only be rotated by using images. In that case, the localization becomes a nightmare, because you need to translate each image in each language you want to support.
- ▶ The "hit test area" for any element is always square. In other words, even if you have an image of a round button with a transparent background, you may activate the action even if you click outside the round area. The mouse cursor will turn to a hand even though the mouse is still on the transparent area.

Thankfully, Silverlight corrects all these shortcomings, and since it can be "blended" among HTML elements, it offers a real possibility to graphically enrich a web page.

## Communicating in a Richer Way with AJAX

In addition to enabling DHTML, JavaScript brought a huge improvement to client-server communication, and to creating a smoother user experience: *Asynchronous JavaScript And XML (AJAX)*, a cumbersome name for a great technology. In short, AJAX allows JavaScript to send a request to the web server, and to get the response without the web page being refreshed. Instead of using the normal post-back mechanism, the whole communication occurs in the background, and the user doesn't notice it.

With this technology, the user can get additional information from the server without losing the context he is working in. The user experience is also much smoother, because new information gets loaded without the “traumatic” reloading of the page and the temporary white screen caused by it.

### TIP

AJAX is not very easy to program. Like you will see later, Silverlight 2 offers a great deal of improvement in this area too and makes background communication much easier.

## Using Third-Party Plug-Ins

With the limitations of DHTML, various third-party additions to the web browser have been created with the explicit intent to enhance the user experience. The most notorious technologies are probably Java applets, ActiveX controls, and Flash applications. This section reviews the advantages and shortcomings of these technologies to understand better how Silverlight compares.

### Using Java Applets

Java applets were extraordinarily popular for a short time and then became unpopular. While some web pages still use the Java technology, their number is decreasing. The main problem with Java is that the runtime is very slow to start. When you load a page with a Java applet on it, the initial loading time makes it a painful experience.

On the plus side, Java is a great programming language. It paved the way for .NET, and many of the best features of C# are greatly inspired by the Java language. It also enables advanced graphical effects, such as the famous ripple effect.

Limited Java-to-JavaScript and JavaScript-to-Java communication is possible. However, the interface used for this communication is not supported by all the web browsers that Java runs into. Also, the syntax is not easy to learn, and communication is awkward.

### NOTE

An annoying shortcoming of all the technologies described in this section is that it is impossible to place HTML content in front of a third-party plug-in. The Java applet, ActiveX control, or Flash application will always be in the top level. Silverlight corrects this problem and makes it possible to have HTML content blending with Silverlight content.

Also, the HTML–JavaScript–Silverlight interaction is better than with any of these older technologies. The boundaries are thinner than ever.

Java makes it possible to open alternative communication ways to the web server. For example, it enables the server to “talk” directly to the client, which is impossible with classic web communication. While this can be a huge technological advantage in certain scenarios, it can also, under certain circumstances, represent a security threat. Because of this, the acceptance of Java applets by big, security-sensitive corporations has been slow and has pretty much “killed” Java in the web browser.

### NOTE

Every book about web technology should make one thing clear: Java and JavaScript have strictly no relationship. When you program JavaScript, you do *not* program Java.

## Using ActiveX Controls

When Microsoft introduced the COM technology in 1993, it also made it possible to create so-called ActiveX controls, “packing” pieces of software and communicating with the external world using the COM interfaces. With this, it is possible to develop controls using classic Windows technology (including fast C++ code), and to place it in a web browser. It is even possible to have some limited interaction with JavaScript and through this to the web page.

The major shortcoming of ActiveX is that it uses an obsolete, almost 15-year-old technology. Also, ActiveX controls can only be used in Internet Explorer on Windows.

## Using Flash Applications

Adobe Flash is probably the most popular third-party plug-in currently available. On the plus side, it allows advanced graphical effects, and it's not rare to find websites programmed entirely in Flash. While Flash content is often called “Flash movies,” it's probably fairer to call them “applications.”

Flash's major shortcoming is that it is difficult to program. A (commercial) editor is needed to create the front-end user interface. Additionally, the code-behind can only be programmed in a language called ActionScript, which is a subset of JavaScript. The fact that Flash has been created primarily for graphics designers makes it difficult for software developers to use the tools and to create compelling content and functionality.

### WARNING

Programming a website entirely in Flash (or in Silverlight) is not a good idea. Many platforms (especially mobile phones, PDAs, and so on) do not allow accessing such sites (or only in a limited manner). This type of technology should only be used to enhance the content of a website, not to replace it completely.

### NOTE

Microsoft doesn't advertise Silverlight as a Flash “killer.” This strategy would probably be doomed anyway, considering how many Flash designers there are, and how much Flash content is available online. Since both technologies can run safely side-by-side, there is really no need to start a religion war about which is the best!

A limited interaction between the Flash application and the containing HTML page is possible through JavaScript. Some interaction between the Flash application and the containing HTML page is possible through JavaScript.



## Running on Multiple Platforms

One important factor when you develop a third-party web technology is to run it on as many platforms as possible. The Web is, by definition, ubiquitous, and it is not rare to find Internet-enabled devices of every shape and power. Adobe Flash is a good example of this: The plug-in is available for a great number of browsers on a great number of operating systems. This makes Flash a powerful platform, and Microsoft understands the need for Silverlight to run in multiple browsers too.

As said earlier, at the time of this writing, the Silverlight runtime is available for Internet Explorer and Firefox on Windows, and for Firefox and Safari on Macintosh. A version for Linux is in preparation. The real challenge is to offer consistent interfaces on all the platforms on which Silverlight runs. So far, this goal has been very successful, and is a great achievement in the quest for a unified web environment.

Some prototypes of Silverlight for smartphones have also been presented, but not much is known about what will be supported on these limited devices. A first version of Silverlight for mobile phone (running on Windows Mobile and also on Nokia mobile phones) should be released by the end of 2008. This should support Silverlight 1 applications, including video.

Because Silverlight runs on so many platforms, some limitations in the functionality (compared to the full .NET platform) are unavoidable. Many factors must be considered:

- ▶ The goal is to keep the runtime small, so that it can easily be installed over Internet connections. The goal in the foreseeable future is to keep it under (or equal to) 5MB. When you compare to the size of the full .NET runtime (and even if you remove all the server-side bits), it's obvious that not everything can be implemented.
- ▶ Features requiring a lot of hardware acceleration (especially 3D animations) will probably be too hard to implement consistently on multiple platforms (not even mentioning small limited mobile devices).
- ▶ Various platforms use various underlying graphic technologies, and some of these technologies

### NOTE

In order to run Silverlight on many platforms in a consistent way, Microsoft is also collaborating with Novell and the makers of the open source .NET framework “Mono.”

### TIP

Hardware acceleration is the process through which some of the complex calculations are executed by hardware processors specializing in graphics (as opposed to software). Hardware graphic processors are much faster than anything that software can calculate.

may not allow all the desired effects. It's already pretty amazing to see how consistent the implementation is on both Windows and Macintosh. For Silverlight to be viable, it is imperative that it remain totally compatible on all supported platforms and browsers.

All these reasons create a lot of intense discussions to decide what will make it to Silverlight and what won't. It will be interesting to observe the development of the platform!

### TIP

Microsoft is listening to you! Don't hesitate to contact evangelists and other Silverlight developers, either through the Silverlight forums (<http://silverlight.net/forums>), or through their blogs (check the section titled "Reading Silverlight-related blogs" in Chapter 24, "Silverlight: Continuing the Journey"). Let them know what you do with the technology, and what your expectations are. Just like third-party developers influenced the way WPF is growing, the same process is true for Silverlight! Let them hear your voice!

## Making the Web Application Secure

Security in web applications is a huge challenge. Even though Microsoft has more often than any other company been the target of virulent criticisms because some of their applications or operating systems were not secure enough, it's fair to say that any popular web technology is exposed and has had security issues (Java, Firefox, the Linux operating system, the new Google Chrome web browser and many others have all been the target of attacks).

Microsoft is taking the security of Silverlight-based applications very seriously. Silverlight applications run in a "sandbox" that limits the functionality of the application and protects the computer it runs on against attacks. Every time a feature is allowed by the sandbox, it is carefully tested to make sure that the change doesn't present a hole for a potential attack.

If humanity was intrinsically good, programming would be easier, but of course, security is not just a matter of attackers, it also has a lot to do with bad code, memory corruptions, and so on. Thankfully, the .NET programming platform (the so-called "managed code") is a secure way of programming. Unlike older unmanaged C++ code, many bugs are eliminated already at the source by the much safer managed programming languages. Memory corruption is almost impossible; memory is automatically freed by the *Garbage collector* when it is not used anymore, making memory leaks much more infrequent.

Obviously it would be naive to believe that Silverlight will never be attacked, or that Silverlight applications will never crash. However, thanks to the experience gathered by the .NET teams, thanks to the managed programming languages, and thanks to the

extra attention invested by Microsoft into that matter, Silverlight should provide a very safe environment.

T

## Introducing Silverlight.net

The community website <http://silverlight.net> contains a collection of valuable information about Silverlight, including quickstarts, samples, tutorials, and so on. Maybe the most interesting section is the gallery, available at <http://silverlight.net/community/communitygallery.aspx>.

Here you will find a lot of samples, made by Microsoft and third-party developers. This is a good starting point to get an idea of what Silverlight is capable of. Currently, the gallery is divided into two subsections: Silverlight 1.0 (with JavaScript) and Silverlight 2 (with .NET). However, if you install Silverlight 2 (as we recommend), you can still execute older applications.

## What Do You Need to Run Silverlight?

Silverlight is an add-on to the web browser. It gets installed separately and adds functionality to your web pages. It is currently available for Internet Explorer and Firefox on the Windows XP and Vista operating systems; it is also available for Firefox and Safari on the Macintosh. At the time of this writing, a version for Linux is in preparation, following a historical agreement between Microsoft and Novell.

To run Silverlight applications, you need a compatible web browser. When you point a Silverlight-capable web browser to a web page with Silverlight content, you get a small Install Microsoft Silverlight icon, as shown in Figure 1.2, instead of the Silverlight content.



FIGURE 1.2 The Install Microsoft Silverlight icon

- ▶ Click on the icon to be taken to a Microsoft web page where you can download and install the Silverlight runtime on your PC.
- ▶ After installing Silverlight in Internet Explorer, you don't even need to restart your web browser. In Firefox, you might have to, but it's not that big a deal.

Alternatively, you can install Silverlight 2 from <http://silverlight.net/GetStarted>.

### WARNING

If you worked with Silverlight before and you already installed a version older than the one required by the current application, you will also see the “Install Microsoft Silverlight” button. You can check the Silverlight version by right-clicking on any Silverlight application, choosing the Silverlight Configuration menu, and selecting the About tab.

## Updating Your Runtime—Automatically

Once the runtime is installed, the Silverlight content will automatically be started. Also, a version check will happen when Silverlight is started; if a newer version is available, you can choose to install it automatically. This can be set up in the Silverlight configuration dialog by following these steps:

1. Navigate to <http://silverlight.net/samples/1.0/Page-Turn/default.html>. This starts one of the earlier Silverlight demos created by Microsoft: The PageTurn sample application.
2. Right-click on the Silverlight application and choose Silverlight Configuration.
3. Click on the Updates tab.
4. Choose the options you want (see Figure 1.3).

### NOTE

Silverlight gets installed only once, and then runs in Firefox or IE (on the Mac, on Safari and Firefox) with the same runtime. If you change the options in Firefox, it will be valid in the other browser installed on the same PC too!

### TIP

You can turn pages by clicking and dragging your mouse on the bottom-right corner of the photo album. If you click on the small open book icon on the bottom right, you see a list of thumbnails.

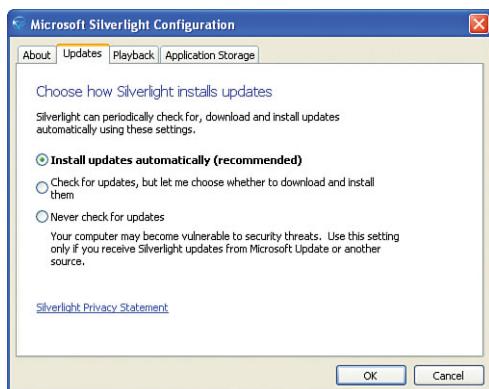


FIGURE 1.3 Silverlight configuration

## Trying Silverlight Demos

The cool thing with Silverlight is that it's so easy to deploy: Simply put the files on a web server, and anyone in the world with a Silverlight-capable browser can see the applications.

The applications developed by Microsoft for demo purposes are also available online. Additionally, third-party developers can also publish their own sample applications to the Silverlight.net website. As mentioned previously Microsoft used two demo applications when they first introduced Silverlight to the public: the Chess application and the Silverlight Airline sample.

## Playing Chess Against Silverlight

At the time of writing, the Chess demo (see Figure 1.4) is not available yet for Silverlight 2. You can, however, see a video illustrating this application at <http://on10.net/blog/tina/Microsoft-Silverlight-Chess>. Also, keep checking the Silverlight.net gallery where the demo and its source code should be posted soon.

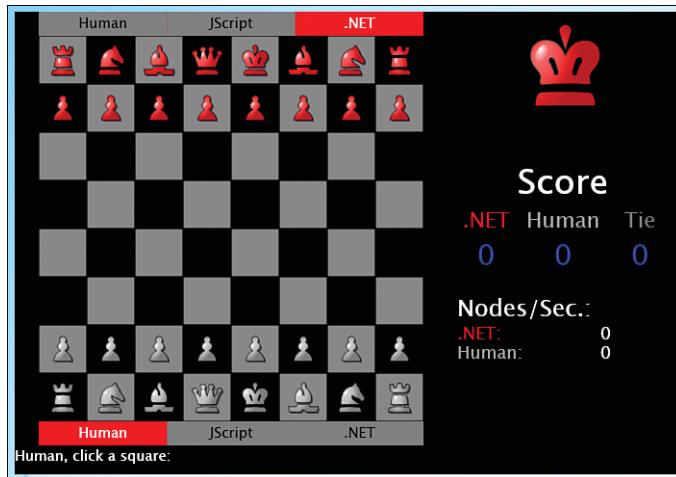


FIGURE 1.4 Chess application

In this application, you get a chance to play against the machine: The Human button is selected on the bottom. On the top, you can choose to play against another human (booooring), against JavaScript (remember, JScript is Microsoft's name for its implementation of the technology) or against .NET.

The implementation for the chess engine is exactly the same in JavaScript and in .NET. So the most interesting thing to do in that application is to let JavaScript play against .NET:

1. On top, make sure that the .NET button is selected.
2. On the bottom, select the JScript button.
3. Observe how both engines are playing against each other. The most interesting data is the number of nodes per second calculated by each engine. This is an indication of how fast each engine can process data. Typically, .NET calculates approximately 1,000 times faster than JavaScript!

Running this application always leads to the same result: .NET wins over JavaScript, because it is just so much faster. This is a good proof (if it was needed) of some of the advantages we will encounter when we use .NET for advanced calculations instead of the JavaScript engine.

## Planning Your Trips with the Airline Application

The airline application shown in Figure 1.5 was also shown on stage at MIX07 and presents what could be an airline reservation system in the (near) future. Here too, the demo is not yet available for Silverlight 2 at the time of writing but should be updated soon in the Silverlight.net gallery. A video showing this application is available at <http://on10.net/blogs/tina/Microsoft-Silverlight-Airlines>.



FIGURE 1.5 Airline application

1. Click and hold the departure city. Then, still holding down the button, move the cursor to the target city. You can release the mouse now.
2. Select a date for your trip in the Calendar on the right.
3. The system calculates a number of possible itineraries for your trip. They are shown under the map. Pass your mouse cursor over one itinerary to see it drawn on the screen, and to see a small plane fly from city to city.

This example is interesting, because strictly speaking, Silverlight doesn't offer new functionality here. Online reservation systems are already available today. However, the user interface is better with Silverlight and makes the user experience better and more attractive.

### NOTE

Thanks to the goodness of vector-based graphics, you can also resize the window and note how the whole user interface gets resized to fit the screen.

## DeepZooming into the Hard Rock Café

At MIX08, probably the most exciting demo was given by the Hard Rock Café and its memorabilia collection of rock souvenirs: How would you like to be able to see high resolution pictures of all the souvenirs normally exposed in various Hard Rock Cafés around the globe (see Figure 1.6)? Even better, how would you like to be able to select them by artist, year, type, and so on? And the most awesome feature of all: Zoom smoothly into the pictures until you are able to see every detail, like small scratches or a handwritten dedication on a guitar! Navigate to <http://memorabilia.hardrock.com> to try it yourself.

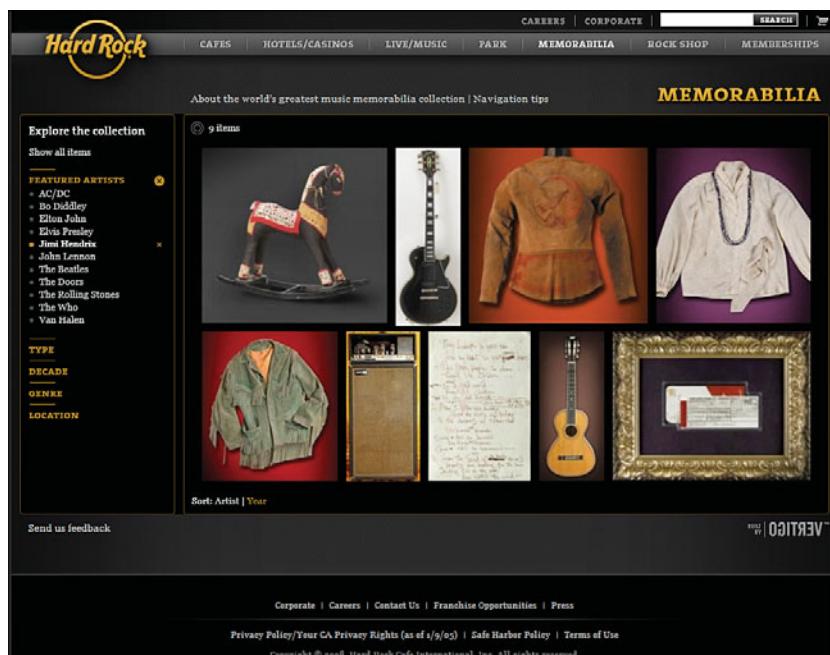


FIGURE 1.6 Hard Rock Memorabilia

This application is enabled by a feature introduced in Silverlight 2, named DeepZoom. A very high resolution picture is prepared by a tool and divided in multiple, lower-resolution pictures. A Silverlight control is loading these images dynamically (according to the level of zoom chosen) and displays them. Additionally, you can pan the image simply by clicking and dragging the mouse, with a lifelike movement. All these seemingly very complex features are made rather easy thanks to Silverlight 2 (though one should not underestimate the value of the developers and designers having created this application!).

## Finding More Demos

Many more samples and demos are available online. Make sure to browse through the Silverlight.net gallery and play with demos to see what Silverlight can do at <http://silverlight.net/community/communitygallery.aspx>.

## What Do You Need to Develop Silverlight?

The term “Silverlight” also qualifies the programming platform, which is developed in parallel to .NET 3.5. Since Silverlight is in development now, not all the .NET classes are available yet. In fact, some of them will never make it to Silverlight, as mentioned earlier. However, because the .NET basis is stable now, we can expect to see big improvements in Silverlight in the coming months as Microsoft releases new versions.

We are not going to create standalone applications here. We will only create applications embedded in a web page and unable to run on their own. The nice thing with this way of programming is that distributing (deploying) your application is as easy as copying some files to your web server. If you are adding Silverlight content to an existing web page, you probably already have a web server. If you don’t, you will need to get web server space eventually to publish your application.

During development, however, you will be able to use the web server IIS built in the Windows operating system. Even better, if you use Expression Blend or Visual Studio 2008, starting a web application is as easy as pressing F5! You learn more about deploying web applications in Chapter 7, “Deploying to a Web Page.”

Developing Silverlight can be done with just a text editor. Notepad (or its Mac equivalent) is an option, seriously. You can write your XAML markup (a new programming language we will introduce later) in there, and then run it in the web browser to see the result. However, developing this way can be painful. Fortunately, we now have the support of great tools to make that experience a real pleasure. The best place to start and install the development environment for Silverlight is from <http://silverlight.net/GetStarted>.

### Expression Studio

When Microsoft announced WPF, it also gave us a new suite of tools named the Expression Studio. Included are three important tools for the Silverlight developer: Expression Design, Expression Blend, and Expression Encoder.

This book uses these tools to create Silverlight content. Expression Blend is studied more in detail in Chapters 4, “Expressing Yourself with Expression Blend,” and 6, “Blending a Little More” (and throughout the book). This great tool is located at the border between designers and developers and is used by both professions. It is an almost unavoidable tool when you create WPF and/or Silverlight content. Unfortunately, at the time of writing there is no free edition of Expression Blend. A fully functional demo version can be downloaded from the Microsoft Expression website at [www.microsoft.com/expression](http://www.microsoft.com/expression).

### WARNING

Even though the Silverlight runtime is available for multiple platforms, the development environment described in this book (programming, designing, compiling, debugging) is available only for the Windows operating system. There is also a development environment for Linux (Moonlight).

### NOTE

Expression Studio also contains Expression Web (used to develop web pages) and Expression Media, a tool used to manage media collections (pictures, videos, and so on).

In addition to Expression Blend, we will be using Expression Encoder in this book. This application can also be downloaded from the Microsoft Expression website.

Expression Design can be used to create XAML-based “assets” (pieces of design). Later, you can import these elements in your Silverlight application. We will talk about this in Chapter 24.

## Visual Studio 2008

Additionally, we will enjoy a better programming tool than Notepad: The powerful Integrated Development Environment (IDE) Visual Studio 2008 is the tool of choice to program Silverlight applications. In this book, we will use mostly Visual Studio 2008 and Expression Blend to program Silverlight applications. A fully functional 90-day demo version can be downloaded at <http://msdn.microsoft.com/en-us/vs2008/products/cc268305.aspx>.

In addition to Visual Studio, you must also install the Silverlight tools available from the “Get Started” page at <http://silverlight.net/GetStarted/>.

You learn more about Visual Studio 2008 in Chapters 9, “Understanding .NET” and 10, “Progressing with .NET” about .NET, and in following chapters, where we will use these tools to create C# code-behind.

Microsoft released so-called Express editions of Visual Studio. Each Express edition presents a limited set of features targeted at a specific type of software development (for example, web applications, C#, VB.NET, and so on). These limited editions are free to download and install. According to Microsoft, you will be able to use these Express editions to program Silverlight. At the time of writing, however, an exact release date was not public.

### TIP

If you are a MSDN subscriber, Expression Blend is included in the premium edition of MSDN, so you don’t need to purchase it.



### NOTE

Even though Silverlight supports VB.NET and other managed languages, the examples in this book are written in C#. I want to personally apologize for this to all the VB.NET developers interested in Silverlight. A translated version of the source code is available on the book’s website at [www.galasoft.ch/SL2U/Code](http://www.galasoft.ch/SL2U/Code).

In addition to C# and VB.NET, Silverlight supports the so-called *dynamic languages* IronRuby and IronPython.

## Reading the Documentation

This book does not contain a full reference documentation about Silverlight but will give you a great head start in this technology. In addition, you will use the Silverlight documentation, available in various forms.

## Browsing the Online Reference

The official Silverlight reference is found on MSDN (Microsoft Developer Network) online. You can start at [http://msdn.microsoft.com/en-us/library/bb404710\(VS.95\).aspx](http://msdn.microsoft.com/en-us/library/bb404710(VS.95).aspx) (for Silverlight 1.0) and [http://msdn.microsoft.com/en-us/library/bb404700\(VS.95\).aspx](http://msdn.microsoft.com/en-us/library/bb404700(VS.95).aspx) (for Silverlight 2).

## Finding More Documentation

With the number of classes available in the .NET framework (and even in the Silverlight subset), and with the number of members that each class contains, a comprehensive documentation is a key factor when you program against this framework. Microsoft released a Silverlight 2 Software Development Kit (SDK). It contains technical documentation, libraries and tools to help you in your Silverlight experience.

1. Navigate to the Get Started page at <http://silverlight.net/GetStarted>
2. Scroll down until you see the link to Microsoft Silverlight 2 SDK. Follow this link and download the SDK setup file.
3. Execute the file “silverlight\_sdk.exe” you just downloaded. This will extract the content to C:\Program Files\Microsoft SDKs\Silverlight\v2.0
4. For more information about the SDK, technical documentation, links to external resources, etc., check the “Welcome” page. You find it in your Start menu, under “All Programs / Microsoft Silverlight 2 SDK”.

The Silverlight documentation is also available as a compiled HTML file that can be installed locally (see Figure 1.7). The download link is on the SDK welcome page.



FIGURE 1.7 Silverlight documentation

The Search tab allows you easy access to a specific topic. The documentation contains many interesting topics, and it is absolutely worth browsing through it.

## Learning with Tutorials

Many tutorial websites are available for WPF, Silverlight, and Blend. The community website Silverlight.net has already been mentioned. Another exciting and innovative site is the Nibbles Tutorials website at [www.nibblestutorials.net](http://www.nibblestutorials.net).

This website does an amazing job of presenting and teaching the technology. Using small, bite-sized tutorials, the Nibbles website allows you to test some of the knowledge acquired in this book.

Additionally, the website's author, Celso Gomes, is a talented graphics designer working for Microsoft, and his tutorials give you the designer's point of view on Silverlight (while this book gives you the developer's point of view).

## WARNING

The “Nibbles” website doesn't display any content if Silverlight is not installed. Install Silverlight before you navigate to this site! Like we mentioned before in this chapter, Silverlight-only websites are not recommended because they are only visible on Silverlight-compatible web browsers which speaks against the principles of the World Wide Web.

## Looking into Silverlight's Future

Web technologies are evolving fast, and any attempt to look into the future and predict where they are heading in the long term will result in mortifying failure for the psychic...Silverlight is no exception. It's almost impossible to predict where the technology will be five or ten years from now.

One thing is sure, though: Microsoft is putting a lot of effort in Silverlight and porting the .NET framework to multiple platforms. Mobile Silverlight will “hit us” in the very close future. And because so many developers are already active on the .NET platform, we will see more and more Silverlight content soon.

It is also safe to write that Silverlight and WPF will become always more compatible with each other. Silverlight 2 is already delivering a great number of WPF-compatible classes. Learning either WPF or Silverlight will offer an easy passage to the other.

## NOTE

At MIX08, in March 2008, Nokia announced a partnership with Microsoft to deliver the Silverlight mobile platform on the Nokia mobile phones.

## Is ASP.NET/HTML/JavaScript Dead?

With the advent of Rich Interactive Applications (RIAs) made with Silverlight or Flash, one can wonder if classic ASP.NET will still be available in the future. One thing is sure: ASP.NET is a very strong and important pillar of Microsoft's client application strategy. In fact, Microsoft also released updates to the ASP.NET framework with its .NET framework 3.5 this year.

With “classic” ASP.NET (HTML/CSS/JavaScript) for “thin clients,” Silverlight for RIAs and WPF for desktop applications, you cover the whole range of client applications. And since all these platforms run .NET, you can use the same programming language of your choice (amongst C#, VB.NET and many others) to create your libraries.

So in short: No, ASP.NET is definitely not dead, and HTML-based web pages will continue to entertain and inform us on the Internet for many years to come.

## Summary

In this chapter, we took a preliminary look at Silverlight and understood where it comes from and what technologies are its “ancestors.” We also started playing with demo applications and installed the development environment.

Now it is time to start getting our hands dirty, and to write code. The next chapters introduce you to the new programming language XAML and give you the occasion to write markup and to see your creations running. Gradually, we will build knowledge and help you create more elaborate code. Eventually, we will build applications together, bringing your knowledge of Silverlight to a productive level.

# Numerics

0-based index, 51  
3D border effects, creating with  
RadialGradientBrush in Expression Blend, 63

## A

abstract classes, 32  
ActiveX controls, 8  
Add Service Reference dialog, 528  
Add Silverlight Application dialog box, 263  
adding  
comments to XML documents, 27  
Full Screen button to Expression video  
player, 256-259  
interactivity to Thumbnails project, 181-185  
media to scenes, 74-79  
overlays to videos, 242  
Adding a namespace listing (18.2), 366  
Adding a SolidColorBrush listing (24.20), 545  
Adding an event handler listing (22.21), 486  
Adding and removing items listing (18.16), 379  
Adding elements to a Grid in code listing  
(15.7), 307  
Adding event handlers listing (22.30), 495  
Adding namespaces listing (19.9), 397  
Adding the SelectionChanged event handler list-  
ing (19.15), 404  
AddUser and ValidateUser methods listing  
(22.31), 495  
AddUser method, 276-277, 493, 495  
AddUserCompleted event handler, 497  
Adobe Flash, 8  
Adobe Illustrator, converters, 551  
adorners, 77  
advanced parameters in Expression Encoder,  
237  
advanced properties, setting for overlays, 242  
affine transforms, 37  
agTweener, 552  
airline application demo, 14

- AJAX, 7**
- Alpha channel, 67-71**
  - ancestors, 180**
  - animations**
    - color animations, 51
    - elements, 204-205
    - key frames, 49, 199
    - PointAnimation, 206
    - properties, editing, 202-203
    - rotating, 46-48
    - scale animations, 49-50
    - Silverlight, adding to Thumbnails project, 210-213, 215-220
    - smooth transitions, 50
    - starting and stopping, 208
  - storyboards**
    - reversing, 206
    - triggering, 201-202
  - synchronizing, 204**
  - testing in Blend, 201**
  - anonymous types, 476**
  - Another button listing (14.9), 268**
  - API keys, obtaining Flickr, 504**
  - App.xaml file, storing resources in, 347-348**
  - Apple Macintosh computers, debugging applications, 465**
  - application assemblies, versioning, 434**
  - Application object, 434-435**
  - applications**
    - client applications, updating, 494-497
    - code, stepping through, 459-461
    - Debug version, 458
    - debugging, 458-465
    - drawing applications
      - creating, 328-335
      - running, 335
    - HTML, blending with, 285
    - initializing JavaScript, 453-454
    - items, scrolling, 422-425
    - JavaScriptDotNet, 439-443
    - media information, removing, 467-468
    - parameters, passing, 470
    - refactoring, 430-434
  - Release version, 458-459**
  - rich interactive applications, 499**
  - running applications, debugging, 464-465**
  - startup screens, adding, 468-469**
  - test applications, creating, 542-544**
  - testing, 480**
  - Thumbnails application**
    - embedding fonts in, 354-356
    - styling, 352-354
    - thumbnails viewer, creating, 387-396
    - web browsers, caching, 439
  - Applying the template listing (19.26), 414**
  - ArrayList (old implementation) listing {20.11}, 436**
  - arrays, JavaScript, 452**
  - ASP.NET control listing (24.10), 536**
  - ASP.NET controls, 536-541**
  - aspect ratio of videos, changing in Expression Encoder, 228**
  - ASPX versus HTML, 160**
  - attached properties, 305-307**
  - Attaching an event listing (21.25), 455**
  - attaching properties, 26**
  - auto-play, disabling for videos, 80**
  - automatic property, 178-179**

**B**

- background parameter (object tag), 110**
- backslash characters, C#, 391**
- Barber, Corrina, 550**
- binary numbering system, 68**
- Binding class, 369-370**
- Binding to an object listing (24.8), 535**
- Binding to the DataContext listing (24.9), 535**
- bindings**
  - Binding class, 369-370
  - Blend, setting in, 531-532
  - Converter classes, 534-535
  - data binding, 363
    - data object creation, 364-369
  - data objects, writing to, 366-367

errors, notifications, 367-368  
objects, 535-536  
special configurations, 531-536  
templates, 427  
    to controls, 359-360  
validation errors, handling, 533-534  
values, converting in, 368-369

**Blend.** See **Expression Blend**

**blog sites**

- Abrams, Brad, 121
- HTML tags, adding, 115
- Silverlight-related blogs, 553

**blueprints, creating custom controls**, 405-406

**borders, surrounding elements with**, 308-310

**Bouquain, Thierry**, 552

**breakpoints, conditional**, 461

**Brush in App.xaml listing (24.24)**, 547

**brushes, storing resources in**, 345

**bubbling exceptions**, 518-519

**Bubbling exceptions listing (23.24)**, 518

**build actions, checking media files**, 282-283

**built-in objects (JavaScript)**, 133

**built-in types (.NET)**, 163

**Button control**, 319-320, 387

**ButtonBase control**, 319-320

## C

**C#**

- backslash character, 391
- Border element, 309
- casting, 185
- elements, 294
- loops
  - do\_while, 170
  - for, 167
  - for\_each, 168-169
  - if\_then\_else, 166-167
  - while, 169
- methods, overloading, 185-186
- namespaces, 176

operators

- conditional, 174
- logical, 173
- mathematical, 171-172

parenthesis, 174  
Popup, 310  
properties, 178-179  
resources, defining, 344  
ScrollViewer, 308  
switch\_case operator, 170-171  
TabControl, 336  
TextBlock control, 317  
type conversion, 165  
types

- floating point types, 164-165
- integer types, 163
- using directive, 176-177
- visibility, 177

**C# Elements listing (15.2)**, 294

c:\temp directory, availability, 489

**Cache panel listing (22.2)**, 469

**cacheFadeOut\_Completed event handler**, 479

**caching applications, web browsers**, 439

**Calendar control**, 325-326

**Call to RegisterScriptableObject listing (14.2)**, 262

**calling JavaScript methods from .NET**, 267-269

**Campbell, Dave**, 553

**Canvas class**, 297-298

- elements, dragging, 298-301

**canvases**

- creating, 25-26
- shapes, drawing on, 311-313

**captions, adding to videos**, 234

**case sensitivity in JavaScript**, 131

**casting**, 165, 185

**catching**

- .NET events, 285-287
- unhandled exceptions, 522-524

**chapters, adding to videos**, 233-234

**characteristics of XML documents**, 22

**CheckBox control**, 320

**CheckBox template**, 358-361

**Checking if someone subscribed listing (23.8),** 506

**Chess demo,** 14

**Class CounterTickedEventArgs listing (14.3),** 263-264

**class libraries**

- classes, moving to, 430
- creating, 429-434

**Class MyDate listing (21.11),** 446

**classes**

- abstract, 32, 292
- Binding class, 369-370
- Canvas, 297-301
- class libraries, creating, 429-432, 434
- Control, 315-317
- Converter, 534-535
- DataFile, 488-491
- decorating, 407
- DependencyObject, 292
- EventArgs, 187
- external assemblies, exporting to, 429-434
- FrameworkElement, 293
- generics, 436-437
- GridSplitter, 324
- hierarchy, 291-293
- HtmlPage, 443-445
- ItemsControls, 335-337
- Media

  - creating, 388-390
  - extending, 390-391

- MediaEx

  - creating, 390-391
  - extending, 481

- MediaExCollection, creating, 391
- MediaInformationFile, 469-472
- MyDate, 446
- Object, 292
- Page, modifying, 278-280
- scriptable classes, creating, 263-264
- ScriptObject class, converting, 449-452
- Settings, 421
- silverlightControlHost, 443

**Style,** 348-352

- creating, 348-349
- priority system, 349-350
- using, 349

**StylusDevice,** 332

**System.Windows.Controls.Panel,** 296-307

**Thumb,** 322-324

**UIElement,** 292-293

**User,** 517

**UserInformation,** 492

**client applications**

- service references, adding to, 494-495
- updating, 494-497

**client-side proxies,** 496

**ClientBin, creating in MediaFiles folder,** 401

**clientPhoto\_OpenReadCompleted event handler,** 514

**client\_DownloadProgressChanged event handler,** 483

**client\_DownloadStringCompleted event handler,** 472

**client\_OpenReadCompleted event handler,** 483-485

**clipping paths,** 94-95

**code**

- Border element, 309
- breakpoints, conditional, 461
- debugging, 458-465
- Grid layout, 303
- intermediate language (IL) code, 429
- .NET code, obfuscation, 530
- Popup, 310
- protecting, 530
- ScrollViewer, 308
- StackPanel, 301
- stepping through, 459-461
- TabControl, 336
- TextBlock control, 317

**variables**

- inspecting, 462-463
- modifying, 462-463

**code-behind**

- Grid, adding elements to, 307
- resources, using in, 345
- XAML compared, 293-294, 296

**Code-behind for dragging listing (15.4), 299-300**

**codecs, allowing viewers to select, 245-250**

**collections, storing objects in, 342-343**

**color animations, 51**

**color picker (Expression Blend), 61**

**Color Slider listing (16.9), 329**

**colors**

- mixing, 67, 69
- opacity, setting, 71
- OpacityMask, creating, 90, 92
- predefined color listing, 68
- transparency, setting, 69-70

**combining shapes, 93-94**

**comments**

- adding to XML documents, 27
- writing in JavaScript, 130

**comparing vector and raster graphics, 72**

**components**

- testing, 432
- versioning, 434

**composing**

- elements, 297
- scenes in XAML, 32-34
- transforms, 45

**conditional breakpoints, 461**

**conditional operators, 173**

**configurations, binding special configurations, 531-536**

**configuring FTP client, 122-123**

**connecting to your web server, 123**

**constructors, User class, 271-272**

**Constructors of User class listing (14.13), 271**

**constructs**

- do\_while, 170
- for, 167
- if\_then\_else, 166-167
- while, 169-170

**content controls, ScrollViewer, 307-308**

**Content Property, 35**

**ContentPresenter for a CheckBox listing (17.8), 361**

**Control class, 315-317**

**controls, 387**

- ASP.NET controls, 536-541
- Button, 319-320, 387
- ButtonBase, 319-320
- Calendar, 325-326
- CheckBox, 320
- custom controls
  - creating, 405-414
  - MediaInfoDisplay custom control, 405-414
- data controls, 363, 372-386
- DatePicker, 326-327
- disabling, 316, 424
- Ellipses, 387
- enabling, 316
- fonts, 317
- HyperlinkButton, 319-320
- Image, 74
- InkPresenter, 327-328, 332-335
- lockless controls, templating, 356-361
- MediaElement, 387
- MediaInfoDisplay control
  - creating templates, 419-420
  - default template creation, 415-417
  - removing test data, 421
  - scenes, adding to 418-419
  - test data, adding to, 417-418
  - transitions, 420-421
  - using, 417-422
  - wiring up, 421-422
- MultiScaleImage, deep zooming, 339
- Paths, 387
- pop-ups, 310-311
- primitives, 387
- property setters, 348-350
- RadioButton, 320
- Rectangle, 387
- RepeatButton, 319-320
  - changing transitions, 426-427
  - disabling, 423
  - event handlers, 423
  - styling, 427-428
  - template creation, 425-428
  - testing, 427

Scrollbar, 320-321  
 skins, 550-551  
 Slider, 320-321, 387  
 styling, 316-317, 348-352  
 templates, binding, 359-360  
 templating, 316-317  
 TextBlock, 317-318, 387  
 TextBox, 318-319  
 third-party controls, 552  
 ThumbnailsViewerControl, packing, 391-393  
 ToggleButton, 319-320  
 user controls, 405  
**UserControl**  
 connecting to real data, 403  
 creating, 387-396  
 DataTemplate, 399-401  
 moving media files, 401-402  
 namespaces, 397  
 removing test data source, 401  
 SelectionChanged event, 403-405  
 XAML media instances, 397-399  
**Convenience setter and getter listing (19.6), 395**  
**conversions (data), 446-448**  
**ConvertBack method, implementing, 368**  
**Converter classes, 534-535**  
**converters**  
 Adobe Illustrator, 551  
 Flash, 551  
**converting**  
 files to XAML, 551-552  
 ScriptObject class, 449-452  
 types to other types, 165  
**cookies, 190**  
**copying Silverlight applications, 124-126**  
**Corby, Karen, 503-504**  
**Correcting the error listing (17.6), 347**  
**CounterButton\_Click event handler, 446**  
**CounterTicked event**  
 creating, 263-264  
 raising, 264-265  
 subscribing to on JavaScript side, 265-267  
 testing, 267  
**createDelegate method, 286**  
**creating**  
 canvases, 25-26  
 LinearGradientBrush in Expression Blend, 61-62  
 objects with JSON, 139-140  
 OpacityMask, 90, 92  
 paths, 92  
 RadialGradientBrush in Expression Blend, 63  
 Silverlight 2 applications in Visual Studio 2008, 156, 161-162  
 compilation errors, finding, 158  
 Silverlight project files, 160-161  
 Web Site project files, 159  
 test pages, 107-108  
 transforms, 89  
 user controls, 98-101  
**Creating a StringReader listing (23.16), 512**  
**Creating a StringReader listing (23.17), 512**  
**creating new projects in Expression Blend, 56**  
**cropping videos in Expression Encoder, 229**  
**cross-domain policy files**  
 Flash, 500  
 Silverlight, 501-502  
**cross-domain requests**  
 placing, 499-503  
 restrictions, 502-503  
**Crumb bar, 352**  
**CSS (Cascaded Style Sheets), 5**  
 DHTML, 6  
 Silverlight, comparing with, 341  
**CSS class silverlightControlHost listing (21.7), 443**  
**CSS rules listing (14.27), 284**  
**custom controls, 405**  
 blueprints, creating, 405-406  
 creating, 405-414  
 DependencyProperties, adding to, 408-410  
 MediaInfoDisplay custom control, creating, 405-414  
**parts**  
 handling, 411-413  
 specifying, 407

- shells, creating, 406-410
  - states
    - handling, 410-411
    - specifying, 407
  - templates, applying, 413-414
  - Custom Exception constructor listing (23.28), 523**
  - Custom Exception constructor listing (23.29), 524**
  - Custom Exception listing (23.26), 520**
  - custom exceptions, creating, 523-524
  - CustomTimestamp property, 446**
- 
- ## D
- data, ItemsControls, presenting, 335-337**
  - data bindings, 363, 388**
    - Binding class, 369-370
    - data objects, creating, 364-369
  - Data bound ThumbItemsControl listing (19.12), 398**
  - Data class MyColor listing (18.12), 372**
  - data controls, 363, 372-386**
  - data conversions, 446-448**
  - data objects**
    - creating, 364-369, 388-391
    - writing to, bindings, 366-367
  - Data property, drawing shapes, 312-313**
  - data sources, XML data sources, 386**
  - data templates, creating, 376-378**
  - DataContext property, setting, 370-371**
  - DataFile classes**
    - moving to servers, 488-489
    - running on servers, 489-491
  - DataGridView**
    - elements, editing, 385-386
    - manually defined columns, 383
    - using, 380-386
  - DataGridView with manually defined columns listing (18.20), 384**
  - DataTemplate, designing for UserControl, 399-401**
  - DataTemplate for a brush listing (18.14), 377**
  - DatePicker control, 326-327**
  - dates, picking, 325**
    - Calendar control, 325-326
    - DatePicker control, 326-327
  - DateTimeConverter class listing (24.6), 534**
  - Debug mode, killing processes, 530**
  - Debug toolbar (Visual Studio), 460**
  - Debug version, comparing with Release version, 458**
  - debugging**
    - applications, 458-465
    - JavaScript, 150, 463-464
    - running applications, 464-465
    - web sites, 463
  - declaring events, 188**
  - decoding**
    - JSON strings, 510-514
    - responses, 509-510
  - decorating classes, 407**
  - deep zooming image collections, 337-339**
  - DeepZoom, 15, 337-339**
  - Default Style for MediaInfoDisplay listing (20.2), 416**
  - default templates, 357**
    - editing, 357-358
    - MediaInfoDisplay control, creating for, 415-417
  - Defining a Style in XAML listing (17.7), 348**
  - Defining resources in C# listing (17.2), 344**
  - Defining resources in XAML listing (17.1), 344**
  - defining XML namespaces, 24-25**
  - delegates, 187-188**
  - demos**
    - airline application, 14
    - Chess, 14
    - Hard Rock Cafe, 15
  - Dependency Object class, 292**
  - DependencyProperties, 292, 393-396**
    - custom controls, adding to, 408-410
    - LongDescription DP, 409
    - MediaInfo DP, 410
    - registration, 394

SelectedItem DependencyProperty, 395-396  
 ShortDescription DP, 408  
**deprecated functionality, 154**  
**derived types, 180**  
**DescriptionPanelPart property, 411-412**  
**DescriptionPanelPart property listing (19.24), 411**  
 designing DataTemplate for UserControl, 399-401  
 detecting Silverlight version number with JavaScript, 149  
 developing Silverlight, requirements, 16-17  
 development web server, 108  
**DHTML (Dynamic HTML), 6**  
**dialogs**  
   Add Service Reference, 528  
   Add Silverlight Application, 263  
   login dialogs, Thumbnails application, 270-289  
   Publish Web Site, 457  
**disabling**  
   auto-play feature for videos, 80  
   controls, 316, 424  
   RepeatButtons, 423  
   ScrollBar, 308  
**Disabling both RepeatButtons listing (20.5), 423**  
**display frame, adding to Thumbnails project, 101-102**  
**Displaying the parameters listing (21.22), 454**  
**DLLs (Dynamic-Link Libraries), 429, 432**  
**documentation for Silverlight, installing, 18**  
**DOM (Document Object Model), 443**  
**DoubleAnimation property, 46-48**  
**DoubleToThicknessConverter class listing (18.6), 368**  
**DoubleToThicknessConverter in resources listing (18.7), 369**  
**downloading videos versus streaming, 239-240**  
**DoWork method, 527**  
**do\_while loop, 170**  
**DPs (DependencyProperties), 50, 393-396**  
   custom controls, adding to, 408-410  
   LongDescription DP, 409

MediaInfo DP, 410  
 registration, 394  
 SelectedItem DependencyProperty, 395-396  
 ShortDescription DP, 408  
**dragging elements on canvas, 298-301**  
**Dragging a Thumb, C# code listing (16.2), 323**  
**Dragging a Thumb, XAML markup listing (16.1), 322**  
**drawing shapes, 311-313**  
**drawing applications**  
   creating, 328-335  
   events, handling, 330-335  
   running, 335  
   scenes preparing, 328-330  
**Dynamic HTML, 443**

## E

**editing**  
   DataGrid elements, 385-386  
   default templates, 357-358  
**editing animation properties, 202-203**  
**elements**  
   aligning Grid panel, 302-305  
   Border, 308-310  
   borders, surrounding with, 308-310  
   ButtonBase control, 319-320  
   C#, 293-296  
   Calendar, 325-326  
   Canvas, 297-301  
   composing, 297  
   Control class, 315-317  
   DataGrid, editing, 385-386  
   DatePicker, 326-327  
   DeepZoom, 337-339  
   DependencyObject, 292  
   dragging, Canvas class, 298-301  
   FrameworkElement, 293  
   Grid panel, 302-307  
   GridSplitter, 324  
   grouping, 95-96

- hierarchy, composing in Expression Blend, 64-65
- HTML elements, attaching to .NET events, 454-457
- InkPresenter, 327-328
- ItemsControls, 335-337
- JavaScript, finding in, 441-443
- MultiScaleImage, 339
- Object, 292
- Popup, 310-311
- reusable UI elements, 405
- Scrollbar control, 320-322
- ScrollViewer, 307-308
- Slider control, 320-322
- StackPanel, 301-302
- TextBlock control, 317-318
- TextBox control, 318-319
- Thumb control, 322-323
- UIElement, 292
- walking the tree, 346-347
- XAML, 293-296
- Ellipses control, 387**
- embedded media files, 401**
- embedding**
  - fonts, Thumbnails application, 354-356
  - HTML applications in existing web pages
    - using iframe, 253
    - using Live Control, 254
- Empty ResourceDictionary listing (20.1), 416**
- enabling controls, 316**
- encoding videos for Silverlight, 235-236**
- Endless loop listing (24.2), 529**
- Enumerating the items and setting brushes listing (22.12), 478**
- enums, parsing, 476**
- equality operators, redefining, 272**
- Error handling listing (22.11), 478**
- errors, 478**
  - bindings, notifications, 367-368
  - handling
    - JavaScript, 524
    - validation errors, 533-534
- Event handler ColorSlider\_ValueChanged listing (16.18), 334**
- Event handler CounterButton\_Click listing (14.6), 265**
- Event handler CounterButton\_Click listing (21.13), 446**
- Event handler handleCounterTicked listing (21.14), 447**
- Event handler handleNewUserClicked listing (14.32), 289**
- Event handler handleSubmitClicked listing (14.31), 288**
- Event handler HtmlButton\_Click listing (21.26), 456**
- Event handler InkPresenter\_MouseLeftButtonDown listing (16.14), 332**
- Event handler InkPresenter\_MouseMove listing (16.15), 333**
- Event handler LoginLink\_Click listing (14.25), 281**
- Event handler SaySomethingButton\_Click listing (14.10), 268**
- Event handler SettingsButton\_Click listing (16.17), 334**
- Event handler SetTitleButton\_Click listing (21.10), 444**
- Event handler SizeSlider\_ValueChanged listing (16.19), 335**
- event handlers, 413**
  - adding to Thumbnails project, 181-182
  - AddUserCompleted, 497
  - cacheFadeOut\_Completed, 479
  - clientPhoto\_OpenReadCompleted, 514
  - client\_DownloadProgressChanged, 483
  - client\_DownloadStringCompleted, 472
  - client\_OpenReadCompleted, 483-485
  - CounterButton\_Click, 446
  - handleCounterTicked, 447, 449
  - JavaScript, 135-136
  - LoginLink\_Click, 281
  - mediaFile\_DownloadProgressChanged, 486
  - \_mediaFile\_Loaded, 477
  - Media\_Loaded, 487
  - Page\_Loaded, 477
  - RepeatButtons, 423
  - SetTitleButton\_Click, 444

- ValidateUserCompleted, 496
- VideoPlayer\_Ended, 539
- Event handlers for RepeatButtons listing (20.6), 423**
- Event handlers**
  - InkPresenter\_MouseLeftButtonUp and LayoutRoot\_MouseLeave listing (16.16), 333
- Event handlers listing (19.25), 413**
- EventArgs class, 187**
- events**
  - adding, 396
  - CounterTicked event
    - creating, 263-264
    - raising, 264-265
    - subscribing to on JavaScript side, 265-267
    - testing, 267
  - declaring, 188
  - drawing applications, handling, 330-335
  - HTML events, handling, 288-289
  - Loaded, 470
  - .NET events
    - attaching to HTML elements, 454-457
    - catching, 285-287
    - handling in JavaScript, 263-267
  - parameters, 187
  - raising, 187
  - SelectionChanged events, 396
  - subscribing to, 189
  - testing, 267
- Exception instance, 521**
- exceptions**
  - bubbling, 518-519
  - custom exceptions, creating, 523-524
  - packing, 520-521
  - re-throwing, 519-521
  - throwing, 517-521
  - unhandled exceptions, catching, 522-524
- exchanging data between JavaScript and .NET, 445-452**
- executing Silverlight applications in browser, 59**
- exporting classes to external assemblies, 429-432, 434**
- Expression Blend**
  - animations
    - properties, editing, 202-203
    - testing, 201
  - bindings, setting in, 531-532
  - color picker, 61
  - elements, grouping, 95-96
  - elements hierarchy, composing, 64-65
  - fonts, embedding in, 354-356
  - installing, 54
  - keyboard shortcuts, 60
  - keyframes, 199
  - launching, 54
  - LinearGradientBrush, creating, 61-62
  - options, 55-56
  - panels, 57-58
  - player template, adding Full Screen button, 256-257, 259
  - Project panel, 58-59
  - projects, creating, 56
  - properties, resetting to default value, 64
  - Properties panel, 60
  - RadialGradientBrush, creating, 63
  - resources, moving in, 354
  - results, testing in web browser, 77
  - running Silverlight applications in browser, 59
  - shapes, working with, 60
  - Style class
    - creating in, 350-351
    - editing in, 351-352
  - timelines, 200-201
  - transforms, creating, 89
- Expression Design, creating XAML resources, 549-550**
- Expression Encoder, 224**
  - advanced parameters, 237
  - captions, adding to videos, 234
  - chapters, adding to videos, 233-234
  - leader/trailer video segments, adding, 230
  - Metadata panel, 232-233
  - options, setting, 224-225
  - panels, 225

- script commands, 254-256
- videos
- cropping, 229
  - encoding for Silverlight, 235-236
  - importing, 226-227
  - playing in full screen, 237
  - profiles, 227-228
  - size of, changing, 228
  - testing before encoding, 231-232
- Expression Studio, 16-17, 549**
- extending**
- Media class, 390-391
  - MediaEx class, 481
  - User object, 270-274
- extension methods, 542, 544-548**
- extensions (markup), 346-347**
- external assemblies, exporting to classes, 429-434**
- ExtractResponse method, 509**
- ExtractResponse method listing (23.11), 509**
- F**
- Fajardo, Jose, 339**
- File clientaccesspolicy.xml, sample 1 listing (23.2), 501**
- File clientaccesspolicy.xml, sample 2 listing (23.3), 501**
- File crossdomain.xml listing (23.1), 500**
- files**
- XAML, converting to, 551-552
  - XML files, loading, 467-477, 479-480
  - Zip files
    - creating, 480
    - loading, 480-483
    - reading, 483-485
- filtering LINQ (Language Integrated Query), 476**
- Flash, 8**
- converters, 551
  - cross-domain policy file, 500
- Flick JSON response listing (23.14), 511**
- Flickr**
- API keys, obtaining, 504
  - requests, 504-508
  - responses, handling, 508-514
  - TOS (terms of service), accepting, 503
  - flickr-search entry listing (23.4), 505**
  - floating point types, 164-165**
  - flow layout, StackPanel, 301-302**
  - Flow layout with StackPanel listing (15.5), 302**
  - folders, References folder, 429**
  - fonts**
    - Blend expression, embedding in, 354-356
    - controls, 317
    - Thumbnails application, embedding in, 354-356
    - Visual Studio, embedding in, 356
  - for construct, 167**
  - foreach construct, 168**
  - formats, passing standard data formats, 448-449**
  - FrameworkElement class, 293**
  - frameworks, unit tests, 541**
  - free-webhosts.com, 121**
  - FTP client**
    - configuring, 122-123
    - web server, connecting to, 123
  - Full Screen button, adding to Expression video player, 256-257, 259**
  - full screen videos, playing, 237**
  - functions (JavaScript), 132**
  - future of Silverlight, 19**
- G**
- Generic class implementation listing (20.14), 437**
- Generic List (new implementation) listing [20.12], 436**
- Generic method implementation listing (20.13), 436**
- generics, 186, 436-437**
- GetColor method, 331**

- GetProperty** method, 452
- Getting and starting the Storyboard listing (22.13)**, 479
- Getting the parameters at startup listing (21.23)**, 454
- GetUser** method, 277
- global object (JavaScript)**, 137-138
- Gomes, Celso**, 19
- GoToState** method, 410
- GoToState** method listing (19.23), 410
- Grid panel**
  - adding elements to, 305-307
  - aligning elements, 302-305
- GridSplitter** class, 324
- grouping elements**, 95-96
- Guthrie, Scott**, 3
  
- H**
- handleCounterTicked** event handler, 447-449
- handleNewUserClicked** method, 289
- handleSubmitClicked** method, 288
- handling**
  - errors, 478
    - JavaScript, 524
  - events, 330-335
    - HTML, 288-289
    - .NET, 263-267
  - images, *UserControl*, 399-400
  - multiple users, 274-277
  - parts, custom controls, 411-413
  - SelectionChanged* event, 403-405
  - states, custom controls, 410-411
  - third-party responses, 508-514
  - validation errors, 533-534
  - videos, *UserControl*, 400-401
- hard disks, placing websites on**, 282
- Hard Rock Cafe demo**, 15
- hardware requirements for Silverlight**, 11
- height attribute (object tag)**, 109
- Heuer, Tim**, 553
  
- hexadecimal numbering system**, 68
- hiding media from mouse input**, 83
- hiding ScrollBar**, 308
- hierarchy**
  - classes, 291-293
  - inheritance, 291
- history of CSS**, 5
- history of JavaScript**, 5
- history of Silverlight**, 4
- HTML**
  - applications, blending with, 285
  - Dynamic HTML, 443
  - elements, attaching .NET events to, 454-457
  - events, handling, 288-289
  - login dialogs, adding applications to, 283-284
  - object tag, 109
    - background parameter, 110
    - height attribute, 109
    - id attribute, 109
    - onerror parameter, 110
    - onload parameter, 110
    - type attribute, 109
    - windowless parameter, 110
  - pages
    - access, preventing, 445
    - .NET, accessing, 443-445
    - versus ASPX, 160
  - HTML control listing (24.11)**, 536
  - HTML login dialog listing (14.26)**, 284
  - HtmlPage** class, 443-445
  - HyperlinkButton** control, 319-320
  
- I**
- id attribute (object tag)**, 109
- IDEs (Integrated Development Environments), Visual Studio**, 457
- iframe**, embedding HTML applications in existing web pages, 253
- if\_then\_else construct**, 166-167

**IL (intermediate language) code**, 429  
**image collections**, deep zooming, 337-339  
**Image control**, 74  
**image pyramids**, creating, 337-339  
**images**, handling with **UserControl**, 399-400  
**implementing interfaces**, 342-343  
**Implementing two event handlers listing (23.22)**, 515  
**importing videos in Expression Encoder**, 226-227  
**Infragistics**, 552  
**inheritance**, 180  
  hierarchy, 291  
  methods, overriding, 447  
**Initialization code listing (16.12)**, 331  
**initialization parameters (JavaScript)**, passing, 453-454  
**InitializeComponent method**, stepping through, 461  
**Initializing the ColorBrushes class listing (18.13)**, 374  
**Initializing the TextBoxes listing (18.17)**, 379  
**InkPresenter control**, 327-328, 332-335  
**inputting text**, 317-319  
**installing**  
  Expression Blend, 54  
  framework unit tests, 541  
  Silverlight documentation, 18  
  Visual Studio, 54, 156  
**instances**  
  Exception, 521  
  JavaScript, 137  
  XAML media instances. creating, 397-399  
**integer types**, 163  
**Integrated Development Environments (IDEs)**, Visual Studio, 457  
**integrating Silverlight with existing web page**, 115-116, 143-149  
  animated logo, creating, 117-118  
  results, testing, 119  
  XAP files  
    referencing from external web servers, 119-120  
    retrieving, 117

**Intellisense**, 35, 162  
**interactivity**, adding to **Thumbnails project**, 181-185  
**interfaces**  
  implementing, 342-343  
  thumbnails viewer, creating, 391-393  
  user interfaces, updating, 486-488  
**intermediate language (IL) code**, 429  
**interoperability**, 261  
  JavaScript in .NET environment, 261-269, 289  
**interpreter**, 141  
**isolated storage**, 190-193  
  multibrowser compatibility, 197  
  reading from, 195  
  writing to, 195-197  
**items, scrolling**, 422-425  
**ItemsControls class**, 335-337  
**ItemsSource properties**, adding, 393-396  
**ItemsSourceProperty DP listing (19.5)**, 395

## J

**Januszewski, Karsten**, 549  
**Java applets**, 7-8  
**JavaScript**, 261  
  arrays, 452  
  built-in objects, 133  
  case sensitivity, 131  
  comments, writing, 130  
  debugging, 150, 463-464  
  detecting, 150  
  DHTML, 6  
  errors, handling, 524  
  event handlers, 135-136  
    multiple event handlers, registering, 440-441  
  functions, 132  
  global object, 137-138  
  history of, 5  
  initialization parameters, passing, 453-454  
  instances, 137

JSON, creating objects, 139-140  
 libraries, 133  
 literal notation, 138-139  
 methods, calling from .NET, 267-269  
 modifying web pages during runtime, 150  
 .NET environment  
   data exchange, 445-452  
   events, attaching to HTML elements, 454-457  
   events, handling, 263-267  
   interoperability, 261-269, 289  
   methods, calling from, 269  
 objects, 136, 452  
 prototype property, 132  
 scripts, loading in web page, 141-142  
**Silverlight**  
   applications, initializing, 453-454  
   elements, finding in, 441-443  
   integrating in existing web page, 143-149  
   version number, detecting, 149  
 simple values, 452  
 statements, 130  
 static members, 140  
 this keyword, 142-143  
 types, 132  
 variables, 131  
   null value, 135  
   undefined values, 134  
**JavaScriptDotNet application, 439**  
   multiple JavaScript event handlers, registering, 440-441  
   Silverlight elements, finding, 441-443  
**JavaScriptDotNet application, JavaScript Code listing (14.7), 265**  
**JavaScriptDotNet application, Page class listing (14.4), 264**  
**JSON (JavaScript Object Notation), 138**  
   objects, creating, 139-140  
   strings, decoding, 510-514

**K**

Kaxaml, testing XAML markup, 29-30  
 key frames, 49  
 keyboard shortcuts for Expression Blend, 60  
 keyframes, 199, 233  
 killing  
   loops, 170  
   processes, 528-530  
**Kinney, Adam, 553**

**L**

**launching Expression Blend, 54**  
**Laying out elements in a Grid listing (15.6), 305**  
**layouts**  
   flow layout, StackPanel, 301-302  
   Grid, 302-307  
   panels, 296-307  
   ScrollViewer, 307-308  
**leader video segments, adding in Expression Encoder, 230**  
**Lee, Gigi, 335**  
**libraries, 133**  
   class libraries, creating, 429-432, 434  
   DLLs, checking, 432  
   third-party, 552  
**LinearGradientBrush**  
   creating in Expression Blend, 61-62  
   gradient vector, changing, 62  
**LINQ (Language Integrated Query)**  
   filtering, 476  
   XML files, reading, 473-476  
**ListBox markup listing (18.15), 378**  
**ListBoxes, 372-376**  
**literal notation, 138-139**  
**Live Control, embedding HTML applications in existing web pages, 254**  
**Loaded event, 470**  
**Loaded event and companion method listing (22.5), 470**

**loading**

XML files, 467-480  
Zip files, 480-483

**Loading the Zip file listing (22.16), 482****LoadMediaFile method, 471-472****LoadMediaFile method listing (22.6), 471****LoadUsers method, 275, 490****LoadUsers method listing (22.25), 490****LoadZipFile method, 482****LoadZipFile method listing (22.17), 482****lockless controls, templating, 356-361****logical operators, 173**

login dialogs, adding to Thumbnails application,  
270-289

login link, adding to XAML user interfaces,  
280-282

**LoginLink\_Click event handler, 281****LongDescription DP, 409****LongDescription DP listing (19.21), 409****lookless objects, 405**

Looping through the photos, placing requests  
listing (23.18), 513

**looping videos, 80-81****loops**

do\_while, 170  
for, 167  
foreach, 168-169  
if\_then\_else, 166  
killing, 170  
while, 169-170

**M**

Macintosh computers, debugging running applications, 465

**Making a LINQ Query listing (22.8), 474****managed codes, 155****manually defined columns, DataGrid, 383****markup extensions, 346-347****mathematical operators, 172****MatrixTransform, 43-44****media**

adding to scenes, 74-79

files, moving with UserControl, 401-402

instances, creating in XAML, 397-399

presenting, 324

removing information from applications,  
467-468

transforms, adding, 86

**videos**

build actions, checking, 282-283

cropping in Expression Encoder, 229

hiding from mouse input, 83

importing with Expression Encoder,  
226-227

overlays, adding, 242

players, allowing viewers to select,  
245-250

profiles, 227-228

reflection, adding, 104

size of, changing in Expression Encoder,  
228

writing with, 84-85

web-based, 86-87

**Media class**

creating, 388-390

extending, 390-391

**MediaDataSource collection listing (19.10), 397****MediaElement control, 387****MediaEx class**

creating, 390-391

extending, 481

**MediaEx class listing (19.2), 390-391****MediaExCollection class, creating, 391****MediaExCollection class listing (19.3), 391****\_mediaFile\_Loaded event handler, 477****\_mediaFile\_Loaded event handler listing  
(22.10), 477****MediaFileLoadedEventArgs class listing (22.4),  
470****MediaFiles folder, creating in ClientBin, 401****mediaFile\_DownloadProgressChanged event**

handler, 486

**mediaFile\_DownloadProgressChanged handler  
listing (22.22), 486**

**MediaInfo class listing (24.3), 531**

**MediaInfo DP, 410**

**MediaInfo DP listing (19.22), 409**

**MediaInfoDisplay constructor listing (19.17), 406**

**MediaInfoDisplay control**

- creating, 405-414
- default template, creating for, 415-417
- parts, handling, 411-413
- scenes, adding to, 418-419
- states, handling, 410-411
- templates
  - applying, 413-414
  - creating, 419-420
- test data
  - adding, 417-418
  - removing, 421
- transitions, creating, 420-421
- using, 417-422
- wiring up, 421-422

**MediaInformationFile class, 469-472**

**MediaPlayer ASP.NET control, 538-541**

**MediaPlayer ASP.NET control with chapters listing (24.15), 540**

**MediaPlayer ASP.NET control with event handler listing (24.14), 540**

**MediaType enumeration and Media class listing (19.1), 388-389**

**Media\_Loaded event handler, 487**

**Media\_Loaded event handler listing (22.24), 487**

**media\_MouseLeftButtonDown event handler listing (19.14), 403**

**MediaInfoDisplay custom control**

- blueprints, creating, 405-406
- DependencyProperties, adding, 408-410
- parts, specifying, 407
- shells, creating, 406-410
- states, specifying, 407

**members (.NET), creating scriptable members, 261-262**

**Metadata panel (Expression Encoder), 232-233**

**metainformation, 407**

**Method AddUser in Page class listing (14.21), 278**

**Method createDelegate listing (14.29), 286**

**method execution context, 142-143**

**Method GetColor listing (16.13), 331**

**Method OnLoginClicked listing (14.24), 281**

**Method saySomething listing (14.8), 267**

**Method saySomething listing (21.6), 442**

**Method ToString listing (21.15), 447**

**Method TryFindResource overloaded for the Application class listing (24.26), 548**

**Method ValidateUser in Page class listing (14.22), 279**

**methods**

- AddUser, 276-277, 493-495
- ConvertBack, 368
- createDelegate, 286
- DoWork, 527
- extension methods, 542-544
  - TryFindResource extension method, 544-548
- ExtractResponse, 509
- generics, 436-437
- GetColor, 331
- GetProperty, 452
- GetUser, 277
- GoToState, 410
- handleNewUserClicked, 289
- handleSubmitClicked, 288
- inherited methods, overriding, 447
- InitializeComponent, 461
- JavaScript methods, calling from .NET, 267-269
- LoadMediaFile, 471-472
- LoadUsers, 275, 490
- LoadZipFile, 482
- .NET methods, calling from JavaScript, 269
- Object class, overriding, 292
- OnLoginClicked, 281
- overloading, 185-186
- ResponseCallback, 508
- SaveUsers, 275-276, 490
- saySomething, 442
- Scriptable, 454-457
- test methods, 449
- TestTryFindResourceInLayoutRoot, 543

ToString, 271, 447-448  
    ValidateUser, 277, 493-495

**Methods** AddUser listing (14.18), 276

**Methods** Equals and GetHashCode listing (14.15), 274

**Methods** GetUser listing (14.20), 278

**Methods** LoadUsers listing (14.16), 275

**Methods** SaveUsers listing (14.17), 276

**Methods** ValidateUser listing (14.19), 277

**MFC (Microsoft Foundation Classes)**, 155

**Microsoft Silverlight Streaming servers**, publishing on, 250-252

**Microsoft Visio**, converters, 552

**mixing colors**, 67-69

**Modified method TryFindResource listing (24.25)**, 547

**Modified test div element listing (21.24)**, 455

**Modified TryFindResource method listing (24.19)**, 545

**modifying paths with splines**, 93

**Modifying the Binding listing (24.7)**, 535

**Modifying the DoWork method listing (24.1)**, 528

**multiple JavaScript event handlers**, registering, 440-441

**multiple users**, handling, 274-277

**MultiScaleImage control**, deep zooming, 339

**multithreading programming**, 509

**MyDate class**, 446

## N

**namespaces**, 23, 176

- changing, 430-431
- defining, 24-25
- UserControl, adding, 397
- using directive, 176-177

**navigating property paths**, 51-52

**.NET**, 154

- code, obfuscation, 530
- deprecated functionality, 154

**events**

- catching, 285-287
- HTML elements, attaching to, 454-457
- JavaScript, handling in, 263-267

**HTML pages**, accessing, 443-445

**JavaScript**

- calling methods from, 267-269
- data exchange, 445-452
- interoperability, 261-269, 289

**methods**, calling from JavaScript, 269

**objects**, 180

- Silverlight applications, registering with, 262-263
- System.Windows.Browser namespace, availability, 262

**RegisterScriptableObject method**, 262-263

**ScriptableMember attribute**, 261-262

**types**, 163

- converting to other types, 165
- floating point types, 164-165
- integer types, 163

**versions**, 154

**New button listing (14.5)**, 264

**New event handler handleCounterTicked listing (21.17)**, 449

**New event handler HtmlButton\_Click listing (21.27)**, 456

**new keyword (JavaScript)**, 132

**New method ToString listing (21.16)**, 449

**New SizeSlider listing (18.4)**, 366

**New SolidColorBrush listing (24.23)**, 547

**nonsquare shapes**, clipping paths, 94-95

**notifications**, binding errors, 367-368

**Notifying the user and getting the stream listing (23.12)**, 510

**null value (JavaScript)**, 135

## O

**obfuscation**, .NET code, 530

**Object class**, 292

- object tag**
  - background parameter, 110
  - height attribute, 109
  - id attribute, 109
  - onerror parameter, 110
  - onload parameter, 110
  - type attribute, 109
  - windowless parameter, 110
- objects**
  - Application object, 434-435
  - binding to, 535-536
  - collections, storing in, 342-343
  - data objects
    - creating, 364-369, 388-391
    - writing to, 366-367
  - global objects, 137-138
  - inspecting, 462
  - JavaScript, 452
  - lookless objects, 405
  - .NET objects, registering, 262-263
  - User objects, extending, 270-274
  - value coercion, 321-322
- Objects and Timeline category (Style class), 351**
- ObservableCollection, 374**
- obtaining web space, 120-121**
  - FTP client, configuring, 122-123
  - web server requirements, 121
- onerror parameter (object tag), 110**
- OnItemsSourceChanged event handler listing (19.4), 394**
- online resources, XAML resources, 550-551**
- onload parameter (object tag), 110**
- OnLoginClicked method, 281**
- OnLongChanged event handler listing (19.20), 408**
- OnSelectionChanged helper method listing (19.8), 396**
- OOP (object-oriented programming), 175**
  - inheritance, 180
  - polymorphism, 32
  - static members, 140
- opacity, setting, 71**
- Opacity property, 50**
- OpacityMask, creating, 90, 92**
- opening Solution Explorer, 160**
- operators**
  - conditional, 173
  - logical, 173
  - mathematical, 172
- options, setting in Expression Encoder, 224-225**
- overlays**
  - adding to videos, 242
  - advanced properties, setting, 242
  - XAML overlays, adding to videos, 243-244
- overloading methods, 185-186**
- Overridden method ToString listing (14.12), 271**
- overriding**
  - inherited methods, 447
  - object class methods, 292

## P-Q

- packing**
  - exceptions, 520-521
  - panels, 296-307
  - ThumbnailsViewerControl, 391-393
- Packing and re-throwing an exception listing (23.27), 521**
- Page class, modifying, 278-280**
- Page\_Loaded event handler, 477**
- Page\_Loaded event handler listing (22.9), 477**
- panels, 296-307**
  - Canvas, 298-301
  - Expression Blend, 57-58
    - Metadata panel, 232-233
  - Expression Encoder, 225
  - Grid, 302-307
  - StackPanel, 301-302
- parameters, passing, 470**
- parsing, 141, 476**
- parts, custom controls**
  - handling, 411-413
  - specifying, 407
- Parts and states listing (19.18), 407**

**passing**  
 parameters, 470  
 standard data formats, 448-449

**Passing initialization parameters in JavaScript listing (21.21), 453**

**passwords, storing on web clients, 270**

**Path Markup Syntax, 95**

**paths**  
 clipping paths, 94-95  
 creating, 92  
 in XAML, 95  
 modifying with splines, 93  
 shapes, combining, 93-94

**Paths control, 387**

**pausing videos, 82-83**

**Pen tool, 92, 426**

**Pencil tool, 92**

**photographs, sending for requests, 513-514**

**placing POST requests, 516-517**

**playing videos, 82-83**

**polymorphism, 32, 166, 180**

**pop-up controls, 310-311**

**POST requests, placing, 516-517**

**predefined color listing, 68**

**presenting**  
 data, ItemsControl, 335-337  
 media, 324  
 text, 317-319

**preventing HTML page access, 445**

**Preview tab UserControl, 399**

**primitives, 387**

**priorities, property setters, 349-350**

**processes, killing, 528-530**

**profiles (Expression Encoder), 227-228**

**programming, multithreading programming, 509**

**progressive downloads, 240**

**project files, 58**

**Project panel (Expression Blend), 58-59**

**projects**  
 classes, moving, 430  
 creating, 56  
 web site projects, debugging, 463

**properties, 178-179**  
 Application object, 435  
 attached properties, 305-307  
 attaching, 26  
 CustomTimestamp, 446  
 DataContex, setting, 370-371  
 DependencyProperties, 393-396  
 adding to, 408-410  
 LongDescription DP, 409  
 MediaInfo DP, 410  
 registration, 394  
 SelectedItem DependencyProperty, 395-396  
 ShortDescription DP, 408  
 DescriptionPanelPart, 411-412  
 Exception instance, 521  
 ItemsSource, adding, 393-396  
 OpacityMask, 90, 92  
 resetting to default value, 64  
 setting in XAML, 30-31  
 stretch, 83  
 User class, 270  
 UserControl, 98-101  
 VideoBrush, 84-85

**Properties panel (Expression Blend), 60**

**Property CustomTimestamp listing (21.12), 446**

**Property in Application object listing (20.10), 435**

**property paths, navigating, 51-52**

**property setters, 348-350**

**protecting source code, 530**

**prototype property (JavaScript), 132**

**proxies, client-side, 496**

**public .NET members, creating scriptable public members, 261-262**

**Publish Web Site dialog, 457**

**publishing**  
 on Microsoft Silverlight Streaming servers, 250-252  
 videos on website, 240-241  
 websites with Visual Studio, 457-458

## R

- RadialGradientBrush**, 63
- RadioButton control**, 320
- raising events**, 187
  - CounterTicked event, 264-265
  - SelectionChanged event, 403-405
- raster graphics**, 72
- re-throwing exceptions**, 519-521
- Re-throwing an exception listing (23.25)**, 520
- reading**
  - XML files, LINQ, 473-476
  - Zip files, 483-485
- Reading the string listing (23.15)**, 511
- Rectangle control**, 387
- recursion, implementing unit testing**, 545-548
- redefining equality operators**, 272
- Redefining the equality operators listing (14.14)**, 272
- refactoring**, 432-434
- referenced assemblies**, 176
- References folder**, 429
- reflection**
  - adding to videos, 104
  - creating under thumbnails, 102-104
- registering multiple JavaScript event handlers**, 440-441
  - Registering and unregistering event handler to a Silverlight element listing (21.4)**, 441
  - Registering one event handler listing (21.1)**, 440
  - Registering two event handlers listing (21.3)**, 441
  - Registering two event handlers listing (23.21)**, 515
- RegisterScriptableObject method (.NET)**, 262-263
- registering**
  - DependencyProperties, 394
  - .NET objects with Silverlight applications, 262-263
- Release version**
  - creating, 458-459
  - Debugged version, comparing, 458
- RenderTransform property**, 37-39
- RepeatButtons control**, 319-320
  - disabling, 423
  - event handlers, 423
  - styling, 427-428
  - templates
    - binding, 427
    - creating, 425-428
  - testing, 427
  - transitions, changing, 426-427
- requests**
  - cross-domain requests, placing, 499-503
  - Flickr
    - preparing for, 504-508
    - sending to, 504-508
  - pictures, sending for, 513-514
  - POST requests, placing, 516-517
  - security checks, 500
  - sending, 471-472
  - triggering, 477
  - WCF, sending to, 488-497
- requirements for Silverlight**, 11, 16-17
- resetting Blend properties to default value**, 64
- ResourceDictionaries**, 343-344
- resources**
  - App.xaml file, storing in, 347-348
  - Blend expression, moving in, 354
  - brushes, storing in, 345
  - code-behind, using in, 345
  - data objects, creating in, 365-366
  - Silverlight-related blogs, 553
  - storing, 343-344
  - Style class, 348
    - creating, 348-349
    - creating in Expression Blend, 350-351
    - editing in Expression Blend, 351-352
    - priority system, 349-350
    - using, 349
  - XAML
    - creating and purchasing, 549-552
    - defining in, 344
    - online resources, 550-551
    - using in, 345-347

- ResponseCallback method, 508**
- ResponseCallback method listing (23.10), 509**
- responses**
  - decoding, 509-510
  - obtaining, 472-473
  - third-party services, handling, 508-514
- restrictions on cross-domain requests, 502-503**
- retrieving JavaScript initialization parameters, 453-454**
- reusable UI elements, 405**
- reusing unit tests, 549**
- reversing storyboards, 206**
- rich interactive applications, 499**
- RotateTransform, 39-40**
- rotating animations, 46-48**
- Roxio Buzz, 223**
- running DataFile classes on servers, 489-491**
- running applications, debugging, 464-465**
  
- S**
- SaveUsers method, 275-276, 490**
- SaveUsers method listing (22.26), 491**
- saving users, 275-276**
- saySomething method, 442**
- Scalable Vector Graphics SVG, 552**
- scale animations, 49-50**
- ScaleTransform, 41, 49-50**
- scenes**
  - composing in XAML, 32-34
  - dragging Canvas class, 298-301
  - drawing applications, preparing, 328-330
  - media, adding, 74-76, 78-79
  - MediaInfoDisplay control, adding, 418-419
  - scrollbars, adding, 97
- schema, 24**
- script commands, firing in Expression Encoder, 254-256**
- scriptable classes, creating, 263-264**
- Scriptable method, 454-457**
- scriptable public .NET members, creating, 261-262**
- ScriptableMember attribute (.NET), 261-262**
- ScriptableMember listing (14.1), 262**
- ScriptObject class, converting, 449-452**
- scripts (JavaScript)**
  - loading in web page, 141-142
  - wiring up, 287-288
- Scrollbar control, 320-321**
- scrollbars**
  - adding to a scene, 97
  - disabling and hiding, 308
- scrolling items, 422-425**
- ScrollViewer, 97-98, 307-308**
- ScrollViewer in code listing (15.9), 308**
- ScrollViewer in XAML listing (15.8), 307**
- ScrollViewer property, 97**
- security**
  - of web applications, 10-11
  - source code, protecting, 530
- SelectedItem DependencyProperty, adding, 395-396**
- SelectedItem DP listing (19.7), 395**
- SelectionChanged event, 396, 403-405**
- SendFlickrRequests method listing (23.9), 507**
- sending requests, 471-472**
- serialization, 270-271**
- serialization languages, XAML, 293**
- servers**
  - DataFile classes
    - moving to, 488-489
    - running on, 489-491
    - users, moving to, 488-489
- Service Reference, updating, 527-528**
- service references, adding to client applications, 494-495**
- services**
  - third-party services, communicating with, 503-516
  - WCF services
    - creating, 491-492
    - implementing, 492-494
- setting**
  - DataContext property, 370-371
  - initialization parameters in JavaScript, 453

- Setting the data context listing (18.9), 371**
- Setting the first column's width listing (18.19), 382**
- Setting the TextBlocks' brushes in code listing (19.13), 402**
- Setting the ThumbDisplay1 listing (22.23), 487**
- Settings Border and Grid listing (16.8), 329**
- Settings Button listing (16.11), 330**
- Settings class, 421**
- Settings class listing (18.1), 364**
- Settings class listing (20.4), 421**
- Settings in resources listing (18.3), 366**
- Settings Popup listing (16.7), 329**
- SetTitleButton\_Click event handler, 444**
- shapes**
  - combining, 93-94
  - drawing, 311-313
  - working with in Expression Blend, 60
- Shapes listing (15.10), 312**
- shells, creating custom controls, 406-410**
- ShortDescription DP, 408**
- ShortDescription DP listing (19.19), 408**
- Silverlight**
  - applications, copying, 124-126
  - CSS (Cascaded Style Sheets), comparing with, 341
  - detecting, 114
  - developing, requirements, 16-17
  - documentation, 18
  - evolution of, 553
  - future of, 19
  - hardware requirements, 11
  - history of, 4
  - integrating with existing web page, 115-116, 143-149
    - animated logo, creating, 117-118
    - results, testing, 119
    - XAP files, retrieving, 117
    - XAP files, referencing from external web server, 119-120
  - running on multiple platforms, 9-10
  - tutorials, 18-19
- Silverlight ASP.NET control listing (24.12), 537**
- Silverlight Cream blog, 553**
- Silverlight Streaming Servers, publishing on, 250-252**
- silverlightControlHost class, 443**
- SilverlightDraw, setting the scene listing (16.6), 328**
- SilverlightPad, testing XAML markup, 27-29**
- Simple DataGrid listing (18.18), 380**
- simple property syntax, 30**
- simple values, JavaScript, 452**
- SimpleUserLogin logic listing (20.9), 433**
- SimpleUserLogin UI listing (20.8), 433**
- size of videos, changing in Expression Encoder, 228**
- Size Slider listing (16.10), 330**
- SkewTransform, 42-43**
- skins, controls, 550-551**
- Slider control, 320-321, 387**
- smooth transitions, 50**
- SoftwareFX, 552**
- solution files, 58**
- Solution Explorer, opening, 160**
- source code, protecting, 530**
- special configurations, binding, 531-536**
- splines, modifying paths, 93**
- stacking elements with StackPanel, 301**
- StackPanel**
  - elements, stacking, 301
  - flow layout, 301-302
- standard data formats, passing, 448-449**
- starting animations, 208**
- startup screens, adding to applications, 468-469**
- statements (JavaScript), 130**
- states, custom controls**
  - handling, 410-411
  - specifying, 407
- States category, modifying, 358-359**
- static members, 140**
- static registrations, classes, 394**
- Status TextBlock listing (23.20), 515**
- stepping through code, 459-461**
- stopping animations, 208**

**storing data**

- in App.xaml file, 347-348
- collections, 342-343
- isolated storage, 190-193
  - multibrowser compatibility, 197
  - reading from, 195
  - writing to, 195-197
- resources, 343-344

**Storyboard to hide the cache listing (22.3), 469****storyboards, 200**

- reversing, 206
- triggering, 201-202

**streaming servers, publishing on Streaming****Silverlight servers, 250-252****streaming videos, 239-240****Stretch property, 83****StringReader, creating, 512****strings, decoding JSON strings, 510-514****Style class, 348-352****Blend expression**

- creating in, 350-351
- editing in, 351-352

**creating, 348-349****Objects and Timeline category, 351****priority system, 349-350****using, 349****XAML, defining in, 348****styles, applying, 341**

- to controls, 316-317, 348-352
- to RepeatButtons control, 427-428
- to TextBlocks, 352-353
- to Thumbnails application, 352-354

**StylusDevice class, 332****subscribing to events, 189, 265-267****Swanson, Mike, 551****switch\_case condition operator, 170-171****synchronizing animations, 204****System.Windows.Browser namespace, .NET****objects availability, 262****System.Windows.Controls.Panel class, 296-307****T****TabControl, 336-337****tablets, writing text with, 327-328****TargetProperty syntax, 48****TDD (Test Driven Development), 458, 541****Teach Yourself ASP.NET 2.0 in 24 Hours, 489****Telerik, 552****TemplateColumn for the Name column listing (18.21), 385****TemplateColumn for the Timestamp column listing (18.22), 385****templates, 316-317, 341****CheckBox, 358-361****controls, binding in, 359-360****custom controls, applying, 413-414****data templates, creating, 376-378****default templates, 357-358****lockless controls, 356-361****MediaInfoDisplay control, creating for, 415-420****RepeatButtons control****binding, 427****creating, 425-428****XAML templates, purchasing, 551****TEMPMediaDataSource collection listing (19.11), 398****test applications, creating, 542-544****test data****MediaInfoDisplay control****adding to, 417-418****removing from, 421****UserControl, removing from, 401****Test div element listing (21.8), 444****Test Driven Development (TDD), 458, 541****Test initialization method listing (24.16), 543****test methods, 449****Test methods listing (21.18), 450****test pages, creating, 107-108****testing****animations in Blend, 201****applications, 480****components, 432**

events, 267  
 RepeatButtons control, 427  
 Silverlight files on web server, 126  
 unit tests, 541  
     creating, 541-549  
     frameworks, 541  
     recursion, 545-548  
     reusing, 549  
     test applications, 542-544  
     test passes, 545  
     TryFindResource extension method, 544-548  
 video in Expression Encoder, 231-232  
 XAML markup  
     with KaXaml, 29-30  
     with SilverlightPad, 27-29

**TestTryFindResourceInLayoutRoot method, 543**

**TestTryFindResourceInLayoutRoot method listing (24.17), 543**

**TestTryFindResourceNotFound method listing (24.22), 546**

**text**  
     inputting, 317-319  
     presenting, 317-319  
     writing, 327-328

**TextBlock and HyperlinkButton listing (14.23), 280**

**TextBlock control, 317-318, 352-353, 387**

**TextBox control, 318-319**

**TextBoxes, initializing, 379**

**TextBox\_BindingValidationError error handler, 533**

**TextBox\_BindingValidationError error handler listing (24.4), 533**

**third-party controls, 552**

**third-party libraries, 552**

**third-party plugins, 7-8**

**third-party services, communicating with, 503-506, 508-514, 516**

**third-party web technologies, running on multiple platforms, 9**

**this keyword (JavaScript), 132, 142-143**

**threading context, 508**

**threads, 509**

**Three event handlers listing (21.20), 451**

**Three TextBlocks listing (21.19), 450**

**throwing exceptions, 517-521**

**Thumb class, 322-324**

**thumbnails, placing into ScrollViewer, 97-98**

**Thumbnails application**  
     display frame, adding, 101-102  
     fonts, embedding in, 354-356  
     HTML events, handling, 288-289  
     HTML login dialogs, adding to, 283-284  
     interactivity, adding, 181-185  
     items, scrolling, 422-425  
     login dialogs, adding to, 270-289  
     media files, build actions, 282-283  
     media information, removing, 467-468  
     .NET event, catching, 285-287  
     reflection, adding under thumbnails, 102-104  
     scripts, wiring up, 287-288  
     Silverlight animations, adding, 210-220  
     startup screen, adding, 468-469  
     styling, 352-354  
     web projects, adding to, 282

**Thumbnails constructor listing (14.28), 285**

**Thumbnails prototype listing (14.30), 286**

**thumbnails viewer**  
     creating, 387-396  
     data objects, creating, 388-391  
     DataTemplate, designing for, 399-401  
     images, handling, 399-400  
     interface, creating, 391-393  
     media files, moving, 401-402  
     namespaces, adding, 397  
     Preview tab, 399  
     real data, connecting to, 403  
     SelectionChanged event, raising and handling, 403-405  
     videos, handling, 400-401  
     XAML media instances, creating, 397-399

**ThumbnailsViewerControl, packing, 391-393**

**ThumbnailsViewerControl\_LayoutUpdated event handler listing (20.7), 424**

**ThumbnailsViewer\_SelectionChanged event handler listing (19.16), 404**

timelines, 200-201  
timeouts, 522  
ToggleButton control, 319-320  
TOS (terms of service), Flickr, 503  
ToString method, 270-271, 447-448  
trailer video segments, adding in Expression Encoder, 230  
TransformGroup, composing transforms, 45  
transforms, 37  
    adding to media, 86  
    composing, 45  
    creating, 89  
    MatrixTransform, 43-44  
    RotateTransform, 39-40  
    ScaleTransform, 41, 49-50  
    SkewTransform, 42-43  
    TranslateTransform, 40  
transitions  
    MediaInfoDisplay control, creating, 420-421  
    RepeatButtons control, changing, 426-427  
Translate animation, 212-220  
TranslateTransform, 40  
transparency, setting, 69-70  
triggering  
    requests, 477  
    storyboards, 201-202  
TryFindResource extension method, 544-548  
TryFindResource method listing (24.18), 544  
Trying to register two event handler listing (21.2), 440  
tutorials for Silverlight, 18-19  
Two event handlers listing (21.5), 442  
Two events and a helper method listing (23.7), 506  
Two methods listing (24.21), 546  
Two new classes listing (23.5), 505  
Two new classes listing (23.6), 505  
Two new properties listing (14.11), 270  
Two new properties listing (22.15), 481  
type attribute (object tag), 109  
types  
    anonymous, 476  
    JavaScript, 132

.NET, 163  
    converting to other types, 165  
    floating point types, 164-165  
    integer types, 163

## U

UIElement class, 292-293  
UIs (user interfaces), updating, 486-488, 514-516  
Undefined resource listing (17.5), 347  
undefined values (JavaScript), 134  
unhandled exceptions, catching, 522-524  
unit tests, 541  
    creating, 541-549  
    framework, installing, 541  
    recursion, implementing, 545-548  
    reusing, 549  
    test applications, creating, 542-544  
    test passes, 545  
    TryFindResource extension method, 544-548  
unmanaged code, 155  
Updated Border listing (18.5), 367  
updating  
    client applications, 494-497  
    Service Reference, 527-528  
    user interfaces, 486-488, 514-516  
User class, 517  
    constructors, 271-272  
    properties, 270  
    ToString method, 270-271  
User class constructor listing (23.23), 517  
user controls, 98, 100-101, 405  
user interfaces  
    describing with XML, 23  
    updating, 486-488, 514-516  
user objects, extending, 270-274  
UserControl  
    creating, 387-388, 390-396  
    data objects, creating, 388-391

- DataTemplate, designing for, 399-401
- images, handling, 399-400
- ItemsSource property, adding, 393-396
- media files, moving, 401-402
- namespaces, adding, 397
- Preview tab, 399
- properties, 98, 100-101
- real data, connecting to, 403
- SelectionChanged event, raising and handling, 403-405
- test data source, 401
- videos, handling, 400-401
- XAML media instances, creating, 397-399
- UserInformation class, 492**
- UserInformation class listing (22.27), 492**
- users**
  - comparing, 272-274
  - saving, 275-276
  - servers, moving to, 488-489
- Using a DatePicker and catching errors, C# code listing (16.5), 327**
- Using a DatePicker and catching errors, XAML markup listing (16.4), 327**
- Using a generic class listing (20.15), 437**
- Using a GridSplitter listing (16.3), 324**
- Using a resource in code listing (17.3), 345**
- using directive, 176-177**
- using statements, 183**
- Using the Bubbly style listing (24.27), 551**
- Using the DoubleToThicknessConverter listing (18.8), 369**
- Using the explicit data context in the Border listing (18.10), 371**
- Using the explicit data context in the Slider listing (18.11), 371**
  
- V**
- ValidateUser method, 277, 493-495**
- ValidateUser method listing (22.28), 493**
- ValidateUserCompleted event handler, 496**
- ValidateUserCompleted event handler listing (22.32), 496**
- validation errors, handling, 533-534**
- Validation properties listing (24.5), 534**
- value coercion, 321-322**
- values**
  - bindings, converting in, 368-369
  - JavaScript, 131
- variables**
  - inspecting, 462-463
  - JavaScript, 452
  - modifying, 462-463
- vector graphics, 72-73**
- versioning**
  - application assemblies, 434
  - components, 434
- versions of .NET, 154**
- VideoBrush property, painting videos with, 84-85**
- VideoPlayer\_Ended event handler, 539**
- VideoPlayer\_Ended event handler listing (24.13), 539**
- videos**
  - auto-play, disabling, 80
  - captions, adding, 234
  - chapters, adding, 233-234
  - cropping in Expression Encoder, 229
  - encoding for Silverlight, 235-236
  - full screen, enabling, 237
  - handling, UserControl, 400-401
  - hiding from mouse input, 83
  - importing with Expression Encoder, 226-227
  - looping, 80-81
  - overlays
    - adding, 242
    - advanced properties, setting, 242
    - XAML overlays, adding, 243-244
  - pausing, 82-83
  - players, allowing viewer to select, 245-250
  - playing, 82-83
  - profiles, 227-228
  - publishing on website, 240-241
  - reflections, adding, 104
  - size of, changing in Expression Encoder, 228

- streaming, 239-240
- writing with, 84-85
- visibility, 177**
- Visual Studio. See also Visual Studio 2008**
  - applications
    - debugging, 458-465
    - refactoring, 430-434
  - Debug toolbar, 460
  - fonts, embedding in, 356
  - installing, 54
  - websites, publishing directly from, 457-458
- Visual Studio 2008, 17**
  - installing, 156
  - Intellisense, 162
  - Silverlight 2 applications, creating, 156-157, 162
    - compilation errors, finding, 158
    - Silverlight project files, 160-161
  - Web Site project files, 159

## W

- Walking the tree listing (17.4), 346**
- walking the tree of elements, 346-347**
- WCF (Windows Communication Foundation)**
  - requests, making, 488-497
  - services
    - creating, 491-492
    - implementing, 492-494
- WCF Unleashed, 489**
- web applications, security of, 10-11**
- web browsers, caching applications, 439**
- web clients**
  - JavaScript, 261
  - passwords, storing on, 270
- web projects, adding applications to, 282**
- web servers**
  - connecting to, 123
  - FTP client, configuring, 122-123
  - requirements, 121
  - Silverlight files, testing, 126
  - web space, obtaining, 120-121

- web sites**
  - debugging, 463
  - publishing with Visual Studio, 457-458
- web space, obtaining, 120-121**
- web-based media, 86-87**
- websites, modifying during runtime, 150**
- websites, placing on hard disks, 282**
- while loop, 169-170**
- windowless parameter (object tag), 110**
- Windows Presentation Foundation, 345**
- wiring up scripts, 287-288, 421-422**
- WPF/E (Windows Presentation Foundation Everywhere), 3**
- writing**
  - to data objects, bindings, 366-367
  - to isolated storage, 195-197
  - text, 327-32

## X-Y-Z

- XAML (eXtensible Application Markup Language), 21-23**
  - Border element, 309
  - code-behind, comparing with, 293-296
  - elements, 294
  - files, converting to, 551-552
  - Grid layout, 303
  - login links, adding to user interfaces, 280-282
  - markup, testing
    - with KaXaml, 29-30
    - with SilverlightPad, 27-29
  - media instances, creating in, 397-399
  - overlays, adding to videos, 243-244
  - paths, 95
  - Popup, 310
  - properties, setting, 30-31
  - resources
    - creating, 549-552
    - defining in, 344
    - online resources, 550-551

- purchasing, 549-552
- using in, 345-347
- scenes, composing, 32-34
- ScrollViewer, 307
- StackPanel, 301
- Style class, defining in, 348
- TabControl, 336
- templates, purchasing, 551
- TextBlock control, 317
- XAML Archive, 551**
- XAML Elements listing (15.1), 294**
- XAML scene for dragging listing (15.3), 298-299**
- XAML test markup listing (21.9), 444**
- .xap files, 110, 119-120
- Xceed, 552
- XML (eXtensible Markup Language)**
  - comments, adding, 27
  - data sources, 386
  - document characteristics, 22
  - files
    - loading, 467-480
    - reading, LINQ, 473-476
    - namespaces, 23-25
    - user interfaces, describing, 23
- XML Media Information File listing (22.1), 468**
- Zip files**
  - creating, 480
  - loading, 480-483
  - reading, 483-485