Pro Silverlight for the Enterprise

ISBN-13 (pbk): 978-1-4302-1867-8

ISBN-13 (electronic): 978-1-4302-1868-5

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

Lead Editors: Matthew Moodie, Ewan Buckingham
Technical Reviewer: Fabio Claudio Ferracchiati
Editorial Board: Clay Andres, Steve Anglin, Mark Beckner, Ewan Buckingham, Tony Campbell, Gary Cornell, Jonathan Gennick, Michelle Lowman, Matthew Moodie, Jeffrey Pepper, Frank Pohlmann, Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh
Project Manager: Kylie Johnston
Copy Editor: Ami Knox
Associate Production Director: Kari Brooks-Copony
Production Editor: Kelly Gunther
Compositor and Artist: Kinetic Publishing Services, LLC
Proofreader: Patrick Vincent
Indexer: BIM Indexing & Proofreading Services
Cover Designer: Kurt Krames
Manufacturing Director: Tom Debolski

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail orders-ny@springer-sbm.com, or visit http://www.springeronline.com.

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

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

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

The source code for this book is available to readers at http://www.apress.com.

PART 1

Exploring Silverlight

CHAPTER 1

Understanding Silverlight

The Web 2.0 concept drives enterprises to develop browser-based Rich Internet Applications (RIAs) that support different market needs, such as platform (operating system, browser, and device) independence, information collaboration, social-networking, rich user interfaces, high performance, and high security. Silverlight 2 is a Microsoft .NET Framework–based technology platform enabling us to develop loosely coupled plug-ins and RIAs at the enterprise level. Silverlight helps enterprises to achieve the Web 2.0 concept by implementing RIAs in a very agile and cost-effective way that can provide maximum customer satisfaction and thus can drive enterprises to transform information system centers from cost centers to profit-making centers.

This chapter is divided into two major sections. It starts by explaining concepts of Web 2.0 and RIAs. I will discuss the architecture and available technologies in the market that basically drive the development of advanced RIAs. The latter part of the chapter focuses on introducing and understanding Microsoft's Silverlight technology and its components that allow organizations to develop advanced enterprise RIAs.

Web 2.0 and RIAs

Today users' expectations are increased in terms of collaboration, usability, performance, flexibility, user-level customization, and security for two reasons. The first reason is the use of the Internet and web-based applications have become part of users' day-to-day life. The second reason is the use of IT to execute and integrate business processes within and across organizations is extensive. Additionally, in today's digital economy era, the increased use of digital media makes all types of information available in digital format and can easily lead to information overload if organizations don't manage that information properly. Organizations also need to implement governance policies for data retention and government compliance. Innovative Web 2.0 and RIA concepts drive organizations to develop and deploy second-generation cross-platform and cross-device web applications meeting many of today's market needs.

Figure 1-1 presents a view of the Microsoft Surface home page (`http://www.microsoft.com/surface`), which basically demonstrates the different opportunities of Web 2.0 and RIAs by providing a rich, interactive user interface to explain how the new Microsoft Surface platform can deliver rich services.

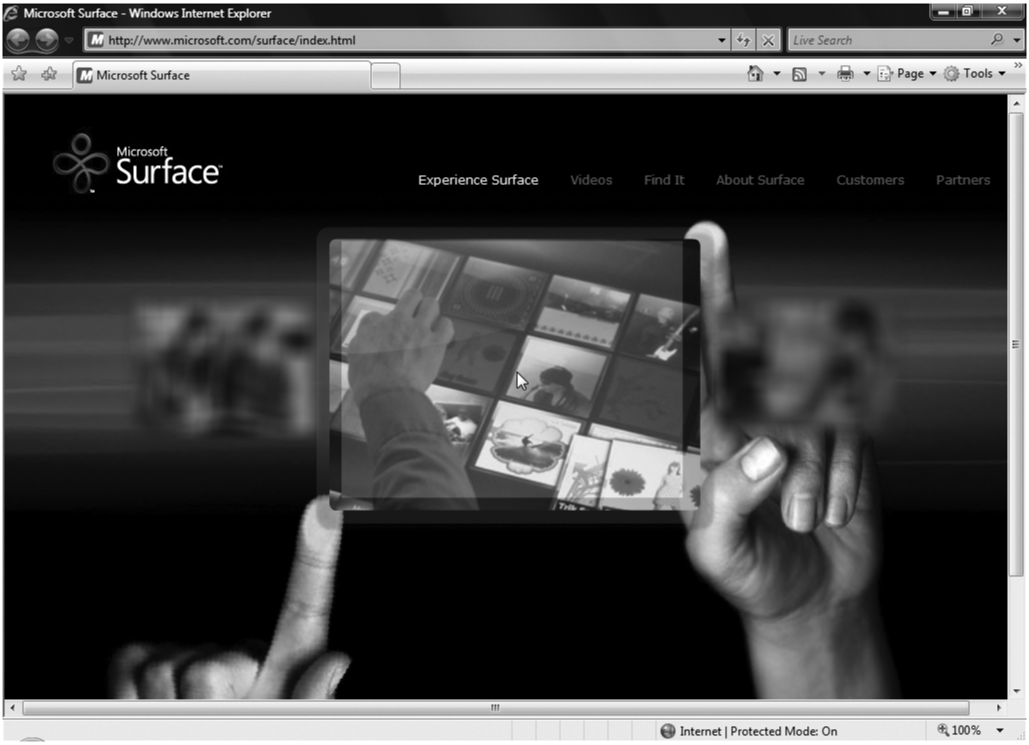


Figure 1-1. *Microsoft Surface home page: an example of an RIA*

Key Attributes of Web 2.0

Although there is no specific definition for Web 2.0, one of the most accepted definitions is by Tim O'Reilly:[1]

> *Web 2.0 is the business revolution in the computer industry caused by the move to the Internet as platform, and an attempt to understand the rules for success on that new platform.*

As shown in Figure 1-2, the following are key attributes of Web 2.0:

- Improved usability via *rich and creative user interface* that provides an interactive rich experience, which is very similar to desktop rich client applications. Web 2.0 applications *process audio and video seamlessly*.
- *Modular* and loosely coupled web-based application architecture that supports flexibility, user-level customization, and seamless content and process integration.
- *High-performing* rich and interactive web-based applications due to the following capabilities:
    - Separation between the *loosely coupled presentation layer and the data access layer*. This enables client-side processing and reduces overall server-side processing by avoiding round-trips.

[1] Tim O'Reilly, `http://radar.oreilly.com/2006/12/web-20-compact-definition-tryi.html`, 2006

  - *Asynchronous communication* in the background to process different data process–related requests between client and server.
  - *Metadata management and data indexing* capabilities to structure and organize data that help to overcome information overload by providing the right information at the right time.

- *Collaboration platform* to share information among diverse and geographically separate groups.
- *Simplified and standard distribution protocols* and *metadata management* enables secured social-networking.
- *Platform-independent* technology that is cross-platform (i.e., available on different operating systems—Microsoft Windows, Apple Mac, Linux), cross-browser (i.e., available on different Internet browsers—Microsoft Internet Explorer, Mozilla Firefox, Apple Safari, Google Chrome), and cross-device (i.e., available on computers, mobile devices).

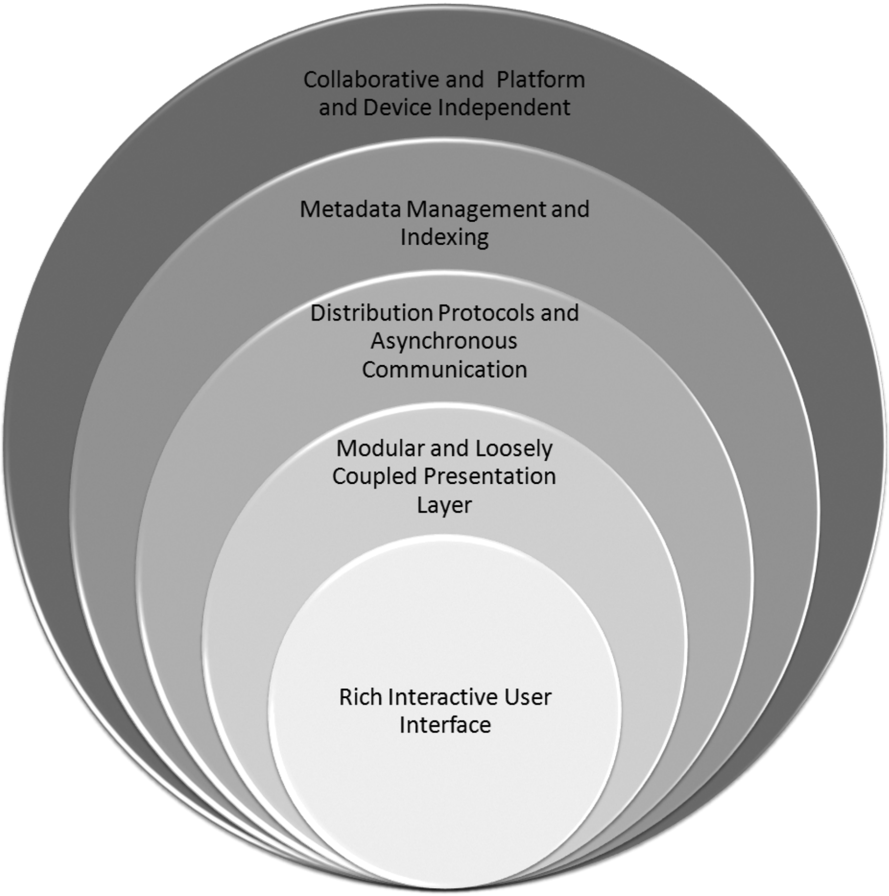

Collaborative and Platform and Device Independent
Metadata Management and Indexing
Distribution Protocols and Asynchronous Communication
Modular and Loosely Coupled Presentation Layer
Rich Interactive User Interface

Figure 1-2. *Key attributes of Web 2.0*

Rich Internet Applications

RIAs are web-based applications incorporating the previously described key attributes of the Web 2.0 concept. Following is a simplified definition of RIA:[2]

> *Rich Internet applications (RIAs) are web applications that have the features and functionality of traditional desktop applications.*

From the end users' perspective, RIAs enable end users to collaborate information (content and media) securely and provide effective data search and social-networking capabilities with rich user interface. RIAs' user interfaces are usually high performing, interactive, and intuitive, and as such deliver rich user experiences.

As defined in a white paper published by Macromedia in 2002,[3] RIAs from a technology perspective should have the following capabilities:

- Provide an efficient, high-performance runtime for executing code, content, and communications.
- Integrate content, communications, and application interfaces into a common environment.
- Provide powerful and extensible object models for interactivity.
- Enable rapid application development through components and reuse.
- Enable the use of web and data services provided by application servers.
- Embrace connected and disconnected clients.
- Enable easy deployment on multiple platforms and devices. If designed correctly, RIAs can support the concept of development and distribution of service-oriented software applications as software as a service (SaaS).

■**Note** SaaS is a software deployment approach for software providers to deploy software as a service for public consumption. The deployed software as a service can be consumed by different consumers based on the subscription policies defined by providers.

RIA Architecture

It is essential to understand the architecture of traditional web-based applications before you understand the architecture of Web 2.0 and RIAs.

Figure 1-3 shows a traditional web application's architecture. As shown in the figure, these traditional client applications follow the client-server model to perform all the user requests and actions on the server side mostly in synchronous mode.

[2] Wikipedia, `http://en.wikipedia.org/wiki/Rich_Internet_application`

[3] Jeremy Allaire, "Macromedia Flash MX—A Next-Generation Rich Client," `http://www.adobe.com/devnet/flash/whitepapers/richclient.pdf`

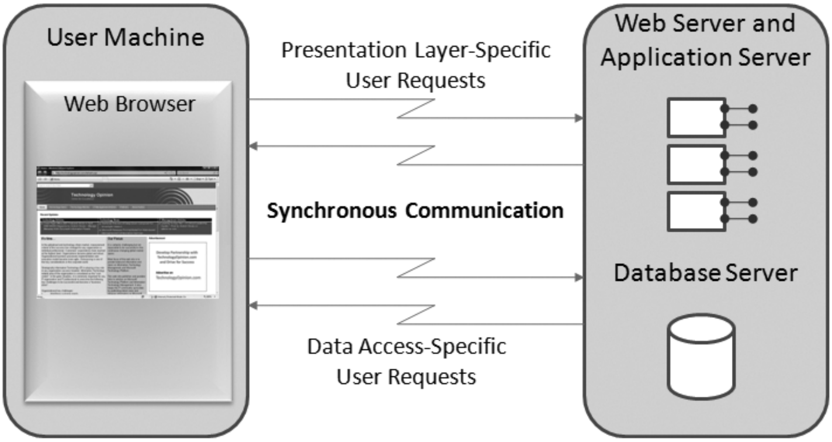


Figure 1-3. *Traditional (thin client) web-based application architecture*

Due to the sequential execution model and limitations of the rendered HTML user interface, traditional web applications lack the rich, creative, and interactive user interfaces of desktop applications. They lack the high-performing and modular application design that allows asynchronous communication between client and server in the background.

Figure 1-4 shows the typical RIA architecture. RIAs follow a hybrid approach to process user requests and actions. The presentation layer–related user requests and actions are performed at the web client side, and the data process–related user requests and actions are mainly performed at the server side asynchronously.

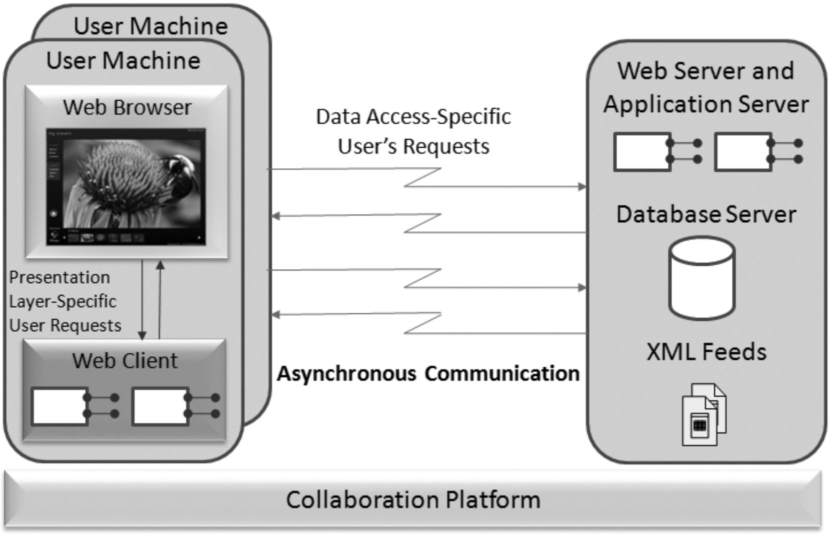


Figure 1-4. *RIA architecture*

The key difference between traditional web-based applications and RIAs is the presence of the web client layer, which is also often called the *client engine*. The web client layer helps to process the RIA's presentation layer–related requests and actions on the user machine and leverages the resources of the end user device in a sandbox environment. It also enables the asynchronous communication between the client and server to fetch and cache data before it is needed.

■ **Note** The *sandbox environment* is a mechanism to provide a restricted, isolated, and secured environment on the client machine to download and execute untrusted RIA components and code securely without accessing and having an impact on key client machine resources.

The Five Usability Dimensions for RIAs

Before developing any application, it is critical to understand the vision and scope of the application. In addition, especially for RIAs, it is important to choose the right technology platform and set of technology components because RIA technology is evolving all the time.

A paper published by Keynote Systems[4] defines four key dimensions of application usability that are essential for measuring the success of a web application—*availability*, *responsiveness, clarity*, and *utility*. I would add *safety* as a fifth usability dimension. Figure 1-5 shows the five dimensions of application usability.

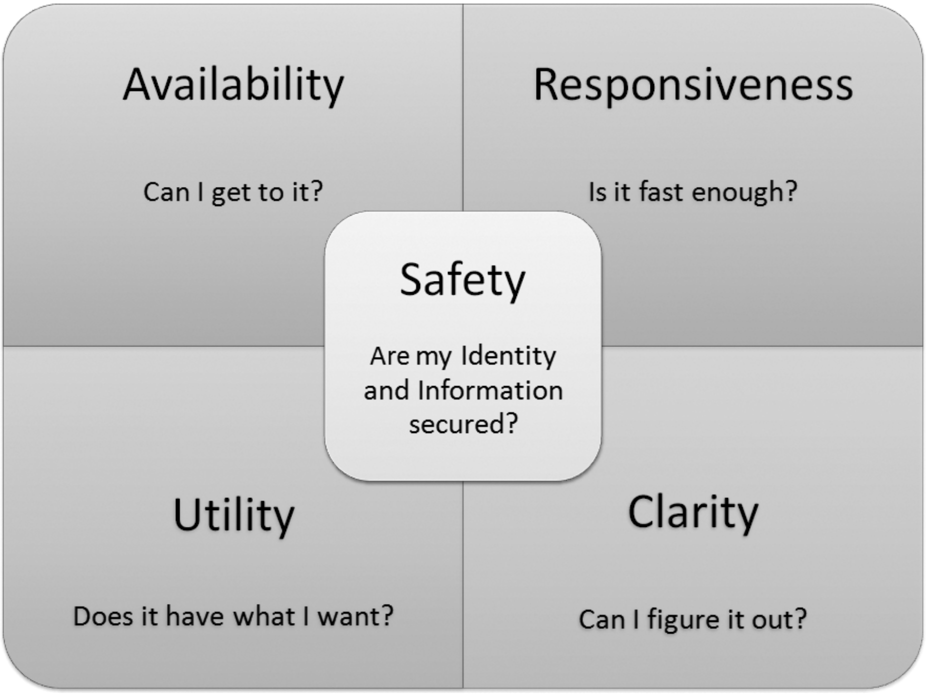


Figure 1-5. *The five dimensions of application usability*

[4] Chris Loosely, "Application usability dimensions—Rich Internet Applications: Design, Measurement, and Management Challenges," `http://www.keynote.com/docs/whitepapers/RichInternet_5.pdf`, 2006

I will cover the five dimensions of application usability in Chapter 4 in more detail. However, I wanted to mention them in this first chapter because I think it is important to know and keep in mind these usability dimensions before you really start exploring the Silverlight technology platform.

Technologies That Support RIAs

So far we have discussed the high-level vision and key attributes of Web 2.0 and the key features of RIAs. Unfortunately, there is no single technology available in today's market to develop RIAs fulfilling the Web 2.0 vision. This section of the book provides a high-level overview of the available technologies supporting the development of rich, interactive RIAs.

Microsoft ASP.NET AJAX

As described on the Microsoft ASP.NET web site:[5]

> *ASP.NET AJAX is a free scripting framework for quickly creating efficient and interactive Web applications that works across all popular browsers.*

Microsoft ASP.NET AJAX is an enhanced version of JavaScript. Its client- and server-side libraries are tightly integrated with Visual Studio 2008 and is included as part of the Microsoft .NET Framework 3.5 SDK. The client-side library allows you to implement client-level processing such as end-user-entered information processing and validation, refreshing a portion of the web page and developing rich, interactive user interfaces. You can also efficiently integrate the client-side library components with the server-side ASP.NET controls library in asynchronous mode.

However, ASP.NET AJAX is not an ideal solution for developing enterprise-level RIAs. The key technology driver of ASP.NET AJAX is scripting. In general, script-based web applications face several challenges due to different browser settings (e.g., JavaScript is not enabled by default) on PCs and mobile devices. ASP.NET AJAX supports only limited features of RIAs and does not support effective multimedia integration, managed code-behind integration, or metadata and information management.

ASP.NET AJAX does have asynchronous communication capabilities. However, due to the nature of the implementation, the web application must continuously pull data from the server to get the latest updates, which ultimately slows down overall application performance.

In conclusion, ASP.NET AJAX is a good place to start to transform your traditional web-based applications to RIAs. Often scripting is not always the best strategy for enterprises to develop secured and scalable RIAs. ASP.NET AJAX's script-based approach and the previously mentioned limitations raise many concerns about the security, compatibility, and maintainability of ASP.NET AJAX web applications. Thus many enterprises may hesitate to develop enterprise-level RIAs using only ASP. NET AJAX technology.

The PageFlakes web site (`http://www.pageflakes.com/Default.asp`) is a very good example demonstrating the use of ASP.NET AJAX. This page demonstrates key RIA features like modular design, asynchronous communication, and user customization with the use of ASP.NET AJAX.

[5] Microsoft ASP.NET AJAX Definition, `http://www.asp.net/ajax/default.aspx?wwwaspnetrdirset=1`

Java Applets

As described on the Java Sun web site:[6]

> *[A Java] applet is a special kind of Java program that a browser enabled with Java technology can download from the Internet and run. An applet is typically embedded inside a web page and runs in the context of the browser.*

A Java applet is a browser plug-in that is cross-browser and cross-platform. Java applets use the Java Virtual Machine (JVM) incorporated within the end user browser to execute and run in a sandbox environment on the end user machine.

Using the Java platform, it is possible to develop interactive and heavy graphical RIAs with offline capabilities. With the use of Java server-side component libraries, they can perform asynchronous communication.

Java applets have limitations similar to the ASP.NET AJAX technology. Users' browsers should be Java enabled, and the Java Runtime Environment (JRE) needs to be installed on users' machines to run Java applets successfully. Java applets support only limited features of RIAs. It would be challenging to develop a rich, interactive user interface with the media streaming and metadata management capabilities of RIAs in rapid application development (RAD) mode using the Java applets and supporting Java technology components.

Adobe Flash, Adobe Flex, and Adobe AIR

As described on the Adobe Flash product site:[7]

> *[Adobe Flash] software is an advanced authoring environment for creating rich, interactive content for digital, web, and mobile platforms.*

Adobe Flash is one of the most popular web browser plug-ins for displaying animation and audio and video streaming on web sites. Adobe Flash provides ActionScript—a scripting language—for developing animation and multimedia streaming.

Adobe Flash does not provide an efficient development platform to build interactive and complex RIAs, but it is the most suitable tool for creating interactive plug-ins for web applications.

To meet the need for an efficient development platform for enterprise-level RIAs, Adobe has developed Adobe Flex, which basically is an enhancement to Adobe Flash. As described on the Adobe Flex product site:[8]

> *[Adobe Flex] is a . . . free open source framework for building and maintaining expressive web applications that deploy consistently on all major browsers, desktops, and operating systems.*

The Adobe Flex development platform provides a rich UI component library and uses MXML, a declarative XML-based language to develop rich, interactive user interfaces. The ActionScript programming language is used to implement the business service layer.

Adobe is making some serious moves to enhance its platform so you can use it not just for developing rich, interactive user interfaces and integrating multimedia elements, but also for

[6] Java Applet Definition, `http://java.sun.com/docs/books/tutorial/deployment/applet/index.html`

[7] Adobe Flash Definition, `http://www.adobe.com/products/flash/`

[8] Adobe Flex Definition, `http://www.adobe.com/products/flex/`

developing desktop applications that can seamlessly integrate with RIAs. In addition to Adobe Flash and Adobe Flex, in February 2008 Adobe introduced Adobe AIR for developing desktop applications that we can extend as RIAs. As described on the Adobe AIR product site:[9]

> *[Adobe AIR] lets developers use proven web technologies to build rich Internet applications that deploy to the desktop and run across operating systems.*

Figure 1-6 shows the eBay desktop application developed utilizing Adobe AIR and Adobe Flex. The eBay application demonstrates the power of the offline capabilities of the RIA.

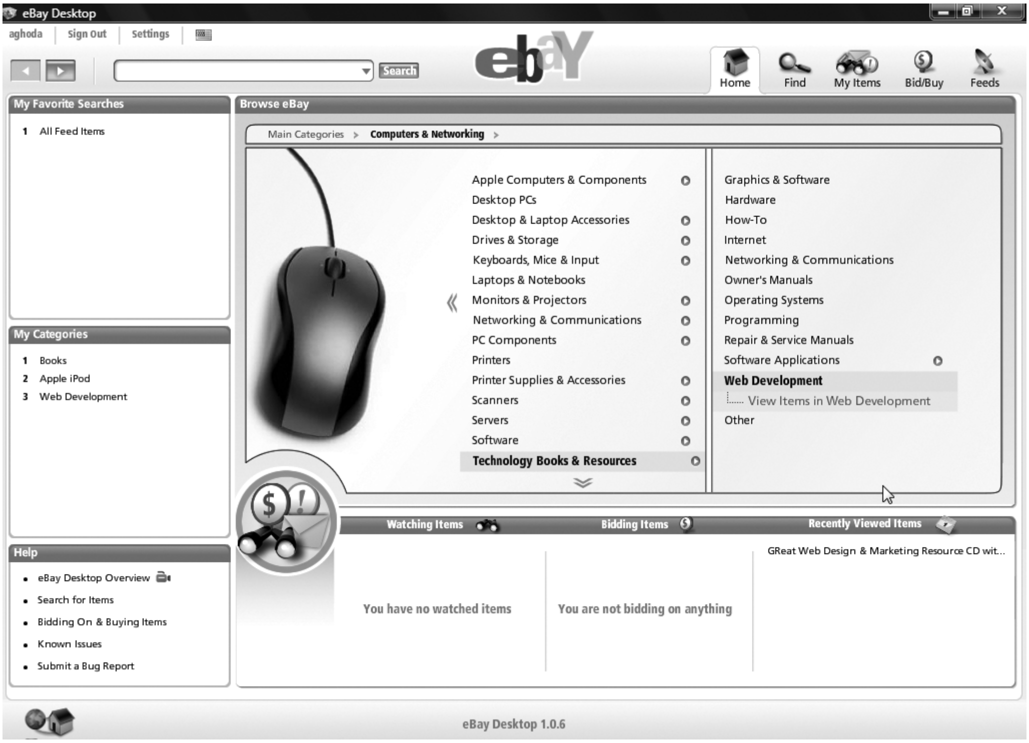


Figure 1-6. *eBay desktop application developed using Adobe Flex and Adobe AIR*

Microsoft Silverlight

As described on the Microsoft Silverlight web site:[10]

> *Microsoft Silverlight is a cross-browser, cross-platform, and cross-device plug-in for delivering the next generation of .NET-based media experiences and rich interactive applications for the Web.*

Microsoft Silverlight is a subset of Windows Presentation Foundation (WPF), which is part of .NET Framework 3.5. Silverlight is integrated with the broad range of Microsoft tools and services

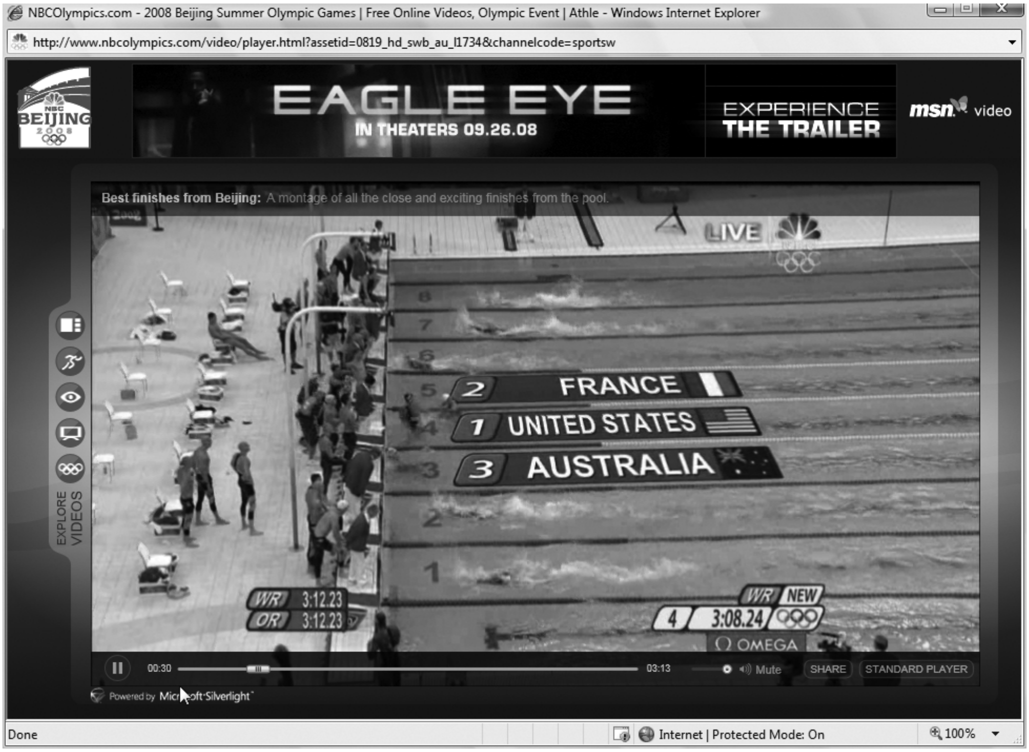
[9] Adobe AIR Definition, `http://www.adobe.com/products/air/`

[10] Microsoft Silverlight Definition, `http://silverlight.net/`

like Microsoft Visual Studio 2008, Microsoft Expression Blend, Microsoft Deep Zoom Composer, and Microsoft Silverlight Streaming by Windows Live for the easy development and deployment of Silverlight-based multimedia RIAs.

Microsoft partnered with NBC Universal for the Beijing 2008 Olympic Games[11] and successfully demonstrated the capabilities and power of the Microsoft Silverlight technology in the commercial RIA market. NBC's Olympics web site (`http://www.nbcolympics.com`) featured more than 2,000 hours of live content and 3,000 hours of on-demand video for Olympics lovers. The Silverlight technology enabled NBC to develop the Control Room feature, presenting video picture-in-picture capability that let viewers watch up to four events at the same time.

Figure 1-7 presents a page from the NBC Olympics web site featuring Silverlight-based video streaming.



NBCOlympics.com - 2008 Beijing Summer Olympic Games | Free Online Videos, Olympic Event | Athle - Windows Internet Explorer

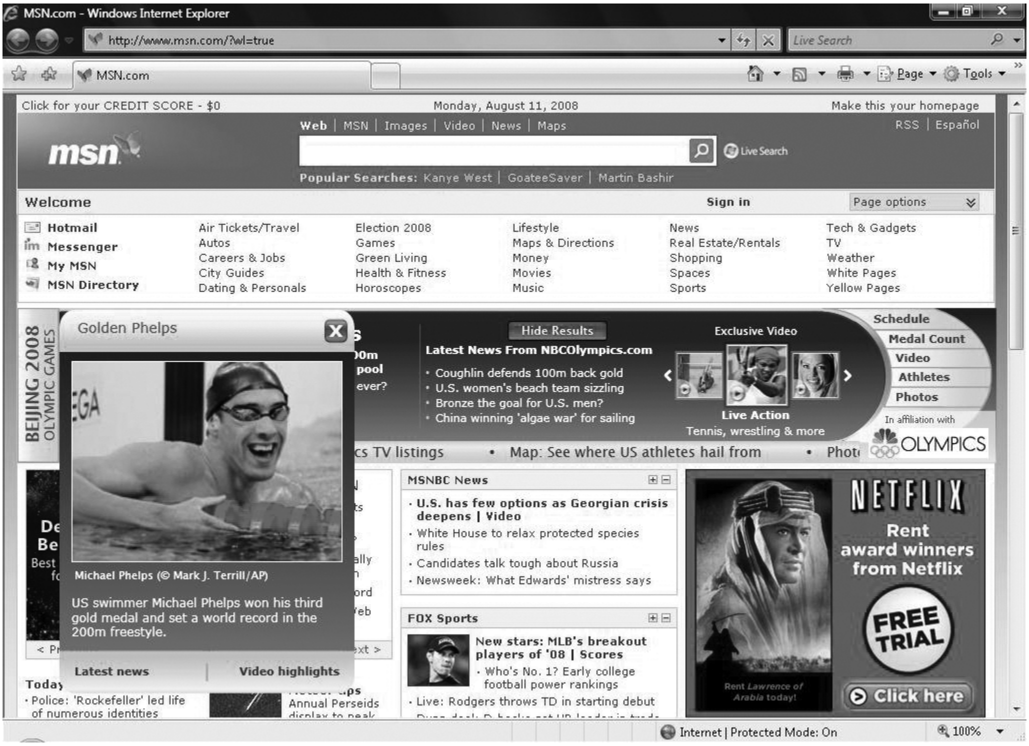
http://www.nbcolympics.com/video/player.html?assetid=0819_hd_swb_au_l1734&channelcode=sportsw

Best finishes from Beijing: A montage of all the close and exciting finishes from the pool.

Figure 1-7. *Beijing 2008 Olympics game videos on NBC's Olympics web site featuring Microsoft Silverlight*

[11] Ina Fried, "What It Takes to Bring the Olympics to the PC," `http://news.cnet.com/8301-13860_3-10003752-56.html?tag=nefd.lede`, 2008

Figure 1-8 shows the Silverlight Olympic 2008 plug-in on the MSN web site (`http://www.msn.com`) during the Beijing 2008 Olympics.



Michael Phelps (© Mark J. Terrill/AP)

US swimmer Michael Phelps won his third gold medal and set a world record in the 200m freestyle.

Latest News From NBCOlympics.com

- Coughlin defends 100m back gold
- U.S. women's beach team sizzling
- Bronze the goal for U.S. men?
- China winning 'algae war' for sailing

cs TV listings • Map: See where US athletes hail from • Phot

- U.S. has few options as Georgian crisis deepens | Video
- White House to relax protected species rules
- Candidates talk tough about Russia
- Newsweek: What Edwards' mistress says

New stars: MLB's breakout players of '08 | Scores

- Who's No. 1? Early college football power rankings
- Live: Rodgers throws TD in starting debut

Today

- Police: 'Rockefeller' led life of numerous identities

Annual Perseids display to peak

Figure 1-8. *Olympics Silverlight plug-in on the MSN web site*

Applying a Balancing Act When Selecting Technology Components

It is very clear from the discussion in the preceding sections that one technology is not enough to develop intuitive RIAs. Although I will cover this in more detail in the later part of the book (in Chapters 3 and 4), in summary it is recommended you apply a balancing act (see Figure 1-9) in the selection of the right set of technology components to develop and deploy RIAs successfully. The selection of technology components must align with your organization's strategic vision, your customers' requirements, the product roadmap, and the technology roadmap for long-term success and to gain maximum ROI.

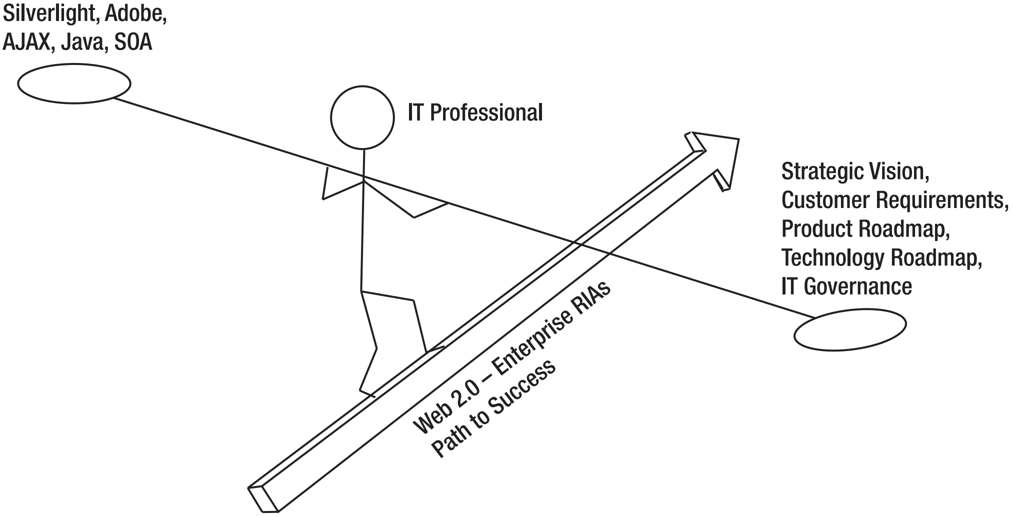


Figure 1-9. *Applying a "balancing act" in the selection of the right Web 2.0 technologies to develop Enterprise RIAs*

Examining the Silverlight Platform

This book is mainly focused on the Microsoft Silverlight technology. You will see how Silverlight can be utilized to develop service-oriented enterprise RIAs that fulfill the Web 2.0 vision. Before we dive into the details of Silverlight for the enterprise, let's take a high-level look at Silverlight and its available versions and key components. This book assumes that you are already aware of the Silverlight technology and how to build basic applications using Silverlight 2. If you are a newcomer to Silverlight and the following section is not sufficient to get started with Silverlight 2, please read the online materials and books mentioned in the "Additional References" section of this chapter.

As described earlier, Microsoft Silverlight provides a platform to develop cross-browser, cross-platform, and cross-device RIAs. All versions of Silverlight are a subset of WPF and use XAML (Extensible Application Markup Language), an XML-based language that facilitates externalized and loosely coupled definitions of the user interface and related style sheets.

Different Versions of Silverlight

At present, two versions of Microsoft Silverlight—Silverlight 1 and Silverlight 2—are available.

Silverlight 1

After having a great showcase at Mix 2007 in May 2007, Microsoft Silverlight 1 (earlier known by the code name WPF/E—Windows Presentation Foundation/Everywhere), along with Microsoft Expression Encoder 1.0, was released in September 2007 as the first version of Silverlight. Silverlight 1 is a very rudimentary release of the product and mainly focuses on the development of Rich Media Internet Applications (RMIAs). It contains rich media presentation framework–related core components like UI controls handling the rendering of 2D vector graphics, media, animation, and simplified user input management. The Silverlight Media Player controls support the WMA, WMV, and MP3 formats of media files. Silverlight 1 is also capable of using JavaScript with Document Object Model (DOM) integration to manipulate the UI.

■Note The RMIA is a type of RIA containing a subset of RIA attributes. RMIA-type applications/plug-ins are mainly focused on media (audio and video) streaming features.

Even though Silverlight 1 has limited features and limited support for development, when it was released, it was highly respected in the commercial market and created a buzz. Along with the Major League Baseball web site (`http://www.mlb.com`), many major commercial sites adopted Silverlight 1 to provide rich media experiences. Microsoft also provided some great sample sites like the Tafiti search front end (`http://www.tafiti.com/`), powered by the MSN search engine to explain the Silverlight technology and RIA concepts.

Silverlight 2

The next landmark version of Silverlight was released as Silverlight 2 in October 2008. It is a big leap from the first basic version to version 2. Silverlight 2 is enhanced in the following areas:

- Enhanced support for the Microsoft .NET Framework 3.5 with the Common Language Runtime (CLR). The support for the CLR enables the integration of Microsoft .NET managed code-behind using default Microsoft .NET class libraries in Silverlight 2 projects.
- Enhanced Base Class Library (BCL) and Language-Integrated Query (LINQ) integration to develop complex enterprise RIAs.
- Additional UI components to support the development of RIAs featuring rich multimedia functionalities.
- Enhanced media management supporting secured multimedia streaming.
- Enhanced networking support including policy-based cross-domain networking to support different types of application deployment.
- Support for the open source and cross-platform Eclipse development platform via Eclipse Tools for Microsoft Silverlight: eclipse4SL (`http://www.eclipse4SL.org`).
- Improvement of the interoperability features of Silverlight due to the Silverlight XAML schema vocabulary specification (MS-SLXV) released under the Open Specification Promise (OSP) (`http://msdn.microsoft.com/en-us/library/dd361850(PROT.10).aspx`).

Silverlight 2 and Development Tools Quick Links

Along with Silverlight 2, you can download the following supporting development tools from `http://silverlight.net/GetStarted/`:

- *Microsoft Silverlight Tools for Visual Studio 2008 SP1* enables you to develop Silverlight 2–based applications using Visual Studio 2008 SP1 IDE.
- *Microsoft Expression Blend 2 SP1* is tightly integrated with Visual Studio 2008 SP1 and allows artists and designers to create rich XAML-based user interfaces for Silverlight applications.
- *Deep Zoom Composer* allows professionals to create and prepare images to implement the Deep Zoom feature within Silverlight applications.
- *Microsoft Expression Encoder 2* contains Silverlight Media Player templates to author, manage, and publish media for Silverlight applications. Visit `http://www.microsoft.com/expression` and look for the Microsoft Expression Encoder within the Express product suite.
- *Eclipse Tools for Microsoft Silverlight (eclipse4SL)* enables development of Silverlight applications using the Eclipse open source and cross-platform development platform. You can install this tool set by visiting `http://www.eclipse4sl.org/download/link`.

Silverlight 3

Microsoft never stops improving its technology. By the time you have this book in your hands, it is very likely that the Silverlight 3 beta release will be available.

As mentioned by the corporate vice president of Microsoft's developer division on his blog,[12] Silverlight 3 is to be released in the latter part of 2009. Based on the currently available knowledge, the new version will mainly include enhancements in graphics capabilities (support for 3D graphics), media management (support for H.264 video), and application development areas (additional controls and enhanced data binding support). There are also plans for the enhancements of the Visual Studio and Expression Blend development tools to develop Silverlight RIAs more effectively.

A few introductory sessions on Silverlight 3 are planned for the MIX09 conference (`http://visitmix.com/News/Silverlight-3-Sessions-at-MIX09`).

■**Note** This book is written in the context of utilizing enterprise capabilities to develop Silverlight-based service-oriented RIAs and adopting Silverlight in your organization. All technical concepts are Silverlight version (specifically, Silverlight 2 and 3) agnostic. Based on the current knowledge on the enhancements in Silverlight 3, the RIAs developed in this book should work with minimal or no changes.

---

[12] Scott Guthrie, "Update on Silverlight 2—and a Glimpse of Silverlight 3," `http://weblogs.asp.net/scottgu/archive/2008/11/16/update-on-silverlight-2-and-a-glimpse-of-silverlight-3.aspx`

Silverlight Architecture

Figure 1-10 shows the architecture of Silverlight. It also highlights new enhancements introduced in the Microsoft Silverlight 2 version.

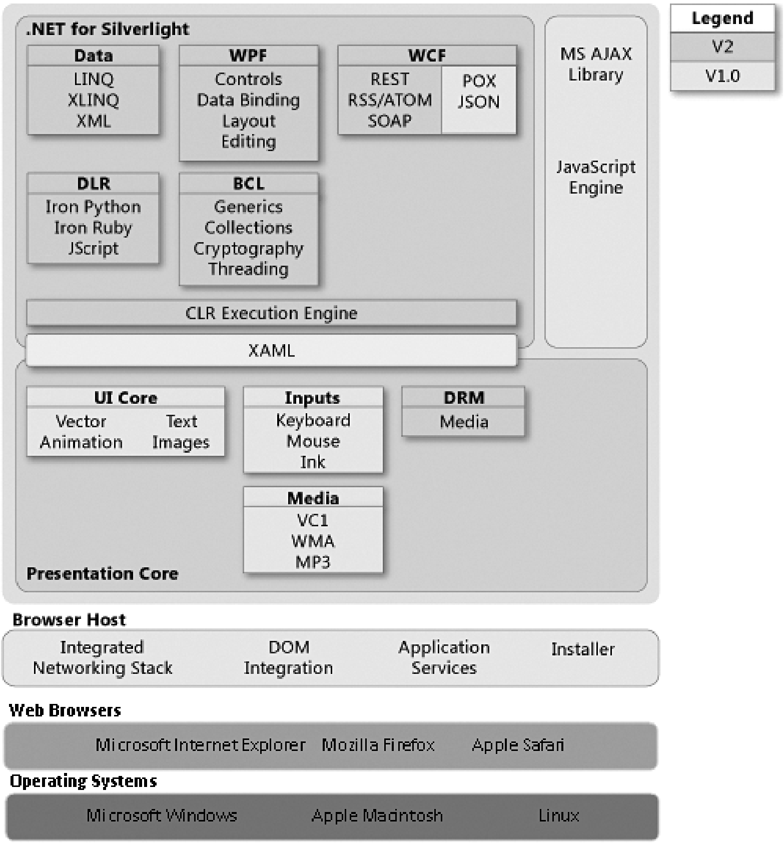


Figure 1-10. *Silverlight architecture*[13]

■ **Note** Figure 1-10 is modified from the original diagram to demonstrate the concept of platform independence.

Silverlight Is Platform Independent

Silverlight is a lightweight, platform-independent technology. Silverlight applications are

- Cross-platform and can run successfully on different operating systems—Microsoft Windows and Apple Mac
- Cross-browser and can run successfully on different well-known web browsers—Microsoft Internet Explorer, Mozilla Firefox, Apple Safari, and Google Chrome
- Cross-device and can run successfully on PCs and handheld mobile devices

[13] Silverlight Architecture, `http://msdn.microsoft.com/en-us/library/bb404713(VS.95).aspx`

Presentation Core

Compared to Silverlight 1, Silverlight 2 includes significant enhancements in the areas of UI framework, media management, user controls and graphics, and animation support to implement the rich, interactive presentation layer of RIAs.

UI Framework

Silverlight 2 delivers a robust and platform-agnostic UI framework, which is based on .NET Framework 3.5, WPF, and XAML. As a subset of WPF, Silverlight ultimately enables us to reuse WPF-based components for web browser–based RIAs and desktop applications such as Vista gadgets. You can also convert Silverlight 2 projects to WPF projects with some effort in order to achieve reusability. However, Silverlight 2 does not support this feature completely, and future releases may possibly extend this reusability feature.

Media Management

Silverlight's capabilities for adaptive media streaming help to improve the synchronization of media, which ultimately provides high-performing and seamlessly integrated RMIAs by automatically adjusting the bit rates based on the network bandwidth. The introduction of digital rights management (DRM) for media streaming encourages enterprises distributing digital media to utilize Silverlight applications to develop media-protected RMIAs.

User Controls

As part of the default Silverlight XAML user controls and Silverlight toolkit, the Silverlight UI controls library contains a set of basic and advanced XAML UI controls, which fall into four main categories:

- Layout management XAML controls
- Form XAML controls
- Data manipulation XAML controls
- Functional XAML controls

These controls are available with full source code and with an OSI license that allows professionals to modify them to meet their expectations.

■**Note** The Open Source Initiative, or OSI (`http://www.opensource.org`), manages the Open Source Definition (OSD) and provides open source licenses to the products that pass the OSI approval process and comply with OSD.

You will see more details in the "Quick Overview of Silverlight 2's Key Components" section, which will walk you through development of a simple and basic user interface to reinforce some key concepts.

With the help of Expression Blend, Silverlight 2 supports user control templates, which allow us to modify the shape and look of control templates very easily. Silverlight 2 also enables the Visual State Manager (VSM) feature with the help of Expression Blend. The visual state and state transition features of VSM enable us to customize the interaction of the Silverlight control templates with their `Normal`, `MouseOver`, `Pressed`, `Disabled`, `Focused`, and `Unfocused` events.

Silverlight 2 downloads the core UI controls as part of the initial application installation to minimize runtime downloads. This feature improves overall application performance. However, you can decide during the design phase how and when to download components. Applying a balancing act, enhanced UI controls like DataGrid and TabControl can be referenced and downloaded at runtime when they are referenced.

Graphics and Animation Support

The Microsoft Silverlight Deep Zoom feature provides us the ability to smoothly present and navigate large amounts of visual information regardless of the size of the data, and optimizes the bandwidth available to download it.[14] A very good example of Deep Zoom is available on the Hard Rock memorabilia web site (http://memorabilia.hardrock.com/), developed by Vertigo and presented during the Microsoft Mix 2008 conference in March 2008.

Along with the Deep Zoom feature, Microsoft Silverlight supports object animation and embedded code-based animation to provide high-performing graphics and animation support.

Microsoft .NET for Silverlight

Compared to Silverlight 1, Silverlight 2 introduces tight .NET integration by providing support for the CLR. This support allows professionals to implement managed code-behind .NET integration and support for WCF/Web Services, LINQ, and service-oriented features.

Managed Code-Behind .NET Integration

Support for the.NET Framework 3.5 with the CLR, and thus support for .NET managed code-behind code using the default .NET class libraries, is a key enhancement in Silverlight 2. The CLR basically provides memory management, garbage collection, type-safety checking, and exception handling. Additionally, the BCL contains a set of components that provide basic programming capabilities like string handling, regular expressions, input and output, reflection, collections, and globalization.

The Dynamic Language Runtime (DLR) supports the dynamic compilation and execution of scripting languages like JavaScript and IronPython to develop Silverlight-based applications. It also includes a pluggable model for adding support for other languages to Silverlight.

■ **Note** IronPython is an implementation of the Python programming language integrated with the .NET Framework, which is compatible with the Python language. For more details, visit http://www.codeplex.com/IronPython.

Data/Information Integration

Silverlight 2 provides us with asynchronous, loosely coupled data-integration capabilities. The rich user controls library is capable of binding and presenting data in a rich manner. It is capable of integrating with WCF and Web Services via REST, WS*/SOAP, POX, RSS, and standard HTTP, enabling us to perform various data transactions with external data (e.g., XML, relational databases). And last but not least is its integration with ADO.NET data services, LINQ, LINQ to XML and XLinq for data transformation, and local data caching with isolated data storage capabilities, which empower the Silverlight technology platform to develop high-performing data and media-centric RIAs.

[14] Jaime Rodriguez, “A Deep Zoom Primer,” http://blogs.msdn.com/jaimer/archive/2008/03/31/a-deepzoom-primer-explained-and-coded.aspx

Networking Support

Silverlight applications are capable of background threading and asynchronous communication, which helps to separate user interface interaction from other server-side processes to improve the overall application performance and provide a rich user experience.

Silverlight also supports JSON-based service integration. LINQ to JSON support enables us to query, filter, and map JSON results to .NET objects within a Silverlight application, which makes integration with existing AJAX-based applications and services easy.

Silverlight supports policy-based application development and deployment with cross-domain networking using HTTP and sockets.

■**Note** JSON (JavaScript Object Notation) is a lightweight data-interchange language-independent text format language. It is based on a subset of the JavaScript programming language, Standard ECMA-262 Third Edition, December 1999. For more details, visit `http://www.json.org/`.

Quick Overview of Silverlight 2's Key Components

Before we develop our first basic application in the next chapter, I want to give you a quick overview of key components of Silverlight 2.

Defining User Interfaces in Silverlight

As Silverlight is a subset of WPF, the key technology for defining Silverlight application user interfaces is WPF and XAML. XAML UI controls are the building blocks to develop RIAs.

Figure 1-11 shows all the default basic Silverlight XAML UI controls available in Visual Studio as part of Silverlight 2. The same controls are also available in Expression Blend.

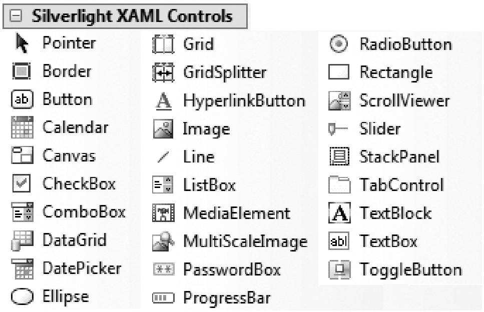


Figure 1-11. *Microsoft Silverlight 2 XAML UI controls*

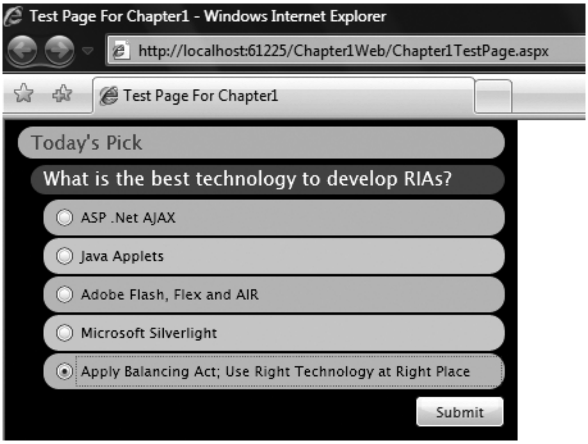
As described earlier, we can divide the default XAML UI controls into four categories:

- *Layout management XAML controls* are the key basic controls of any Silverlight application and act as the containers for hosting and positioning different UI controls including other layout controls (as subcontainers). Three types of layout controls are available in the XAML controls library:
    - `Canvas` is a very basic and straightforward layout control that hosts user controls with absolute position relative to the canvas area.

  - `StackPanel` layout controls are used to stack/group user controls horizontally or vertically in relative position.
  - `Grid` layout controls are the most commonly used layout control and can group user controls in defined row and column positions (the same as tables do in HTML).
- *Form XAML controls* help users to perform different actions like entering text or selecting options. `TextBox`, `RadioButton`, and `CheckBox` controls are examples of form XAML controls.
- *Data manipulation XAML controls* manage data manipulation, and you can use them to display data in whatever format you choose. `DataGrid` and `ListBox` controls are examples of data manipulation XAML controls.
- *Functional XAML controls* help end users display information in a usable manner to perform different actions such as date selection to control application behavior (using the Slider UI control). `Calendar`, `DatePicker`, `ScrollViewer`, and `Slider` XAML controls are examples of functional XAML controls.

You can extend the scope of the Silverlight XAML controls library by using the Silverlight toolkit (downloadable from `http://www.codeplex.com/Silverlight`) and third-party controls (explained later in this chapter).

I'm assuming that you know how to create and test a Silverlight project. However, let's create a very simple Silverlight application as an example just to illustrate the different types of XAML user controls to get you up and running. Figure 1-12 presents the simple Silverlight survey application user interface we are going to develop. Hopefully, the question shown in the figure will also refresh your memory about the discussion we had on the use of which technology to develop RIAs with.



Test Page For Chapter1 - Windows Internet Explorer
http://localhost:61225/Chapter1Web/Chapter1TestPage.aspx
Test Page For Chapter1
Submit

Today's Pick

What is the best technology to develop RIAs?

○ ASP .Net AJAX

○ Java Applets

○ Adobe Flash, Flex and AIR

○ Microsoft Silverlight

● Apply Balancing Act; Use Right Technology at Right Place

Figure 1-12. *XAML voting screen of the Silverlight web application*

For any Silverlight application, we need to select the main layout XAML controls. For this application, we have options to use the `StackPanel` or `Grid` layout control. In this case, let's use the `Grid` control. The following code snippet defines these aspects of the application:

- The background of the main `Grid` container is black.
- We need seven rows in the `Grid` each of height 30, which we define with the `RowDefinition` control.
- Each row will have one column with a width of 400 defined by the `ColumnDefinition` control. Since our sample application needs only one column, the width of which is also matched with the main user control's width, the column definition is optional for us.

```
<Grid x:Name="LayoutRoot" Background="Black">
   <Grid.RowDefinitions>
     <RowDefinition Height="30" />
     <RowDefinition Height="30" />
     <RowDefinition Height="30" />
     <RowDefinition Height="30" />
     <RowDefinition Height="30" />
     <RowDefinition Height="30" />
     <RowDefinition Height="30" />
     <RowDefinition Height="30" />
   </Grid.RowDefinitions>

   <Grid.ColumnDefinitions>
     <ColumnDefinition Width="400"/>
   </Grid.ColumnDefinitions>

</Grid>
```

Now we have to define in each row the other XAML controls that we need for our sample application.

The first two rows contain simple text with different background colors. For that we use the TextBlock XAML control to display text as demonstrated in the following code snippet:

```
<Grid Grid.Row="0" Margin="10,5,10,0">
  <Border Grid.Column="0" CornerRadius="12" Background="Aqua">
    <TextBlock Text="Today's Pick" Foreground="Blue" Margin="10,2,0,0"/>
  </Border>
</Grid>

<Grid Grid.Row="1" Margin="20,5,10,2">
  <Border Grid.Column="0" CornerRadius="12" Background="Blue">
    <TextBlock Text="What is the best technology to develop RIAs?"
               Foreground="White" Margin="10,0,0,0"/>
  </Border>
</Grid>
```

As shown in the code snippet:

- The Grid.Row and Grid.Column attributes define in which row and column the control should be placed.
- The Margin property of the control is used to define the position of that control.
- The Border control is used to provide the border and/or background of the other control. Here we have used Border to define the border of the TextBlock control.
- The TextBlock control's Text property is used to define the static text in the first two rows.
- The CornerRadius property of the control is used to define the rounded angle of the control.

We'll use the RadioButton control to define all the options. These options are grouped within the Group1 named group by using the GroupName property of the RadioButton. Each option's description is defined using the Content property of the RadioButton control.

The following is a sample snapshot of the first option in our application. You can define the other options in a similar way.

```
<Grid Grid.Row="2" Margin="30,2,10,0">
   <Border Grid.Column="0" CornerRadius="12" Background="LightGreen">
      <RadioButton Content="ASP .Net AJAX" GroupName="Group1" Foreground="Black"
  Margin="10,0,0,0"/>
```

```
   </Border>
</Grid>
```

Finally, we need a button to submit the user's selection. For that we'll use the Button XAML control and name it with the x:Name property. The value defined in the x:Name property will be used to identify the control in the code-behind file.

```
<Button x:Name="btnSubmit" Content="Submit" Grid.Row="7" Width="70"
        Height="25" Margin="310,5,0,0"/>
```

Third-Party Silverlight Controls

In addition to the broad range of XAML controls provided by Microsoft, a number of well-known third-party Microsoft Partners have developed extended Silverlight XAML controls. This section provides information about Microsoft Partners providing extended XAML controls for Silverlight.

ComponentOne

ComponentOne has released Studio for Silverlight, providing enhanced controls covering layout, navigation, data input, grids, charts, data binding, data compression, and enhanced functionality (e.g., MessageBox, ImageRotator, Book, Map, DragDropManager). You can get more details on these controls by visiting http://labs.componentone.com/.

Infragistics

Infragistics' NetAdvantage for Silverlight Data Visualization provides about 28 chart types (including area, bar, column, line, spline, scatter, point, pie, doughnut, and stack charts) and gauges to enable data-driven RIAs. You can get more details on these controls by visiting http://www.infragistics.com/ and clicking "Experience Silverlight."

Telerik

Telerik provides about 19 extended Silverlight controls to feature enhanced layout, form input, and data binding to help with rich user experiences and data virtualization. You can get more details on these controls by visiting http://www.telerik.com/products/silverlight/overview.aspx.

Developer Express

Developer Express (Devexpress) provides the enhanced Silverlight Grid control, which provides rich data integration. You can get more details on this control by visiting http://www.devexpress.com/Products/NET/Controls/Silverlight/Grid/.

Event Integration: Managed Code and Scripting Programming Models

XAML is a declarative language defining objects and related attributes/properties for the application user interface. In addition, it is capable of integrating event handling as JavaScript or code-behind managed code. x:Class is the key attribute for defining the attached programming model of a XAML-based page. If the x:Class attribute is defined at the root level of the XAML page, the managed code-behind file must exist and contain the event handlers related to the XAML user controls defined in the XAML page. If you don't define the x:Class attribute at the root level, there is no need for a code-behind file, and the application will use JavaScript. In this case, the event functions are not compiled at design time like managed code, but the function will be called at runtime when the end user initiates the event. You'll see an example of each of these approaches in the upcoming text.

The x:Class attribute at the root element of the XAML page defines the CLR namespace and class name to implement the code-behind event integration. In the preceding example, we use a managed code-behind file for event integration. To achieve that, the x:Class attribute of the root-level element UserControl is defined with the Chapter1.Page value, defining the CLR namespace as Chapter1 (see the following code snippet).

```
<UserControl x:Class="Chapter1.Page"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    Width="400" Height="250">
```

As an example, the Submit button would need to have a button click event to submit the user's entered information. To achieve that, you can add the Button "btnSubmit" Click event in the XAML code as shown in the following code snippet:

```
<Button x:Name="btnSubmit" Content="Submit" Grid.Row="7" Width="70"
        Height="25" Margin="310,5,0,0" Click="btnSubmit_Click"/>
```

Following is the code snippet of the associated Submit button click event, btnSubmit_Click, of the associated Page code-behind class:

```
    private void btnSubmit_Click(object sender, RoutedEventArgs e)
    {
         //Your Custom Code Goes Here.
          //=============================
    }
```

You can achieve the same type of functionalities using the JavaScript. Let's see how the JavaScript event handling works. The following code snippet shows that the Canvas layout control is at the root level and there is no x:Class attribute. The function name getUserProfile() is defined in the JavaScript file HandleXAMLEvents.js linked to in the <script> tag. The getUserProfile() function will be called on the Loaded event of the TextBlock XAML user control.

```
<html>
   <head>
        ----

        <!-- JavaScript File Managing XAML Controls Events -->
          <script type="text/javascript" src="HandleXAMLEvents.js"></script>

        ----
   </head>

   <body>
     ----

     <!-- Define XAML content. -->
     <script type="text/xaml" id="xamlContent"> <?xml version="1.0"?>

       <Canvas xmlns=http://schemas.microsoft.com/client/2007>

         <TextBlock Canvas.Left="10" FontSize="12" Loaded="getUserProfile"/>

       </Canvas>

     </script>
```

```
    ----
  </body>

</html>
```

Data Binding

The data binding feature of Silverlight enables data display and user interaction with that data for Silverlight applications. Figure 1-13 shows the conceptual flow of data binding between the data object and a Silverlight control's data-bound property. Based on the value of the binding `Mode` property, the flow of the data between the data object and control is determined as follows:

- `OneTime` binding updates the control (the binding target) with the data object (the binding source) only when the binding is created.
- `OneWay` binding updates the control (the binding target) with the data object (the binding source) when the binding is created and in the future whenever the data is changed at the binding source level.
- `TwoWay` binding updates the control (the binding target) and data object (the binding source) whenever the user changes the data at the binding target level or data changes at the data source level.

For `OneWay` and `TwoWay` bindings, the binding source object must implement the `INotifyPropertyChanged` interface in the `System.ComponentModel` namespace for it to notify the binding target of any changes. The default value of the binding mode depends on the control type. For most form XAML user controls that are usually user-editable, the default binding mode is set to `TwoWay`. The XAML user controls that are not user editable have the default binding mode set to `OneWay`.

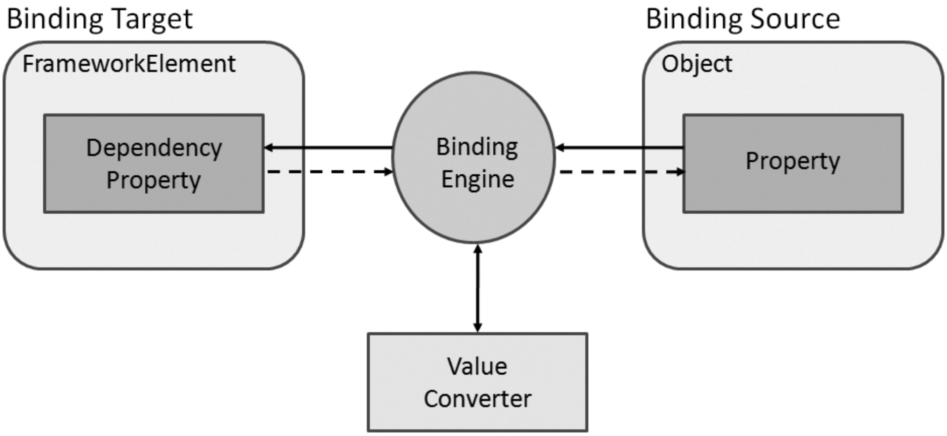


Figure 1-13. *Silverlight data binding*[15]

Note Figure 1-13 is modified from the original diagram to demonstrate `TwoWay` data binding.

[15] Data Binding for Silverlight, `http://msdn.microsoft.com/en-us/library/cc278072(VS.95).aspx`

The following code snippet demonstrates OneWay data binding with the TextBlock control and its Subject type field. The Text property is bound to the Subject value.

```
<TextBlock Name="Subject" Grid.Row="0" Grid.Column="0"
           Text="{Binding Path=Subject, Mode=OneWay}"/>
```

The following code snippet demonstrates OneWay data binding with the ListBox control. The ItemsSource property is bound to the Courses collection.

```
<ListBox x:Name="Courses" ItemsSource="{Binding Path=Courses, Mode=OneWay}"
         Height="60" Width="200" Grid.Row="1" Grid.Column="0" />
```

As explained earlier, for OneWay and TwoWay bindings, the data source object must implement the INotifyPropertyChanged interface. The following code snippet presents the implementation of the INotifyPropertyChanged interface for the properties Subject and Courses used for the data source binding in our preceding example:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;

    public class Training : INotifyPropertyChanged
    {
        private string trainingSubject;
        private List<string> trainingCourses;
        // Declaring PropertyChanged event
        public event PropertyChangedEventHandler PropertyChanged;

        //Properties used for Data Source Binding

        public string Subject
        {
            get { return trainingSubject; }

            set
            {
                trainingSubject = value;
                NotifyPropertyChanged("Subject");
            }
        }

        public List<string> Courses
        {
            get { return trainingCourses; }

            set
            {
                trainingCourses = value;
                NotifyPropertyChanged("Courses");
            }
        }

        //Raise the PropertyChanged event passing
        //the source property that is being updated.

        public void NotifyPropertyChanged(string propertyName)
        {
            if (PropertyChanged != null)
            {
                PropertyChanged(this, new PropertyChangedEventArgs(propertyName));
            }
        }
    }
```

Microsoft Tools to Developing Silverlight Applications

The main goal of the Microsoft Silverlight technology platform is to develop RIAs in the RAD mode, making the lives of developers, designers, and artists easier. This section will give a high-level overview of the different development and designer tools provided by Microsoft to develop Silverlight-based RIAs.

Microsoft Visual Studio 2008 SP1

Microsoft Silverlight Tools for Visual Studio 2008 SP1 enables developers to develop Silverlight 2–based applications using the very familiar Visual Studio 2008 development platform. Microsoft Visual Studio 2008 allows you to develop Silverlight applications using the following templates:

- *The Silverlight Application project template* is used to develop Silverlight 2–based applications and includes all the required configuration, assembly references, and code-behind files to support development, testing, and deployment of Silverlight applications.
- *The Silverlight Class Library project template* is used to develop reusable user interface and code-behind components as library assemblies that can be included and deployed in multiple Silverlight applications as in-package (included in the Silverlight application deployment package) and on-demand (referenced assemblies downloadable at runtime upon request) files.

Microsoft Expression Studio 2 SP1

Microsoft Expression Studio is a set of graphical tools for graphical designers and artists to manage and develop digital media and content and present it in Silverlight-based RIAs. It has the following components:

- *Microsoft Expression Blend* helps user interface designers and artists to develop vector-based 2D and 3D XAML-based web and desktop user interfaces (for inclusion in Silverlight applications). It is also integrated with Visual Studio, enabling designers and developers to rapidly develop, debug, and test Silverlight applications in integrated mode.
- *Microsoft Expression Design* is a vector and raster graphic design tool that can be well integrated with Expression Blend.
- *Microsoft Expression Media* is a media management tool that allows users to build and track a digital media catalog. It also enables professionals to edit, tag, archive, back up, and deploy digital media, as well as transform it from one format to another. The deployed digital media can be used within Silverlight applications.
- *Microsoft Expression Encoder* contains several Silverlight Media Player templates to write, manage, and publish media in VC-1 quality for Silverlight applications.

Note VC-1 is a SMPTE (Society of Motion Picture and Television Engineers) standard codec that is a part of both HD-DVD and Blu-ray high-definition optical disk specifications.

- *Microsoft Expression Deep Zoom Composer* allows professionals to create and export high-resolution Deep Zoom composition tiled image files that can be deployed to enable in-place zooming and panning features for images in Silverlight applications.

Microsoft Silverlight Streaming by Windows Live

The Microsoft Silverlight streaming service is a free service for media streaming and hosting solutions (up to 10GB) for Silverlight-based applications provided by Windows Live. You can visit `http://silverlight.live.com/` for more details.

Summary

This introductory chapter started with details about the next-generation Web 2.0 technology platform and RIAs. RIAs are

- Digital multimedia and content rich, with seamless multimedia and content integration providing desktop application–like rich, interactive user interfaces
- Platform independent (cross-operating system, cross-browser, and cross-device)
- Modular, loosely coupled, flexible, and customizable
- Metadata rich
- Collaborative
- High performing (with client-side processing and client-side local caching and asynchronous communication capabilities)

It is crystal clear from the preceding discussion that, just as the Internet (Web 1.0) and globalization impact broke the boundaries between countries, Web 2.0–based RIAs break the barrier between web-based applications and desktop applications and empower different communities to collaborate on digital information (content and multimedia) in the most user-friendly and secured way.

Microsoft Silverlight is one of the most promising technology platforms in the rising era of Web 2.0 and RIAs. After releasing a very basic scripting-oriented Silverlight 1, Microsoft made a giant leap by delivering Silverlight 2 with support for Microsoft .NET Framework 3.5 with the CLR, and thus support for .NET managed code-behind using the default Microsoft .NET class libraries. Silverlight provides a platform to develop cross-browser, cross-platform, and cross-device RIAs and is a subset of the WPF. As such it uses XAML, which facilitates an externalized and loosely coupled definition of the Silverlight application user interface and related style sheets.

To support an agile rapid application development model, Microsoft provides seamless and fluent integration between Visual Studio 2008 SP1 as the developer's tool and Expression Studio 2 (including Deep Zoom Composer) as the designer's tool to support development of Silverlight RIAs.

In the next chapter, we will create a Silverlight-based RIA, My Album, as a base application for this book. In Chapter 5, we will revise the My Album application to make it suitable to deploy at the enterprise level, and then we will transform it into an Enterprise Training Portal RIA.

Additional References

Links from the Microsoft Web Site

- Silverlight home page, `http://silverlight.net/default.aspx`
- ASP.NET AJAX Overview, `http://msdn.microsoft.com/en-us/library/bb398874.aspx`
- Events and Delegates, `http://msdn.microsoft.com/en-us/library/cc189018(VS.95).aspx`
- Data Binding, `http://msdn.microsoft.com/en-us/library/cc278072(VS.95).aspx`
- Developing a Silverlight Library Assembly, `http://msdn.microsoft.com/en-us/library/cc296243(VS.95).aspx`

Apress Reference Books

- *Silverlight 2 Visual Essentials* by Matthew MacDonald (2008)
- *Beginning Silverlight 2: From Novice to Professional* by Robert Lair (2009)
- *Foundation Expression Blend 2: Building Applications in WPF and Silverlight* by Victor Gaudioso (2008)
- *Pro Silverlight 2 in VB* 2008 and *Pro Silverlight 2 in C#* 2008 by Matthew MacDonald (2008)