

1

FitnessTrackerPlus

An Overview of the FitnessTrackerPlus Application

In the fall of 2008, Microsoft released an update to its new media-centric browser plug-in Silverlight. This update probably wasn't very exciting to end users who may have had Silverlight installed to watch the 2008 summer Olympics, but to developers who make use of Microsoft web technologies, a monumental change had occurred. Silverlight 2.0 finally included the long-awaited support for C#/VB.NET programming directly in the Silverlight world. No longer were developers forced to utilize the plug-in with a JavaScript-based API only. With the addition of C#/VB.NET support, it seemed like Silverlight may have finally made the necessary jump required in order to support line-of-business-style applications. Although a dramatic improvement from its predecessor, Silverlight 2 still left a lot to be desired in terms of line-of-business controls. Luckily, Microsoft has heard developers loud and clear and made several important enhancements in version 3 that specifically addressed line-of-business needs. Although the Silverlight 3 release meant developers had a powerful toolkit at their disposal for creating rich internet applications, some features were still omitted that were absolutely crucial in transitioning line-of-business applications from the desktop to the web. However, with the latest release of Silverlight version 4, these features have finally made it into the runtime, including a new Printing API, RichTextArea, Clipboard API, mouse wheel support, Implicit theming, and right-click event handling to name a few. Features like printing, right-click menus, and clipboard support offer some of the final pieces of the line-of-business missing from the Silverlight puzzle and may, in fact, make this latest release of Silverlight the most exciting yet for developers.

It is the primary intent of this book to introduce you to some of these great new features as well as show you some advanced techniques that you can apply to the development of your own Silverlight-based line-of-business solutions. In this book I will be following the Wrox Problem-Design-Solution style so that each chapter contains a detailed problem statement followed by a design analysis and solution implementation. I will not be discussing detailed API information for Silverlight and the various controls; however, all of this information is available by downloading the official Silverlight 4 documentation from <http://www.silverlight.net>. This book will be taking you through the creation of a fully functional Silverlight 4 line-of-business

application while touching on several important topics such as scalable design, social networking, SEO, revenue generation, rich user interfaces, and others.

In this chapter, you learn the following:

- How to take a great idea and build a design to make it into an even better web application.
- What pages you will need for the application.
- What the new Silverlight 4 features are that you'll use to build your application.
- How you'll actually build the application through membership, personalization, journaling, and social networking. You also learn how to monetize the site, as well as test and deploy it.
- What software you'll need to download and install and what the basic setup instructions are for the site.

It is an exciting time for Silverlight development, and with the plug-in installed on millions of computers and newly announced support in MySpace, now is a great time to learn how to incorporate the technology into your own solutions.

PROBLEM

After a few years of full-time software development, I realized that between sitting in a chair for hours on end, and the not-so-great foods that are consumed during those crunch-time marathon coding sessions, I would probably need to consider some healthy habits to offset what can otherwise be a very sedentary work environment and lifestyle. The first step for me was to join a local gym, and watch what I eat when not on a marathon coding session. After a few weeks of working out I started to wonder, “Am I getting the results I was looking for? Are these foods really good for me? In fact, how can I even measure if I am making any progress while spending all of these hours in the gym?” Perhaps most importantly, the programmer in me decided to ask, “How can I write some software to help me keep track of what I do here?”

Most people would have just gone home, hopped on the Internet, and found some software or a web site that does this very thing. As programmers, however, we can just fire up the IDE and write our own instead. I decided that I wanted to create a new Silverlight-based web application that enables people to keep track of diet, exercise, and measurements. It would also be nice if the site could foster a community of users who share success with others through public facing journals hosted on the site, and integration with social networking sites such as Facebook and MySpace. Because this seemed like it could become a pretty popular site that could generate enough revenue to sustain itself, I also wanted to integrate some revenue streams through some form of pay-per-click advertising and monthly premium memberships for access to enhanced features of the site. At this point, the only thing missing was a name for the site and, of course, the implementation. For the name, I decided on FitnessTrackerPlus. I figured the “Plus” suffix will let users know that they are signing up for a more feature-rich site than traditional ASP.NET sites they may have used in the past. Or at least I will attempt to convince myself of that since I couldn't think of a nice catchy Web 2.0 name like Google or Twitter that works well for a fitness-tracking web site.

DESIGN

In this book, you will be creating a Silverlight 4 web application and will be utilizing the new features included in Silverlight 4 to provide the user with a rich user interface that is more responsive than a traditional ASP.NET web site. While the focus will be on creating an online fitness-tracking system, many of the features and techniques discussed can be applied to other line-of-business applications as well. Like most line-of-business applications, the primary purpose of the site is to allow users to enter data and view reports on that data. You will also be adding a social networking aspect to the site as well. In the end, you should be able to leverage the data entry, charting, site structure, SEO, and social networking components when building your own line-of-business applications in Silverlight.

Determining Requirements for the Site

The first step in the design phase of FitnessTrackerPlus is to get a detailed list of requirements for the site. I briefly touched upon some of the most basic high-level features that will be required by the site, but now it's time to flesh out everything. As with any application, it's best to collect as much detail about requirements as possible before writing a single line of code. Changes will always be easier to make at this stage of the game, as opposed to once you have already started coding a solution. The following sections list the major features that FitnessTrackerPlus will include.

Scalable

The more popular a site becomes, the more important it is that the new traffic can be handled without causing problems for existing users. In order to facilitate this, a multi-tiered architecture should be used. This allows the site to grow as the site's popularity grows, as well as keeps a clean separation between user interface, business logic, and data access code.

Rich User Interface

Users will log on to the site to quickly enter daily exercise, foods, and measurements. A rich user interface that utilizes Silverlight helps to ensure that the amount of time performing data entry tasks is minimized. The less users have to spend entering data, the more time they will have to view reports, measure progress, and to interact with others. The key to making this process simple and painless is to ensure that the pages are not cluttered with large data entry forms. The data entry for foods, exercises, and measurements should be simple enough so that users can quickly figure out what they need to do in order to enter information. It is also common in web sites now to provide multiple themes for users so that they can select a favorite color scheme or site layout. Your site should provide a similar theme selection feature that also remembers theme preference so it can be restored on the next login.

Dashboard

Once users successfully log into the site, they should be presented with a dashboard that shows basic fitness information for the current day. The dashboard should provide a simple non-cluttered look and give the user quick access to current measurements, nutrition, and exercise summaries, and any site announcements that you want users to be aware of. Site announcements can be a good tool to

let users know about upcoming site maintenance or new features, as well as any features that are currently being worked on for future releases. Such announcements enable you to avoid bulk e-mails that could possibly end up being interpreted as spam.

Nutrition Log

The goal of the nutrition log is to provide a simple way for users to enter the foods they eat and view basic summaries of the nutrients that they consumed during the course of the day. Because that is the primary goal, any control or feature that does not directly make this process easier should not be added to the screen. All too often, you'll find what you think will be a useful web site that provides a service you're looking for only to sign up and find out that the developers have crammed in as many features as possible with little regard to the effect on the overall user interface. It should not require a training manual for people to use any part of your site, especially the data entry. It will be difficult to show your users how they are progressing if they don't come back to the site daily to enter foods. The only way they will consistently do so is if the process is quick and easy.

In order to help users track the foods they eat, you should provide a database of foods that the user can search. With millions of foods on the market, it is not practical or realistic to include every food in your database, so you'll also need to provide a mechanism for users to create and manage custom foods, complete with custom nutrition facts for the given food. The user doesn't want to browse an entire food database on the site in order to find a specific food, so you need to provide an assisted search method. Most sites now have some sort of auto-suggest functionality in search boxes that helps guide the user to the correct entry in the database. Your site should provide similar functionality by taking the first few characters typed and suggesting foods that match the search criteria in the food database. In most cases, you want your users to type in a few letters of a food, click the matching item, and move on to the next food they ate.

Exercise Log

The exercise log will be very similar in functionality to the nutrition log. Again, the primary goal is to provide the user with a quick and easy way to enter daily exercises and workout information. The database should contain a list of well-known exercises that users can search. You should add auto-suggest functionality to the search box as you did with the food search. In addition, to make exercise entry easier, you should break down exercises into various muscle groups and provide users with the capability to browse exercises based on those muscle groups. Just as you probably can't populate your database with every known food, you most likely will miss some exercises that people perform as well, so users should be able to create and manage their own custom exercises.

Measurements Log

One of the most important aspects of the site will be that users can keep track of their current measurements. You want to allow for data entry of a few standard measurements such as weight, BMI, waist, and legs. As you did with the exercise and nutrition logs, you will provide a way for users to create custom measurements that they would like to keep track of. Although keeping track of numerical measurements can be an effective way to measure fitness progress, it would also be beneficial to give users a visual representation of that progress. In order to satisfy this requirement, the site should have a mechanism to upload images of the user when they are entering measurements.

Public Journal Page

Once your users have taken the time to update their journal, they may want to share their updates with other users as well as friends and family. Getting positive feedback when working toward any goal can provide powerful motivation. Your site should allow users the option to share information with others. You can do this by providing a public-facing version of the journal similar to other social networking sites such as MySpace and Facebook. What you want to avoid is forcing the user to share a long, convoluted URL that friends and family have no chance of remembering; you need the site to give the user a friendly URL in the form of `FitnessTrackerPlus/Journals/username`. When users enter that URL in the browser, it should take them directly to the public-facing journal page that matches the username specified. Because some users may not want to share all the information in their journal, you should give them the option of sharing everything or any combination of foods, exercises, measurements, and journal entries. Users should be able to disable sharing of the data or make their journals private again at any time.

Social Networking Integration

Although you are trying to create primarily a data entry application, it would not be wise to ignore the explosion of social networking sites that have been created in the last few years. Most content-based sites have found ways to integrate with popular sites such as MySpace, Facebook, Twitter, and others. Providing this integration further promotes the sharing of information and the ability to gather feedback about the information being shared. Because it's been established already that you are trying to build a new online community right here at FitnessTrackerPlus, you also want to take advantage of what some of the other major social networking sites currently have to offer for integration. In addition to offering your users with the public journal page, you should also provide them with the option to share their fitness information with others on an existing social networking site like MySpace. Now that MySpace offers official support for Silverlight applications, you should be able to create a MySpace application that allows users to share data right from their own MySpace profile page.

Browser History Support

One of the complaints over the last couple of years about Rich Internet Applications, especially AJAX-based solutions, has been the lack of support for basic browser navigation functions. All too often, users would be engaged in a web application that utilized an AJAX library and they would click the Back button on the browser only to be brought all the way back to the home page or login screen. Your application should fully support browser history and navigation features so that at no time will an accidental click of the mouse redirect the user back to a login screen or some other page on the site that is completely unrelated to what they were working on at the time.

Generate Revenue

One issue to consider when starting a site like this is that if it gains in popularity and you want to use a hosting provider that offers technologies such as SQL Server and ASP.NET, you will have to cover the costs associated with that hosting. When starting out, you can most likely get away with one of the many shared hosting providers available online. As the site grows, you may need to look at a dedicated hosting solution. In either case, it would be best if you didn't have to pay for this out of

your own pocket. You will need to come up with a way to create revenue to pay for the costs of running and maintaining this site. You can do this by utilizing a pay-per-click advertising service. In addition to the advertisements, you can look at charging a small monthly fee in order for users to have access to the site.

Feel Like a Desktop Application

The most important reason for choosing Silverlight for this site's user interface layer is that it provides users with a rich experience. The best case scenario is for FitnessTrackerPlus to look and feel like a traditional desktop application, complete with features that users have come to expect from those applications, such as mouse wheel support, right click menu options, fluid user interface transitions, and so on. As you develop FitnessTrackerPlus, you need to ensure that you use these common user interface features, now available in Silverlight, throughout the application to give the users the impression that this is not your typical web site.

Silverlight 4 Features

As part of the design, you should also think about what technology you will be using to implement the solution. For this site, the decision has already been made to utilize the new Silverlight 4 runtime to provide the rich user interface required for the application. The latest version of Silverlight provides some long-awaited features that will really help in the creation of a rich data entry site. For example, the public journal feature would have been much harder to implement with the previous version of Silverlight, but the new Navigation Framework makes this type of thing relatively easy to implement.

Let's take a look at some of the improvements in this latest release of Silverlight that will make programming this site much easier than it would have been with an older version.

Navigation Framework

A major benefit of sticking with the existing ASP.NET technology for creating a web site was that pages could be bookmarked and shared, and users could easily navigate from one page to the next using standard features of the web browser. Applications that were written in older versions of Silverlight could not easily mimic this functionality. Although you could create a rich user interface, it was terribly difficult to do some basic things such as support browser history, and Forward and Back buttons. It was also very difficult to have direct links to pages written in XAML. The new navigation framework takes care of all of those issues by allowing developers to create pages in XAML that have full support for browser navigation features as well as deep linking so it is now easy to share XAML-based page URLs. Also included with the new navigation framework is the capability to map long, complex URLs to simple, easy-to-remember ones. This will become important as you work on the public journal page and allow others on the Web to access those pages using an easy-to-remember URL such as `FitnessTrackerPlus.com/username`. The URL mapping located in the `App.xaml` file is handled by the new `UriMapper` class shown in the following code:

```
<navigation:UriMapper x:Key="uriMapper">
  <navigation:UriMapping Uri="/{user}"
    MappedUri="/Views/PublicJournal.xaml?user={username}" />
</navigation:UriMapper>
```

In this case, the URI will be the default `FitnessTrackerPlus.com/username` where the `{user}` will be replaced with an actual username. The `MappedUri` will be the actual path to the XAML page that contains the public journal. As you can see, the Navigation Framework allows for individual XAML pages to behave like a typical ASP.NET page complete with their own query string. The Navigation Framework will also allow you to simulate some of the ASP.NET master page behavior. Using navigation frames, you will be able to create user controls for each feature you are implementing while sharing a common navigation menu and top banner across all of the pages. The Navigation Framework is a significant enhancement to Silverlight and really takes you a large step closer to being able to create web applications that combine a rich user interface with the standard web functionality available in traditional ASP.NET-based web sites.

New Data Controls

With the new release, some important new data controls have been made available including the `DataForm` and `DataPager`. The `DataGrid` control previously had no support for paging data and it became a pretty large limitation of the control, especially if binding to a potentially large result set. The `DataPager` provides paging capabilities to any data set that implements the `IPagedViewCollection`. Because the `ObjectDataSource` supports this interface, you can set the `DataPager` to work in conjunction with the `DataGrid` and provide an efficient paging mechanism for large result sets. The new `DataForm` provides a very powerful way to display detailed information about a data item in a standard data entry form. The `DataForm` has similar functionality to the `DetailsView` in ASP.NET, and in this application, you will use it to provide a data entry screen not only for custom foods, exercises, and measurements, but also for details about individual nutrition and exercise journal entries that are being displayed in the `DataGrid`.

In addition to the `DataForm` and `DataPager` controls, there have been important enhancements to the `DataGrid` control itself, including optimizations to speed up the overall load time of the control, and a variety of column sizing options to prevent horizontal scroll bars from appearing when users resize various column headers.

Control Toolkit

The Silverlight Control Toolkit has been available since version 2 of the runtime was released but the latest release includes some new controls and promotes some other controls into the Stable band. Controls in the Stable band are considered pretty much ready for prime time and have been thoroughly tested. These controls are updated and modified only during bug fix cycles and typically are safe from breaking changes. Preview band controls are subject to modifications that include breaking changes, so there is some minimal risk to using them in a production application. Experimental band controls are really intended for evaluation only. These experimental controls should not be used in production applications, as in most cases they are not feature complete and it is a pretty safe bet that breaking changes will be made between releases. Table 1-1 lists the controls available in the toolkit as well as the current stability status:

TABLE 1-1: Controls Available in the Control Toolkit

TOOLKIT	AVAILABLE CONTROLS
Stable Quality Band	DockPanel Expander HeaderedContentControl Label NumericUpDown Viewbox WrapPanel
Preview Quality Band	Accordion Charting DomainUpDown ImplicitStyleManager LayoutTransformer Rating TimePicker TimeUpDown Eleven themes
Experimental Quality Band	TransitioningContentControl GlobalCalendar TreeMap Drag & Drop Busy Indicator
Available Themes (All are considered part of the Preview Quality Band)	Bubble Crème Bureau Black Expression Dark Expression Light Rainier Purple Rainier Orange Shiny Blue Shiny Red Twilight Blue Whistler Blue

Developers can get access to the source code over at the CodePlex site using the following URL: <http://silverlight.codeplex.com>. The project has been made open source so you can feel free to make changes to the code and integrate those changes into your own projects. If you are not interested in the source code, you can also download just the raw binaries and add them to your Silverlight project. This toolkit provides some great controls and features that are not available from the Silverlight 4 runtime. You will be using them extensively throughout the FitnessTrackerPlus application. The application will make use of the `Label`, `DockPanel`, `Charting`, `TimePicker`,

TimeUpDown, GlobalCalendar, BusyIndicator, and all of the available themes. In the following chapters, I go into detail about each one of these controls and demonstrate how they really provide a large amount of user interface functionality with a very minimal level of development effort.

WCF RIA Services

These new services are an important addition to Silverlight and help provide n-Tier data support based on the new ADO.NET data services. The services also provide a means to perform data entry validation on the client along with paging, sorting, and querying data. WCF RIA Services will also help with the integration of Silverlight applications with the ASP.NET authentication and role management services. Perhaps one of the biggest benefits to this new feature is providing a way to handle change tracking between tiers, which was not straightforward in earlier Silverlight versions when using ORM technologies such as LINQ to SQL or the Entity Framework.

Dynamic, Implicit, and BasedOn Styling

The new dynamic and implicit styling support in Silverlight 4 provides a way to change the currently applied theme at runtime. When attempting to achieve this functionality in previous versions of Silverlight, developers were forced to worry about applying the theme elements to the entire visual tree of controls manually. Dynamic styling also provides the capability to change the theme more than once, a difficult and error prone process before, which sometimes required rebooting the Silverlight application to apply the new theme changes. BasedOn styling from which other style definitions can inherit allows you to create base styles and then styles that inherit the settings from that base style. Now you can create a style with some basic settings and when you need to change only one aspect of that style to use in another control, you don't have to copy the entire original style to include that change.

Finally, implicit styling also gives you the ability to declare a style that applies to all controls of the specified type. This gives you a much easier way to share styles across controls and is very similar to CSS based styling. For example, implicit styling allows you to style all `TextBox` controls with a thick border using code similar to the following:

```
<Style TargetType="TextBox">
    <Setter Property="BorderThickness" Value="5" />
</Style>
```

Additional Features

In addition to the major feature enhancements I have mentioned in the previous sections, the following enhancements have also been added to the Silverlight 4 runtime:

- Webcam/microphone access
- ICommand support
- HTML Hosting in Offline mode
- Elevated trust applications
- Local file access

- COM Interop
- Notification API in Offline mode
- Network authentication
- Cross-domain networking support
- Keyboard access in full screen mode
- Text trimming
- Right-to-left, BiDi and complex script
- Offline DRM
- H.264 protected content
- Support for using the Silverlight plug-in as a drag & drop target
- Managed Extensibility Framework
- Support for Google Chrome web browser

SOLUTION

In the “Solution” section of each chapter, I will take you through the complete implementation details required to build the features highlighted during that chapter. This section of the chapter will typically contain the majority of the code snippets along with brief discussions and explanations about what the code is doing and why it is doing it. For this first chapter I won’t be getting into any real code, but instead I’ll give you a quick overview of what you can expect to see in Chapters 2 through 12.

Chapter 2

The second chapter is all about architecture and a discussion of the various technology choices that are available to Silverlight developers. The goal, of course, is to make sure that choices are made that will allow you to keep a multi-tiered design and implementation with an emphasis on both scalability and performance. This chapter will be broken down into discussions about both the Physical N-Tier design as well as the Logical N-Tier design. I will briefly cover the following technologies before ultimately deciding which ones will be used for FitnessTrackerPlus, along with the reasons why:

- Silverlight Toolkit
- Silverlight Extensions
- ASMX Web Services
- WCF Web Services
- WCF RIA Services
- ADO.NET

- Entity Framework
- LINQ to SQL
- ADO.NET Data Services
- SQL Server
- Oracle
- MySQL

Chapter 3

In Chapter 3 I will focus on how to integrate the ASP.NET Membership, Profile, and Role services into the FitnessTrackerPlus Silverlight application. You will see how the new WCF RIA Services Framework makes integration with these services simple. After completing this integration process, you will then add user registration and login capabilities to the application. Finally, I will take a look at how you can prepare an alternate view of the FitnessTrackerPlus site for those users who may have not yet downloaded and installed the Silverlight plug-in. This way, you will ensure that users without the plug-in can still access the landing page and find out more information about the site before making a decision about installing Silverlight on their own machine.

Chapter 4

In Chapter 4 I will cover the initial design and implementation of the dashboard page. The dashboard page will be the first page that users see after successfully creating a new account and logging into the site. The dashboard page will also provide users with the ability to dynamically select a theme for the site. In this chapter I will show you how to combine the theme files from the Silverlight Control Toolkit with the new Dynamic and Implicit Styling features of Silverlight in order to provide the users with the ability to dynamically change the current site theme. You will also begin to make use of the ASP.NET Profile provider to make sure that the theme selection is saved in the user's profile so that it can be restored upon the next successful login. Finally, you will also see how to use the new `ChildWindow` control included in Silverlight in order to provide users with site announcements.

Chapter 5

Chapter 5 begins the first of three data entry chapters. In this chapter you will be designing and implementing the food log page where users will be able to enter the foods they eat on a daily basis. I will cover how to provide a user-friendly way to search for foods with the new `AutoCompleteBox` control. You will also see how the new `GlobalCalendar` control in the Silverlight Toolkit can be used to provide users with access to previous food log entries.

Chapter 6

In Chapter 6 you will design and implement the exercise log page. Instead of relying on the `AutoCompleteBox` control, you will be making use of a cascading `DropDownBox` solution in order to

provide users with an easy way to select exercises for their log. You will also see how to make use of the new `DomainDataSource` control to provide easier data binding with data that you make available through the WCF RIA Services created in your business logic layer.

Chapter 7

In Chapter 7 you will create the final data entry page of the application. The only page left to work on is the measurement log page. On this page, you will be providing users with the ability to keep track of various fitness measurements. To help assist users with some of the calculations, you will be building a plug-in system that will display a special modal calculator control that can be used to calculate the user's BMI value based on parameters supplied by the user.

Chapter 8

In Chapter 8 I will take another look at the dashboard page in order to complete some of the work done earlier. Part of any meaningful dashboard is to provide the users with some visual feedback related to their data. In this case, you will be adding some basic charting components to the dashboard page. As you will see, the latest version of the Silverlight Toolkit makes adding charts to your Silverlight application a breeze.

Chapter 9

In this chapter you will see how to use the features of the new Navigation Framework in order to provide a public-facing version of the user's fitness journal. After your users perform all of the hard work of dieting and exercising, they may want to share their success stories with family members or other users of the site. Now, just in case they don't yet have their own blog or social networking page, you will be creating one right here on FitnessTrackerPlus that they can use. By using the Navigation Framework, you will be able to give every user their very own unique URL to share with others that will lead to a public journal page that includes the user's food, exercise, and measurement log entries. Finally, in order to provide visitors with a means of leaving feedback to the users, you will also build a commenting system that allows for HTML-based comments. That's right—HTML content embedded in a Silverlight application. The best part is that it's not really as difficult as you may think.

Chapter 10

Chapter 10 will cover how to incorporate social networking into FitnessTrackerPlus. There is no doubt that social networking has become a major component of any successful website, and just because you will be developing in Silverlight does not mean you can't jump on the bandwagon as well. In this chapter I will show you step by step how to create a version of the public journal page in the form of a Silverlight MySpace application. This application will be available to any FitnessTrackerPlus user that has a MySpace page. Thanks to the now official support for Silverlight from MySpace, you will see how easy it is to create a Silverlight application using the new MySpace OpenSocial API and the Silverlight extensions for the API.

Chapter 11

In Chapter 11 you will see some possible techniques that you can use in order to generate revenue for the FitnessTrackerPlus application. Whether you deploy FitnessTrackerPlus to a shared hosting provider or you decide to have your own dedicated servers set up to host the site there will most certainly be a cost associated with it. Of course, generating revenue and making a profit off of the site is never usually a bad thing either. In this chapter I will show you how to integrate Pay-Per-Click advertisements directly into the Silverlight application. In case you decide that you don't want to subject your users to any kind of advertisements but still want to generate some money to offset your hosting costs, I will also show you how to charge recurring monthly fees for access to the site by using the Subscriptions feature of the PayPal developer API.

Chapter 12

In Chapter 12 I will cover how to perform the final build steps for the application as well as show you step by step how to deploy the site to a live shared hosting provider in Discount ASP.NET. You will see how Discount ASP.NET provides several unique tools that make deployment of a Silverlight-based site simple and painless. Some of these tools include utilities to manage IIS, SQL Server and more.

Getting Started

Before you move on to the next chapter on application architecture, I wanted to get you started with the project. Silverlight 4 development can be done with the full version of Visual Studio or the freely available Express edition that can be downloaded at <http://www.microsoft.com/Express>. For this book I will be using Visual Studio 2010 Professional Edition as well as the free SQL Server 2008 Express edition with Advanced Services. When downloading the free edition of SQL Server Express, choose the Advanced Services option which includes the SQL Server Management Studio tool, which you will be using throughout this book to make modifications to the application database.

Once you have your development environment set up, you will need to download several installation packages that are required for Silverlight development. All of the required packages are available for download at <http://silverlight.net>. This web site is considered the main source of Silverlight programming information for developers, and contains very useful starter tutorials as well as webcasts and online forums. Once on the site, you will see the following downloads:

- Silverlight 4 Tools for Visual Studio 2010
- Microsoft Expression Blend 3 Trial
- Silverlight Toolkit
- WCF RIA Services

Downloading the Expression Blend Trial is considered optional and I will be providing raw XAML for any user interface code that is presented in the book. Visual Studio 2010 now includes complete designer support for XAML pages, and the Expression tools provide designers with some advanced tools geared specifically towards website designers. As a developer, I find that the improved tooling in Visual Studio 2010 is more than sufficient for developing XAML pages, and will not cover the Blend tool in this book.

After downloading all of the required software, you should install the Silverlight 4 Tools for Visual Studio 2010 first. The installation process for this part of the toolkit can be quite lengthy, so feel free to brew some coffee while waiting. Once the download is complete, you will want to install the WCF RIA Services; this installation is much faster, so don't go anywhere. Finally, you will install the Silverlight Toolkit, which again is not a lengthy installation process. If you have decided to download Blend, feel free to install that at this time along with the offline help file.

At this point, you should have everything you need to get started, so it's time to fire up Visual Studio and create a new project. The code in this book will be written in C#, so I will be using the C# project templates, but if you prefer VB.NET, feel free to work with that language as it will not be terribly difficult to follow the C# code and convert it to corresponding VB.NET code. With Visual Studio open, you will want to create a new Silverlight project. Specifically, you want to use the new Silverlight Navigation Application template, as shown in Figure 1-1.

When you click OK, you're presented with the dialog shown in Figure 1-2, which includes some additional project options. In this dialog, you are asked to supply a name for the ASP.NET web project that will link with the Silverlight application. You can go ahead and leave the default name of FitnessTrackerPlus.Web. The Web project type option should stay set at ASP.NET Web Application Project, and the Silverlight Version option should stay at 4. Finally, be sure to select the Enable WCF RIA Services option before proceeding.

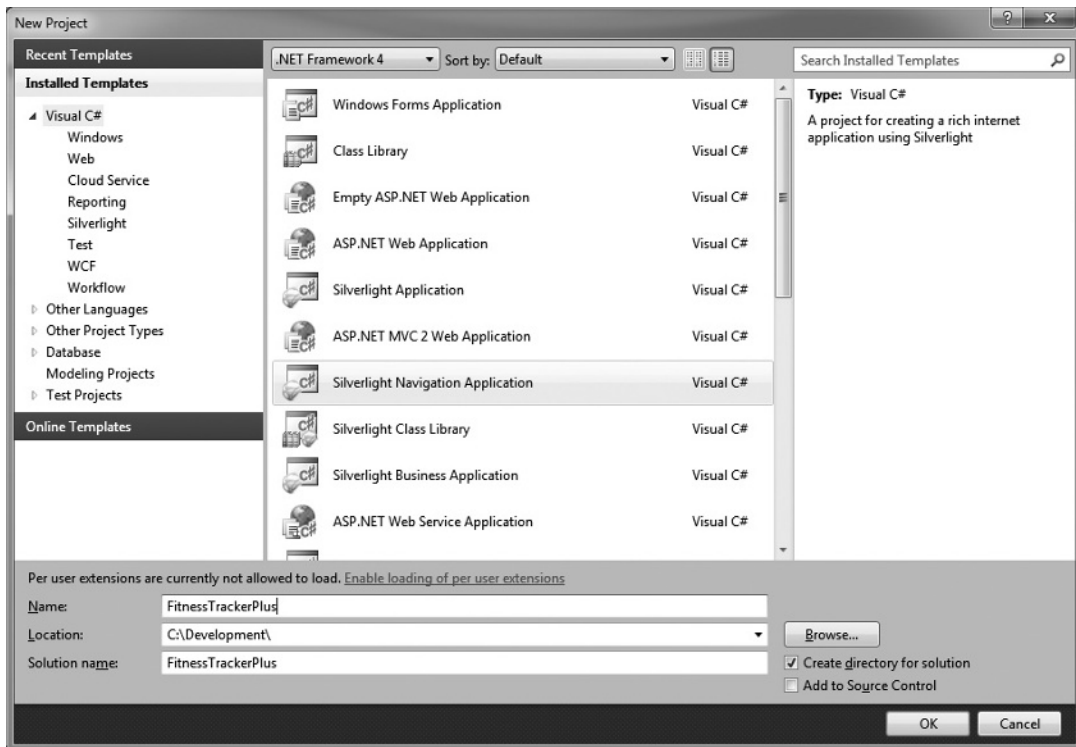


FIGURE 1-1

The Navigation Application project template creates two new projects, one for the ASP.NET web site that will host the Silverlight application called `FitnessTrackerPlus.Web`, and the Silverlight application project itself named `FitnessTrackerPlus`. In the Silverlight project, you will notice that a new folder called `Views` is generated along with `App.xaml`, and `MainPage.xaml`. In the `Views` folder, you will see three new pages called `About.xaml`, `ErrorWin.xaml`, and `Home.xaml`. You will be replacing these with your own pages, but you can see from this sample site that was generated that, unlike previous versions of Silverlight, with help from the Navigation Framework you can finally have real page navigation from XAML pages. At this point you can see the browser button support by compiling and running the application. Figure 1-3 shows the newly generated site running in the browser.

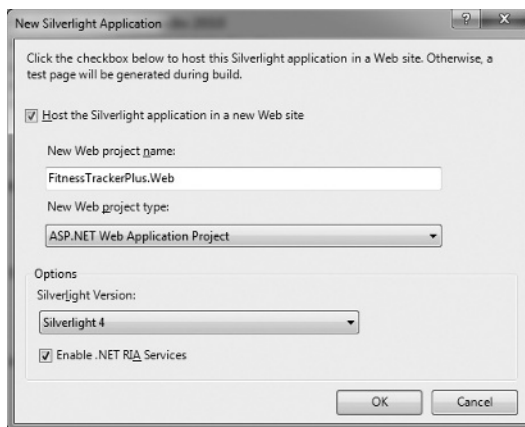


FIGURE 1-2

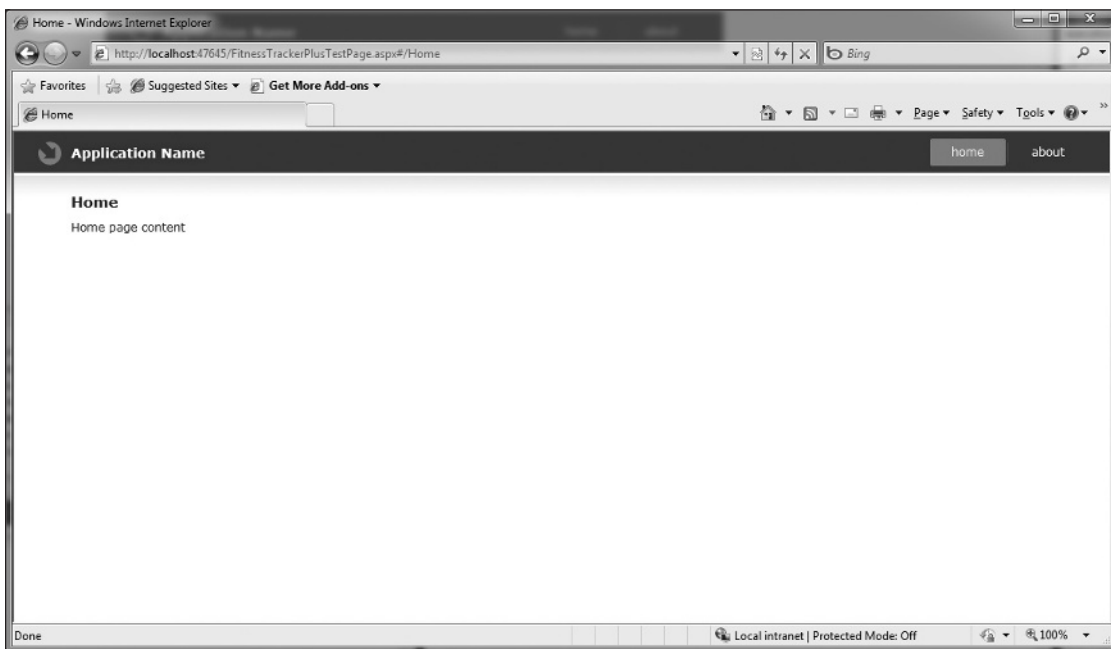


FIGURE 1-3

You should note how using the browser's Back and Forward buttons enables you to navigate between XAML pages. The URLs generated for these pages support deep linking and will allow users to bookmark and return to the exact XAML page that they were viewing at a later time. The Navigation Framework is probably one of the largest and most important enhancements included in this latest release of Silverlight and will allow you to implement some of the most important features of FitnessTrackerPlus.

Using the Navigation Framework requires very little code to implement. The majority of the work is handled by the framework and all you need to do to utilize it is use the new `System.Windows.Controls.Navigation.Frame` class. In the following code from the `MainPage.xaml` file, the `Frame` control has its `Source` property set to a relative URL called `/Home`. The `/Home` is translated into the full `/Views/Home.xaml` path by the embedded `uriMapper` object, which is covered later. For now, just know that setting the `Source` property to a valid `.xaml` file is all it takes to load up the `Frame`.



Available for
download on
Wrox.com

```
<navigation:Frame x:Name="ContentFrame" Style="{StaticResource ContentFrameStyle}"
    Source="/Home" Navigated="ContentFrame_Navigated"
    NavigationFailed="ContentFrame_NavigationFailed">
  <navigation:Frame.UriMapper>
    <uriMapper:UriMapper>
      <uriMapper:UriMapping Uri="" MappedUri="/Views/Home.xaml" />
      <uriMapper:UriMapping Uri="{pageName}"
        MappedUri="/Views/{pageName}.xaml" />
    </uriMapper:UriMapper>
  </navigation:Frame.UriMapper>
</navigation:Frame>
```

Code snippet MainPage.xaml

The actual navigation occurs by setting the `NavigateUri` property of the `HyperlinkButton` controls to the destination page desired. In the following code, the project, Home, and About pages are made accessible by setting the `NavigateUri` property to `/Home` and `/About` respectively.



Available for
download on
Wrox.com

```
<Border x:Name="LinksBorder" Style="{StaticResource LinksBorderStyle}">
  <StackPanel x:Name="LinksStackPanel" Style="{StaticResource LinksStackPanelStyle}">
    <HyperlinkButton x:Name="Link1" Style="{StaticResource LinkStyle}"
      NavigateUri="/Home" TargetName="ContentFrame" Content="home" />
    <Rectangle x:Name="Divider1" Style="{StaticResource DividerStyle}" />
    <HyperlinkButton x:Name="Link2" Style="{StaticResource LinkStyle}"
      NavigateUri="/About" TargetName="ContentFrame" Content="about" />
  </StackPanel>
</Border>
```

Code snippet MainPage.xaml

As you move further into the FitnessTrackerPlus application, I will go into more detail about the navigation system and all of its benefits, including the new URI mapping feature. Before moving on to the next chapter, you should feel free to launch the default site and see how seamless page transitions are and how you can move forward and backward through the pages using the browser without any additional code, all of it courtesy of the new Silverlight Navigation Framework.

SUMMARY

In this chapter, you have learned some of the history behind the creation of FitnessTrackerPlus, and, more importantly, how you will be leveraging the latest and greatest features of Silverlight 4 to make a rich line-of-business-style application. I briefly outlined the new features that you will be utilizing including the Navigation Framework, Data Controls, Silverlight Toolkit and more. In the Design section I covered the complete list of functional requirements that the site will have to meet, and in the solution section provided you with an overview of what to expect in the coming chapters.

You should also now have all the software required for Silverlight development downloaded and installed, and be familiar with the default navigation project that is created by the project template in Visual Studio. You pretty much have everything in place to really get started on this project. Although this book discusses the creation of an online fitness-tracking site, I strongly believe the techniques used in this site can be applied across a variety of line-of-business web applications. At the end of this book, you will have created a feature-rich Silverlight web application that has a rich data entry system, social networking aspects, and community features, and is even capable of potentially generating revenue from advertising and premium membership subscriptions. It's even possible that after developing and using this application, you might just rethink ordering that large pizza during your next all-night coding session.

