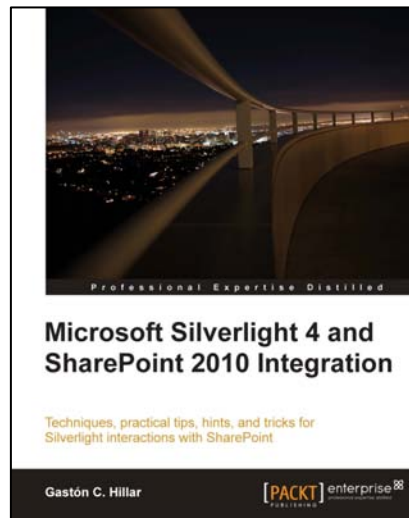


# Microsoft Silverlight 4 and SharePoint 2010 Integration

**Gastón C. Hillar**



## Chapter No.1 "Integrating Silverlight 4 with SharePoint 2010"

## In this package, you will find:

A Biography of the author of the book

A preview chapter from the book, Chapter NO.1 "Integrating Silverlight 4 with SharePoint 2010"

A synopsis of the book's content

Information on where to buy this book

## About the Author

**Gastón C. Hillar** has been working with computers since he was eight. He began programming with the legendary Texas TI-99/4A and Commodore 64 home computers in the early 80s.

He has a Bachelor's degree in Computer Science in which he graduated with honors and he also has an MBA (Master in Business Administration) in which he graduated with an outstanding thesis. He has worked as a developer, an architect, and project manager for many companies in Buenos Aires, Argentina. Now, he is an independent IT consultant and a freelance author looking for new adventures around the world. He also works with electronics (he is an electronics technician). He is always researching about new technologies and writing about them. He owns an IT and electronics laboratory with many servers, monitors, and measuring instruments.

He has written two additional books for Packt Publishing, *C# 2008 and 2005 Threaded Programming: Beginner's Guide* and *3D Game Development with Microsoft Silverlight 3: Beginner's Guide*.

**For More Information:**

[www.PacktPub.com/microsoft-silverlight-4-and-sharepoint-2010-integration/book](http://www.PacktPub.com/microsoft-silverlight-4-and-sharepoint-2010-integration/book)

He contributes to Dr. Dobb's Go Parallel programming portal <http://www.ddj.com/go-parallel/> and he is a guest blogger at Intel Software Network <http://software.intel.com/>.

In 2009, he was awarded an Intel®Black Belt Software Developer award.

He is the author of more than 40 books in Spanish about computer science, modern hardware, programming, systems development, software architecture, business applications, balanced scorecard applications, IT project management, the Internet, and electronics.

He lives with his wife, Vanesa and his son, Kevin. When not tinkering with computers, he enjoys developing and playing with wireless virtual reality devices and electronics toys with his father, his son, and his nephew Nico.

You can reach him at [gastonhillar@hotmail.com](mailto:gastonhillar@hotmail.com).

You can follow him on Twitter at <http://twitter.com/gastonhillar>.

Gastón's blog is at <http://csharpmulticore.blogspot.com>.

**For More Information:**

**[www.PacktPub.com/microsoft-silverlight-4-and-sharepoint-2010-integration/book](http://www.PacktPub.com/microsoft-silverlight-4-and-sharepoint-2010-integration/book)**

# Microsoft Silverlight 4 and SharePoint 2010 Integration

Silverlight is a powerful development platform for creating engaging, interactive user experiences for web, desktop, and mobile applications. Integrating Silverlight RIAs in SharePoint 2010 offers amazing opportunities to combine the power and flexibility offered by SharePoint. It is easy to create great user experiences when you have a step-by-step guide to implement Silverlight 4 applications on SharePoint 2010 sites, isn't it?

This book begins with the fundamental concepts of integrating Silverlight 4 with SharePoint 2010, such as the preparation of the development environment to create applications using Silverlight 4 and the addition of one or more Silverlight RIAs to a SharePoint site.

Then, it moves into the SharePoint Silverlight Client Object Model world, using step-by-step examples to combine a Silverlight application and a SharePoint module. It explains to deploy and debug the Silverlight application while it runs as a Silverlight Web Part in a SharePoint page.

Then, it is time to interact with SharePoint lists to perform CRUD (Create, Read, Update, and Delete) operations and retrieve metadata information for the fields that compose lists. The book teaches to take advantage of the new features offered by Visual Studio 2010 to browse SharePoint lists.

Once you have control over the SharePoint Silverlight Client Object Model and its asynchronous operations in Silverlight applications, it is time to access external databases through the new Business Connectivity Services (BCS) and interact with workflows. Then, the book explains how to perform CRUD operations by consuming the new SharePoint 2010 WCF Data Services in Silverlight.

Then, consuming different kinds of multimedia files from the new SharePoint 2010 Digital Asset Libraries offers the final touches to the Silverlight 4 and SharePoint 2010 integration learning experience.

In each example, the book teaches to take advantage of unique features offered by Silverlight in order to create impressive User eXperiences that interact with SharePoint 2010. For example, themes, effects, animations, data-binding, and Silverlight 4 Out-of-Browser capabilities.

## What This Book Covers

*Chapter 1, Integrating Silverlight 4 with SharePoint 2010*, briefs you about integration of Silverlight 4 applications with SharePoint 2010 sites and solutions. In this chapter, you will learn to prepare a development environment and look at the tools to work with

**For More Information:**

[www.PacktPub.com/microsoft-silverlight-4-and-sharepoint-2010-integration/book](http://www.PacktPub.com/microsoft-silverlight-4-and-sharepoint-2010-integration/book)

Silverlight 4 RIAs. You will also configure the SharePoint 2010 server and add Silverlight Web Parts to a new page. Finally, you will create your first Silverlight RIA and then make it available in a SharePoint site.

*Chapter 2, Deploying and Debugging Techniques for Silverlight and SharePoint*, will help you create a new Silverlight 4 RIA that retrieves data from SharePoint through the SharePoint Silverlight Client Object Model. In this chapter, you will create your first Silverlight RIA that interacts with a list of tasks in SharePoint and learn a way to run and debug asynchronous queries to the SharePoint server run by using multiple threads.

*Chapter 3, Interacting with Data on the SharePoint Server*, covers advanced interaction with data on the SharePoint Server. In this chapter, you will create a new Silverlight 4 RIA that allows us to insert new items into a remote SharePoint list and later enhance this simple application to retrieve metadata information for the fields. You will also perform delete and update operations to the remote SharePoint list through the SharePoint Silverlight Client Object Model.

*Chapter 4, Creating Dynamic Business Solutions*, guides you to create dynamic business solutions by accessing external databases in a Silverlight RIA included in a SharePoint solution and interacting with workflows. You will also work with the new SharePoint 2010 approval workflows and enhance an existing Silverlight RIA to retrieve workflow status information and display it in a column of a grid.

*Chapter 5, Working with WCF Data Services*, helps you develop Silverlight 4 applications in SharePoint 2010 sites that interact with data in lists by performing insert, update, and delete operations with WCF Data Services. In this chapter, you will use Fiddler to inspect the communication between the Silverlight RIA and the SharePoint server. Finally, you will analyze the performance and resource usage information for pages and Web Parts with SharePoint Developer Dashboard.

*Chapter 6, Interacting with Rich Media and Animations*, will help you access asset libraries in a Silverlight RIA rendered in a SharePoint Visual Web Part. In this chapter, you will learn to take advantage of Silverlight 4 rich media features to add effects and interactive animations to images and videos.

**For More Information:**

[www.PacktPub.com/microsoft-silverlight-4-and-sharepoint-2010-integration/book](http://www.PacktPub.com/microsoft-silverlight-4-and-sharepoint-2010-integration/book)

# 1

## Integrating Silverlight 4 with SharePoint 2010

We want to include Silverlight 4 RIAs (**Rich Internet Applications**) in SharePoint 2010. RIAs provide rich experience for users, both through their browsers and outside them. Integrating Silverlight RIAs in SharePoint 2010 offers amazing opportunities to combine the power and flexibility offered by SharePoint with great user experiences. In fact, many interfaces shown in SharePoint 2010 are developed in Silverlight. We want to integrate Silverlight RIAs into SharePoint 2010. First, we must understand some of the fundamentals that are related to various tools and their configurations. In this chapter, we will cover many topics to help us understand the new tools and techniques involved in creating Silverlight RIAs for SharePoint 2010 sites. We will:

- Understand the benefits of integrating Silverlight with SharePoint
- Prepare the development environment to develop applications for SharePoint 2010 using Silverlight 4
- Prepare the SharePoint 2010 server to host Silverlight applications
- Create a Silverlight Line of Business RIA
- Learn to add a Silverlight RIA to a SharePoint site
- Understand the advantage of creating rich user experiences for SharePoint solutions
- Work with shared documents to store a Silverlight application in SharePoint sites
- Work with many Silverlight applications in a single page
- Learn the differences between client and server code

**For More Information:**

[www.PacktPub.com/microsoft-silverlight-4-and-sharepoint-2010-integration/book](http://www.PacktPub.com/microsoft-silverlight-4-and-sharepoint-2010-integration/book)

## Understanding the benefits of integrating Silverlight with SharePoint

The following list shows many benefits of integrating Silverlight with SharePoint 2010:


- **Rich UX:** Silverlight RIAs can offer a rich user experience. You can take full advantage of the rich visual capabilities offered by Silverlight and include them in a SharePoint site. The rich and interactive content offers an incredible new world of possibilities in SharePoint. For example, you can offer an interactive balanced scorecard with animated graphs, rich navigation capabilities, and context menus.
- **Code runs on the client:** You can take advantage of the power of the client computers accessing the SharePoint server. You can use threading and asynchronous calls to offer responsive user interfaces and to take advantage of modern multi-core microprocessors found in client computers. You can offer great response times without the need to wait for the server to load another page. You can take advantage of rich controls, animations, and exciting multimedia effects. The processing removes load from the server and enables you to use both the server and the client in your solutions. Additionally, Silverlight 4 is cross-browser capable and we can take advantage of the improved Out of Browser features to create applications that interact with the SharePoint 2010 server but run in the Windows desktop, out of the web browser.
- **Efficient applications:** As you can work with the power offered by the client, you can process data without the need to make requests to the server all the time. This way, you can create load-balanced solutions.
- **Access to the Client OM (Client Object Model):** When you have to access data and services offered by the SharePoint 2010 server, you don't need to create your own complex infrastructure. There is no need to add additional layers. You can take advantage of the new Client Object Model, also known as Client OM. As you can work with asynchronous calls to the Client OM, you can still offer great responsive applications when consuming services from the server. Users can interact with SharePoint data without requiring server calls as they would from traditional pages. Lots of the processing can be pushed down to the client. This way, as previously explained, you can remove load from the SharePoint server and create load-balanced solutions.
- **Leverage your existing Silverlight knowledge, components, and applications:** You can build new capabilities quickly from existing Silverlight components and applications, integrating them with SharePoint 2010.

Considering the aforementioned benefits, we will work hard to learn all the possibilities offered by the integration of Silverlight and SharePoint 2010 in later parts of the book.

## Creating a SharePoint solution


Now, when we design a new SharePoint 2010 solution, we will be able to consider Silverlight RIAs as new components for the global solution. We have to consider the aforementioned benefits of integrating Silverlight with SharePoint and decide which parts would be convenient to create as Silverlight RIAs.

This way, we can focus on preparing the SharePoint 2010 infrastructure and then we can access data and services offered by the server through Silverlight RIAs. For example, you can view the images found in an assets library defined in SharePoint through a Silverlight application.

 Once you start integrating Silverlight with SharePoint, you will find a new exciting way of enhancing SharePoint solutions.

## Preparing the development environment

We want to take full advantage of modern technologies. First of all, we must install the latest tools and begin working on configurations. Later, we will be able to use our existing knowledge to create different kinds of RIAs for SharePoint 2010, using Silverlight 4 – the newest kid-on-the-block from Microsoft.

 Silverlight 4 is backward-compatible with its previous version, Silverlight 3. Therefore, when an example uses a feature found only in Silverlight 4, you will find a note explaining this situation. Most of the examples work for both Silverlight versions. However, we will also take advantage of some of the new features found in Silverlight 4.

The only requirements underpinning the development and integration of RIAs into SharePoint 2010 sites are understanding the basics of the C# programming language, ASP.NET, XAML code, and the Visual Studio IDE. We will cover any other requirements in our journey through the creation of many different kinds of RIAs to run in a SharePoint 2010 site. First, we must download and install various Silverlight development tools. We need Visual C# 2010 Professional, Premium, or Ultimate installed, in order to successfully complete the installations explained in the following section. Visual C# 2010 allows us to choose the desired Silverlight version (for example, version 3 or version 4). The following sections and chapters will show Visual Studio 2010 Ultimate screenshots. If you use other versions, some elements that appear in the screenshots could be different but the steps are all valid for the aforementioned versions.

## Setting up the development environment

Follow these steps to prepare the development environment:

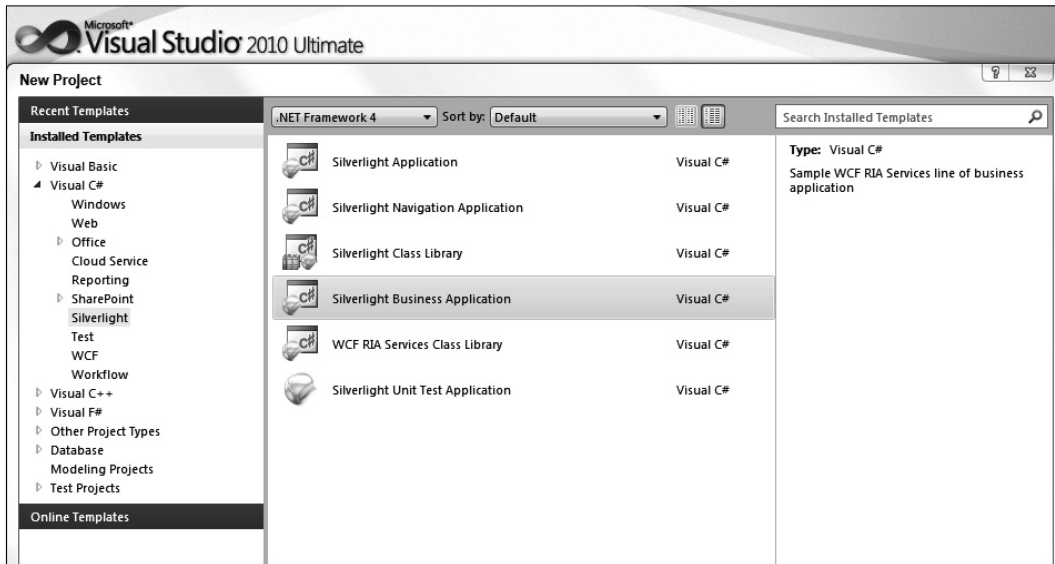
1. Download the following files:

Application's name	Download link	File name	Description
Silverlight 4 Tools for Visual Studio 2010	<a href="http://www.microsoft.com/downloads/details.aspx?FamilyID=eff8a0da-0a4d-48e8-8366-6ddf2ecad801&amp;displaylang=en">http://www.microsoft.com/downloads/details.aspx?FamilyID=eff8a0da-0a4d-48e8-8366-6ddf2ecad801&amp;displaylang=en</a>	Silverlight4_Tools.exe	We must install Silverlight 4 Tools in order to create Silverlight 4 applications in the Visual Studio 2010 IDE, using XAML and C#. It will co-exist with previous Silverlight SDKs (Software Development Kits). This new version of Silverlight Tools also includes the WCF RIA Services package.

Application's name	Download link	File name	Description
Silverlight 4 Offline Documentation (in CHM format)	<a href="http://www.microsoft.com/downloads/details.aspx?familyid=B6127B9B-968C-46C2-8CB6-D228E017AD74&amp;displaylang=en">http://www.microsoft.com/downloads/details.aspx?familyid=B6127B9B-968C-46C2-8CB6-D228E017AD74&amp;displaylang=en</a>	Silverlight_Documentation.EXE	We must download and run this file to decompress its content, because we will need access to Silverlight 4 official documentation in due course.
Expression Blend for .NET 4	<a href="http://www.microsoft.com/downloads/details.aspx?FamilyID=88484825-1b3c-4e8c-8b14-b05d025e1541&amp;displaylang=en">http://www.microsoft.com/downloads/details.aspx?FamilyID=88484825-1b3c-4e8c-8b14-b05d025e1541&amp;displaylang=en</a>	Blend_Trial_en.exe	This tool will enable us to create content that targets Silverlight 4 and to create rapid prototypes with the SketchFlow tool.
Silverlight Toolkit (Updated for Silverlight 4 compatibility)	<a href="http://codeplex.com/Silverlight">http://codeplex.com/Silverlight</a>	Silverlight_4_Toolkit_April_2010.msi	It is convenient to download the latest stable release. This toolkit provides a nice collection of Silverlight controls, components, and utilities made available outside the normal Silverlight release cycle. It will be really helpful to use these controls to provide even more attractive user interfaces. Besides, it includes more Silverlight themes.

2. Run the installers in the same order in which they appear in the above table and follow the steps to complete the installation wizards.
3. Once the installations have successfully finished, run Visual Studio 2010 or Visual Web Developer 2010 (or later).

4. Select **File | New | Project...** or press *Ctrl+Shift+N*. Select **Visual C# | Silverlight** under **Installed Templates** in the **New Project** dialog box. You will see many Silverlight templates, including **Silverlight Business Application** and **WCF RIA Services Class Library**, as shown in the following screenshot:



## Discovering the rich controls offered by the Silverlight Toolkit

Silverlight Toolkit is a Microsoft project offering many rich controls, components, and utilities that can help us to enhance our Silverlight UI (User Interface). As we want to create a very attractive UI for SharePoint, it is convenient to get familiar with its features. Follows these steps to see the controls in action and to change the values for many of their properties.

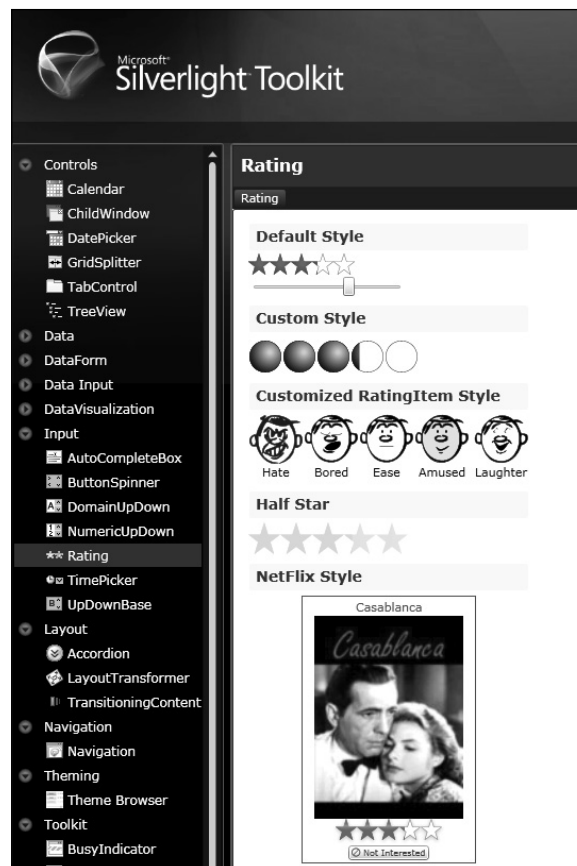
1. Select **Start | All Programs | Microsoft Silverlight 4 Toolkit April 2010 | Toolkit Samples** and your default web browser will display a web page with a Silverlight application displaying a list of the controls organized in ten categories as follows:
  - Controls
  - Data
  - DataForm
  - Data Input

- DataVisualization
- Input
- Layout
- Navigation
- Theming
- Toolkit



By default, the default .htm web page is located at C:\Program Files (x86)\Microsoft SDKs\Silverlight\v4.0\Toolkit\Apr10\Samples in 64-bit Windows versions.

2. Click on a control name under the desired category and the right panel will display the control with different values assigned for its properties, creating diverse instances of the control. For example, the following screenshot shows many instances of the Rating control under the **Input** category.



- Click on the buttons shown at the bottom of the web page and you will be able to see both the XAML and the C# code used to create the sample for the control. For example, the following screenshot shows the XAML code for the DataGrid control example, DataGridSample.xaml. You can also click on DataGridSample.xaml.cs and check the C# part. This control appears under the **Data** category.

The screenshot displays the Visual Studio IDE with the DataGrid control configuration window open. The configuration window shows various properties for the DataGrid, including AreRowDetailsFrozen, CanUserReorderColumns, CanUserResizeColumns, CanUserSortColumns, FrozenColumnCount, IsReadOnly, IsValid, RowDetailsVisibilityMode, and SelectionMode. A data grid is visible in the background, showing columns for Age, Complaint, FirstName, and FullAddress.

Age	Complaint	FirstName	FullAddress
8	Closed	BGGQSTPK14	WVMFVPQP
38	Resolved	CKXKMGJU13	IBGMTHUI
63	Resolved	DTXEHGMT16	YTXSSSJQ
11		DXERURET5	BCNSJIRQ
10	Resolved	HKRQGKWF10	JUOEFUFG
88	Resolved	HXGUFJCB24	UGODHLDV
2	Active	IAXGSJW18	AEQFTEXP
57		Jane	24, place Kléber, 67000 Str
		Jill	Geislweg 36, Salzburg 5020
	Closed	Joe	Strada Provinciale 124, 421
35	Closed	John	Obere Str. 57, Berlin 12209

The XAML code in the bottom pane is as follows:

```

// All other rights reserved.
-->
<UserControl x:Class="System.Windows.Controls.Samples.DataGridSample"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:datagrid="clr-namespace:System.Windows.Controls;assembly=System.Windows.Controls.Data"
  xmlns:input="clr-namespace:System.Windows.Controls;assembly=System.Windows.Controls.Input.Toolkit"
  xmlns:sys="clr-namespace:System;assembly=mscorlib">
  <StackPanel>
    <ContentControl Content="DataGrid" Style="{StaticResource Header}" />
    <Grid>
      <Grid.ColumnDefinitions>
        <ColumnDefinition />
        <ColumnDefinition />
      </Grid.ColumnDefinitions>
      <StackPanel Grid.Column="0">
        <CheckBox Content="AreRowDetailsFrozen" IsChecked="{Binding AreRowDetailsFrozen, ElementName=dataGrid, Mode=TwoWay}" />
        <CheckBox Content="CanUserReorderColumns" IsChecked="{Binding CanUserReorderColumns, ElementName=dataGrid, Mode=TwoWay}" />
        <CheckBox Content="CanUserResizeColumns" IsChecked="{Binding CanUserResizeColumns, ElementName=dataGrid, Mode=TwoWay}" />
        <CheckBox Content="CanUserSortColumns" IsChecked="{Binding CanUserSortColumns, ElementName=dataGrid, Mode=TwoWay}" />
        <TextBlock Style="{StaticResource ApiName}" Text="{Binding FrozenColumnCount, ElementName=dataGrid, Mode=TwoWay}" />
        <input:NumericUpDown VerticalAlignment="Center" Minimum="0" Margin="4" Value="{Binding FrozenColumnCount, ElementName=dataGrid, Mode=TwoWay}" />
      </StackPanel>
    </Grid>
  </StackPanel>
</UserControl>

```

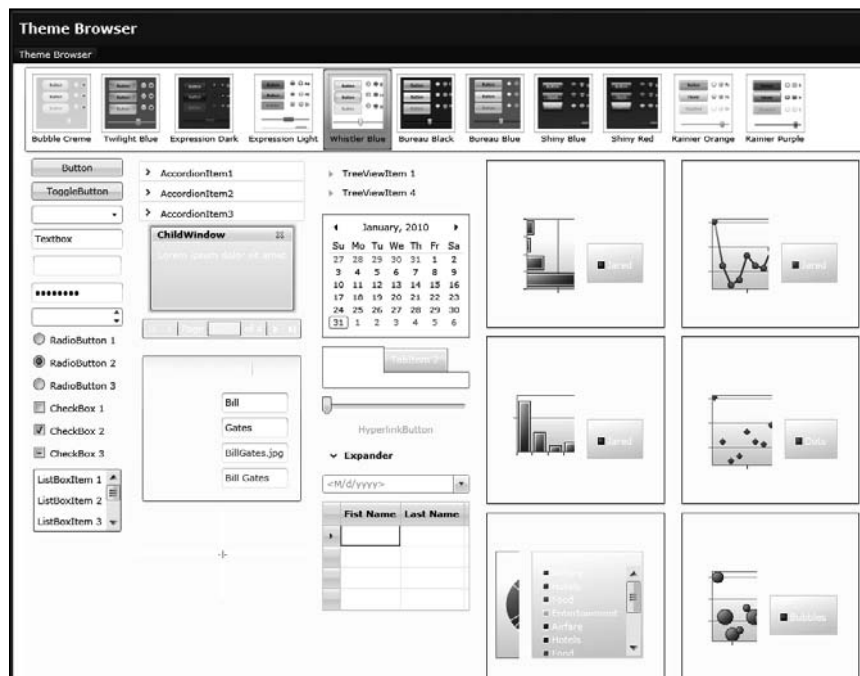
**For More Information:**  
[www.PacktPub.com/microsoft-silverlight-4-and-sharepoint-2010-integration/book](http://www.PacktPub.com/microsoft-silverlight-4-and-sharepoint-2010-integration/book)

## Browsing themes with sample controls

Silverlight Toolkit also includes 11 themes that allow us to change and improve the overall look-and-feel for our Silverlight UI. They are:

- Bubble Creme
- Bureau Black
- Bureau Blue
- Expression Dark
- Expression Light
- Rainier Purple
- Rainier Orange
- Shiny Blue
- Shiny Red
- Twilight Blue
- Whistler Blue

Click on **Theme Browser** under the **Theming** category and you will be able to select one of the themes shown in the previous list to preview the look-and-feel of many controls. The following screenshot shows the preview for the **Whistler Blue** theme:



**For More Information:**

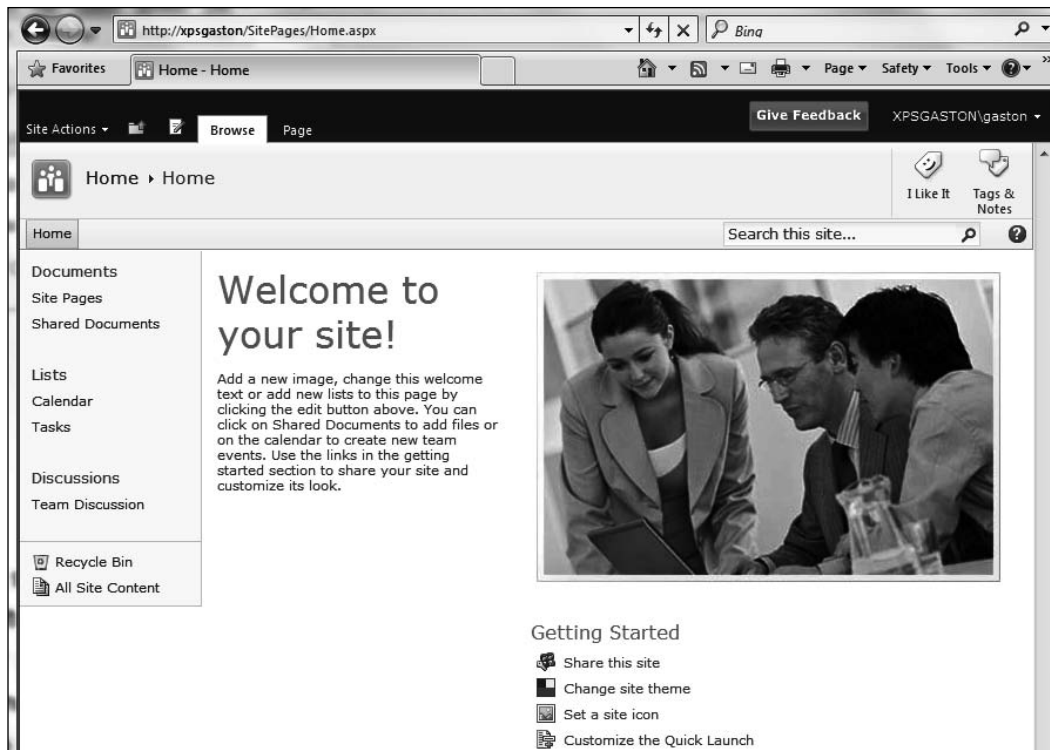
[www.PacktPub.com/microsoft-silverlight-4-and-sharepoint-2010-integration/book](http://www.PacktPub.com/microsoft-silverlight-4-and-sharepoint-2010-integration/book)

## Preparing the server

So far, we have prepared the development environment. Now, it is time to make sure that we have the necessary configuration for the SharePoint 2010 server in which we are going to integrate Silverlight applications.

In order to complete all the examples that we will develop throughout this book, you must be an administrator of a SharePoint site collection. SharePoint Server 2010 or SharePoint Foundation 2010 must be installed in the same computer that runs Visual Studio 2010. You can check the necessary steps to perform a SharePoint Server 2010 or SharePoint Foundation 2010 installation for your development computer at <http://msdn.microsoft.com/en-us/library/ee554869.aspx>. Follow these steps to ensure that you are a site collection administrator:

1. Open your default web browser, view the SharePoint site, and log in with your username and password. You have to enter the SharePoint server URL. In our examples, we will use `http://xpsgaston` as our default SharePoint 2010 site. However, you have to replace it with your SharePoint 2010 site URL. Your default site will appear, in this case, `http://xpsgaston/ SitePages/Home.aspx`, as shown in the following screenshot:



2. Click **Site Actions | Site Permissions** and a list of users with their permission levels will appear.
3. Now, click on **Site Collection Administrators** in the ribbon. A new dialog box with the names of the users with administrator rights on this site collection will appear.



4. If your user name appears in the text box, you are a Site Collection Administrator.

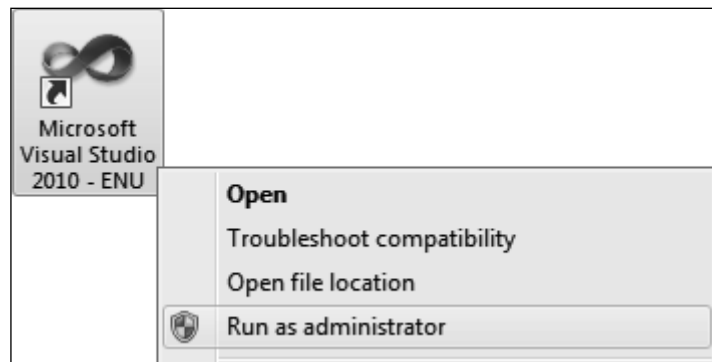


If you cannot see the **Site Collection Administrators** button in the ribbon, it means that you don't have site collection administrator privileges on the site. In this case, you have to request this permission from the SharePoint site administrator.

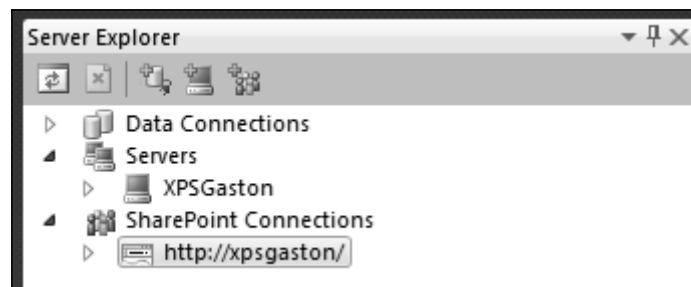
## Browsing SharePoint Site collections

Once we have ensured that our username is a Site Collection Administrator, we can use **Server Explorer** in Visual Studio to browse a SharePoint site.

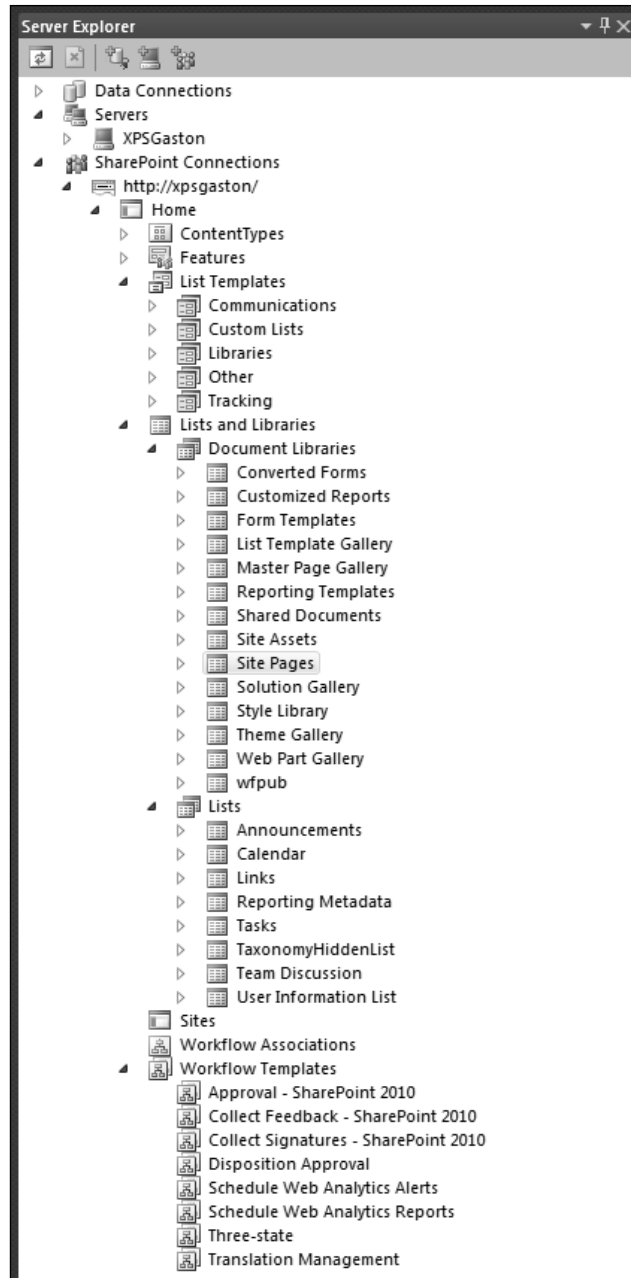
1. Start Visual Studio as a system administrator user. In Windows Server 2008 R2, 2008, and 2003, if you are already logged as Administrator on the machine you can simply run the application. However, in Windows 7 or Windows Vista, you can do it by right-clicking on its shortcut and selecting **Run as administrator** in the context menu that appears, as shown:



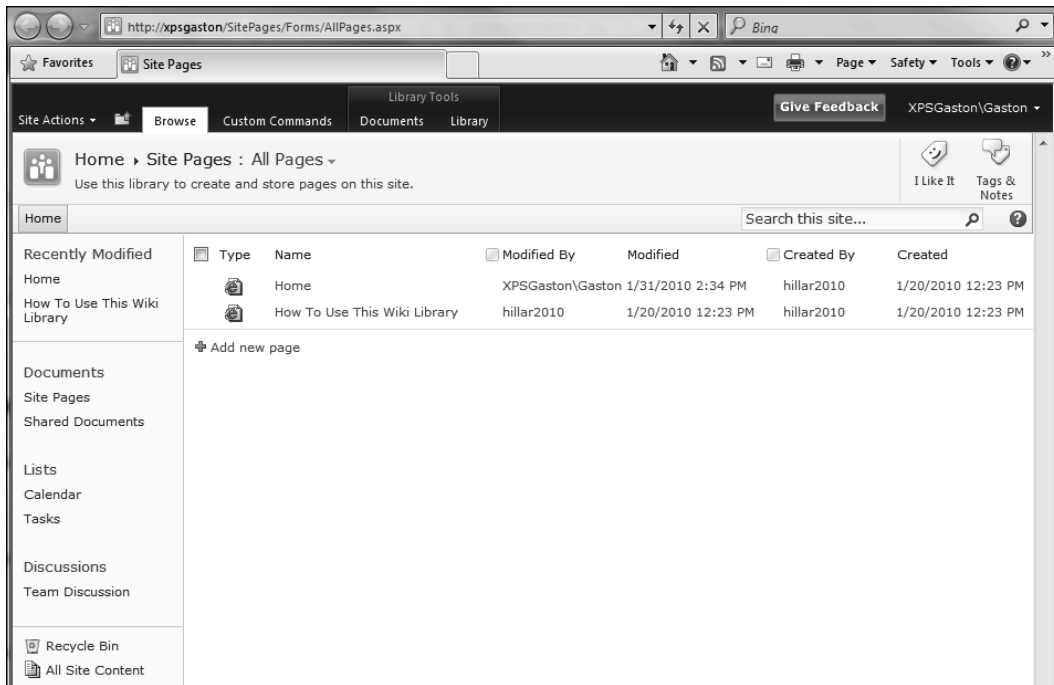
2. Activate the **Server Explorer** palette. If it isn't visible, you have to select **View | Server Explorer** in the main menu.
3. Click on the expand button for **SharePoint Connections**. If the name of your desired SharePoint 2010 server doesn't appear in the list, you can manually connect to the server. You can do it by right-clicking on **SharePoint Connections** and selecting **Add Connection...** in the context menu that appears. Then, you have to enter the URL for the server, for example, `http://xpsgaston` and click **OK** in the dialog box that appears. If your user has the previously explained privileges, the server will appear in the list.



- Now, click on the expand button for the SharePoint server and you will be able to browse its different nodes. Every component of a SharePoint site is represented by a node in the **Server Explorer** tree view. You can inspect the properties for each node, as shown in the following screenshot:



5. You can view some lists in your default web browser by right-clicking on a node and then selecting **View in Browser** in the context menu that appears. For example, you can do it for the node **Home | Lists and Libraries | Site Pages** and your default web browser will display all the pages. In this case, the URL shown is `http://xpsgaston/SitePages/Forms/AllPages.aspx`.

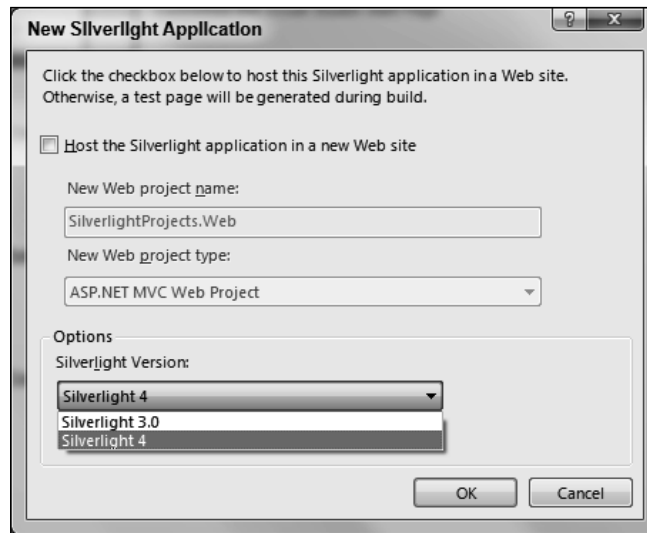


Remember to run Visual Studio as a system administrator user in order to interact with SharePoint for all the examples covered in this book.

## Creating a Silverlight LOB (Line-of-Business) RIA

Now, we are going to create a very simple Silverlight LOB (Line-Of-Business) RIA that retrieves data, displays a grid with a list of projects, and allows the users to navigate through the data. Then, we are going to integrate this Silverlight UI in SharePoint.

1. Create a new Visual C# project using the **Silverlight | Silverlight Application** template. Use `SilverlightProjects` as the project's name.
2. Deactivate the **Host the Silverlight application in a new Web site** checkbox in the **New Silverlight Application** dialog box. We want the Silverlight application to run in a simple HTML web page. As you have installed Silverlight 4 Tools, the dialog box will offer you a combo box with the possibility to choose the desired Silverlight version. Select **Silverlight 4** as we want to take advantage of the new features offered by this version.



3. Add a new XML file to the project, `Projects.xml`. The following lines define properties and values for five project instances. This way, we have some data in XML format for our simple LOB application.

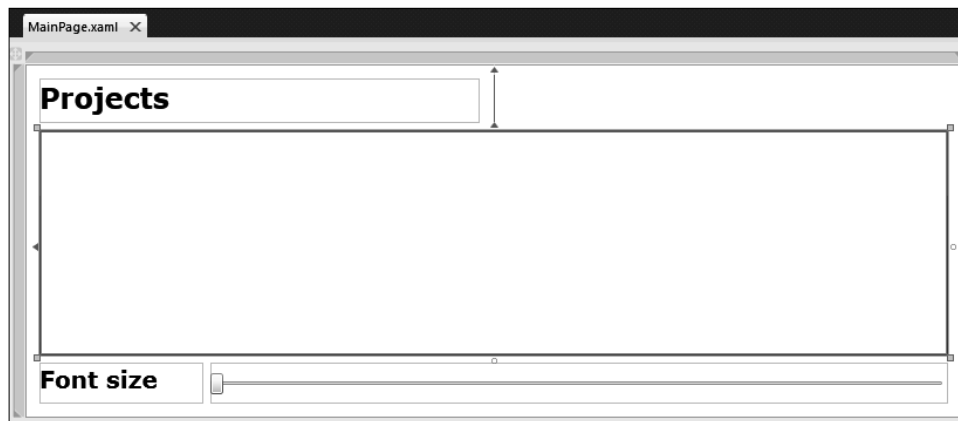
```
<?xml version="1.0" encoding="utf-8" ?>
<projects>
  <project projectId="0">
    <title>Creating a Silverlight 4 UI</title>
    <estimatedDaysLeft>4</estimatedDaysLeft>
    <status>Delayed</status>
    <assignedTo>Jon Share</assignedTo>
    <numberOfTasks>5</numberOfTasks>
  </project>
  <project projectId="1">
    <title>Creating a Complex Silverlight LOB RIA</title>
    <estimatedDaysLeft>5</estimatedDaysLeft>
    <status>Delayed</status>
```

```
<assignedTo>James Point</assignedTo>
<numberOfTasks>35</numberOfTasks>
</project>
<project projectId="2">
  <title>Creating a New SharePoint Site</title>
  <estimatedDaysLeft>3</estimatedDaysLeft>
  <status>Delayed</status>
  <assignedTo>Vanessa Dotcom</assignedTo>
  <numberOfTasks>8</numberOfTasks>
</project>
<project projectId="3">
  <title>Installing a New SharePoint 2010 Server</title>
  <estimatedDaysLeft>3</estimatedDaysLeft>
  <status>Delayed</status>
  <assignedTo>Michael Desktop</assignedTo>
  <numberOfTasks>25</numberOfTasks>
</project>
<project projectId="4">
  <title>Testing the New Silverlight LOB RIA</title>
  <estimatedDaysLeft>4</estimatedDaysLeft>
  <status>Delayed</status>
  <assignedTo>Jon Share</assignedTo>
  <numberOfTasks>35</numberOfTasks>
</project>
</projects>
```

4. Add a new class to the project called `Project` in a new class file, `Project.cs`. The following lines define the new class, with six properties. This way, you will be able to create instances of this class to hold the values defined in the previously created XML file.

```
public class Project
{
    public int ProjectId { get; set; }
    public string Title { get; set; }
    public int EstimatedDaysLeft { get; set; }
    public string Status { get; set; }
    public string AssignedTo { get; set; }
    public int NumberOfTasks { get; set; }
}
```

5. Open `MainPage.xaml`, define a new width and height for the `Grid` as 800 and 600, add the following controls located in the **Toolbox** under **All Silverlight Controls**, and align them as shown in the screenshot. Remember that Visual Studio 2010 allows us to drag-and-drop controls from the toolbox to the Silverlight `UserControl` in the design view and it will automatically generate the XAML code.
  - Two `Label` controls.
  - One `DataGrid` control and set its name to `dataGridProjects`. Set its `AutoGenerateColumns` property to `true`.
  - One `Slider` control, `sliGridFontSize`. Set its `Minimum` property to 8, `Maximum` to 72, and `Value` to 11.



6. Apply data binding to the font size for the `DataGrid` control, `dataGridProjects`. In order to do so, select `dataGridProjects`, activate the **Properties** panel, display them in alphabetical order, right-click on the **FontSize** property, and select **Apply Data Binding** in the context menu that appears. Then, select `ElementName` in **Source**, `sliGridFontSize`, and then `Value` in **Path**. This way, when the user moves the slider, the font size for the data grid will change. The code that defines the data binding is as follows:

```
FontSize="{Binding ElementName=sliGridFontSize, Path=Value}"
```

7. The complete XAML markup code for `MainPage.xaml` will be similar to the following lines:

```
<UserControl x:Class="SilverlightProjects.MainPage"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/
  presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
  xmlns:mc="http://schemas.openxmlformats.org/markup-
  compatibility/2006"
  mc:Ignorable="d"
  d:DesignWidth="800" d:DesignHeight="600" xmlns:data=
  "clr-namespace:System.Windows.Controls;
  assembly=System.Windows.Controls.Data" xmlns:dataInput=
  "clr-namespace:System.Windows.Controls;
  assembly=System.Windows.Controls.Data.Input">

  <Grid x:Name="LayoutRoot" Background="White">
    <data:DataGrid AutoGenerateColumns="True" Height="491"
      HorizontalAlignment="Left" Margin="12,56,0,0"
      Name="dataGridProjects" VerticalAlignment="Top"
      Width="776" FontSize="{
      Binding ElementName=sliGridFontSize, Path=Value}" />

    <dataInput:Label Height="38" HorizontalAlignment="Left"
      Margin="12,12,0,0" Name="label1"
      VerticalAlignment="Top" Width="376"
      FontWeight="Bold" FontSize="24"
      Content="Projects" />

    <Slider Height="35" HorizontalAlignment="Left"
      Margin="158,553,0,0" Name="sliGridFontSize"
      VerticalAlignment="Top" Width="630" Value="11"
      Maximum="72" Minimum="11" />

    <dataInput:Label Content="Font size" FontSize="20"
      FontWeight="Bold" Height="35" HorizontalAlignment="Left"
      Margin="12,553,0,0" Name="label2"
      VerticalAlignment="Top" Width="140" />
  </Grid>
</UserControl>
```

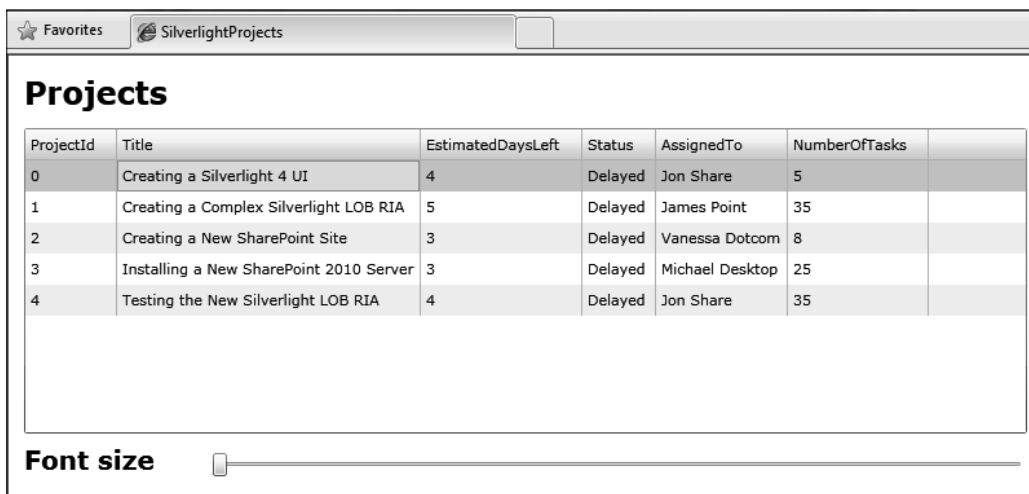
8. Now, it is necessary to add code to retrieve data from the XML file and assign a value to the `ItemsSource` property of the `DataGrid` control. First, you have to add a reference to `System.Xml.Linq.dll`. Then, you can add the new `InitializeGrid` method and call it from the class constructor as shown in the following lines for `MainPage.xaml.cs`.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Animation;
using System.Windows.Shapes;
// Added
using System.Xml.Linq;

namespace SilverlightProjects
{
    public partial class MainPage : UserControl
    {
        private void InitializeGrid()
        {
            XDocument docProjects =
                XDocument.Load("Projects.xml");
            var projectsData = from el in
                docProjects.Descendants("project")
                select new Project
                {
                    ProjectId = Convert.
ToInt32(el.Attribute("projectId").Value),
                    Title = Convert.ToString(el.
Element("title").Value),
                    EstimatedDaysLeft = Convert.
ToInt32(el.Element("estimatedDaysLeft").Value),
                    Status = Convert.ToString(el.
Element("status").Value),
                    AssignedTo = Convert.
ToString(el.Element("assignedTo").Value),
                    NumberOfTasks = Convert.
ToInt32(el.Element("numberOfTasks").Value)
                };
            dataGridProjects.ItemsSource = projectsData;
        }
        public MainPage()
```

```
    {  
        InitializeComponent();  
        InitializeGrid();  
    }  
}
```

9. Build and run the solution. The default web browser will appear showing a grid with headers and the five rows defined in the previously added XML file, as shown in the following screenshot:



It is a very simple Silverlight LOB RIA displaying a data grid with rows that are read from the XML file included in the project, `Projects.xml`.

First, we added the XML file with the definitions for the five projects. Then, we added a class with the necessary properties to hold the values defined in this XML file.

The `InitializeGrid` method loads the projects from the `Projects.xml` XML file (embedded and compressed in the `.xap` file).

```
XDocument docProjects = XDocument.Load("Projects.xml");
```

Then, it uses a LINQ to XML query to create instances of the `Project` class and assign values to their properties. Finally, it assigns this query to the `ItemsSource` property of the `DataGrid`:

```
dataGridProjects.ItemsSource = projectsData;
```



C# 3.0 (Visual C# 2008) introduced LINQ and it is very useful for processing queries for many different data sources. The features of LINQ and its usage in real-life scenarios are described in depth in *LINQ Quickly (A Practical Guide to Programming Language Integrated Query with C#)* by N. Satheesh Kumar from Packt Publishing.

## Creating rich User eXperiences (UX)

We can click on one of the headers and the grid will sort the data in ascending order. Then, we can click again to sort the data into descending order.

When we drag the slider located at the bottom, the font size for the grid will change, as shown in the following screenshot:

Projects	
ProjectId	Title
0	Creating a Silverlight 4 UI
1	Creating a Complex Silverlight
2	Creating a New SharePoint Site
3	Installing a New SharePoint 20
4	Testing the New Silverlight LOI

Font size

As previously explained, we can also take advantage of the themes included in Silverlight's Toolkit to offer the user a more exciting UI. Follow these steps to apply a theme to the main `UserControl` for the Silverlight UI:

1. Select **Start | All Programs | Microsoft Silverlight 4 Toolkit | Binaries** and Windows will open the folder that contains the Silverlight Toolkit binaries. By default, they are located at `C:\Program Files (x86)\Microsoft SDKs\Silverlight\v4.0\Toolkit\Apr10\Bin` in 64-bit Windows versions. Its parent folder contains the Themes sub-folder, `C:\Program Files (x86)\Microsoft SDKs\Silverlight\v4.0\Toolkit\Apr10\Themes`. In 32-bit Windows versions, the default folders are `C:\Program Files\Microsoft SDKs\Silverlight\v4.0\Toolkit\Apr10\Bin` and `C:\Program Files\Microsoft SDKs\Silverlight\v4.0\Toolkit\Apr10\Themes`.

**For More Information:**

[www.PacktPub.com/microsoft-silverlight-4-and-sharepoint-2010-integration/book](http://www.PacktPub.com/microsoft-silverlight-4-and-sharepoint-2010-integration/book)

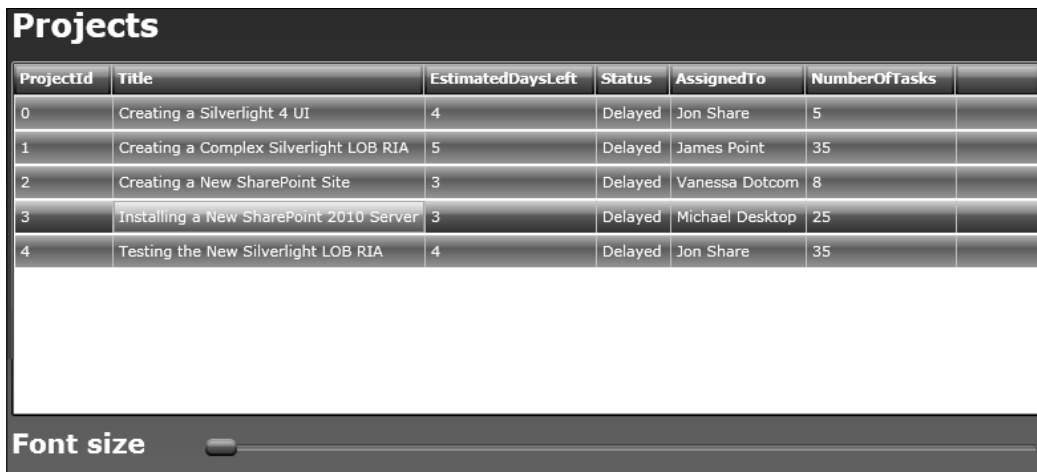
2. Add a reference to `System.Windows.Controls.Theming.Toolkit.dll`. Remember that it is located in the aforementioned `Bin` sub-folder.
3. Add a reference to the DLL for the desired theme in the `Themes` sub-folder. For example, if you want to apply the `ShinyBlue` theme, add `System.Windows.Controls.Theming.ShinyBlue`, located in the aforementioned `Themes` sub-folder.
4. Add the following line to include the namespace that defines the theme in the `UserControl` defined in `MainPage.xaml`:  

```
xmlns:shinyBlue=  
    "clr-namespace:System.Windows.Controls.Theming;  
    assembly=System.Windows.Controls.Theming.ShinyBlue"
```
5. Add the following line before the definition of the main `Grid, LayoutRoot`:  

```
<shinyBlue:ShinyBlueTheme>
```
6. Add the following line after the definition of the main `Grid, LayoutRoot`:  

```
</shinyBlue:ShinyBlueTheme>
```

This way, the `ShinyBlue` theme will be applied to the main `Grid, LayoutRoot` and all its child controls. It wasn't necessary to make great changes to offer a more attractive rich user experience. Let's see the revised look in the following screenshot:



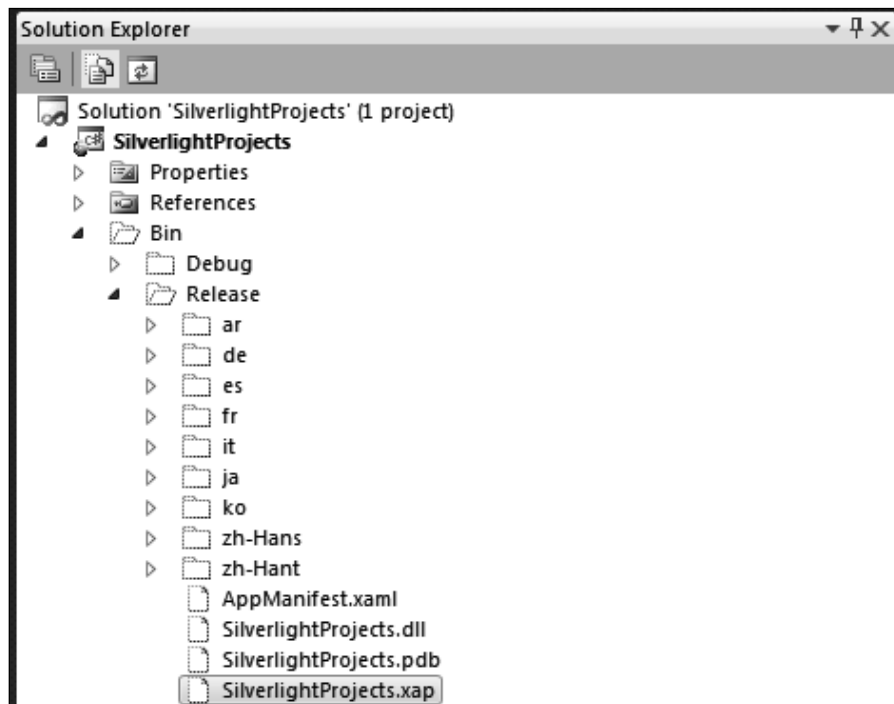
## Building a Silverlight 4 RIA

When we build a Silverlight project, Visual Studio or Expression Blend creates many folders with a lot of sub-folders and files.

In this case, we want to add the simple Silverlight RIA to a page in SharePoint 2010. Thus, we are interested in the `SilverlightProjects.xap` file. This is a compressed file, that is, a ZIP file with a `.xap` extension, and it contains all the necessary files for the Silverlight application.

You can find this file in the `Debug` or the `Release` sub-folder, according to your active solution configuration. For example, if the project is located in a `SilverlightProjects` folder, the relative path for the release `SilverlightProjects.xap` file would usually be `..\SilverlightProjects\Bin\Release`. If you want to be sure about the location of the generated `.xap` file, you can follow these steps after building the project:

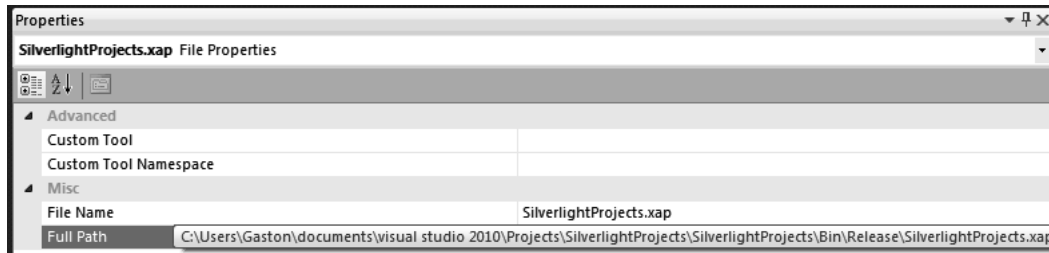
1. Activate the **Solution Explorer**.
2. Click on the **Show All Files** button, the second button located at the top of the **Solution Explorer** palette. Now, expand the `Bin` folder and then the `Debug` or `Release` sub-folder, according to your active solution configuration, as shown in the following screenshot:



**For More Information:**

[www.PacktPub.com/microsoft-silverlight-4-and-sharepoint-2010-integration/book](http://www.PacktPub.com/microsoft-silverlight-4-and-sharepoint-2010-integration/book)

3. Right-click on the `SilverlightProjects.xap` file and, and select **Properties** in the context menu that appears or press *F4*. The **Properties** palette will appear and you will be able to see the value for its **Full Path** property. This way, you can get the exact path for this file, as shown in the next screenshot. You will need it later to integrate it with SharePoint.



## Adding a Silverlight Web Part

We now want to add the Silverlight RIA to a SharePoint site. This Silverlight application doesn't interact with the SharePoint services. It displays information about projects saved in the same `.xap` file.

In this case, we can add a **Silverlight Web Part** to a SharePoint site.

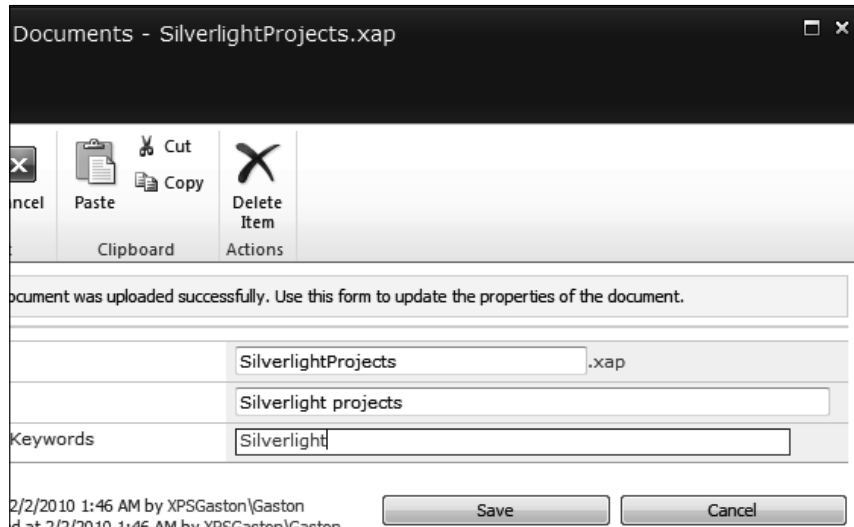
## Adding a Silverlight RIA as a shared document

First, follow these steps to add the Silverlight RIA as a shared document in a SharePoint site:

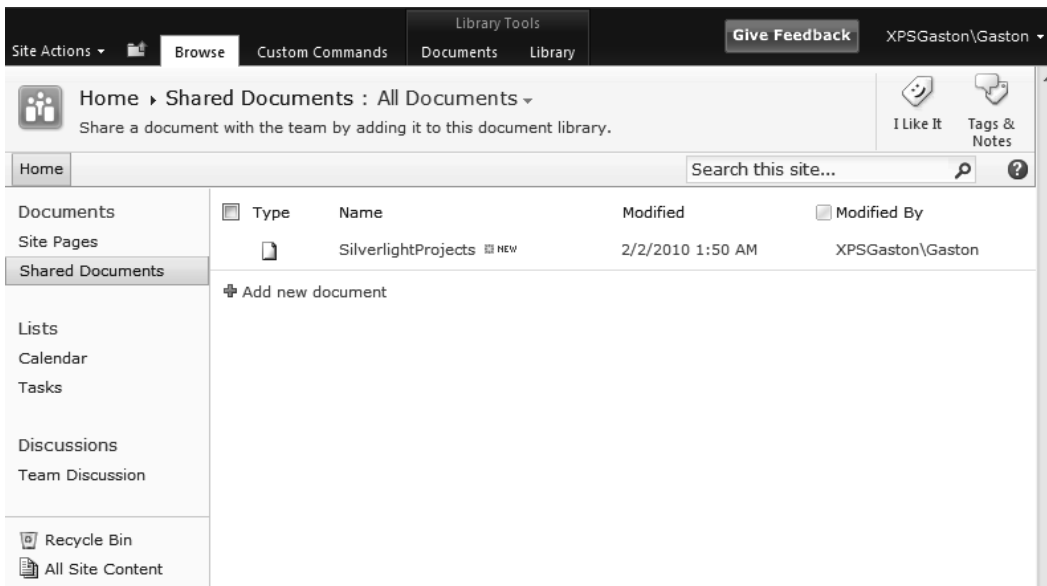
1. Open your default web browser, view the SharePoint site, and log in with your username and password.
2. Click **Site Actions | View All Site Content** and SharePoint will display all sites, lists, and libraries in the active site.
3. Click on **Shared Documents** under **Documents** on the panel located at the left of the web page. SharePoint will display the shared documents library. These are the documents shared with the team.



5. Click **OK**. Once the file finishes the upload process, the dialog box to define its properties as a shared document will appear. Enter **Silverlight projects** in **Title** and **Silverlight** in **Managed Keywords**.



6. Click on **Save** in the ribbon or on the button located at the bottom of the dialog box. The .xap file will appear in the **Shared Documents** list.

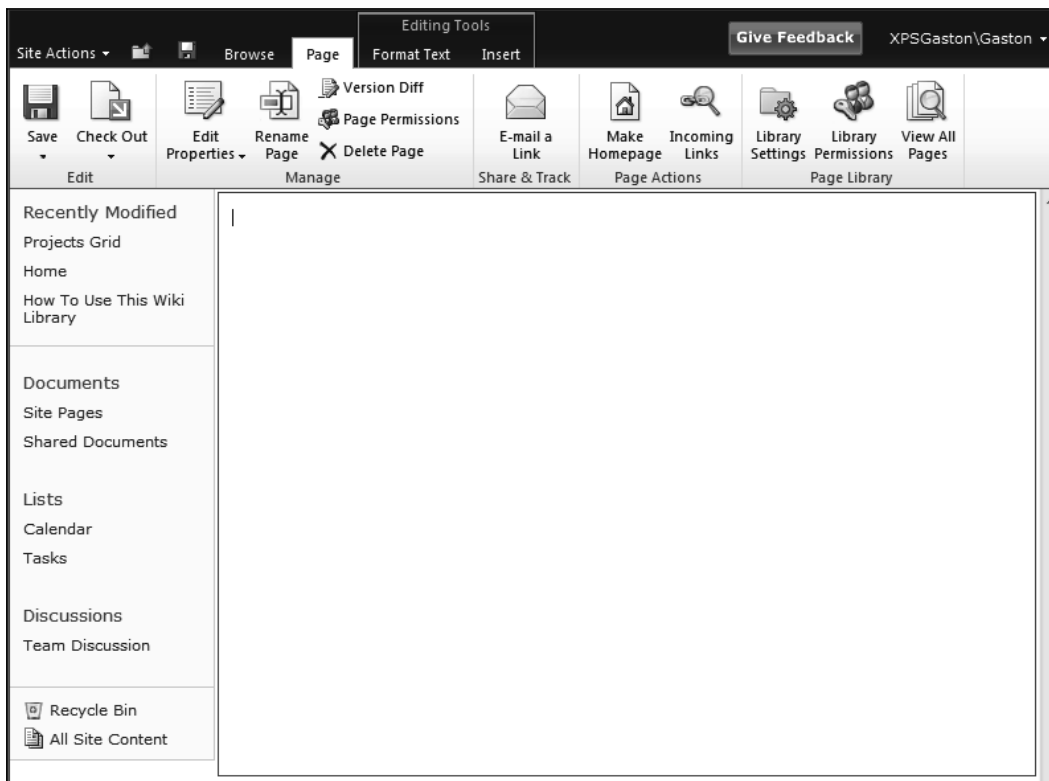


7. Right-click on the new document name, `SilverlightProjects`, and select **Copy Shortcut** from the context menu that appears. This way, you will copy the URL for this shared document in the clipboard and you will be able to paste it when SharePoint asks you for its URL. In this case, the copied URL is `http://xpsgaston/Shared%20Documents/SilverlightProjects.xap`.

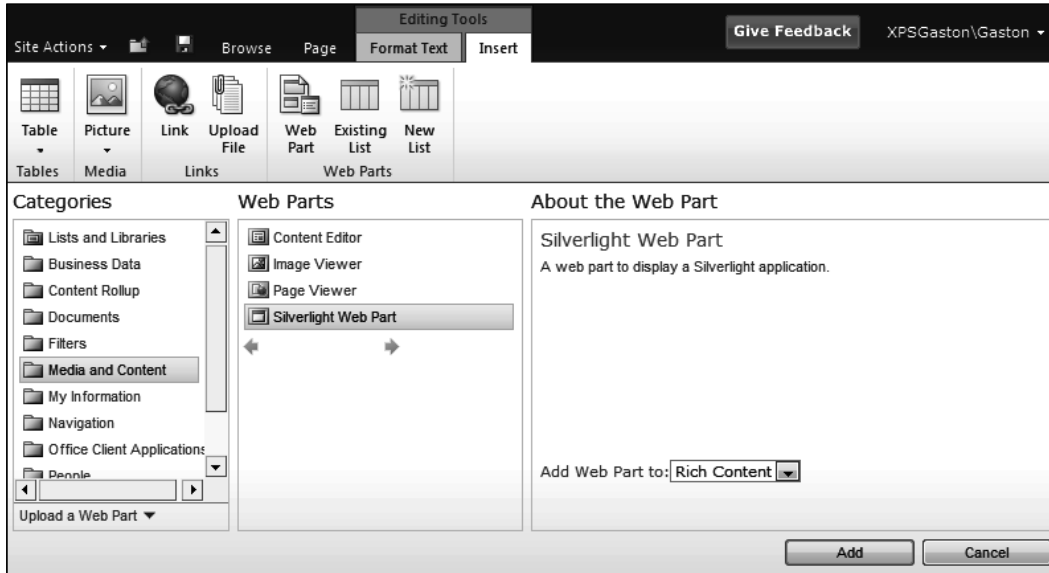
## Adding a Silverlight Web Part to display a Silverlight RIA

The `.xap` file is available as a shared document. Now, follow these steps to add a Silverlight Web Part to display the Silverlight RIA in a SharePoint site:

1. Click **Site Actions | New Page** and SharePoint will display a new dialog box requesting a name for the new page. Enter `Projects Grid` and click on **Create**. SharePoint will display the **Editing Tools** for the new page.



2. Click **Insert | Web Part** in the ribbon and a new panel will appear. Select **Media and Content** in **Categories** and then **Silverlight Web Part** in **Web Parts**, as shown in the following screenshot:



3. Click **Add**. The **Silverlight Web Part** dialog box will appear. Paste the previously copied URL for the shared document, the .xap file. In this case, it is `http://xpsgaston/Shared%20Documents/SilverlightProjects.xap`, as shown in the next screenshot:





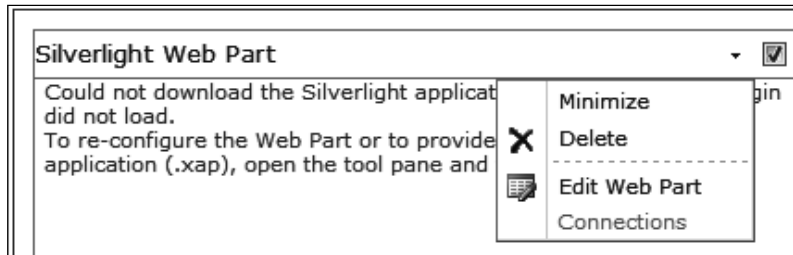
If you don't have the URL for the the .xap file copied to the clipboard, you can open a new tab or a new browser window and access the shared documents as previously explained. Then, you can right-click on the desired document name and select **Copy Shortcut** from the context menu that appears. This way, you will be able to paste the URL in the **Silverlight Web Part** dialog box.

4. Click **OK**.

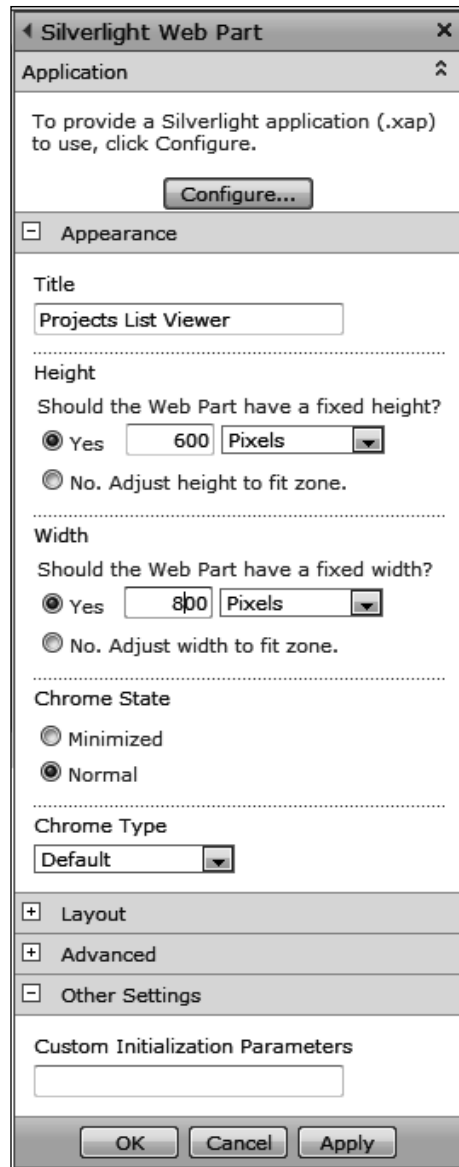


In certain SharePoint 2010 pre-release versions, the following error message could appear, **Could not download the Silverlight application or the Silverlight Plugin did not load. To re-configure the Web Part or to provide a different Silverlight application (.xap), open the tool pane and then click Configure.**

5. If the error appears, click on **open the tool pane** or click on the down arrow, located at the top and then select **Edit Web Part**.



6. The **Silverlight Web Part** pane will appear on the right. It will enable us to define many properties that affect the appearance and behavior for the Silverlight UI.
7. Enter `Projects List Viewer` in **Title**.
8. Click on **Yes** in **Should the Web Part have a fixed height?** and enter `600` in **Pixels**.
9. Click on **Yes** in **Should the Web Part have a fixed width?** and enter `800` in **Pixels**.



10. Click **OK**. The title for the Web Part will change but the same error message will appear. Don't worry about that.

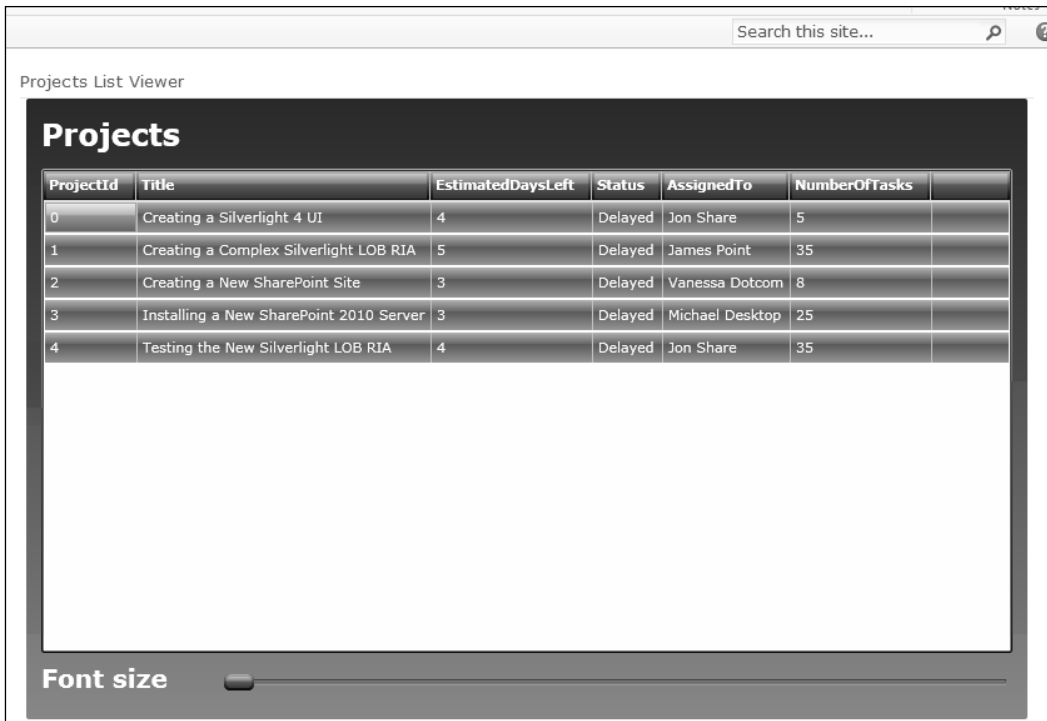
- Click on the **Save** button in the ribbon. Now, the new page will appear displaying the previously created Silverlight RIA. However, the title won't appear as expected, as shown in the next screenshot:

ProjectId	Title	EstimatedDaysLeft	Status	AssignedTo	NumberOfTasks
0	Creating a Silverlight 4 UI	4	Delayed	Jon Share	5
1	Creating a Complex Silverlight LOB RIA	5	Delayed	James Point	35
2	Creating a New SharePoint Site	3	Delayed	Vanessa Dotcom	8
3	Installing a New SharePoint 2010 Server	3	Delayed	Michael Desktop	25
4	Testing the New Silverlight LOB RIA	4	Delayed	Jon Share	35

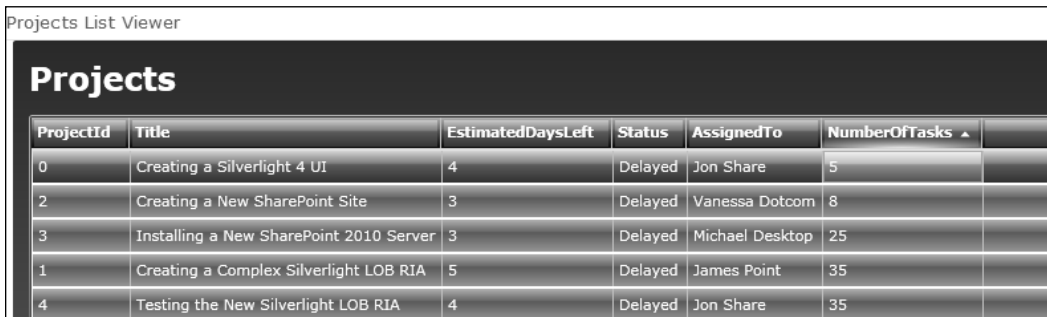
- Click **Site Actions | Edit Page**. This time, SharePoint will display the Silverlight RIA alive in the editing mode instead of the previously shown error message.
- Click on the down arrow, located at the top, and then select **Edit Web Part**. The **Silverlight Web Part** pane will appear on the right.
- Click on **No. Adjust width to fit zone.** in **Should the Web Part have a fixed width?**
- Click **OK** and then on the **Save** button in the ribbon. Now, the new page will appear displaying the previously created Silverlight RIA with the title appearing as expected.

**For More Information:**

[www.PacktPub.com/microsoft-silverlight-4-and-sharepoint-2010-integration/book](http://www.PacktPub.com/microsoft-silverlight-4-and-sharepoint-2010-integration/book)



16. The Silverlight Web Part added to the page holds a running and active Silverlight application. Click on one of the grid rows and you will see the gradient animations for the selected theme. Click on a column header and the rows will be sorted according to it.



**For More Information:**  
[www.PacktPub.com/microsoft-silverlight-4-and-sharepoint-2010-integration/book](http://www.PacktPub.com/microsoft-silverlight-4-and-sharepoint-2010-integration/book)

## Working with many Silverlight Web Parts in a single page

Following the previously explained steps, we can upload many Silverlight RIAs as shared documents and then add various Silverlight Web Parts to a single page. For example, we can rename the .xap file for the previous version of the `SilverlightProjects` application, without the application of the theme, to `SilverlightProjectsNoTheme.xap`. Then, we can follow the necessary steps to edit the page and add it as a new Silverlight Web Part to allow the users to choose between the different looks-and-feels to display the list of active projects.

Projects List Viewer (No Theme)

### Projects

ProjectId	Title ▲	EstimatedDaysLeft	Status	AssignedTo	Num
1	Creating a Complex Silverlight LOB RIA	5	Delayed	James Point	35
2	Creating a New SharePoint Site	3	Delayed	Vanessa Dotcom	8
0	Creating a Silverlight 4 UI	4	Delayed	Jon Share	5
3	Installing a New SharePoint 2010 Server	3	Delayed	Michael Desktop	25
4	Testing the New Silverlight LOB RIA	4	Delayed	Jon Share	35

Font size

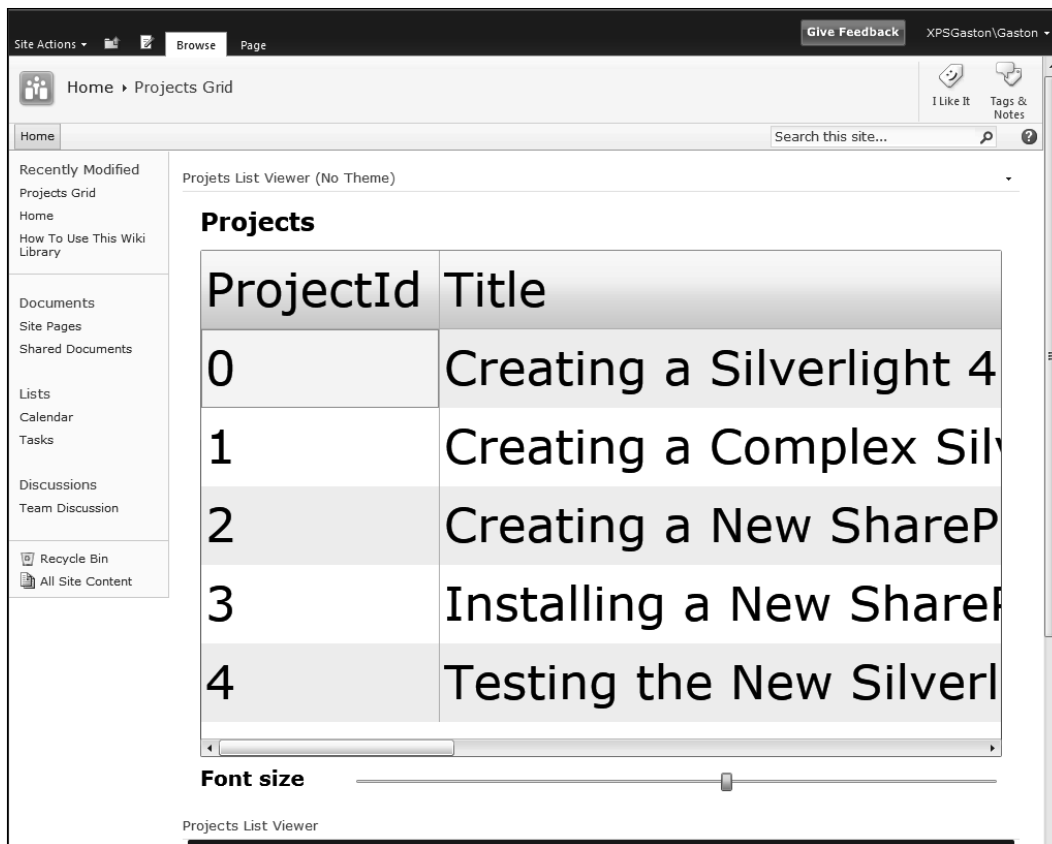
Projects List Viewer

### Projects

ProjectId	Title	EstimatedDaysLeft ▲	Status	AssignedTo	NumberOfTasks
2	Creating a New SharePoint Site	3	Delayed	Vanessa Dotcom	8
3	Installing a New SharePoint 2010 Server	3	Delayed	Michael Desktop	25
0	Creating a Silverlight 4 UI	4	Delayed	Jon Share	5
4	Testing the New Silverlight LOB RIA	4	Delayed	Jon Share	35
1	Creating a Complex Silverlight LOB RIA	5	Delayed	James Point	35

This way, you can make many Silverlight applications available to SharePoint users by following very simple steps. SharePoint 2010 added the new Silverlight Web Part to simplify adding Silverlight applications as part of SharePoint pages.

When the user moves the slider, the font size for the data grid will change, as previously experienced with the Silverlight RIA running in a simple web page. However, this time, the application is running as a Silverlight Web Part in a SharePoint site, as shown. Each Silverlight Web Part offers a very responsive application to the user because they don't have to wait for server responses to refresh the Web Part.



It is easy for a SharePoint user to consume a Silverlight Application. Now, all the users that have access to the new page will be able to interact with the Silverlight UIs added to this page.

**For More Information:**  
[www.PacktPub.com/microsoft-silverlight-4-and-sharepoint-2010-integration/book](http://www.PacktPub.com/microsoft-silverlight-4-and-sharepoint-2010-integration/book)

It is possible to add Silverlight applications to SharePoint through many different alternatives:

- **As a Silverlight Web Part**
- **Included in a SharePoint Module**
- **Included in a SharePoint Visual Web Part**

The previous steps explained how to work with the first alternative. In the next chapters we will analyze the other two alternatives.

## **Understanding client and server code**

As previously explained, when we integrate Silverlight with SharePoint, there is going to be code running on both the client and the server. The Silverlight applications appear on the SharePoint pages. However, once the .xap files are downloaded, their code runs on the client. Each client will require the Silverlight 4 client runtime installed in order to be able to run the application. If it isn't installed, a message will appear indicating that it requires Silverlight.

When you see an animation in a Silverlight RIA, it is consuming processing power found in the client and it is not adding load to the SharePoint server. When you add code to interact with the SharePoint server, the requests consume processing power from the server. However, when the Silverlight application processes the results from a request this code is running on the client.

All the code that presents graphics on the screen runs on the client. When you move the mouse cursor over an element, the animations and the events run on the client. Therefore, you don't add huge processing power from the server when adding Silverlight applications to a SharePoint solution.

## Summary

We learned a lot in this chapter about the integration of Silverlight 4 applications with SharePoint 2010 sites and solutions. Specifically, we prepared a development environment and the tools to work with Silverlight 4 RIAs. We configured the SharePoint 2010 server and added Silverlight Web Parts to a new page. We understood the differences between client and server code and the benefits of integrating Silverlight with SharePoint.

We created our first Silverlight RIA and then, we made it available in a SharePoint site. Now, we are ready to begin adding simple Silverlight RIAs as part of SharePoint solutions.

Now that we have learned about the principles of the integration of Silverlight with SharePoint, we are ready to learn the techniques to work with deploying and debugging for this integration process, which is the topic of the next chapter.

## Where to buy this book

You can buy Microsoft Silverlight 4 and SharePoint 2010 Integration from the Packt Publishing website: <https://www.packtpub.com/microsoft-silverlight-4-and-sharepoint-2010-integration/book>.

Free shipping to the US, UK, Europe and selected Asian countries. For more information, please read our [shipping policy](#).

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.



[www.PacktPub.com](http://www.PacktPub.com)

**For More Information:**

[www.PacktPub.com/microsoft-silverlight-4-and-sharepoint-2010-integration/book](http://www.PacktPub.com/microsoft-silverlight-4-and-sharepoint-2010-integration/book)