CHAPTER 13

Developing Windows Store Applications

In the previous chapters, you learned how to build a data-centric application based on a traditional Windows client (WPF) user interface and a web-based (ASP.NET) user interface. In this chapter, you will learn how to build a user interface using the new Windows Store app. Windows Store apps are designed to run on Windows 8 devices. Windows Store apps have a new look and feel designed to dynamically support different display sizes and devices. As tablets and phones increasingly become the most popular devices for interacting with information, business users are demanding the ability to use these devices for interacting with their information stores. Fortunately, Microsoft offers a variety of application programming interfaces (APIs) that can be used to create applications to run on these devices. They have exposed a managed .NET framework that allows you to use C# for the code behind and a XAML framework for developing the user interface. Developing these apps is a cross between developing WPF apps and web apps.

After reading this chapter, you will be comfortable performing the following tasks:

- using XAML markup to design a user interface
- working with layout controls
- working with display controls
- responding to control events
- using data-bound controls

Building the User Interface

Building the UI of a Windows Store app is very similar to developing WPF applications. If you recall from Chapter 11, the visual interface of an application contains objects. These objects, like most objects you work with in object-oriented languages, expose properties, methods, and events. In a Windows Store app, the main object is a page. A page has properties (for example the Background property), methods (for example the Focus method), and events (for example the DoubleTapped event).

Controls are components with visual interfaces that give users a way to interact with the program. A page is a container control, which hosts other controls. You can place many different types of controls on pages. Some common controls used on pages are TextBoxes, TextBlocks, Buttons, ListViews, and GridViews.

Just like WPF, Windows Store user interfaces are built using the declarative markup language XAML. For example, the following markup defines a button control inside a Grid.
Notice the Grid needs a formal closing tag because it contains the Button control. Since the Button control does not contain any other controls, you can use a forward slash (/) in front of the end bracket to close it. Notice the properties of the controls are set by using attribute syntax.

Controls are positioned either absolutely or relatively using grid rows and columns. Notice the code above uses absolute positioning of the button in the grid which is achieved using the Margin attribute. The code below shows the button positioned relatively by using grid rows and columns.

```xml
<Grid Background="#E5951D1D">
    <Grid.RowDefinitions>
        <RowDefinition Height="140"/>
        <RowDefinition Height="*"/>
    </Grid.RowDefinitions>
    <Grid.ColumnDefinitions>
        <ColumnDefinition Width="140"/>
        <ColumnDefinition Width="*"/>
    </Grid.ColumnDefinitions>
    <Button Content="Button" HorizontalAlignment="Left"
        Grid.Column="1" Grid.Row="1"
        VerticalAlignment="Top" Background="#FF2457A0"
        Height="78" Width="162"/>
</Grid>
```

Figure 13-1 shows the page with the button created by the previous XAML code.

**Figure 13-1.** A window created with XAML