

Although the origins of this design language can be found in earlier versions of the Windows Media Center and the Zune media player applications, it was with Windows Phone 7 that Microsoft fully committed to this new direction and made this language prominent. Since then, the dashboard of Xbox 360 has been adapted to this design, and Windows 8 relies heavily on it as well. In the future, you will probably see more and more applications adopting these concepts.

**NOTE** For Windows 8, you can create two kinds of applications: classic desktop applications and new Windows 8 style applications. You are not forced to use Windows 8. However, the Windows 8 concepts may be much more usable on devices with touch support. Thus, if you are targeting tablets or other devices where users can control the application with touch gestures, consider using Windows 8.

## General Design Principles for Windows 8 Applications

The first thing you probably notice about Windows 8 is that it is definitely different from what you are used to. Windows 8 breaks with the old conventions, and you must be open-minded if you want to accept it.

The first thing you have to take into account is that software and devices are part of a digital world. If you see a photo of a book on your phone or computer, it reminds you of reading, because your brain connects the digital photo to the physical book. But your brain doesn't need a perfect photo or a realistic image to have that connection, because you know that it is just a digital copy of an analog object on your device. With that, you can simplify the image — as you can see in Figure 5-1 — while keeping its original purpose.

Your software doesn't need to copy the analog world into the digital world. It doesn't have to look like something else — it actually is a digital application. Authenticity and simplicity can help users to understand programs faster, which is very important in the infographic design style.

The second thing you have to realize is why you create your applications. It is primarily for producing and consuming information. In this case, you must put the information into the center, and get rid of everything that doesn't help the production and the consumption of it (remember, simplicity). The focus should be on the content, and not the frame that holds it. If your users are working with the content, then all the windows, borders, lines, and background images — the so-called *chrome* — are not so important to them. You can use them if they help your original goal, but not simply to make your application look fancy.



**FIGURE 5-1:** A Windows 8 style icon with simplified graphic

**NOTE** Remember, most people use mobile devices these days, and all the extra chrome you use in your application consumes power and can degrade the precious battery life.

Just take a look at any of the built-in Windows 8 style applications. These are not running within a window, but instead are running in full screen to give space for the content. Windows 8 style applications even hide the classic Windows taskbar to free up some screen real estate (remember, mobile devices have small screens), and help the users to concentrate on the current application.

Today, users are flooded with data from various sources, so they really appreciate when the information is tailored to their needs. On a tablet device, the application may know who you are, where you are, in which position you hold the device, who is around you, and what information are you interested in right now. The application can customize itself to deliver content that is personal, relevant, and connected. If the device supports touch gestures, you can interact directly with the content, which is another level of user experience that rises above using the classic mouse and keyboard combo.

In the Windows 8 design language, content is king, and if you want to use it wisely, you need good typography. Typefaces, font sizes, colors, and whitespace are the keys to displaying the information in a easily readable format. Microsoft provides recommendations about the best practices, and even provides controls and typefaces you can use for free without extra licensing.

Animations are also important in the Windows 8 design language, because motion can further help the user to understand the software. The consistent use of animations makes your application more intuitive, even if the user doesn't consciously care about them. You can use fast and fluid animations to signal to the user that your application is responsive, and bring the interface to life.

But wait a minute! What happens with the unique appearance of your application if you use the same layout, fonts, animation, and style as any other application? Windows 8 provides a set of design principles that Microsoft believes can make applications better by answering the usability challenges. This set is not a rigorous list of expectations your application should meet in order to run on Windows 8. You can definitely brand your application while you are following the Windows 8 design principles. Just look at the built-in applications or the applications that are available from the Windows Store, and you will see that they are different and reflect the brand of the author. Be inspired by the design language, but balance between Windows 8 and your own style.

## Application Structure and Navigation Models

Windows 8 style applications are next-generation software in the sense that they are targeting not only desktop PCs and laptops, but also the latest handheld tablet devices. Tablet devices support touch gestures, and they often come without any other input device (including mouse and keyboard). The Windows 8 operating system is fully prepared for running on tablets (for example, it has a virtual keyboard that enables you to enter text on a touchscreen), and you should also design your application for gesture-based control. In fact, Microsoft recommends designing your Windows 8 style application for touch-first experience, meaning that you should design and optimize for gestures as the primary input method.

**NOTE** *Although touch gestures are the primary input method for Windows 8 style applications, you can use them fully with mouse and keyboard as well.*

Unfortunately, gestures present two problems:

- Users can't work with their fingers as precisely as they can with a mouse. This means you must enlarge the touch targets on your UI.
- Touch gestures are not easy to explore. This type of input works beautifully only if the gestures are used consistently across the system, because, in this way, users must learn them once, and can use the same gestures everywhere, even in your applications.

Fortunately, Microsoft developed a navigation system for Windows 8 style applications that solves these problems.

**NOTE** *You can read more about touch gestures in Chapter 14, "Advanced Programming Concepts."*

The basic unit of navigation in a Windows 8 style application is the *page*. As you may guess, the name comes from the web, because these application pages work very much like web pages. The user works with one page at a time, and navigation means going from one page to another. No windows or pop-ups appear that could break this consistency and complicate the UI.

You can organize your pages into a flat system or into a hierarchical system.

In the flat system, all pages reside at the same hierarchical level. This pattern is best for applications that contain only a small number of pages and the user must be able to switch between the tabs. Games, browsers, or document-creation applications often fall into this category. In these applications, the user can switch between the pages using the navigation bar (or *nav bar*) on the top of the screen, as shown in Figure 5-2.

You can represent your pages in any way on the navigation bar, but a typical technique is the use of thumbnails.

The hierarchical system is more familiar to people because it follows the same pattern in which web-site pages are organized. The hierarchical pattern is best for applications that have a large number of pages that can be grouped into multiple sections.

The root of the hierarchy is a *hub page*, which is the entry point of the application. On the hub page, users can get a glimpse of what is available in the sections of the application, so this first page should engage the users and draw them into the section pages. You can see an example of the hub page in Figure 5-3 from the Bing Finance application. This welcome page is an at-a-glance summary of the most important and relevant data in the application.

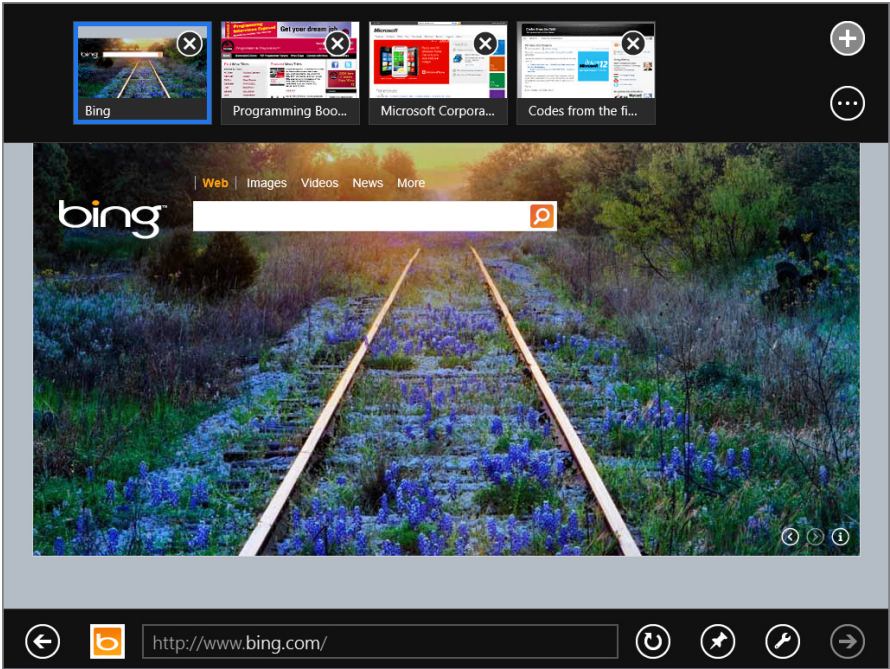


FIGURE 5-2: The Windows 8 style Internet Explorer with the tab navigation bar



FIGURE 5-3: The hub page of the Bing Finance application

From the hub page, users can navigate to the second level of the application, the *section pages*. Section pages often contain lists that can be grouped, sorted, and filtered. In Figure 5-4, you can see the section page of the Bing Finance application that lists the stocks available in the application.

From the list on the section page, users can select a single item and navigate to the third level of the hierarchy, the *detail page*. Detail pages may contain all information about the selected item, and the layout may vary depending on the type of the content. In Figure 5-5, you can see the detail page of a single stock in the Bing Finance application.

Users can go deeper in the hierarchy by activating (touching) an item on the screen, or can go back by using the Back button at the top of the page. Naturally, if your application is simpler, you can omit the hub page or the section page to make a more shallow hierarchy.

In Chapter 2, “Using Windows 8,” you learned about the charm bar and the app bar. The charm bar can slide in from the right, and the app bar can slide up from the bottom of the screen to display more options to the user. Both bars are extensible, and you can add your application-specific options to them.

The app bar should contain contextual commands, and can even pop-up menus when they have too many commands. The charm bar includes the Settings charm, which provides a single access point to all settings that are relevant in the user’s current context. This means you should add your application’s settings to the settings charm as well. In Figure 5-6, you can see how the built-in Weather application extends the Settings charm to include its options.



**FIGURE 5-4:** The Watchlist section page of the Bing Finance application



FIGURE 5-5: The detail page of a stock in the Bing Finance application

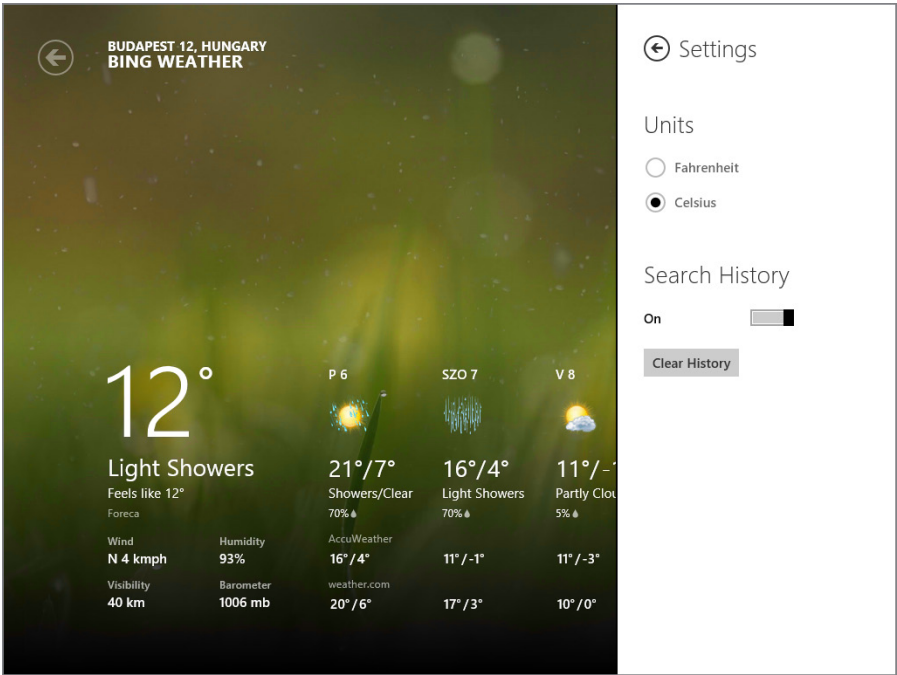


FIGURE 5-6: The Settings charm of the Weather application



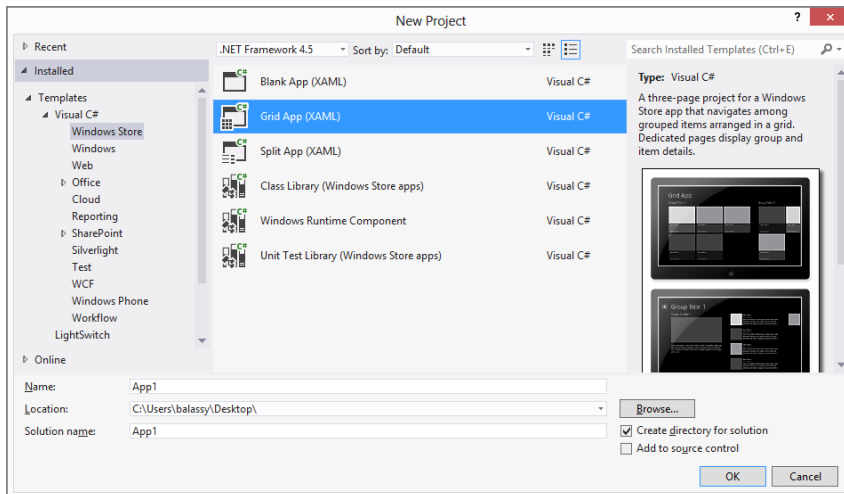
At first, you may think that the navigation concept with the app bar, charm bar, nav bar, hierarchy, and Back button is really complex. Perhaps that's true. However, if all these concepts are used consistently across the operating system and the applications, there's no doubt that the users will sooner or later master them, and, at the end of the day, hopefully they will even enjoy them.

Now let's take a look at how to create a basic Windows 8 style application.

## TRY IT OUT Creating a Basic Windows 8 Application

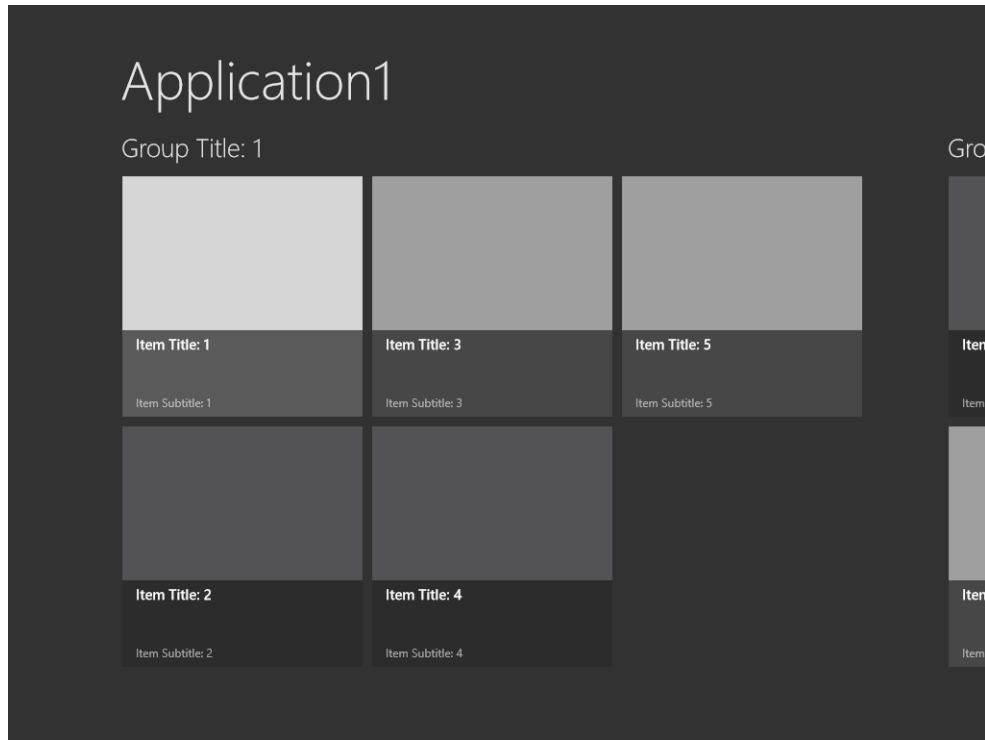
To create your first Windows 8 style application, follow these steps:

1. Start Visual Studio 2012 by clicking its icon on the Start screen.
2. Select File ⇨ New Project (or press Ctrl+Shift+N) to display the New Project dialog box.
3. Select the Installed ⇨ Templates ⇨ Visual C# ⇨ Windows Store category from the left tree, and then select the Grid Application project type in the middle pane, as shown in Figure 5-7.



**FIGURE 5-7:** The New Project dialog box

4. Click OK. Visual Studio creates an empty project skeleton and opens the main `App.xaml.cs` file in the code editor.
5. Start the application by clicking the Debug ⇨ Start Without Debugging menu item. Visual Studio compiles and starts the application, and you will have a Windows 8 style application running, as shown in Figure 5-8.
6. View the groups in the application by scrolling from left to right.
7. Click one of the group titles to open the group detail page.



**FIGURE 5-8:** The default Grid Application type

8. Review the content on the group detail page by scrolling from left to right, and then click one of the items to navigate to the item details page.
9. View the content of the item detail page, and then navigate back to the home screen by clicking the Back button twice, which is the big left arrow on the top of the screen.

### *How It Works*

According to the project template you selected, Visual Studio generated an empty skeleton that contains everything you need to start creating your Windows 8 style application in C#. You can open the Solution Explorer window by clicking the View ⇄ Solution Explorer menu item (or simply pressing Ctrl+Alt+L), where you can see all the files within your project, as shown in Figure 5-9.

The application contains three pages: the homepage, the group detail page, and the item detail page. In the Solution Explorer window (and also in your project folder), you can find the `GroupedItemsPage.xaml`, the `GroupDetailPage.xaml`, and the `ItemDetailPage.xaml` files that represent these pages.

The generated files also contain code to implement the navigation between the pages. You can locate these code lines by double-clicking the `GroupedItemsPage.xaml.cs` file. (You may need to open the hierarchy by clicking the arrow triangle on the left of the `GroupedItemsPage.xaml` file.) Scroll down to the lines that start with `this.Frame.Navigate`.



