From a .NET developer’s point of view, one of the most significant releases in 2012 is Visual Studio 2012. But from Microsoft’s point of view, the most significant release would have to be that of Windows 8. Applications specifically written for Windows 8 bring a big paradigm shift not only in how users experience applications, but also in how developers write them. This chapter talks about how you can get started creating these new styles of applications using Visual Studio 2012.

Windows 8 Apps

Windows has been the predominant operating system used in PCs for a long time. Microsoft originally thought that they could put this OS onto other devices such as PDAs, mobile phones, and other handheld devices and the same success would follow. In fact, I used to think that some of the older smart phones, which ran a modified version of the Windows operating system for the phone, were really cool—they even ran applications such as Excel on them!

I, like several other users, didn’t know any better until the iPhone was launched. The iPhone provided a user experience that incorporated touch, fluid animation, and sensors such as a gyro, accelerometer, proximity sensors, and GPS in such a way that it left its competitors far behind. When the iPhone, and later the Android smart phones, started dominating the market, Microsoft took notice. Microsoft went back to the drawing board and came up with a brand-new design and OS to run on its smart phones. This new design had the following design principles:

- **Typography**: The designers believed that type is beautiful and well-placed type can lead to more content. They believed that apart from being visually pleasing, type can also be functional.

- **Motion**: The designers believed that motion is a very key part of bringing an application to life—so transitions and animations are very important in a UI design.

- **Content, not chrome**: Running the phone on a small form factor meant that the emphasis had to be on the content—the information the user needs and cares about—more than the chrome, which includes all the UI elements such as toolbars and menus that surround it.

- **Honesty**: The phone delivers digital content, and the designers believed that the design should be authentically digital (in other words, not pretend to be something else or use metaphors in design to signify that it isn’t digital).
Although the Windows Phone 7, which uses the new design for its apps, is still a distant third behind Android and iPhones in terms of market share in the smartphone segment, Microsoft saw the potential of new-style apps. The success of iPad and Android tablet devices also opened up a fast growing market segment that Microsoft could just not ignore—touch-based devices.

To counter these threats, Microsoft has come up with a strategy to create and run new-style applications on the next version of its operating system—Windows 8. Windows 8 will run on both tablets and desktops, and it brings a shift in the way that applications are developed, how they are installed, and how users interact with these applications.

The principles behind the new design, which Microsoft now refers to as Modern UI (see sidebar), extend what was originally designed for the phone into Windows 8 and are listed as follows:

- **Show pride in craftsmanship**: Make sure UI designers and developers focus on the smallest of things in the application and are pixel perfect in every way.
- **Be fast and fluid**: Be extremely responsive to user interaction and use things such as touch and animation to create a compelling user experience.
- **Be authentically digital**: Embrace the fact that application runs in a digital medium and create applications accordingly.
- **Do more with less**: Focus more on the content, remove clutter, and ensure that the user does not have any distractions while working with the application.
- **Win as one**: Seamlessly combine with other applications as well as the devices present in the system to work as one.

These applications will be sold via the Windows Store, and they are also referred to as Windows Store apps.

**MODERN UI, WINDOWS 8 APPS, WINDOWS STORE APPS**

As you read through this book and other material that is available in the public domain (including documentation from Microsoft), you will come across applications written for Windows 8 being referred to as Windows 8 apps or Windows Store apps. You will also see the term Modern UI being used.

**Classic Apps**

*So, what happens to all the existing applications that have been written in .NET? Does all this new Windows 8/Modern UI stuff mean we have to throw them away? Not at all. That would make all the previous chapters in this book redundant, wouldn’t it?*

You can still write applications for .NET, and they will continue to run on Windows 8. There are some apps that will be really well suited for the Modern UI principles, some that may require quite a few changes, and some that may just not be suitable. For example, running Visual Studio as a Windows Store app may not work at all. But there are several line-of-business apps that may fit really well with the Modern UI’s design principles.

The non-Windows 8 apps—the ones that will continue to run in .NET or Silverlight—will coexist with the newer style of apps. Some people call them legacy apps, but I prefer the term classic apps.