Preface

The amount of built-in functionality included in the .NET Framework is amazing. One could create complex applications with lots of functionality by simply assembling the pieces you're offered for free. Among those pieces you'll find lots of powerful, flexible, and configurable controls. Who would ever need to build their own controls, right?

While some can get away without ever needing to build a custom control, many don't, or don't want to. Sometimes building a custom control is a necessity, because what you want to achieve doesn't exist, or is too expensive to buy. This is particularly true for controls with customized shapes, which need to be drawn instead of reusing other existing controls.

In some other cases, developers build custom controls as part of their architecture, allowing them to reuse more efficiently common functionality (and code) that they trust and like. There are many circumstances where building a custom control can make a programmer's life easier.

Unfortunately, programmers frequently avoided learning how to build custom controls because learning all the complexities consume lots of time and energy. Comprehensive and advanced books on the subject well exceed 1,000 pages, and the excessive amount of information can indeed be intimidating.

This book takes a lighter approach, guiding you step by step into building your first custom controls, and writing quality code. You'll build a new example in each chapter, and in the end you'll build a completely functional custom control where you'll apply most of what you've learned in the book.

What This Book Covers

Chapter 1: *Introduction to Custom Controls* will be your introduction to the world of .NET custom controls. You'll learn what controls are, why they are useful, what they are made of, and towards the end of the chapter you'll also create a simple yet functional custom control called TinyNoiseMaker.

Chapter 2: *Introduction to GDI+* introduces you to the basics of drawing with GDI+. You will meet a few namespaces, classes, and events that form the foundations of drawing with .NET, and you'll see how to paint the surface of a custom control.

Chapter 3: *Basic Drawing* teaches you more about the coordinate system of GDI+, drawing lines and polygons using pens, brushes, and colors, and guides you to build a control named GradientLabel.

Chapter 4: *Drawing Complex Shapes and Using Transformations* teaches you how to use graphics paths, regions, and transformations to build complex shapes. You'll then use the theory to build a Clock control.

Chapter 5: *Drawing Control Parts, Borders, and Adornments* explores using the `ControlPaint` class to implement common functionality and adds finishing touches to your control. To demonstrate the theory you'll build a simple custom control named GradientButton.

Chapter 6: *Working with Images* covers common techniques for manipulating images. You'll build a control called ImageWarper that scales, skews, and rotates an image.

Chapter 7: *Printing* introduces this very important area of GDI+. In many circumstances you'll want to add printing support to your controls, and you'll see exactly how to do so by creating the PrintableRichTextBox control.

Chapter 8: *Collections* teaches more details about .NET collections, including the new .NET 2.0 generics. Collections are very useful when building custom controls, and as an example you'll build a Font Picker control.

Chapter 9: *Double Buffering* introduces this advanced technique that can make a big difference in improving the speed and responsiveness of your control. Scrolling is one such area where double buffering could make a difference, and you'll end the chapter by implementing a control that displays a scrolling text.

Chapter 10: *Handling Mouse Events* deals with a very important topic for any desktop application you'll ever write. Luckily enough, all controls that ship with .NET have integrated mouse support, but at times you'll need to customize the features. In this chapter, you'll implement two applications: one will allow you to drag an image inside a delimited area, and the second lets you drag pictures from your Windows system into your form.

Chapter 11: *Implementing Design-Time Support* shows you how to make your user controls designer friendly. This way, your control will be friendly not only to the end users working with it but also to developers as well.

Chapter 12: *Designing Intuitive Interfaces* is a high-level overview of some human interface aspects that, handled correctly, can help you build controls and applications that are easier and more fun to use.

Chapter 13: *The PieChart Control* is a comprehensive case study, showing you how to develop a complete custom control in several stages, each time adding features, fixing bugs, and improving functionality.

Appendix A: *Distributing Custom Controls* shows you how to compile a custom control into a separate DLL file, which can be reused later in other projects. You'll be shown how to do this with both Visual Studio 2005 and Visual C# 2005 Express Edition, which offers different built-in features.

Conventions

In this book, you will find a number of styles of text that distinguish between different kinds of information. Here are some examples of these styles, and an explanation of their meaning.

There are three styles for code. Code words in text are shown as follows: "We can include other contexts through the use of the `include` directive."

A block of code will be set as follows:

```
private void SetValues()
{
  totalCount = 0;
  if (mySlices != null)
  {
    foreach (Slice slice in mySlices)
    totalCount += slice.GetSliceRange();
  }
  // mySlicesPercent.Clear();
}
```

When we wish to draw your attention to a particular part of a code block, the relevant lines or items will be made bold:

```
private void SetValues()
{
  totalCount = 0;
  if (mySlices != null)
  {
    foreach (Slice slice in mySlices)
    totalCount += slice.GetSliceRange();
  }
  // mySlicesPercent.Clear();
}
```

New terms and **important words** are introduced in a bold-type font. Words that you see on the screen, in menus or dialog boxes for example, appear in our text like this: "clicking the Next button moves you to the next screen".

Warnings or important notes appear in a box like this.

Tips and tricks appear like this.

Reader Feedback

Feedback from our readers is always welcome. Let us know what you think about this book, what you liked or may have disliked. Reader feedback is important for us to develop titles that you really get the most out of.

To send us general feedback, simply drop an email to `feedback@packtpub.com`, making sure to mention the book title in the subject of your message.

If there is a book that you need and would like to see us publish, please send us a note in the SUGGEST A TITLE form on `www.packtpub.com` or email `suggest@packtpub.com`.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide on `www.packtpub.com/authors`.

Customer Support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

Downloading the Example Code for the Book

Visit `http://www.packtpub.com/support`, and select this book from the list of titles to download any example code or extra resources for this book. The files available for download will then be displayed.

> The downloadable files contain instructions on how to use them.

Errata

Although we have taken every care to ensure the accuracy of our contents, mistakes do happen. If you find a mistake in one of our books—maybe a mistake in text or code—we would be grateful if you would report this to us. By doing this you can save other readers from frustration, and help to improve subsequent versions of this book. If you find any errata, report them by visiting `http://www.packtpub.com/support`, selecting your book, clicking on the Submit Errata link, and entering the details of your errata. Once your errata have been verified, your submission will be accepted and the errata added to the list of existing errata. The existing errata can be viewed by selecting your title from `http://www.packtpub.com/support`.

Questions

You can contact us at `questions@packtpub.com` if you are having a problem with some aspect of the book, and we will do our best to address it