The .NET Framework is a powerful platform that includes a managed runtime, compilers for several languages, and a comprehensive class library upon which you can build applications. The .NET Compact Framework delivers a powerful subset of the .NET framework in only 2.5 megabytes of disk space. If you want to develop applications for the Pocket PC or Windows Mobile-based Smartphones, the .NET Compact Framework is the ideal choice. This book introduces the .NET Compact Framework and includes several projects to get you up to speed quickly. In this book, you'll learn:

* The basics of programming for the .NET Compact Framework using Visual Studio .NET 2003; you'll be up and running with Hello, World in no time at all.
* How to consume web services from .NET Compact Framework applications.
* How to cache on-line data for use when the mobile device is out of range of a network.
* How to write applications that use Bluetooth.

Whether you're new to mobile programming or new to Visual Studio .NET 2003, the .NET Compact Framework Pocket Guide will teach you how to write applications that fit in your pocket.
Contents

Part I. The .NET Framework and Mobile Devices

The .NET CLR 3
Windows Powered Mobile Devices 4
Tool and Language Support 11
Hello, Pocket PC 13
Debugging the Application 21
Troubleshooting 28

Part II. User Interface Design

Using the Windows Forms Controls 30
Design Considerations for Smartphone Applications 34
<table>
<thead>
<tr>
<th>Part III. Projects</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Project A: Currency Converter</td>
<td>42</td>
</tr>
<tr>
<td>Project B: Book Ordering System</td>
<td>53</td>
</tr>
<tr>
<td>Project C: Bluetooth Chat</td>
<td>68</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Part IV. Deploying .NET Compact Framework Applications</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Packaging CAB Files</td>
<td>86</td>
</tr>
<tr>
<td>Redistributing the .NET Compact Framework</td>
<td>92</td>
</tr>
<tr>
<td>Deploying Smartphone Applications</td>
<td>93</td>
</tr>
</tbody>
</table>
Introduction

The .NET Framework includes a runtime called the Common Language Runtime (CLR), compilers for languages such as C# and Visual Basic .NET, and the Framework Class Library (a large, factored class library that exposes many capabilities).

Browse the entries in the Class Library reference, and you’ll find support for web services, networking, input and output, XML wrangling, and more. With all the capabilities in the .NET Framework, it’s no surprise that the runtime weighs in as a 20 MB download (when it’s extracted and installed, it’s much larger than that). What’s surprising, though, is how many capabilities are retained in the Microsoft® .NET Compact Framework, a scaled-down version of the .NET Framework for memory-constrained devices. In roughly 2.5 MB of file storage, and as little as 1.25 MB of runtime memory, you’ll find support for core and extended .NET APIs, including collections, networking, ADO.NET data access, XML, and web services.

Conventions Used in This Book

The following typographical conventions are used in this book:

*Italic*  
Used to indicate new terms, URLs, filenames, file extensions, directories, commands and options, and program names. For example, a path in the filesystem will appear as C:\windows\system.
Acknowledgments

I would like to thank Jonathan Wells, Product Manager of the .NET Compact Framework at Microsoft, for his support and review of this book. My editor Brian Jepson deserves a special mention for his attention to details and for making this book a better read. Thanks to Bill Takacs from O’Reilly, who made this book possible.

I would also like to thank Lori Piquet from DevX (a division of Jupitermedia Corp) who has kindly agreed to let me use some of my articles published at DevX.com for this Pocket Guide.
The .NET CLR

Runtimes like the CLR are called managed execution environments because they exert much more control over components than the operating system does. Although the operating system can usually guarantee that a rogue application won’t bring the whole system down, modern applications are increasingly made up of cooperating components. The failure of a single component may not crash a modern operating system, but it can render an application unusable until the component is fixed. A managed environment like the CLR makes it easier to build reliable components by controlling component versioning, runtime execution, memory management, security, exception handling, and more.

Obtaining the .NET Compact Framework

The Microsoft® .NET Compact Framework is included with Visual Studio .NET 2003. You can deploy the .NET Compact Framework runtime directly to a supported device from within the Visual Studio .NET environment. The .NET Compact Framework is also built into any device based on Windows Mobile 2003.
Windows Powered Mobile Devices

There are several kinds of Mobile Devices that run a version of the Windows Mobile operating system, which is optimized for memory-constrained devices. The current incarnations of this operating system are descendants of the original Windows CE, and they include:

Pocket PC 2000/2002
This is a version of Windows that is designed for small devices with small screens (commonly 240 × 320 pixels) and no keyboard. Instead of a keyboard, there is a stylus and touch-sensitive screen, five-way navigation button, and a few buttons.

The previous version was Pocket PC 2000; many devices running the Pocket PC operating system can be upgraded to Pocket PC 2002. The .NET Compact Framework supports both Pocket PC 2000 and Pocket PC 2002 devices. Earlier versions of Windows CE, such as 2.11, are not supported by the .NET Compact Framework.

Pocket PC 2002 Phone Edition
This version of the Pocket PC 2002 operating system is for Pocket PCs with integrated phones. They still have the same screen, but they include support for voice telephony, SMS messaging, and General Packet Radio Service (GPRS) networking. Figure 1-1 shows the Pocket PC phone from T-Mobile. At the time of this writing, some carriers were rolling out upgrades that update Pocket PC 2002 phones to the Pocket PC 2003 Phone Edition (see “Windows CE .NET 4.2,” later in this section).

Microsoft Smartphone
Microsoft Smartphones look less like a palm-sized computer and more like a phone. The navigation button is still there, along with buttons, but the Smartphone also has a telephone numeric keypad. The screen is smaller (for example, 176 × 220 pixels in the Motorola MPx200 Smartphone).
Smartphones do not come with a stylus or touch-sensitive screen. Most devices in the market use the Smartphone 2002 operating system, although, at the time of writing, Smartphone 2003 was becoming available. The newer Smartphone 2003 operating systems come with the .NET Compact Framework (in the ROM), which allows applications to be developed using the .NET Compact Framework and the Smartphone 2003 SDK. Figure 1-2 shows the EverLink Wireless Smartphone, running the Smartphone 2003 operating system.

**TIP**

You must use Embedded Visual C++ to develop for the older Smartphone 2002 devices. Only the newer Smartphone 2003 devices support the .NET Compact Framework.
tool and language support, 11–13
ToolBar control, 33
Toolbox, 17
touch-sensitive screens, 34
TrackBar control, 33
TreeView control, 32

U
Update button (currency converter project), 52
updateMessageLog() method (Bluetooth chat project), 76
UpdateValue() event (currency converter project), 50
user input, 36
user interface design, 30–41

V
Video option (emulator.exe), 26
Visual Studio .NET 2003 modes for testing, 11
Visual Studio Integrated Development Environment (IDE), 11
VScrollBar control, 32

W
WayPort hotspots, 9
web services (book ordering system project), 54–56
web sites, deploying applications via, 85
Wi-Fi, 9
hotspots, 9
Windows Application project type, 13
Windows CE .NET 4.2, 6
Windows CE .NET Web pad, 27
Windows CE applications, 13
Windows Mobile 2003, 6
developing applications for, 15
Windows Mobile operating system, 4
Windows powered mobile devices (see mobile devices)
wireless connectivity limitations, 8
wireless network access long-range, 9
short-range, 8
WriteFile() function (coredll.dll), 72