

Preface



So, you bought a Pocket PC. It's a cool little device, with Bluetooth, WiFi, a bright screen, and a reasonably functional interface. Sure, it needs a reset once in a while, and it really only wants to synchronize with Microsoft applications, but who cares? You bought it because you wanted to write your own apps on it, and it looked like the Microsoft .NET Compact Framework might be a good thing - might even feel roomy, after the J2ME or cell phone programming that are the main alternatives. Heck, Microsoft even offers a version of SQL Server for the device that can sync to larger databases. Sounds good, right?

And then you notice a few problems:

- To develop on the .NET Compact Framework, you have to work in Visual Studio .NET, an IDE with a mind of its own and a price tag to match.
- That SQL Server CE bit costs a chunk of change itself, and isn't the speediest thing around.
- Web Services are a lot clunkier than advertised.
- All the books on the .NET Compact Framework seem to assume that you're already a .NET developer intimately familiar with the regular .NET Framework.
- When the same books wander beyond the .NET Compact Framework, they figure you're familiar with Win32 APIs.

None of this likely bothers you if you happen to be a .NET developer already. If you are, other people have already written books for you. In fact, you're a privileged person, since at times it feels like Microsoft sees .NET Compact Framework as the pinnacle of .NET development, something to be attempted only by people who already know what they're doing elsewhere and just need to downsize their applications.

On the other hand, if developing apps for handheld devices seems to you like something that ought to be simpler than developing desktop or server apps, and especially if you're

interested in the world beyond Microsoft's offerings, read on. I can't promise that I'll solve all of your software development problems, or even your .NET problems, but we'll start slow, work through some projects using the cheapest tools available, and hopefully get you to the point where you're comfortable writing .NET Compact Framework applications. You'll still be stuck in Visual Studio .NET (at least until Microsoft releases a new version without that requirement), but everything else will be free or cheap, and hopefully after a while Visual Studio won't matter that much either.

Who should read this book

If you want to write programs for the Pocket PC or related Windows CE devices, you'll learn something from this book. You should already know programming, preferably object-oriented programming, to some extent. You may want to have some background in Visual Basic .NET before you start, though hopefully even readers with minimal programming experience will be able to follow it easily enough at the pace we'll be taking. A bit of background in databases, SQL, and possibly XML will also be helpful.

I was originally going to do this in C#, but VB.NET feels more accessible to a larger number of readers. If you want to see this done in C# eventually, send me an email (simonstl@simonstl.com) and let me know.

If you already know your way around .NET, good for you. You will be bored a few times here, but you may still see things you haven't happened upon before. Don't be afraid to skip around.

Finally, if you prefer that your computer books just tell you how the manufacturer wants you to do something, without criticizing their tools and without looking outside their environment, you should probably go find a different book. (You may have already guessed that.)

What we'll do

Rather than dive into "here's the .NET Compact Framework and all the ways it's smaller and weirder than the regular .NET Framework," we'll start with some really simple and not-so-weird stuff, and build from there:

- First, we'll build some really basic GUI applications using the tools in Visual Studio .NET, and run them on the emulators and a Pocket PC. At the simplest level, you can draw a form, add a bit of code, and have something nice to look at on the Pocket PC. This will give you a chance to get familiar with Visual Studio, too.
- Debugging applications is something easier done sooner than later, so once we have some basic applications going, we'll create some problems and see what they look like in the Visual Studio .NET debugger, using both an emulator and a real device.
- After we get the simple applications going, we'll create some more complex applications with multiple forms, more parts, and some forms. We'll also save some data to a file on the device, so the application saves state.

- Once we have a basic application running, we'll take a look at ways to make the GUI more interesting, extending some of the classes in the .NET Compact Framework GUI, and creating things like reusable custom panels and spaces for drawing pictures.
- Now that the GUI looks nice, we'll go back to data structures, using the SQLite database to store information. It's free (heck, public domain!), it works, and it's reasonably fast.
- When we put a fair amount of information into the device, we have a problem. How do we get it out? SQL Server CE had a built-in solution for that, so long as you had a copy of SQL Server handy, but we'll concoct our own way to get data off the Pocket PC that will work with a lot more environments. Fortunately, the .NET Compact Framework includes great support for HTTP, the protocol that underlies the Web, as well as some other network protocols, so you can get your data to whatever Microsoft or non-Microsoft environment you like.
- If you're just using your Pocket PC to collect information more portably than a laptop, you might be done at this point. If you want to use the Pocket PC in connection with another device, you might want to know about how to access serial port data, even though the .NET Compact Framework doesn't support it.
- Finally, we'll take a look at a variety of features in the OpenNETCF.org projects that can help you in future development.

It's not the usual .NET Compact Framework tour, but hopefully it will be more interesting, and more accessible. It shouldn't be that hard, and it won't be that hard.

Other books

As you get deeper into the .NET Compact Framework, you'll undoubtedly have questions that don't get answered here. This isn't a complete reference by any means. Despite the "Compact" in the name, a complete reference for all of the parts in this framework would be very large, and it's mostly out there at MSDN. I recommend these books which have helped me along the way, though no doubt there are others as well:

- *.NET Compact Framework Pocket Guide*, by Wei-Meng Lee. Published by O'Reilly Media in 2004, this tiny 105-page book convinced me that the parts I needed to build applications on this device were available. It's a great quick tour.
- *Microsoft .NET Compact Framework Core Reference*, by Andy Wigley and Stephen Wheelwright. Published by Microsoft Press in 2003, this 860-page hardcover reference explores the parts of the Framework in detail, but doesn't spend a lot of time assembling them.
- *.NET Compact Framework Programming with C#* and *.NET Compact Framework Programming with VB.NET*, by Paul Yao and David Durant. Published by Addison-Wesley in 2004. These are some big heavy books, weighing in at 1350+ pages. There's an awful lot of information in here, but the authors assume that you're thoroughly familiar with both .NET and the Win32 API.
- *Mastering Visual Studio .NET*, by Ian Griffiths, Jon Flanders, and Chris Sells. Published by O'Reilly Media in 2003. Although most of this book covers Visual Studio .NET features that don't apply to .NET Compact Framework development, and notes that "Detailed discussion of palmtop development is beyond the scope of

this book," there's still a lot of useful information here, especially in the first 100 pages.

- *Learning Visual Basic .NET*, by Jesse Liberty. Published by O'Reilly Media in 2003. Again, it's not specifically about the .NET Compact Framework, but if you need to learn Visual Basic .NET - "what's a Protected Friend?" - this is a good place to start.
- *VB.NET Language Pocket Reference*, by Steven Roman, Ron Petruscha, and Paul Lomax. Published by O'Reilly Media in 2003. While it's not specifically about the .NET Compact Framework, it also has the major advantage for .NET Compact Framework developers that "It does not attempt to document the core classes of the .NET Framework Class Library," so it's not full of possibilities that don't apply in the Compact environment.

License and Republishing

For now, this book is licensed under the Creative Commons Attribution-NonCommercial-NoDerivs 2.0 license. More details about this license are available at <http://creativecommons.org/licenses/by-nc-nd/2.0/>. In general, it means that you can use this material to learn about .NET Compact Framework programming without paying me, but you have to include my name on any redistribution, you can't make money off the redistribution, and you can't create derived works from this. (I don't mind you using the code, of course - this restriction applies to the book.) I expect to keep this version of the book available under those terms, and may liberalize them at some point if I see good reason to do so.

I have some mild hopes that eventually some version of this book may arrive in print on bookstore shelves, but I don't expect that to happen for this release of Visual Studio .NET and the .NET Compact Framework, as the next version is already on its way.