### Getting Started with C#

You can’t do any programming without having the right tools for the job!

This chapter looks at what you need to get started with C#. We will cover both ends of the spec- trum, from simple, no-cost tools to a cheap tool that will make programming in C# easier, all the way to the top-of-the-range tools that will set you back a small fortune!

##### Getting Into C# is Cheaper Than You Think!

When most people think of C#, they instantly think “Microsoft.” Then they start to think about how much it’s going to cost them to make use of the language — after all, Microsoft is in the busi- ness of selling software, and that software can cost a lot.

The truth is that you can start to use C# for absolutely nothing. Many people find this hard to believe at first, but it’s absolutely true. You can create C#-based applications for nothing. If you go to the other end of the cost spectrum, however, you can also spend a lot of money, buy expensive development environments, and use those to develop C# applications.

What end of the cost spectrum you choose to work with is entirely up to you and is based on your needs.

The Cheap End of the Spectrum

At the cheap end are the no-cost C# development tools. And don’t be fooled — these are Microsoft tools.

The bare minimum that you need to get started with C# programming are:

* + A text editor (like Windows Notepad)
  + The Microsoft .NET Framework

The Text Editor

You’ve probably already guessed why you need the text editor — it allows you to type the C# code that will be compiled.

Windows Notepad, as shown in Figure 2-1, is a good place for many to start for a number of reasons:

* It’s free.
* It’s familiar.
* It’s darn simple to use!

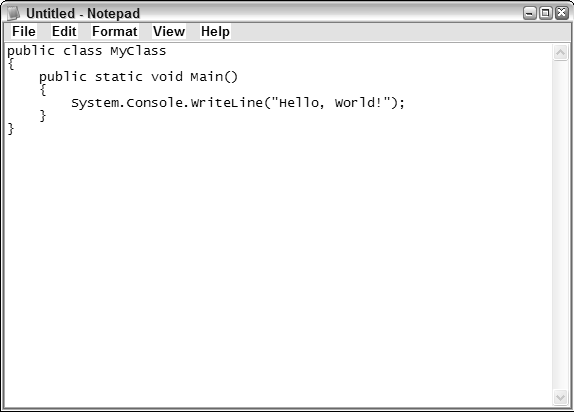


Figure 2-1

There are a number of quirks to Windows Notepad. The main one is that it always wants to save files with the .txt file extension as opposed to the .cs file extension preferred for C# source code (see Figure 2-2).

The other problem with Notepad is that it offers only very basic features. It’s a plain-text editor and nothing more. There are no features designed specifically for the programmer at all (or anyone else for that matter).

That said, if you are looking for a cheap way to get into C#, Windows Notepad is an automatic solution — if you are a Windows user (and we’re going to assume that you are), Notepad is already installed on your PC, ready for you to begin coding with.

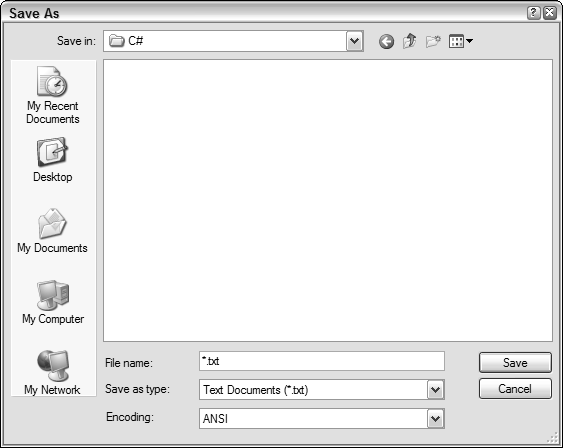


Figure 2-2

The Microsoft .NET Framework

For basic C#, the only thing in addition to a text editor you need to have installed on your PC is the Microsoft .NET Framework. Given that this has been around for some time now, it’s more than likely that you have it installed. The easiest way to check is to look for the files it uses. Using Windows Explorer, go to C:\WINDOWS\Microsoft.NET\Framework and see if you have folders there. (In

Figure 2-3, you see three folders: v1.0.3705, v1.1.4322, and v2.0.50727. The names of these folders correspond to the version numbers of the .NET Framework you have installed.)

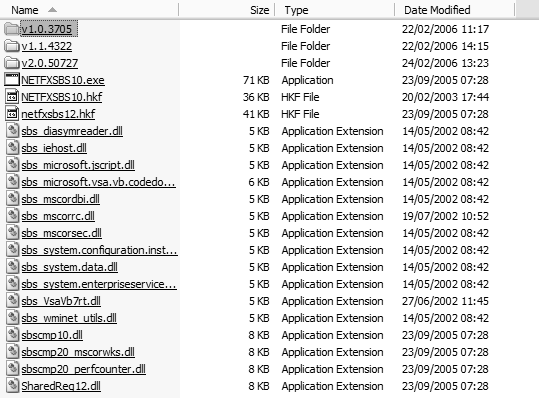


Figure 2-3 9

For the purposes of this book, we are going to assume that you have the latest version of the .NET Framework installed (which at the time of writing is v2.0.50727). If you don’t have this installed (or want to reinstall the latest version just to be on the safe side), you can download it from the Microsoft website at [http://msdn.microsoft.com/netframework/.](http://msdn.microsoft.com/netframework/)

That’s it! That’s the basic kit that you need to leverage C#.

How to Leverage Free C# Tools

OK, you have your Windows Notepad at the ready, and you’ve got the latest and greatest version of

.NET Framework installed. How do you start making use of these and get some results with C#?

Writing Code

Well, it’s pretty obvious that you type the C# code into Notepad (some simple code is shown in Figure 2-4).

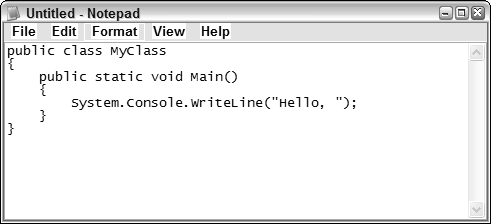


Figure 2-4

The process for using these free tools goes like this:

* 1. Type the code into Notepad (see Figure 2-5).

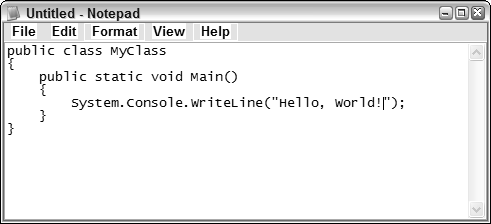


Figure 2-5

* 1. Save the file, remembering to give it the .cs file extension (see Figure 2-6). We also recommend that you save it in the .NET Framework folder for the latest version of the Framework, in our case v2.0.50727 (at least until you get comfortable using the command-line compiler, which comes next).

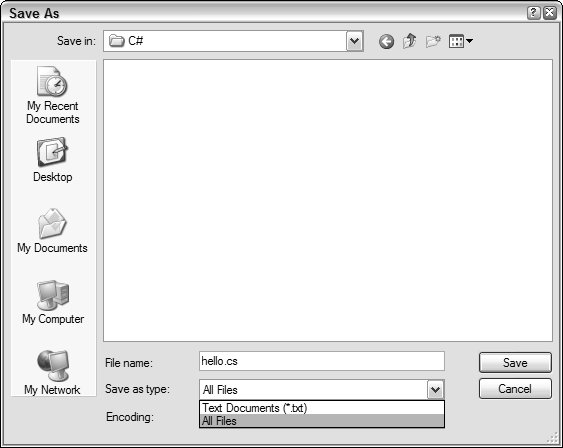


Figure 2-6

* 1. Open the Command Prompt (Start➪Run and type cmd and click OK) and navigate to the folder where you saved the file (see Figure 2-7). Alternatively, you can use the Windows XP Open Command Window Here PowerToy and right-click the folder in Windows Explorer and chose Open Command Window Here. You can download this PowerToy from the Microsoft website at [http://www.microsoft.com/windowsxp/downloads/powertoys/xppowertoys.mspx.](http://www.microsoft.com/windowsxp/downloads/powertoys/xppowertoys.mspx)

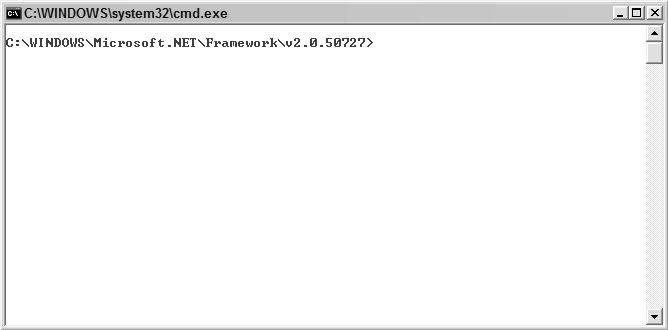


Figure 2-7

11

* 1. Now you’re ready to compile the source code. To do this, you will use the C# command line compiler that ships with the .NET Framework. The compiler is named csc.exe and is in the root folder for the .NET Framework: v2.0.50727. The syntax for compiling the code is simple:

csc.exe source.cs

In our example, the source code is called hello.cs. This means that to compile the code you use the following at the command line (see Figure 2-8):

csc.exe hello.cs

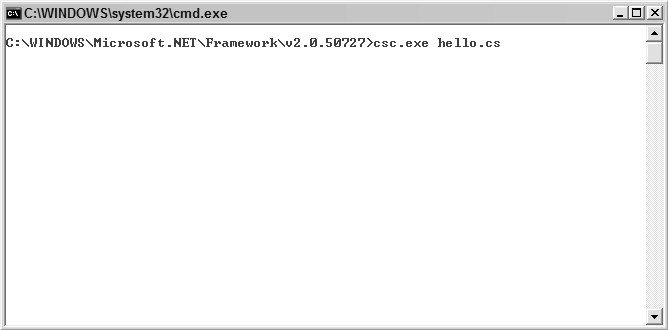


Figure 2-8

* 1. The source code should now be compiled into an executable. The name of the executable is the same as for the source code, except that the .cs is replaced with .exe. In this example, the exe- cutable is called hello.exe.

To run the executable, type the following at the command line:

hello.exe

The executable file will be executed and the message displayed onscreen (as shown in Figure 2-9).

That’s it! It really is that simple to compile a C# application developed using Notepad with the com- mand-line compiler. It’s very quick and very simple, and about the only stumbling block that can trip people up is using the Command Prompt — something that we’ve had to use less and less over the past decade!

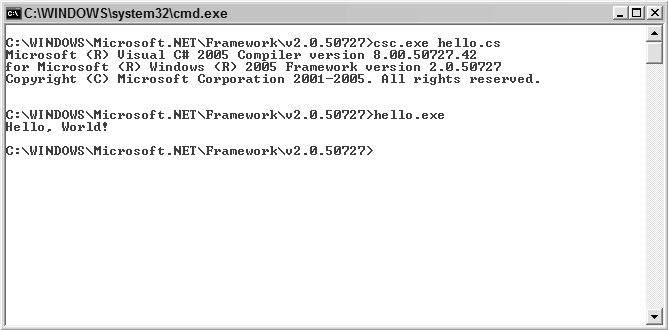


Figure 2-9

##### One Cheap Utility That Makes Life Easier!

We could go on for pages and pages listing dozens of different tools and software utilities that could make your C# programming experience easier and more fun. However, we’re not going to do that! Partly because it’s boring, but mostly just because it’s just as easy for you to fire up a web browser, take a trip over to a search engine, and do a search (for example, a search for “C# tools” on Google brings up 8.9 million results).

There is, however, one tool that we are going to recommend if you think that a lot of your time with C# is going to be spent at the cheap end of the cost spectrum: a text editor called UltraEdit (see Figure 2-10).

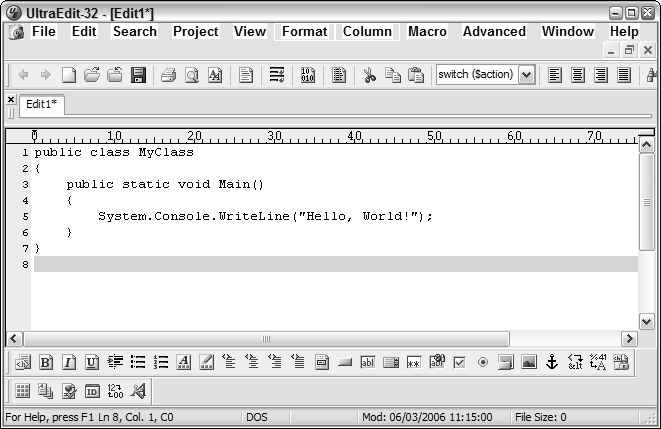


Figure 2-10 13

Why do we recommend UltraEdit? Quite simply because it is the best text editor you are likely to come across and because it has features specifically designed for programmers. Some of these features include:

* **Code folding.** This allows you to fold or collapse functions and structure in C# code simply by clicking [+] and [-] that appear in the interface next to the code (see this in action in Figure 2-11).

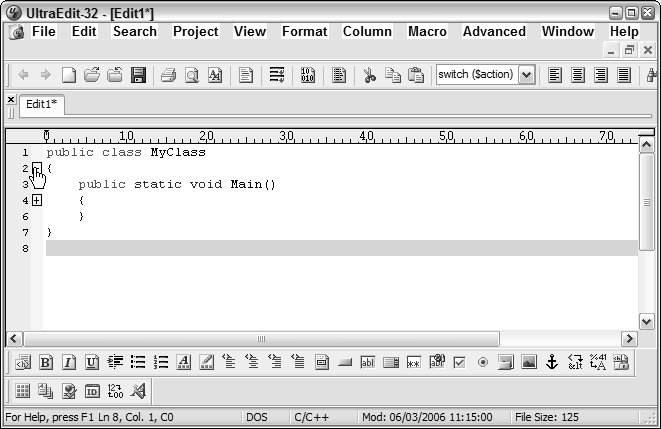


Figure 2-11

* **Spell-checker.** Can be handy!
* **Syntax highlighting.** Preconfigured syntax highlighting within the document, making C# code easier to follow
* **Bookmark facility.** Allows you to place bookmarks within code so you can get to them easily later on
* **Support for ASP.** This means that you can use it for web-enabled .NET applications.
* **Line numbering.** A very useful feature indeed, especially when trying to track down an error!
* **Support for big files.** By *big*, we mean over 4GB.
* **Excellent Search.** Can make use of regular expressions for precise searching
* **Large line lengths.** Notepad wraps lines after 1,024 characters; UltraEdit supports lines up to 9,000 characters (not that you’re going to need that very often).

There’s a free 45-day trial version of UltraEdit available. A single license for UltraEdit costs $39.95. We think that this represents very good value for the money.

For more information on UltraEdit, visit [http://www.ultraedit.com.](http://www.ultraedit.com/)

*If you want to try other text editors, a quick search of the Internet will bring dozens to your attention. A good selection is available of both free text editors (think of them as replacements for Windows Notepad) and commercial ones. We find that UltraEdit works for us, but we want you to find the text editor that works for you!*

##### Alternative Text Editors and C# Tools

There are a number of alternative text editors and C# tools that you could choose to use. Here is a short- list of a few that you might like to try out:

* + Crimson Editor — <http://www.crimsoneditor.com/>
  + EditPad Lite — <http://www.editpadpro.com/editpadlite.html>
  + NoteTab — <http://www.notetab.com/>
  + Notepad ++ — <http://notepad-plus.sourceforge.net/>
  + EditPad Pro — <http://www.editpadpro.com/>
  + Antechinus C# Editor — <http://software.ivertech.com/AntechinusCEditor_soft-> ware4151.htm
  + Programmer’s Notepad — <http://www.pnotepad.org/>

##### Enterprise Tools - Visual Studio and Visual C#

Finally, we shift our attention from no-cost and low-cost tools to high-cost tools for C# development.

We’re not going to spend much time covering enterprise tools in this book, because you really can’t do them justice in a few pages. And anyway, Wrox has a number of other books specific to the Visual Studio/Visual C# platforms. As far as these enterprise tools are concerned, it’s likely that you already have them and know how to use them or that you don’t have them and aren’t that interested in spend- ing a lot of cash on them.

*For a complete list of Wrox titles, visit our website at* http://www.Wrox.com*!*

Briefly, two products from Microsoft allow you to program using C#. These are:

* + **Microsoft Visual Studio.** This is the flagship programming package from Microsoft that incorporates:
    - .NET Framework
    - ASP.NET
    - Visual Basic .NET
    - Visual C++

15

* Visual J#
* Visual C#
* **Microsoft Visual C#.** This is the standalone version of the C# development environment included in Visual Studio.

Visual Studio (current version is called Visual Studio 2005) comes in a number of different editions:

* **Team System Edition.** Allows for collaboration among software development teams. This is the flagship edition of Visual Studio.
* **Profession Edition.** Comprehensive development environment aimed at individual developers
* **Visual Studio Standard Edition.** Simplified version of the Professional Edition
* **Visual Studio 2005 Tools for Microsoft Office.** Tools to build robust Microsoft Office solutions

There are huge benefits in terms of speed of development and ease of use to having and using these tools, and they are pretty much a must if you want to really get down and leverage the Windows operat- ing system. However, they represent a huge learning curve to anyone not familiar with them (the inte- grated development environment, while making the life of the professional developer easier, isn’t all that user friendly to newcomers).

To smooth this over, we are going to assume that you’re either already experienced in using these tools and don’t need us to tell you how to do that or that you aren’t using these just yet and don’t need them right now.

Microsoft also has a low-cost/no-cost version of Visual C# called Visual C# 2005 Express Edition aimed specifically at the hobbyist, novice, or student developer. This is a great solution for those who want to get into professional development but don’t want to spend a lot of money on software.

For more details on Microsoft Visual Studio visit, [http://msdn.microsoft.com/vstudio/.](http://msdn.microsoft.com/vstudio/)

##### Summary

This chapter looked at a span of tools that you can use for C# development, ranging from free tools that will enable you to do basic C# development, all the way up to sophisticated development environments such as Visual Studio that are very powerful but also very expensive.

From this point on, we are going to try to remain “tool neutral,” but forgive us if we sometimes use or refer to a particular application (more than likely, this will be in screenshots). You are free to use what- ever software application or combination of tools best suits your needs.

Chapter 3 provides an overview of the C# programming language.