

BA 272 Notes on Compiling in Visual Studio

Remember the simple program from before?

```
using System;
class ICan
{
    static void Main(string[] args)
    {
        Console.WriteLine("I can compile and run programs I write");
        Console.Write("Type something:");
        Console.WriteLine("You Typed:" + Console.ReadLine()); // show what was typed
        Console.ReadLine(); // this makes the program wait for you press enter
    }
}
```

While it is possible to compile the program on the command line as we did before, most .Net development is done using Visual Studio. Using this IDE (Integrated Development Environment) to develop programs has many benefits including syntax assistance, one-click compiling, management of multiple code files, and debugging support, just to name a few.

However, to accomplish all those complex and helpful things, Visual Studio manages the simpler source code files we have seen so far in a “project.” Actually, projects are also managed as parts of “solutions.” The source codes we use in BA272 are usually stored in a single file. However, for even relatively simple real programs a variety of files have to be managed together. A project can include several separate .cs source code files, related files containing data, and image files. Very importantly, libraries of code built by others or for different projects often need to be combined to create a useful application. Effective Java, C++, and .Net programmers almost all use an IDE or different code management tools to keep these many parts organized and to simplify the build process that turns source codes into executable programs.

***** New 10/5/2009 *****

First: when Visual Studio was installed on these computers, the “settings” were adjusted to best support Visual Basic programmers. The function used in this walk through requires you to return Visual Studio to General Development settings as follows:

- On the *Tools* menu click on “*Import and Export Settings...*”
- Choose “*Reset all settings*” and click *Next*>.
- Choose “*No, just reset settings, overwriting my current settings*” and click *Next*>.
- Choose “*General Development Settings*” and click *Finish*.
- Close the Wizard.

***** New 10/5/2009 *****

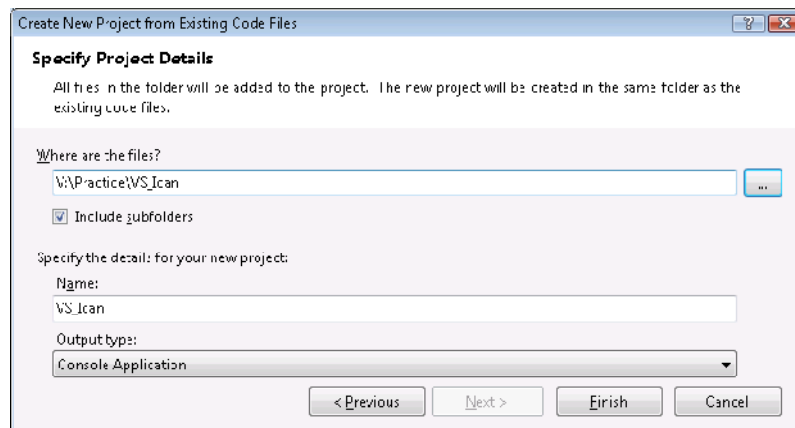
Perform and understand the following steps:

- Map drive v: to your class work area
 - o Open a command prompt (Start → Run cmd press enter)
 - o Type: Net use v: \\bus\storage\ClassFolders\2009_Fall\BA272\001\yourONIDID
 - o Press enter. That runs the “Net” program, passing in 3 parameters (use, v:, and \\bus\storage\ClassFolders.....)
 - o Actually you can paste the command into the command prompt window: select the text above (Net use) and do ctrl-c for copy, click on the c:\ icon in the command prompt window and select edit/paste
- DON’T FORGET, YOU HAVE TO USE YOUR ONIDID in place of “yourONIDID”

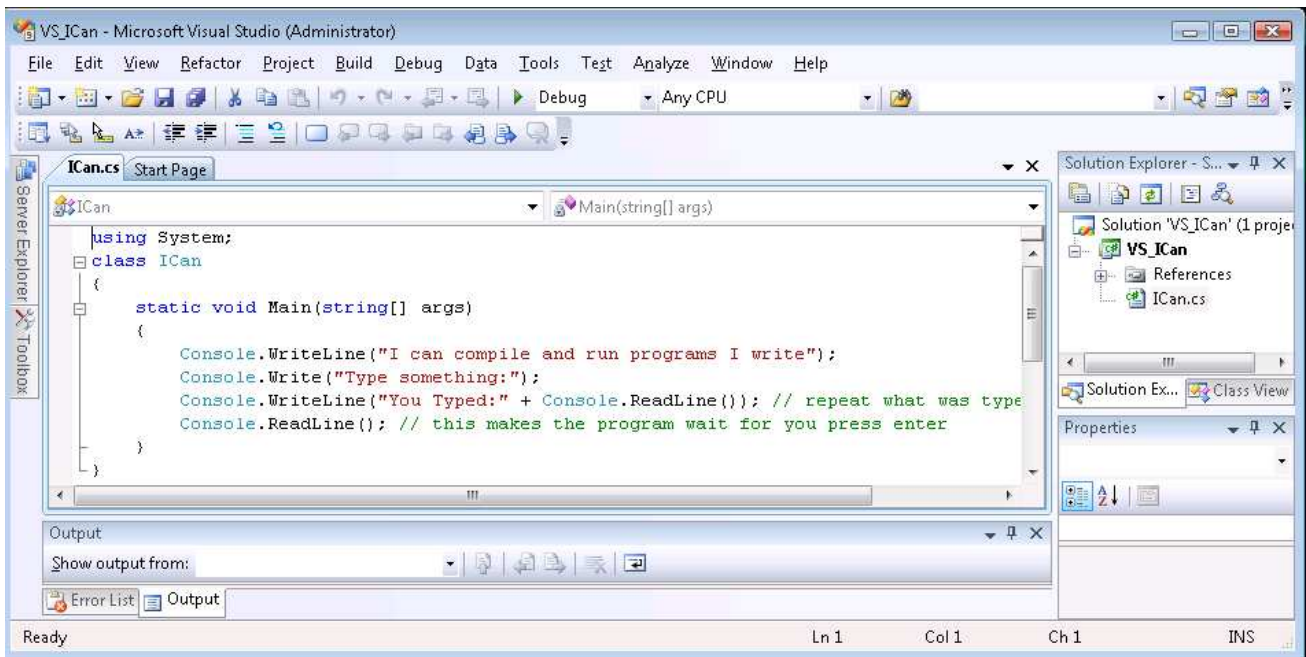
- Open a new file in notepad. Paste the program above into the notepad file.
 - o Save the file in a new directory called Practice\VS_ICan as VS_ICan.cs (This is a source code file)
- Open Visual Studio 2008
- Start → All Programs → Microsoft Software → Microsoft Visual Studio 2008 (the folder) → Microsoft Visual Studio 2008 (the icon)
- Have Visual Studio Create a new project from code for your source file
 - o File → New → Project From Existing Code



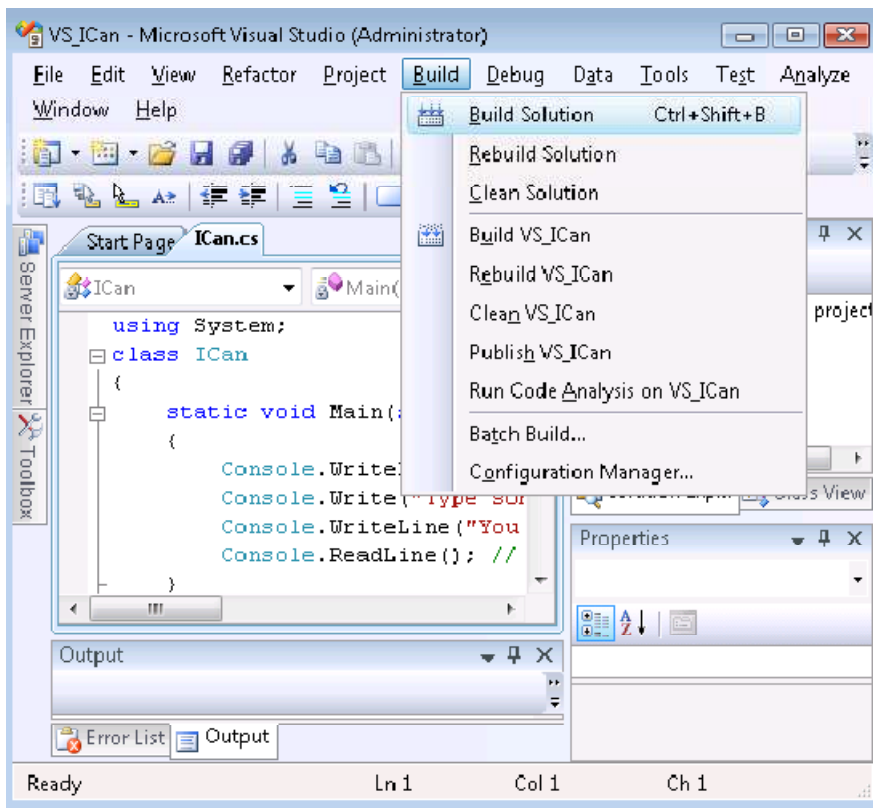
- o This opens a “Wizard” which asks you what type of Project. Choose “Visual C#” and click Next
- o Specify Details for your new project: “Name” = VS_ICan and Output type: “Console Application” as shown below
- o Use the button [...] next to the box for “Where are the files?” to navigate to your new directory. It should show as V:\Practice\VS_ICan as shown below. Click “Finish”



Click the Finish button. It may give you a security warning because the network drive is not on your local computer. Hopefully that won't cause too much trouble. You should see something like this:



To compile the code now, you click Build → Build Solution.



You can run the program by clicking the green arrow or by pressing F5. This actually runs the program in “Debug” mode which we will talk about later this term.

The build also creates a new `VS_ICan.exe` in the `V:\Practice\VS_ICan\bin\Debug` directory. It could be executed from a command prompt.

Importantly, the file `V:\Practice\VS_ICan\VS_ICan.cs`, like the `ICan.cs` file we made in a previous exercise, can be compiled on the command line. For several assignments in BA272 we will pass these `.cs` files around, expecting student reviewers (and instructors!) to be able to put that code into a Visual Studio project for compiling and running.

You now know two ways to compile, from the command line for a simple `.cs` file and with Visual Studio. It is important that you realize this key idea: Visual Studio is a tool to help develop, manage and compile applications. Source code and settings are simply stored in text files. As we will see later in the course, Visual Studio includes some especially cool tools to help generate nice graphical user interfaces. But in the end, all it really does is process a set of source code generated by the developer.

We won't talk much in this course about the power tools organizations used to manage development projects. But it suffices to say that Visual Studio and related tools (such as Team Suite) allow for management of complex development processes. These tools enhance programmer productivity, protect source code, help in coordinating the efforts of programming teams, help manage the software lifecycle, and help in integrating database, web services, and other available systems. Visual Studio is not the only IDE. Other products exist for non-Microsoft environments as well such as java and C++.

Hint: These instructions are short, but I can find a number of quiz/test questions hiding in there.