

PART 1: “Getting Started”

Introduction

Welcome to the first article in the C++ FASTTRACK tutorial series! These tutorials are designed to take you from zero to a decent entry level in a somewhat smooth fashion. We start at the absolute basics: all that you need to get started is a laptop or PC, a fair bit of time, and quite a bit of dedication.

Regarding dedication: These tutorials are supposed to take you anywhere between two and four hours each. You are supposed to complete *two* tutorials per week, in order to meet the deadline (the end of the block). That means that you have an *extra* workload of 4-8 hours. No-one said this was going to be easy, right?

If you need support while going through these articles, head to <http://dgdarc.slack.com>. You'll have no problems getting answers to your newbie questions there. Lecturers and experienced programmers also roam this channel.

About the authors

This course was originally created for the IGAD program of the NHTV University of Applied Sciences in Breda; see <http://made.nhtv.nl>. It was recently updated for the students of the Utrecht University; see <https://www.uu.nl/en/organisation/faculty-of-science/education/bachelors-programmes>.

This tutorial was written by Jacco Bikker, Brian Beuken, Nils Deslé, and Carlos Bott and updated for Visual Studio 2015 by Jacco Bikker.

Getting the stuff you need

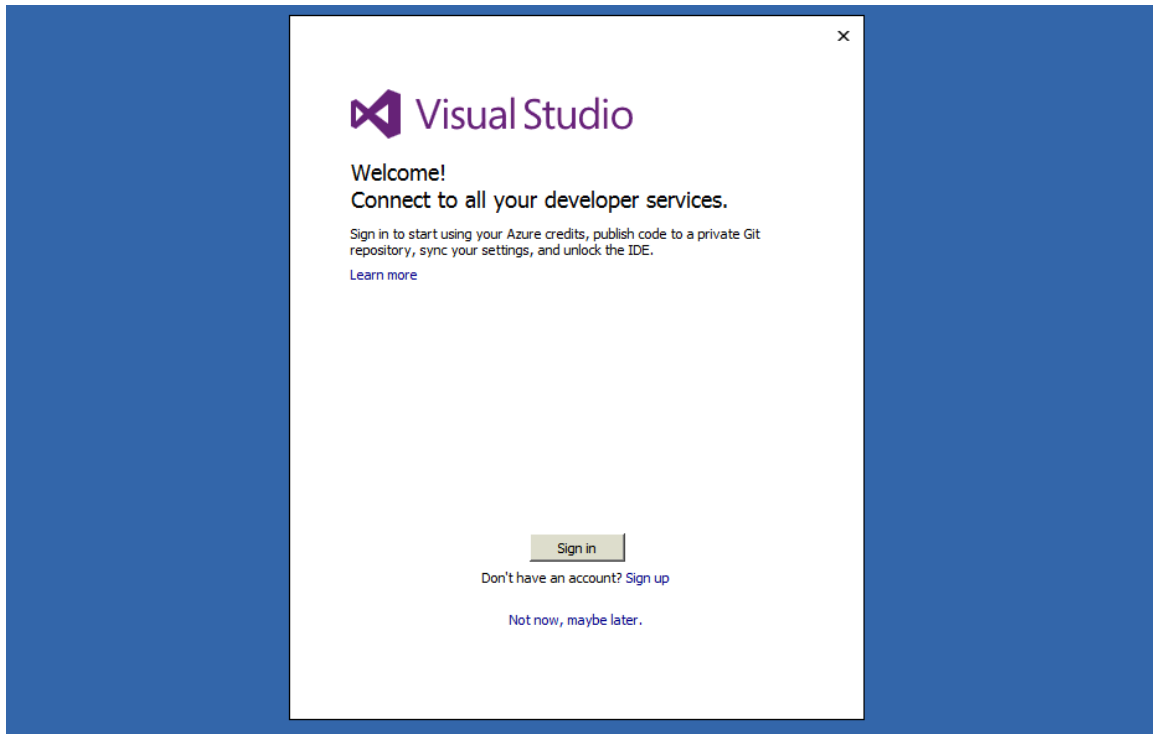
For the purpose of this course, I will assume you develop your games on PCs. The preferred development environment is Microsoft's Visual Studio 2015. A free version of this software is available from Microsoft, via <https://www.visualstudio.com/downloads>.

Checklist:

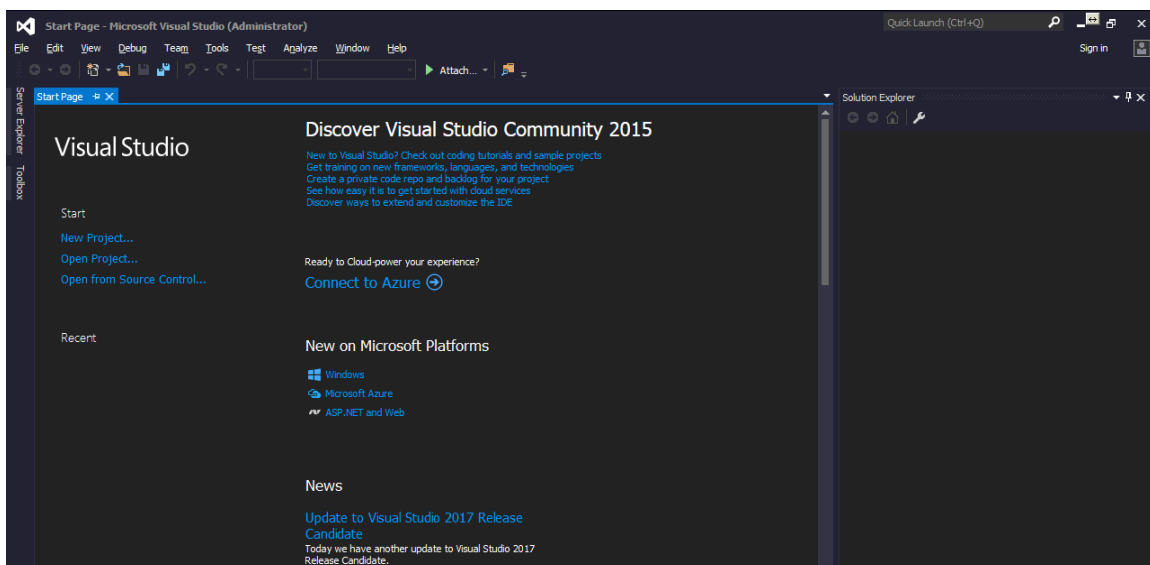
- ✓ A PC or laptop, pretty much any CPU and GPU will do
- ✓ Windows 7 (with SP1), 8 or 10, 32-bit or 64-bit
- ✓ Visual Studio 2015 Community Edition, installed with default settings

Starting Visual Studio 2015

When you start Visual Studio 2015 for the first time you are greeted with this window:



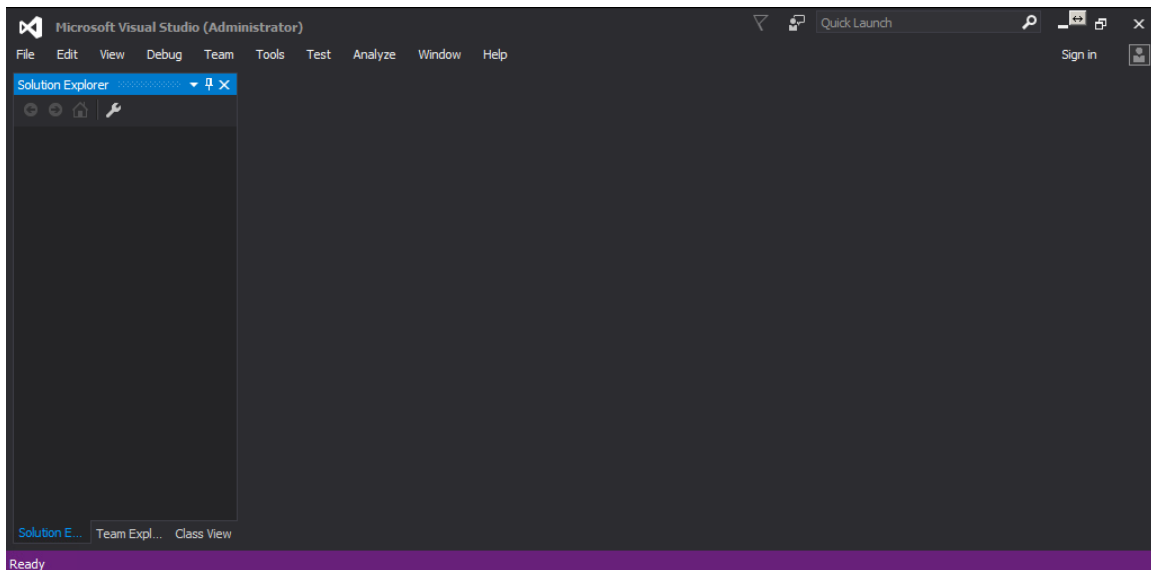
Selecting 'not now, maybe later' is luckily a perfectly viable option, so let's go for that (no need to leave any more personal information with a tech giant than strictly necessary, right?). In the next window, you get to choose a theme. As the window says, you can always change this later, so let's go with 'dark'. Now we finally get into the 'IDE' (which stands for 'integrated development environment'):



As you can see, there is a lot of stuff on the screen. Let's make some room:

- Close the 'start page'.
- Click on 'Server Explorer', then close it to get rid of it.
- Do the same to 'Toolbox'.
- Drag the 'Solution Explorer' to the left of the screen.
- In the main menu, click on View, go to Toolbars, and disable 'standard'.

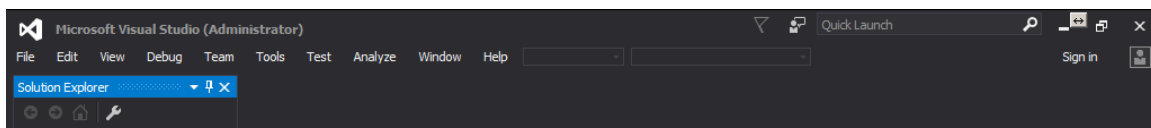
You now have something like this:



I often see people work on the area of a stamp (not just when programming) due to all the toolbars: this layout prevents that. We didn't remove anything that we need (apart from one thing), and we assumed control of the software in the process.

The one thing we are missing is two drop-down boxes (OK, so two things). We can store these next to the menu bar, so they don't need their own toolbar. Do this: click on Tools, then Customize, and select the Commands tab. Now click on Add Command, select the 'Build' category, and choose the Solution Configurations 'command'. This adds the drop-down box to the left of your menu bar; click the 'Move Down' button a few times, until it is all the way to the right. Repeat this for the Solution Platforms command.

The top of the screen now looks like this:



And that is as tidy as it gets for now.

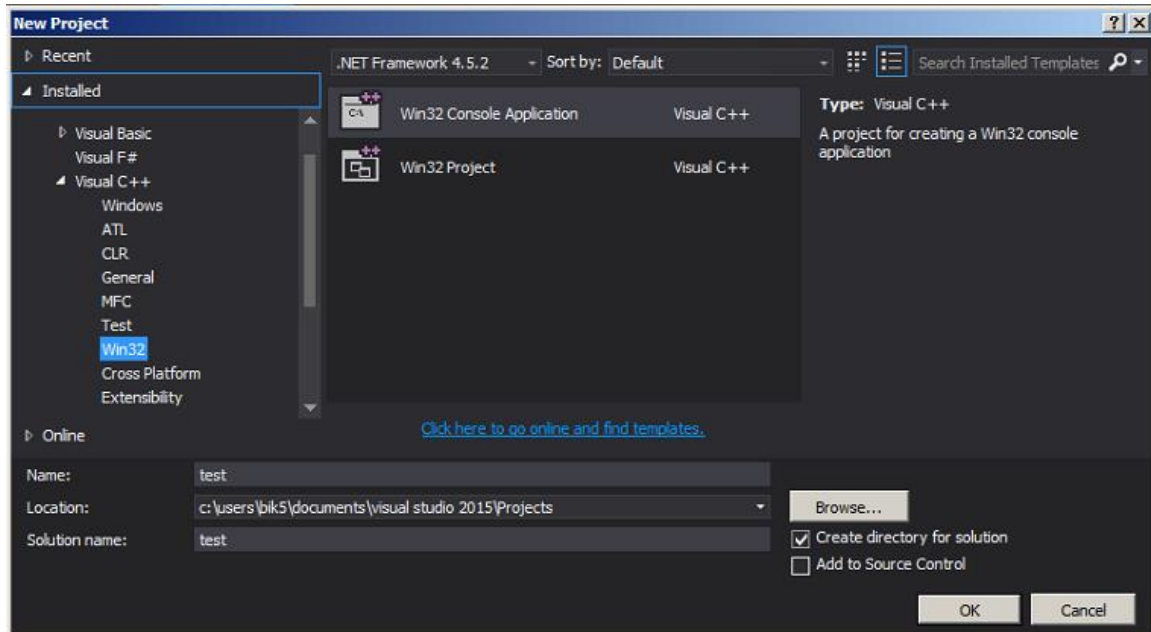
Creating a program

Before we go on:

“This is the first and only article in these series that is this boring. Starting with part 2, you will be coding, not downloading + installing + adjusting settings. Promise.”

That being said, let’s get coding!

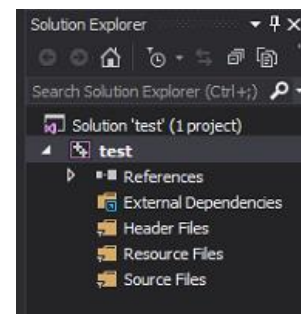
From the File menu, select New, then Project:



You need to change various things in this dialog. Set the name of your project to ‘test’ (or use your imagination). Set the location to `c:\my_projects` (so you can actually find your files). Unselect ‘create directory for solution’. It will create enough directories anyway. Finally, make sure you selected ‘Win32’ and ‘Win32 Console Application’. Then click OK. In the next window, click ‘Next’ (*not Finish*). In the next window, tell VS that you want an empty project, without any ATL, MFC or SDL stuff. Now click Finish.

Your ‘solution’ (you didn’t solve anything, but whatever) should look like the image on the right now.

Now, let’s delete some things we don’t need: Select all three folders (Header files, Resource files and Source files – you can’t delete ‘external dependencies’) and delete them. You can also kill the properties panel that may have appeared.



Now it’s finally time to add a source file to play with. Right-click on ‘test’ and select Add → New Item. In the dialog, select ‘C++ File (.cpp)’, and name it `main.cpp` (or anything else really, as long as it ends with `.cpp`). Then click ‘Add’. The new file is immediately opened, and we’re ready for some coding.

Hello world

A tutorial on C++ is not complete without a 'hello world' example, so here we go. Enter the following program:

```
#include "stdio.h"

void main()
{
    printf( "Hello world!\n" );
    getchar();
}
```

You can start the program by hitting F5. Visual Studio will ask you if you want to build the program; you want this so click 'yes'.

Discussion

When you go to `c:\my_projects\test`, you will see that Visual Studio created a large amount of files for you. The most interesting one is in the 'Debug' folder, and it's called `test.exe`. When you start this application, you will see that you created it yourself.

Apart from this file, we have:

- `test.sln`: your 'solution file', which points to
- `test.vcxproj`: your 'project file', which points to
- `main.cpp`: the 'source file'.

A project can have multiple source files, and a solution can have multiple projects. Other files:

- `test.vcxproj.filters`: stores the folders in your project;
- `test.VC.db` and `opendb`: some internal information;
- various files in the Debug folder: `.obj`, `.ilk`, `.pdb`.

Quite confusing perhaps, but all that matters right now is the `.cpp` file in `c:\my_projects\test` and the executable that it was turned into.

Name ^	Date modified	Type	Size
.vs	1/27/2017 7:05 PM	File folder	
Debug	1/27/2017 7:36 PM	File folder	
main.cpp	1/27/2017 7:35 PM	CPP File	1 KB
test.sln	1/27/2017 7:05 PM	Microsoft Visual Stu...	2 KB
test.VC.db	1/27/2017 7:05 PM	Data Base File	1,480 KB
test.VC.VC.opendb	1/27/2017 7:05 PM	OPENDB File	1 KB
test.vcxproj	1/27/2017 7:35 PM	VC++ Project	8 KB
test.vcxproj.filters	1/27/2017 7:35 PM	VC++ Project Filter...	1 KB

Quick advice: perhaps your explorer window does not show file extensions (`.sln`, `.db`, ...), as in the above screenshot). Fix this, you need this info. This is done in the windows control panel: go to Folder Options, then the 'View' tab, and disable 'Hide extensions for known file types'.

Assignment

Every article in this series will end with a small practical assignment. You need to complete this assignment, and send the result to one of the programming teachers. Once you have done so, you may continue with the next part.

So, here's your task:

Create an executable that prints your name 10 times, then waits for a key, then prints 'Coding is awesome!' 10 times, then waits for a key again, then exits.

END OF PART 1

Next part: "The PR1 Template"