0.5 — Installing an Integrated Development Environment (IDE)

BY ALEX ON MAY 28TH, 2007 | LAST MODIFIED BY ALEX ON JULY 9TH, 2018

As mentioned in the previous section, an **Integrated Development Environment (IDE)** contains all of the things you need to develop, compile, link, and debug your programs. So let's install one.

The obvious question is, "which one?". Keep in mind that you can install multiple IDEs, so there is no "wrong decision" here. During the course of these tutorials, we will be showing you some of the nice features of your IDE, such as how to do integrated debugging. All of our examples will be done using both **Microsoft's Visual C++** (for Windows), and **Code::Blocks** (for Linux or Windows). Thus we highly recommend you pick one of these.

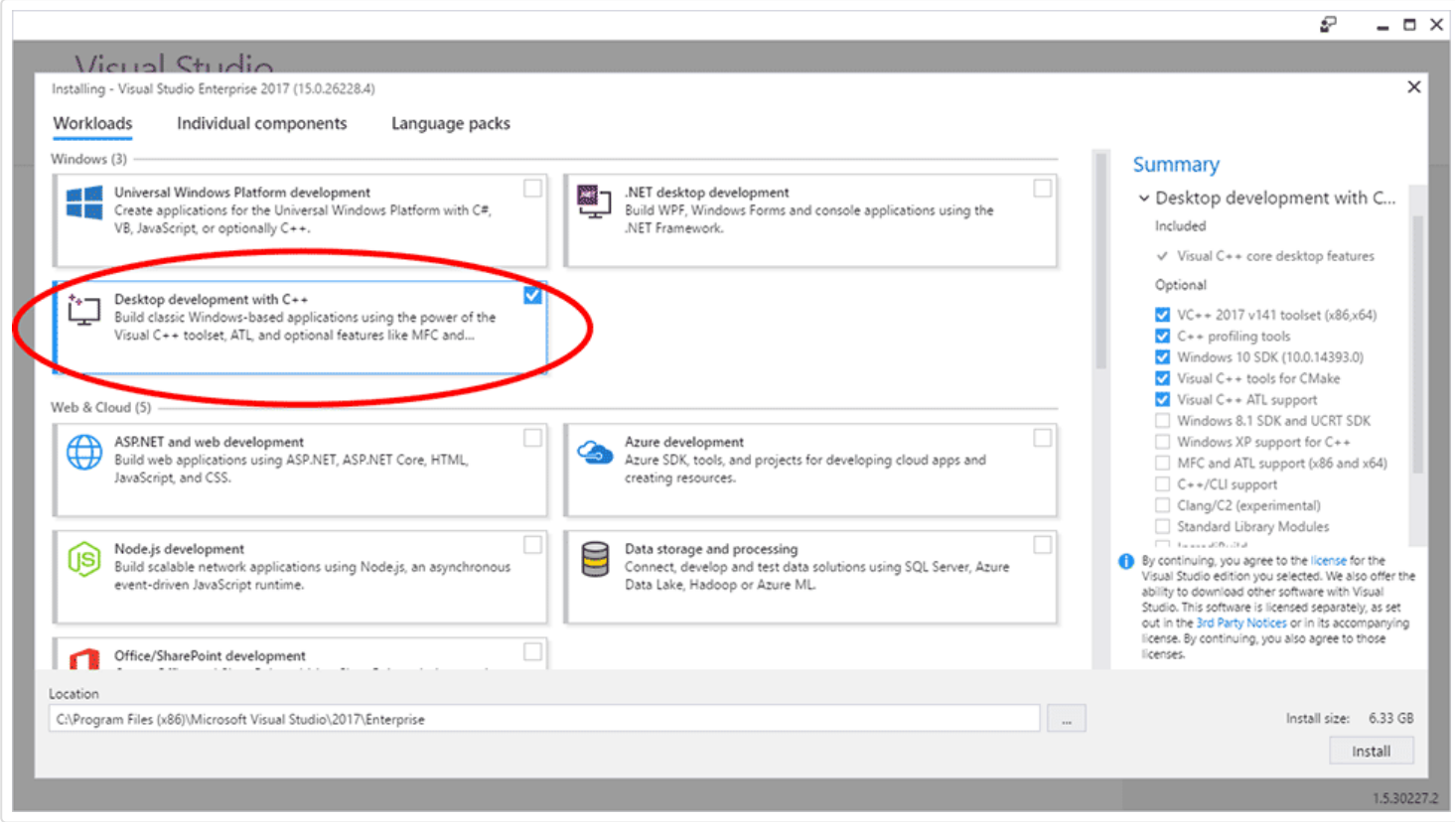
However, if you would like to try a different IDE, you are free to do so. The concepts we show you will work for any IDE -- however, different IDE's use different keymappings and different setups, and you may have to do a bit of searching to find the equivalent of what we show you.

Windows IDEs

If you are developing on a Windows machine (as most of you are), then you have two choices:

1) If disk space and/or download size are not a constraint, then we recommend **Visual Studio Community 2017**. When you run the installer, you'll eventually come to a screen that asks you what workload you'd like to install. Choose "Desktop development with C++".

The default options selected on the right side of the screen should be fine, but please ensure that the Windows 10 SDK is selected. The Windows 10 SDK can be used on older versions of Windows, so don't worry if you're still running Windows 7 or 8.



This will take about 6.3 gigs of drive space.

2) If disk space and/or download size are a challenge, then we recommend Microsoft's free **Visual Studio Express 2015 for Windows Desktop**, which you can find towards the bottom of the page.

The installer that you download off of Microsoft's web page is actually a downloader. When you run it, it will download the actual IDE from Microsoft and install it.

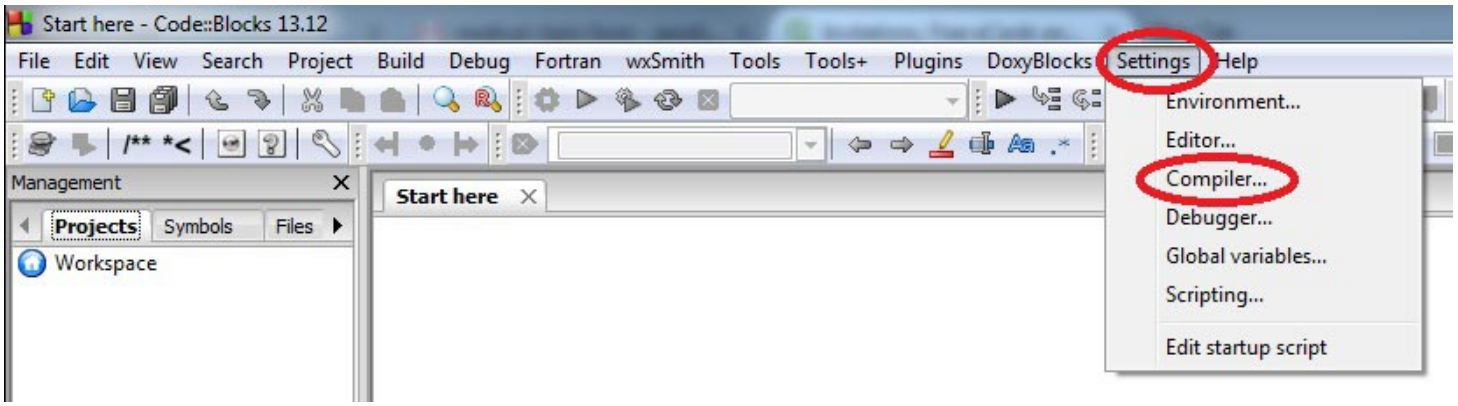
Note: This tutorial was originally written when Microsoft was distributing the 2005 version of Visual C++. Consequently, some references and screenshots are targeted to that version. Running any later versions (such as 2013, 2015, 2017, etc…) are fine, however, your screens may look slightly different.

Linux or Windows IDEs

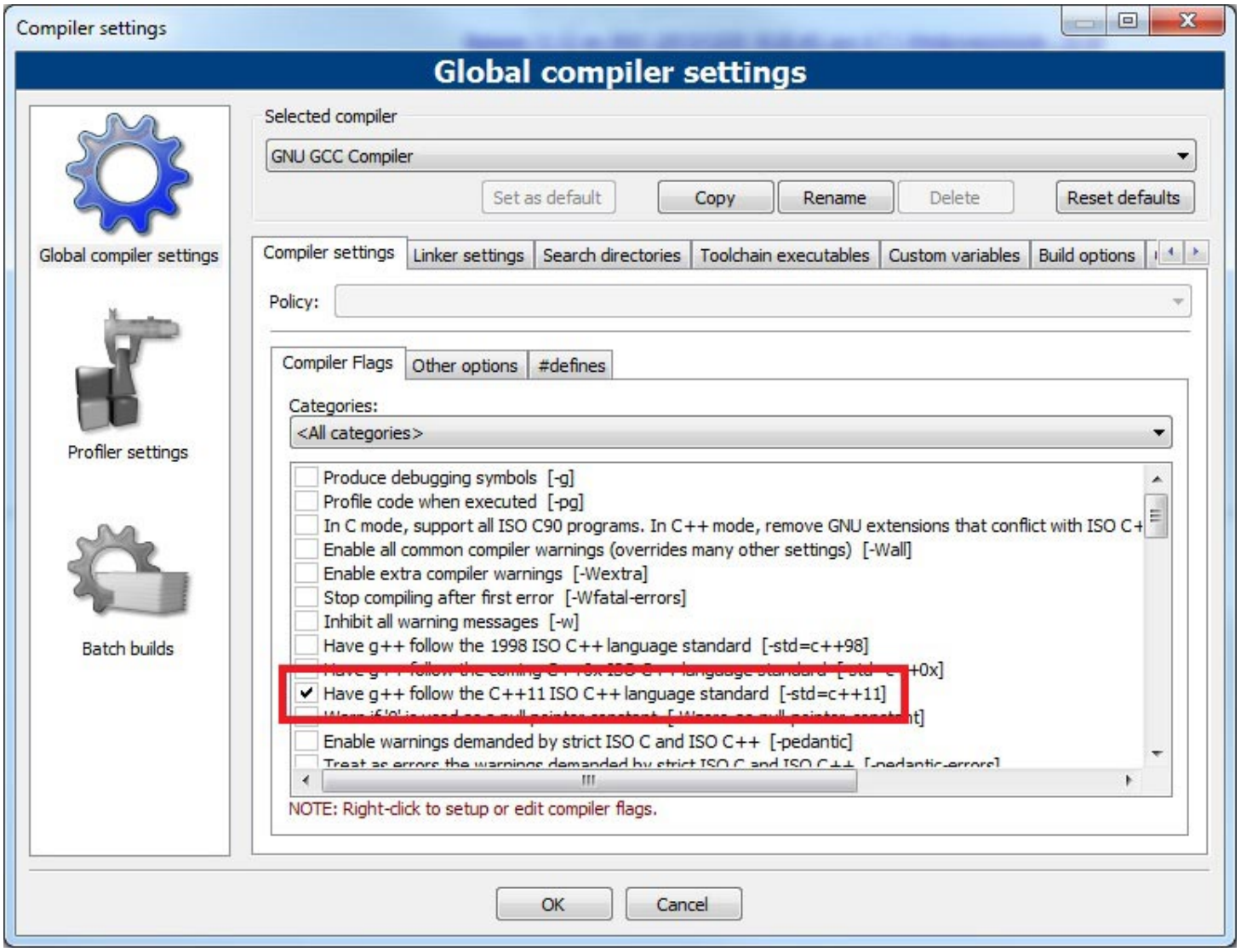
If you are developing on Linux (or you are developing on Windows but want to write programs that you can easily port to Linux), we recommend **Code::Blocks**. Code::Blocks is a free, open source, cross-platform IDE that will run on both Linux and Windows.

Windows users: make sure to get the version with MinGW bundled.

With Code::Blocks, C++11/C++14 functionality may be disabled by default. You'll definitely want to check and turn it on. First, go to Settings->Compiler:



Then check the box marked "Have g++ follow the C++11 ISO C++ language standard [-std=c++11]:



Note: If “Have g++ follow the C++14 ISO C++ language standard [-std=c++14]” exists for your version of Code::Blocks, use that instead.

After installing Code::Blocks, some users have been getting an error message “Can’t find compiler executable in your configured search paths for GNU GCC Compiler”. If you run into this, try the following:

1. In you’re on Windows, make sure you’ve downloaded the version of Code::Blocks WITH MinGW. It’s the one with “mingw” in the name.
2. Try doing a full uninstall, then reinstall.
3. Try going to settings, compiler, and choose “reset to defaults”.
4. Try a different compiler.

Alternately, some people prefer to use **Bloodshed’s Dev-C++**, which also runs on both Windows and Linux.

Mac OSX IDEs

Popular Mac choices include **Xcode** (if it is available to you), or **Eclipse**. Eclipse is not set up to use C++ by default, and you will need to install the optional C++ components.

Although Visual Studio for Mac has been released, it currently does not support C++, so at this time we can not recommend it.

Can I use a web-based compiler?

Yes, for some things. While your IDE is downloading (or if you’re not sure you want to commit to installing one yet), you can continue this tutorial using a web-based compiler, such as the one at **TutorialsPoint**.

Web-based compilers are fine for dabbling and simple exercises. However, they are generally quite limited in functionality -- many won’t allow you to save projects, create executables, or effectively debug your programs. You’ll want to migrate to a full IDE when you can.

When things go wrong (aka. when IDE stands for “I don’t even…”)

IDE installation seems to cause its fair share of problems. Installation might fail outright (or installation might work but the IDE will have problems when you try to use it due to a configuration issue). If you encounter such issues, here’s what to do next:
1) Check lesson **0.7 -- A few common C++ problems** to see if there’s already a fix or workaround there.
2) Uninstall the IDE (if it installed in the first place), reboot your machine, disable your antivirus or anti-malware temporarily, and try the installation again.

If you’re still encountering issues at this point, you have two options. The easier option is to try a different IDE. The other option is to fix the problem. Unfortunately, the causes of installation and configuration errors are varied and specific to the IDE software itself, and we’re unable to effectively advise on how to resolve such issues. In this case, we recommend copying the error message or problem you are having into a Google search and trying to find a forum post elsewhere from some poor soul who has inevitably encountered the same issue. Often there will be suggestions on things you can try to remedy the issue.

Moving on

Once your IDE is installed (which is one of the hardest things this tutorial will ask you to do), or if you’re temporarily proceeding with a web-based compiler, you are ready to write your first program!