

GATS Companion to Installing Visual C++ 2022 Community Edition

Author: Garth Santor

Editors: Trinh Hân

Version/Copyright: 1.1.0 (2023-08-26)

Overview

This document will show how to acquire and install, configure, and use Visual Studio 2022 Community Edition with its C/C++ workload¹. Currently, it will only discuss *console application* development.

Introduction

Back in 1993, Microsoft released Visual C++, an amazing development tool intended to replace several of its compiler products:

- Microsoft C/C++ 7.0 (MS/PC-DOS, their professional tool)
- Microsoft Quick C 2.5 (MS/PC-DOS IDE, for power users)
- Microsoft QuickC for Windows 1.0 (Windows, for power users)

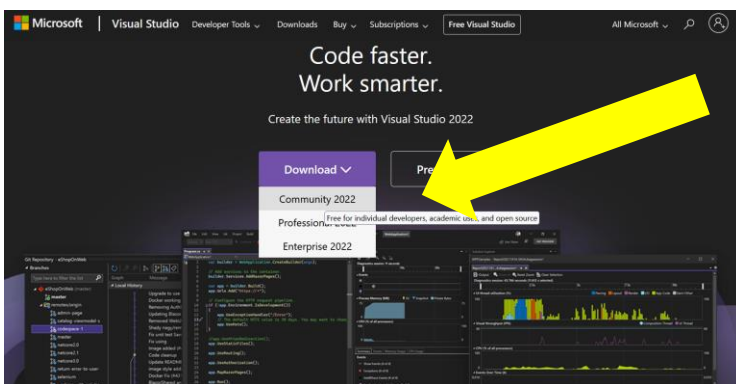
It blended the ease of using a visual integrated development environment (IDE) with its professional back-end compiler.

Since then, *Microsoft Visual C++* has become part of *Microsoft Visual Studio*, arguably the most successful commercial IDE. The current edition of the product is the 2019 edition, with a new edition arriving in 2022.

Currently, Visual C/C++ supports the ISO C17 and C++ 20 standards, with the ISO C++ 23 standard being nearly complete. The community edition is particularly appealing as it is free for the following purposes: individual developers, Open-Source projects, academic research, and educational use.

Acquiring Visual Studio 2022 Community Edition

Let's start at the Visual Studio Website found [here](#).



Select the Community 2022 Edition from the drop-down list control.

¹ Visual Studio *workloads* refer to the collection of tools that support development in a particular language or platform. Examples of workloads are: *Python development*, *Desktop development with C++*, and *Game development with Unity*.

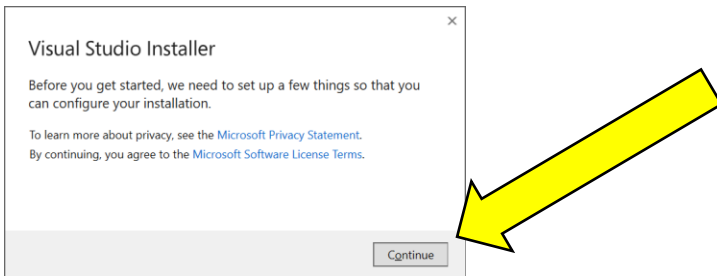
The site will automatically start a download of the installer **VisualStudioSetup.exe**. The installer isn't the full software package, but an install manager. Your security software may block the download (its an executable file), so you may have to unblock or indicate that the file is safe to download.

We will have to run this program, select the components.

Run **VisualStudioSetup.exe**

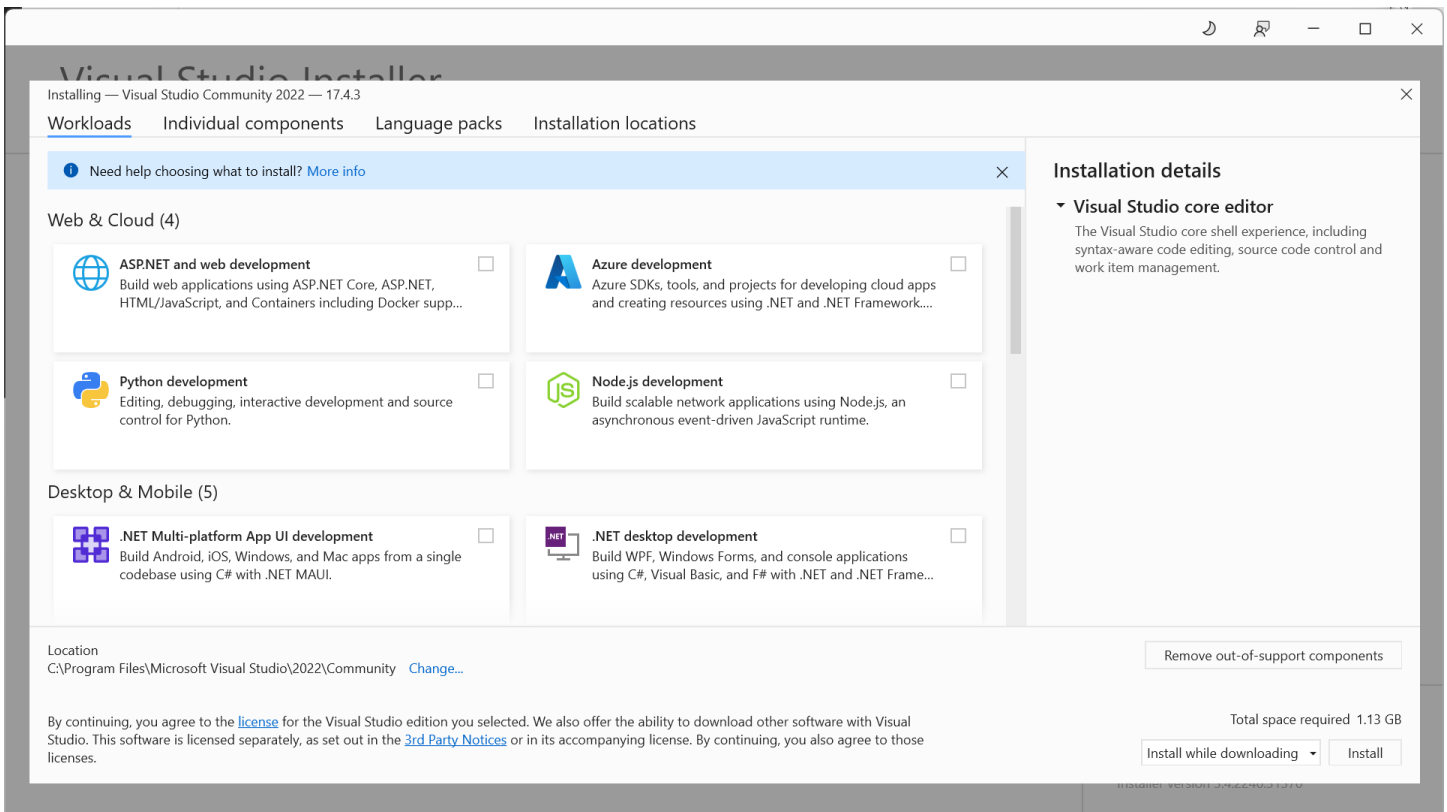
Each user may have slightly different experiences during the installation. These are likely due to difference in operating system version and previously installed software.

You'll typically see:



Click Continue.

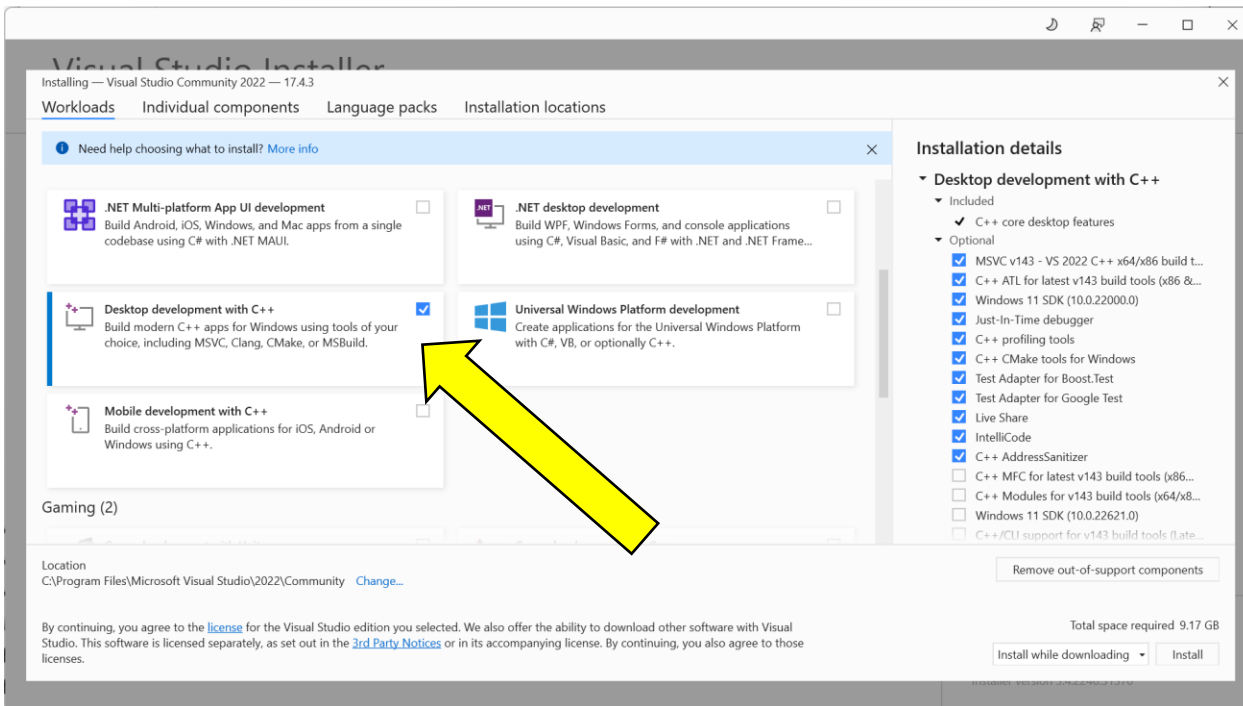
This will take you to the major installation options window:



There are many options here and you are free to choose as many as you would like. I'm just going to focus initially on the essentials for C/C++ console development. If you already have Visual Studio installed, then just verify that the following options are included or added.

Adding the C++ workflow

With the Workloads tab selected, scroll down until you see Desktop development with C++. Select that workflow.



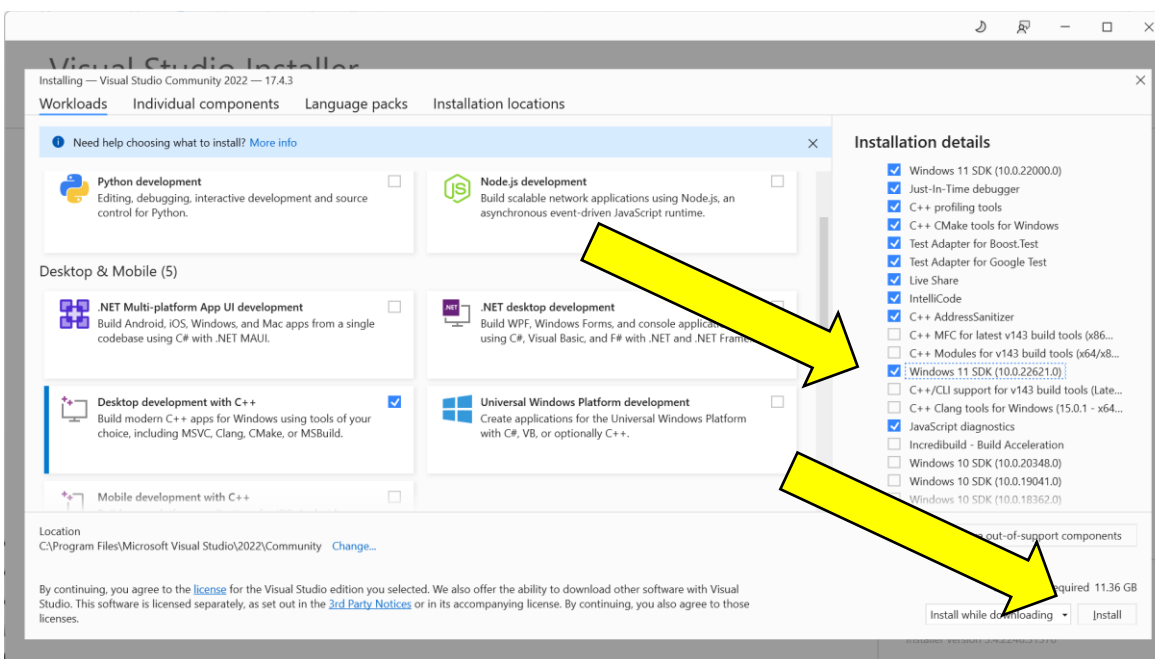
Additional Options

The panel on the right side contains the specific elements that you have selected. Let's add a few more.

Scroll the right-side list and add:

- Windows 11 SDK (10.0.22621.0)
- Windows 10 SDK (10.0.20348.0)

Then click **Install**.



The Visual Studio Installer will now download the C++ development workload while installing it.

Note that this configuration will require approximately 12 GB of space.

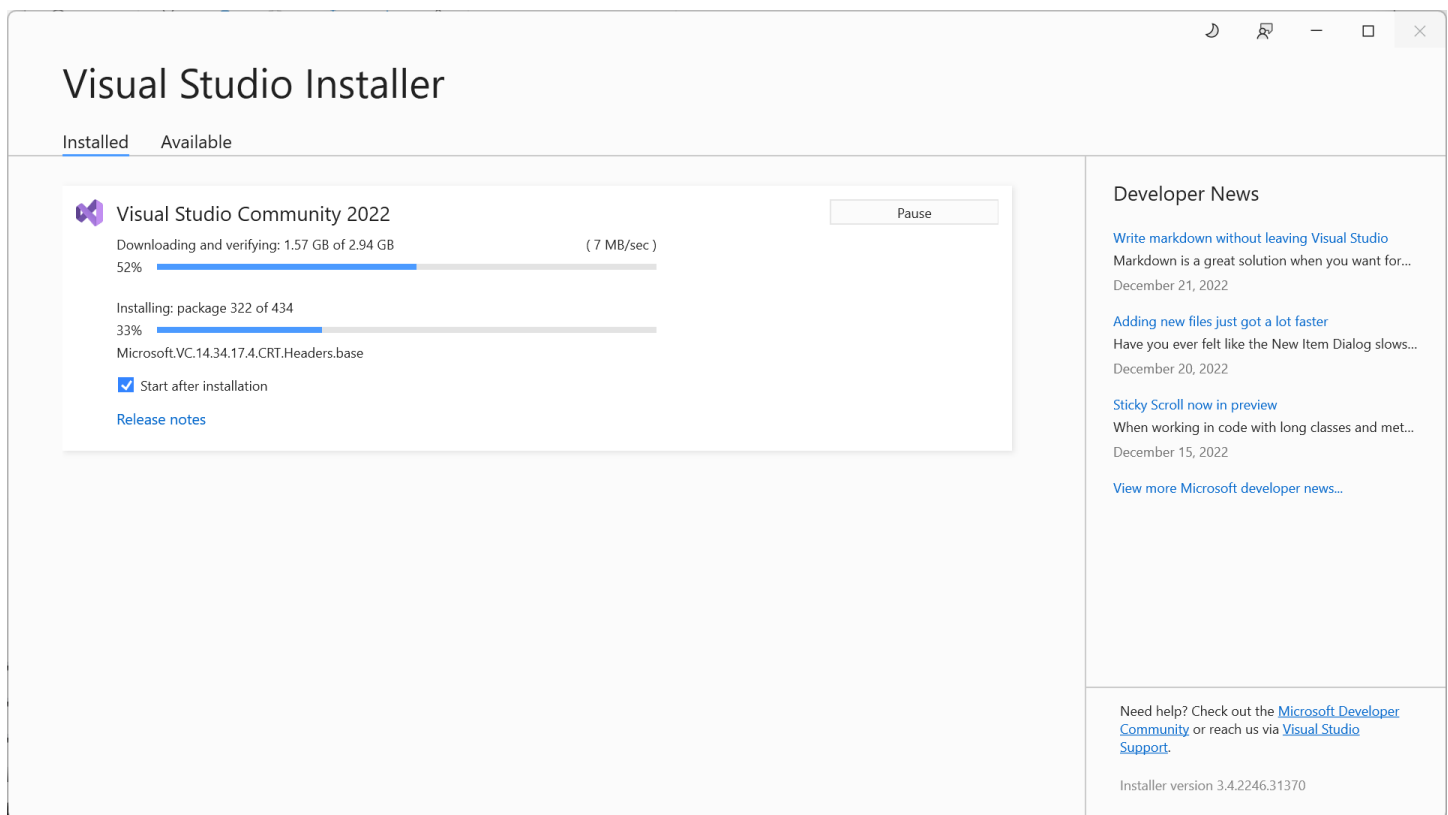
Why the Windows 10/11 SDKs?

The Windows SDK (Software Development Kit) for C++ provides headers, libraries, metadata, and tools for building Windows applications utilizing the Universal Windows Platform (UWP) or Win32 (the core Windows API). Without them, you would be restricted to ISO console applications only. Windows 11 has more features than Windows 10, so we install both.

Firstly, don't worry about the version numbers. Just pick the most recent for each of Windows 11 and Windows 10 (the highest version numbers presented).

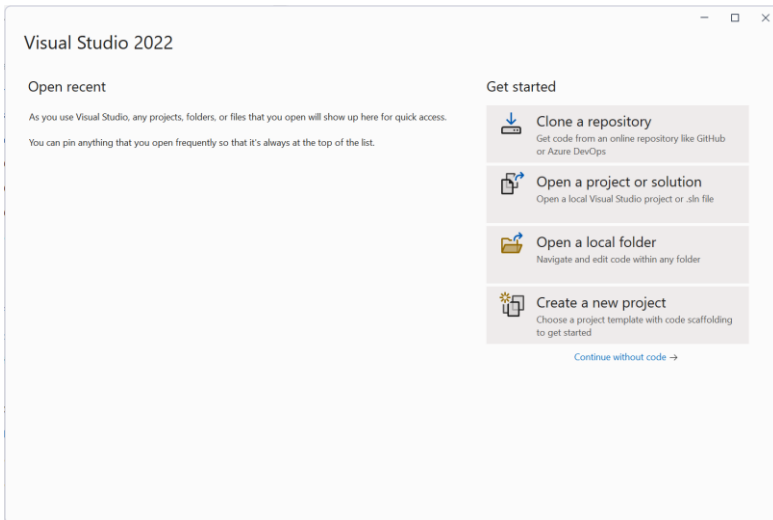
The wait...

The installation time will depend on the speed of your CPU, amount of RAM, disk storage system, network, Internet Service Provider, and how busy the Microsoft Servers are today. Typically, this installation completes in 15 to 20 minutes².



Visual Studio will automatically start if you left the 'Start after installation' option checked. If not, manually start Visual Studio 2022.

² This could take hours if you selected *everything*!



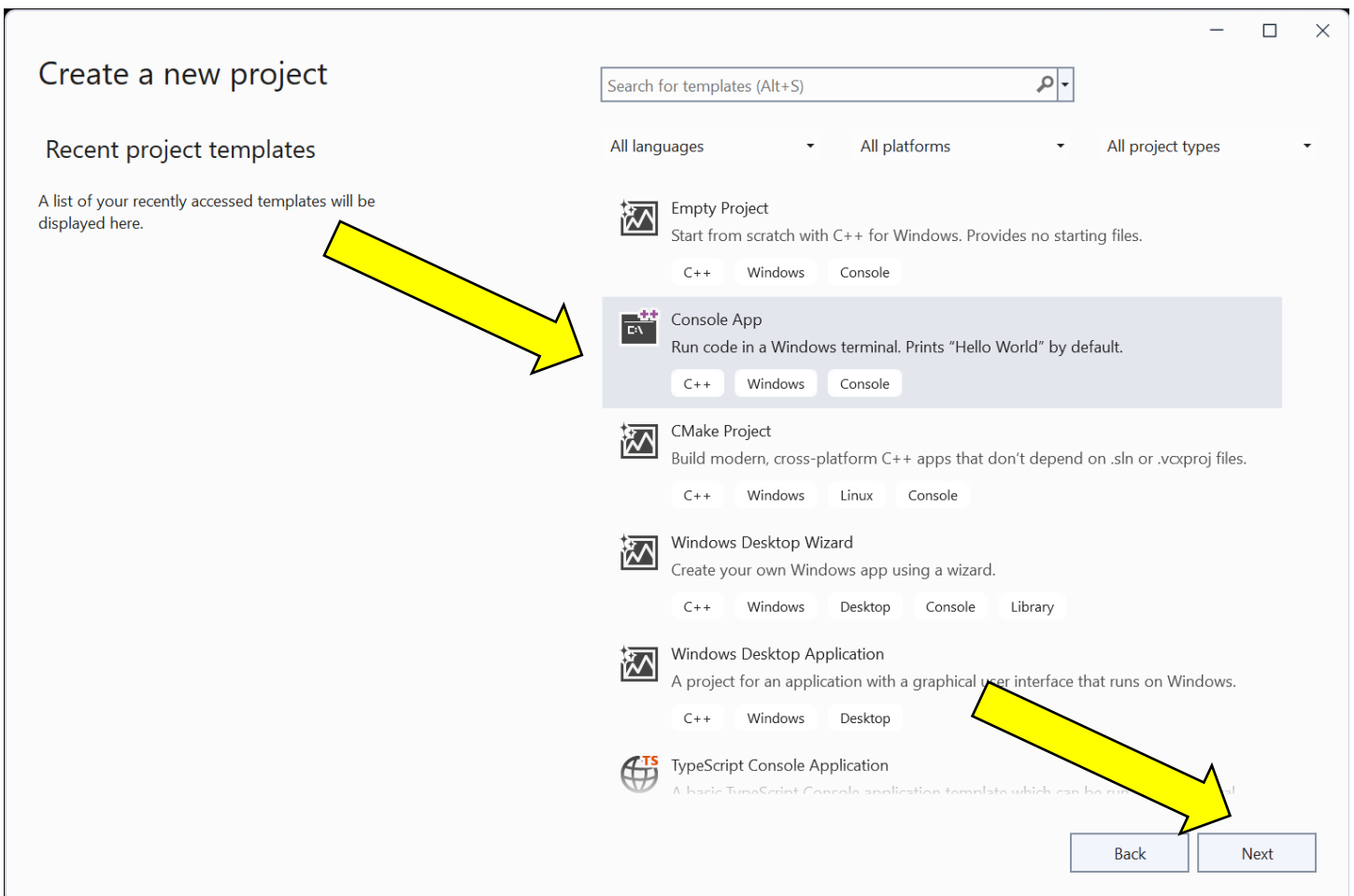
You can now close the Visual Studio Installer.

Testing your installation

If Visual Studio is not running, launch it.

From the 'Get started' window, choose 'Create a new project'.

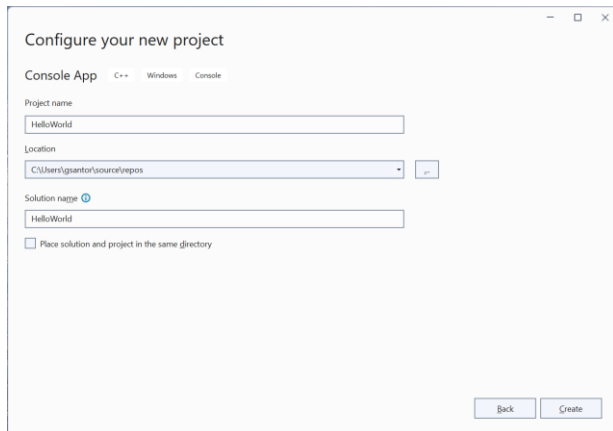
Choose 'Console App', then click 'Next'.



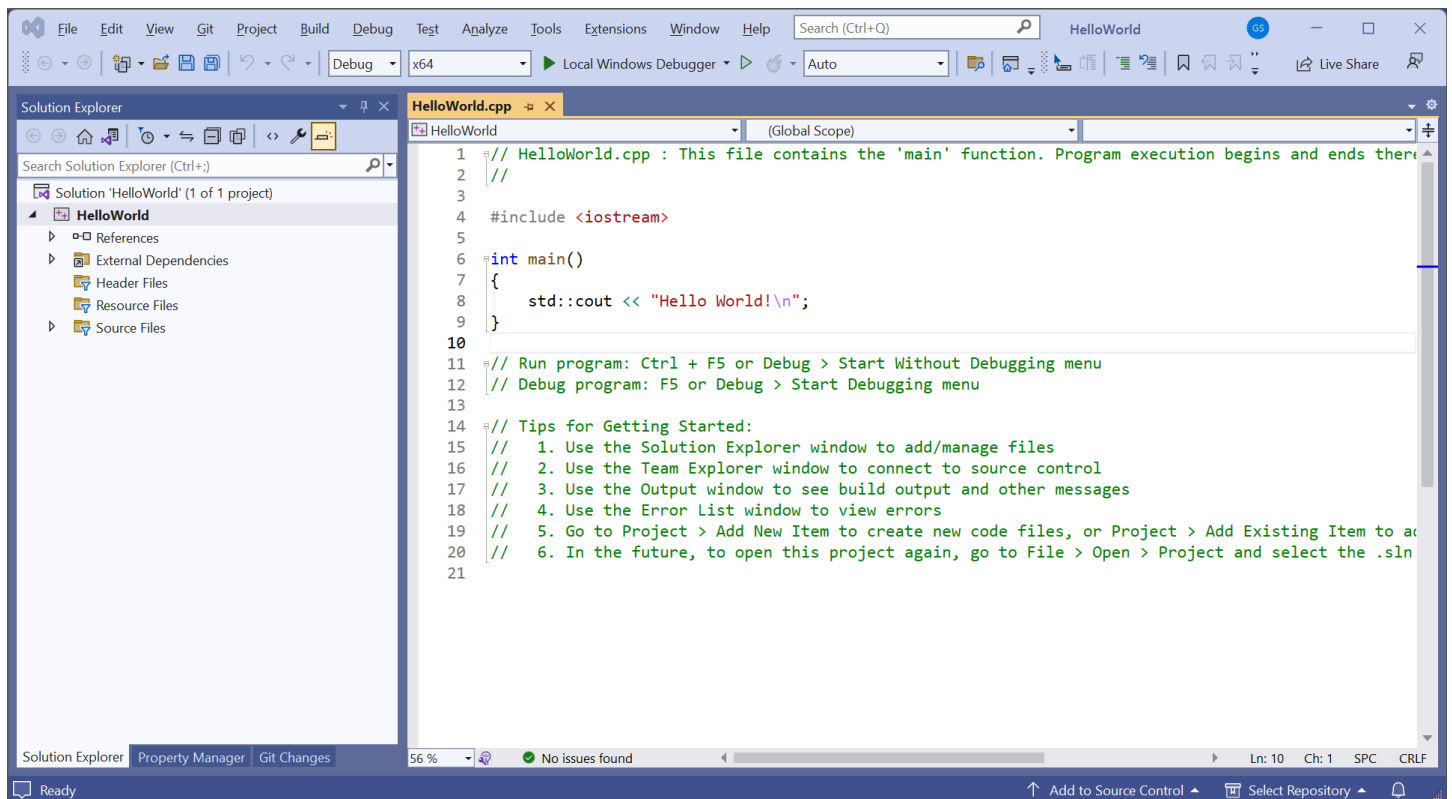
You'll now see the project configuration window.

You can change the name of the project to HelloWorld, and the location to something more suitable if you wish.

Click **Create**.



The IDE will now load with your project and present the automatically generated test program for C++ (its Hello, World).



Building and running the project is as simple as clicking either run button (▶ or ▶▶).

If all is successful, you should see:

```
Microsoft Visual Studio Debu  x + v
Hello World!
C:\Users\gsantor\source\repos\HelloWorld\x64\Debug\HelloWorld.exe (process 19752) exited with code 0.
Press any key to close this window . . .
```

Command-line compiling.

Microsoft Visual Studio provides command-line access to its tool chain via a set of *batch* files. There are several batch files, each configuring the environment for a different target space.

Examples:

File	Environment
vcvars32.bat	32-bit Intel-compatible
vcvars64.bat	64-bit Intel-compatible
vcvarsall.bat	All environments
vcvarsamd64.bat	64-bit AMD-compatible
vcvarsamd64_arm.bat	AMD to ARM cross-compile environment
vcvarsamd64_arm64.bat	AMD to ARM64 cross-compile environment
vcvarsamd64_x86.bat	AMD to Intel 32-bit cross-compile environment
vcvarsx86_amd64.bat	32-bit Intel-compatible to AMD64 cross-compile environment
vcvarsx86_amd.bat	32-bit Intel-compatible to ARM cross-compile environment
vcvarsx86_arm64.bat	32-bit Intel-compatible to ARM64 cross-compile environment

I've found it easiest to add a path to the directory containing the batch files. The programmer selects the build environment, then operates from the command line.

Setting the environment

Adding to the PATH

We'll use control panel to modify the path.

1. Locate the folder containing the file `vcvars64.bat`. They are typically found in the folder `C:\Program Files\Microsoft Visual Studio\2022\Community\VC\Auxiliary\Build`.

- 1.1. Copy the path

2. Launch the Systems Settings app:

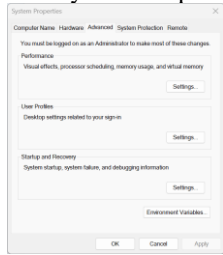
- 2.1. Right click the start button



- 2.2. Select: System

3. Click: Advanced System Settings

3.1. The System Properties dialog will be presented



4. Click: Environment Variables...
5. From the "System Variables" box, select: Path
6. Click: Edit...
7. Click: New
8. Enter³: C:\Program Files\Microsoft Visual Studio\2022\Community\VC\Auxiliary\Build
9. Click: OK
10. Click: OK

Testing the Environment

1. Launch the CLI shell (command-line interface operating system shell): `cmd.exe`
 - 1.1. Type: `⌘+R`
 - 1.2. Type: `cmd``↵`
2. Type: `vcvars64``↵`

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.19041.804]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\gsantor>vcvars64
*****
** Visual Studio 2019 Developer Command Prompt v16.9.1
** Copyright (c) 2021 Microsoft Corporation
*****
[vcvarsall.bat] Environment initialized for: 'x64'

C:\Users\gsantor>
```

Create a C++ source file

Create a text file containing:

```
#include <iostream>

int main() {
    std::cout << "Hello, world!\n";
}
```

Running the compiler

1. Launch the CLI shell in a folder with your `.cpp` files.
2. Type: `vcvars64``↵`
3. Type: `cl /std:c++latest *.cpp``↵`

³ You may have a slightly different path if you installed Visual Studio in a different location.


```
C:\Windows\System32\cmd.exe
D:\usr\fanshawe\in_class\05>vcvars64
*****
** Visual Studio 2019 Developer Command Prompt v16.9.1
** Copyright (c) 2021 Microsoft Corporation
*****
[vcvarsall.bat] Environment initialized for: 'x64'

D:\usr\fanshawe\in_class\05>cl /std:c++17 /EHsc *.cpp
Microsoft (R) C/C++ Optimizing Compiler Version 19.28.29912 for x64
Copyright (C) Microsoft Corporation. All rights reserved.

cppdir.cpp
Microsoft (R) Incremental Linker Version 14.28.29912.0
Copyright (C) Microsoft Corporation. All rights reserved.

/out:cppdir.exe
cppdir.obj

D:\usr\fanshawe\in_class\05>
```

4. Verify the output files. Type: dir↵

References

- [Microsoft Visual C++ on Wikipedia](#)
- [Microsoft Visual Studio on Wikipedia](#)

Document History

Version	Date	Activity
0.0.0	2021-06-27	Document created (for Visual Studio 2019)
1.0.0	2023-01-02	Updated for Visual Studio Community 2022
1.1.0	2023-08-26	Added command-line configuration.