

Forum:[Hack Forums](#)

Thread: [Visual Studio vs CLI](#)

Hackers - March 18, 2024, 7:54 am

Do you prefer Visual Studio (or similar) when it comes to software dev? Or do you prefer CLI?

I have started learning C++ a few months ago and mostly use manual CLI for g++ / gcc / etc compile commands.

However, this is tedious and can be very annoying if you don't document your commands (and need them later). Visual Studio is something I've been using more over the last few weeks and it's great for having requirements/deps/build configs stored, and nice for pulling in dependencies easier.

I think once I am more comfortable in VS I'll probably never go back to manual compiling with CLI, but I don't regret it. Doing it the manual way has taught me a lot more about how things interact, how static / dynamic linking works, and how to properly construct a project. I think if I had started on VS I would've used a lot more default settings and not been exposed to as much about C++ development.

So for me, I think:

- start with manual compiling and CLI to start, to learn things - even if frustrating
- use Visual Studio once you understand how to do it manually

ttbr0 - March 18, 2024, 11:56 pm

agreed. learning to compile programs manually is definitely something that is good to know at the beginning to understand how compiler options work, or even to get experience with using build systems. though i think developing windows programs with visual studio is great because intellisense is really helpful for clearing up my mental capacity to learn other things rather than remembering 200+ macros and function arguments. i never use anything but vs and its ctrl+b build hotkey for windows development.

DarkVision - March 19, 2024, 8:33 am

At work, i am forced to use the wizard. For private projects, i prefer to build with the command line and Notepad++.

The latter preference is mainly based on an inherent dislike for the additional files produced in a VS project.

Euaggelistes - March 19, 2024, 8:54 am

Knowing both is very advantageous. For complex projects, unless you create a compile script of some sort, the IDE is your friend so you don't forget any flags and steps.

I love the CLI, but hear ya on the pain at times.

Fludems - March 19, 2024, 2:02 pm

I'd try JetBrains CLion instead of VSCode.

I recently switched to PHPStorm(another Jet Brains IDE) from VSCode, and it was far more intelligent than VSCode.

IntelliJ by JetBrains has always been excellent too.

With CLion, you can probably automate half the shit you've been doing, but learning how to do it manually was important and wasn't wasted time.

Sajuuk - March 20, 2024, 7:31 am

I agree that compiling from command line from the get go is the most beneficial in the long run. I still compile small side projects via command line, but linking many libraries at a time makes the process tedious enough for me to use an IDE. VS Code has made the process of using command line within an IDE so much more streamlined and, while I do appreciate that, I dislike how bloated it's become. Unsurprisingly much in the same way the entire Visual Studio product line has.

Bearing in mind situations may arise where command line might be needed, we may as well learn at least the basics of how to use our compiler/linker/etc from the console sooner rather than later.

If anyone's curious, this is what I use to compile via command line from within both Notepad++ and VS Code:

PHP Code

```
<?php
@echo off
cd %~dp0
REM // change location to match your install path of "vcvarsall.bat"
REM // Also, ensure the install path of the VS Build Tools has no spaces in it.
REM // I changed mine from "Microsoft Visual Studio" to "vsbuildtools".

REM call Developer Prompt for Win64 native
call C:\vsbuildtools\2019\BuildTools\VC\Auxiliary\Build\vcvarsall.bat x64
REM compile with extended exception handling enabled
REM output executable with same name
cl /Fe"%~n1.exe" /EHsc %1
REM // delete object file; omit the command below if you need it
del %~n1.obj
exit
```

webdev77 - March 20, 2024, 2:54 pm

(Mar 20, 2024 07:31 AM)Sajuuk Wrote: [\[CITE\]](#) I agree that compiling from command line from the get go is the most beneficial in the long run. I still compile small side projects via command line, but linking many libraries at a time makes the process tedious enough for me to use an IDE. VS Code has made the process of using command line within an IDE so much more streamlined and, while I do appreciate that, I dislike how bloated it's become. Unsurprisingly much in the same way the entire Visual Studio product line has.

Bearing in mind situations may arise where command line might be needed, we may as well learn at least the basics of how to use our compiler/linker/etc from the console sooner rather than later.

IDEs offer debugger integration, which is something you cannot live without for things beyond simple projects. Well, I guess could, but shouldn't. VS Community Edition is a great tool and I prefer it over VS Code, although VS Code is truly free for everyone so there's that.

Fludems - March 20, 2024, 6:27 pm

(Mar 20, 2024 02:54 PM)webdev77 Wrote: [\[CITE\]](#) IDEs offer debugger integration, which is

something you cannot live without for things beyond simple projects. Well, I guess could, but shouldn't. VS Community Edition is a great tool and I prefer it over VS Code, although VS Code is truly free for everyone so there's that.

With NeoVIM and the likes you obviously can be really productive and efficient. But to get to that point (at least for me) would take an extremely long time and I don't think I value the time investment enough to use something like that. I'd obviously be slower than I am currently in an IDE for a decent amount of time too, so the cost there would be quite great.

I say that because I just wanted to make the point that. With an IDE it doesn't take too long to become a way more efficient developer than you probably are in just the CLI. Refactoring, code completion, live templates, automations and a whole load of other shit that near enough works just out of the box is a god send.

I also see people install an IDE change a couple of things and leave it at that. Almost using it like it wasn't much different to a text editor. If you haven't you should absolutely invest time into configuring your IDE and looking at how other people use theirs. If you don't you're leaving so much on the table, and you'll actually have a better experience as a developer.

One quick tip I'd suggest is:

If you're routinely doing things that reduces time spent on actually writing code, try and figure out if there's a way you can reduce that.

For example: If you are often looking at your codebase's folder structure in the left hand side of VSCode to get to the file you want to edit. You'll probably save an hour of your time within a week(as fulltime dev) if you just started pressing ctrl/cmd P and typing in the file name

Sajuuk - March 20, 2024, 6:56 pm

(Mar 20, 2024 02:54 PM)webdev77 Wrote: [\[CITE\]](#) IDEs offer debugger integration, which is something you cannot live without for things beyond simple projects. Well, I guess could, but shouldn't. VS Community Edition is a great tool and I prefer it over VS Code, although VS Code is truly free for everyone so there's that.

Those small projects I mentioned usually don't require a whole lot of debugging. Or, at least the ones I've cooked up didn't.

Having integrated debugging definitely makes the entire development process more streamlined, but learning how to do it via command line first definitely would've given me a better understanding of what exactly is happening to the code I write.

I didn't learn how to compile/link/debug via command line until a few years after beginning to learn C++. Researching that led me to these videos:

<https://youtu.be/PzsXHGVTScl>

<https://youtu.be/N2y6csonlI4>

My interest was piqued, and the rest is history. Oh, how desperately I wish I'd have learned all that way back when, believe me. It'd be well worth the time spent early on to jump into the deep end with a more solid foundation. IDEs all the way from there on out.

And yes, VS Community is a solid choice as well. Only reason I use VS Code is because I've been out of the programming loop for a while now and it provides just enough flexibility for what I do today. No pun intended.

Fludems - March 20, 2024, 7:49 pm

(Mar 20, 2024 06:56 PM)Sajuuk Wrote: [\[CITE\]](#) Those small projects I mentioned usually don't require a whole lot of debugging. Or, at least the ones I've cooked up didn't.

Having integrated debugging definitely makes the entire development process more streamlined, but learning how to do it via command line first definitely would've given me a better understanding of what exactly is happening to the code I write.

I didn't learn how to compile/link/debug via command line until a few years after beginning to learn C++. Researching that led me to these videos:

<https://youtu.be/PzsXHGVTScl>
<https://youtu.be/N2y6csonlI4>

My interest was piqued, and the rest is history. Oh, how desperately I wish I'd have learned all that way back when, believe me. It'd be well worth the time spent early on to jump into the deep end with a more solid foundation. IDEs all the way from there on out.

And yes, VS Community is a solid choice as well. Only reason I use VS Code is because I've been out of the programming loop for a while now and it provides just enough flexibility for what I do today. No pun intended.

Honestly, try one of JetBrains IDE's if you get the chance.

Hackers - March 22, 2024, 4:35 am

(Mar 20, 2024 06:56 PM)Sajuuk Wrote: [\[CITE\]](#) Those small projects I mentioned usually don't require a whole lot of debugging. Or, at least the ones I've cooked up didn't.

Having integrated debugging definitely makes the entire development process more streamlined, but learning how to do it via command line first definitely would've given me a better understanding of what exactly is happening to the code I write.

I didn't learn how to compile/link/debug via command line until a few years after beginning to learn C++. Researching that led me to these videos:

<https://youtu.be/PzsXHGVTScl>
<https://youtu.be/N2y6csonlI4>

My interest was piqued, and the rest is history. Oh, how desperately I wish I'd have learned all that way back when, believe me. It'd be well worth the time spent early on to jump into the deep end with a more solid foundation. IDEs all the way from there on out.

And yes, VS Community is a solid choice as well. Only reason I use VS Code is because I've been out of the programming loop for a while now and it provides just enough flexibility for what I do today. No pun intended.

Yeah for small software I think manually compiling is fine. I haven't had too many issues doing it this way, it's just more tedious but overall it's not that difficult to just manually compile for small things. However, using a hotkey/button to build 3+ files, generate resource files, and provide proper error messages is a huge benefit of Visual Studio, so I'd like to transition off of manual command line. I will still do some manually, but you guys will probably start to see me sharing Visual Studio configs, or cmake files more often rather than (what I've done so far) g++ commands being used manually.

I will still use VS Code for projects on the web, but for software dev I want to try to stick to Visual

Studio as much as I can until I feel more comfortable in there. Productivity should increase that way and it will be easier for others to clone my projects.

Sajuuk - March 22, 2024, 8:08 am

(Mar 22, 2024 04:35 AM)nodes Wrote: [\[CITE\]](#) Yeah for small software I think manually compiling is fine. I haven't had too many issues doing it this way, it's just more tedious but overall it's not that difficult to just manually compile for small things. However, using a hotkey/button to build 3+ files, generate resource files, and provide proper error messages is a huge benefit of Visual Studio, so I'd like to transition off of manual command line. I will still do some manually, but you guys will probably start to see me sharing Visual Studio configs, or cmake files more often rather than (what I've done so far) g++ commands being used manually.

I will still use VS Code for projects on the web, but for software dev I want to try to stick to Visual Studio as much as I can until I feel more comfortable in there. Productivity should increase that way and it will be easier for others to clone my projects.

CMake, solutions, etc all exist for a reason. Plus the fact that even CodeBlocks can import MSVC projects and solutions. All for ease of use. And time is money, after all. If I have to link/include anything beyond the stdlibs I do everything from within an IDE. Knowing ahead of time what exactly is being done behind closed doors will come in handy sooner or later.

I was once all about doing everything via command line, but then I had to learn how to shoehorn in resources via windres/rc. Learning how that process works was so tedious that I just chalked it up to the straw that broke the camel's back, but it surely made me incredibly grateful for IDEs that do all that (and more) for us.

I must admit I've grown lazier over the years. Drag-n-drop Batch scripts work, but being able to press F9 and have the finished product in a few seconds is a luxury too cheap to pass up. :)

PulseLink - April 9, 2024, 4:09 pm

after using visual studio for awhile now, i honestly prefer nano and gcc and a simple windows test environment, its the best way to get constant compilations and learning the gcc commands was very helpful for me to compile and understand things

Hackers - April 10, 2024, 5:49 am

(Apr 9, 2024 04:09 PM)PulseLink Wrote: [\[CITE\]](#) after using visual studio for awhile now, i honestly prefer nano and gcc and a simple windows test environment, its the best way to get constant compilations and learning the gcc commands was very helpful for me to compile and understand things

I now have a lot more hours in Visual Studio since I made this thread and I am loving it. The pre-build steps, post-build, multiple program builds, etc are very useful. I can't see myself going back to CLI with g++ compiling now that I've gotten very comfortable in VS. However, I wouldn't change a thing about how I started - I would have no idea what's going on in visual studio project/solution settings if I hadn't spent so much time doing the compiling manually. I recommend that anyone learning C/C++ starts with a manual compiler & Hello World rather than cloning a git repo and letting visual studio (or another) take care of everything.

Novo - December 11, 2024, 4:31 am

This depends; for more customization, sometimes you have to go with manual, but I see no reason to not compile through VS. I used to use VS to compile C# code, but when that didn't work on Linux, as VS does not know Linux, I switched to using the cli. When I used Windows again, I did use CLI as I got used to it. VS for debugging, etc., but I do use VS for compiling sometimes as I am too lazy.

Hackers - December 11, 2024, 7:20 am

(Dec 11, 2024 05:31 AM)NovoDev Wrote: [\[CITE\]](#) This depends; for more customization, sometimes you have to go with manual, but I see no reason to not compile through VS. I used to use VS to compile C# code, but when that didn't work on Linux, as VS does not know Linux, I switched to using the cli. When I used Windows again, I did use CLI as I got used to it. VS for debugging, etc., but I do use VS for compiling sometimes as I am too lazy.

I'm not sure about Visual Studio, but I know that VS Code does have a WSL mode you can use. I'll have to look and see what can be done about regular VS though - not sure.

Novo - December 11, 2024, 7:30 am

(Dec 11, 2024 08:20 AM)nodes Wrote: [\[CITE\]](#) I'm not sure about Visual Studio, but I know that VS Code does have a WSL mode you can use.

I'll have to look and see what can be done about regular VS though - not sure.

Never used WSL mode in vsc

Hackers - December 11, 2024, 7:33 am

(Dec 11, 2024 08:30 AM)NovoDev Wrote: [\[CITE\]](#) Never used WSL mode in vsc

It's not bad - if you go into your specific wsl folder instead of the Windows wrapper for it, it's basically as fast as a real Linux OS for file writing/read/etc.

I like it, makes it easier to share a git between a server, Mac users, and Windows. WSL takes care of any concerns over file ending symbols, case-sensitive/insensitive file naming schemes, etc, so it really normalizes things for teams, or local Windows development for a Linux server deployment.

Novo - December 11, 2024, 7:36 am

(Dec 11, 2024 08:33 AM)nodes Wrote: [\[CITE\]](#) It's not bad - if you go into your specific wsl folder instead of the Windows wrapper for it, it's basically as fast as a real Linux OS for file writing/read/etc.

I like it, makes it easier to share a git between a server, Mac users, and Windows. WSL takes care of any concerns over file ending symbols, case-sensitive/insensitive file naming schemes, etc, so it really normalizes things for teams, or local Windows development for a Linux server deployment.

Holy shit, that's dope! As a system administrator myself, using WSL generally in windows just makes life like 50000x easier.

Hackers - December 11, 2024, 7:45 am

(Dec 11, 2024 08:36 AM)NovoDev Wrote: [\[CITE\]](#) Holy shit, that's dope! As a system administrator myself, using WSL generally in windows just makes life like 50000x easier.

yeah exactly, sometimes I don't use it for personal projects, but for work I've been getting in a habit of hounding the Windows devs to just save themselves the trouble & spend the extra 2 minutes at the start to set up a WSL folder if needed for the project instead.

Novo - December 11, 2024, 7:47 am

(Dec 11, 2024 08:45 AM)nodes Wrote: [\[CITE\]](#) yeah exactly, sometimes I don't use it for personal projects, but for work I've been getting in a habit of hounding the Windows devs to just save themselves the trouble & spend the extra 2 minutes at the start to set up a WSL folder if needed for the project instead.

That completely makes sense, i mostly use Linux, but when i do use windows on the side, i always love using WSL for file management.

also, if you like linux commands, you can even add the git binaries folder to your PATH on windows and get commands like: vim, nano, cat, ls, echo (unix), and more!

padrino - December 11, 2024, 4:11 pm

As someone who has worked as a software engineer for 7 years at this point I'm always going for Visual Studio. It's great to know how the compile through the command line but man, a good IDE makes life so much easier. Also, Intellisense has gotten crazy good in the past few years so it

actually saves a lot of time. thats just me tho.