

# Visual Studio Code vs. Visual Studio: How to choose

**Deciding between Visual Studio Code and Visual Studio may depend as much on your work style as on the language support and features you need. Here's how to decide.**

For decades, when I got to work in the morning, I would start Microsoft Visual Studio (or one of its predecessors, such as Visual C++ or Visual InterDev), then brew tea and possibly attend a morning meeting while it went through its laborious startup. I would keep the IDE open all day as I went through develop/test/debug cycles to avoid another startup delay. When I worked on a C++ project with ~2 million lines of code, I also jump-started each day's work by automatically running a batch script that did a code checkout and full rebuild of the product in the wee hours.

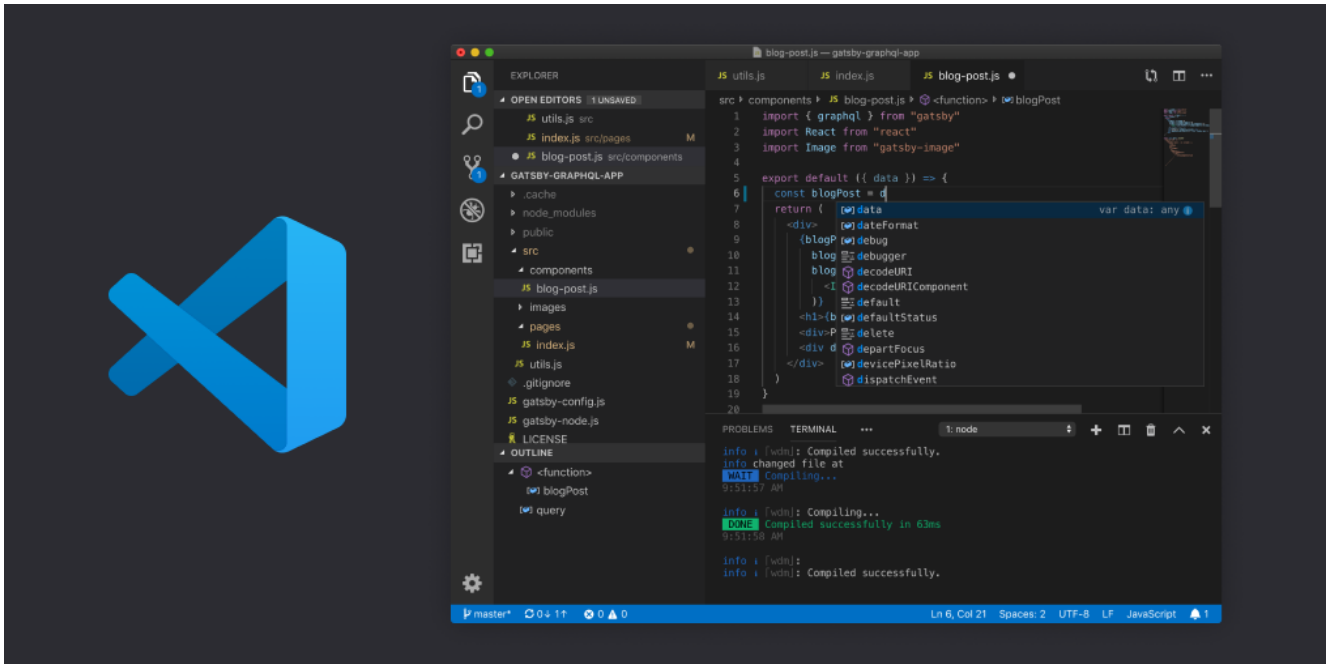


These days, I *don't* feel the need to open my code projects first thing every morning, or to keep them open all day. Visual Studio Code usually starts up quickly enough that I can be productive in a few minutes, even for large projects. I said usually, not always: Visual Studio Code itself needs a monthly update, and the many extensions I have installed often need their own updates. Still, even updating a dozen extensions in Visual Studio Code takes much less time than Visual Studio takes to rebuild the symbol tables of a large C++ project.

Still, choosing between Visual Studio Code and Visual Studio is not as simple as choosing between lightweight editor and heavyweight IDE. While Visual Studio Code is highly configurable, Visual Studio is highly complete. Your choice may depend as much on your work style as on the language support and features you need. Let's take a look at the capabilities and the trade-offs of these two development tools.

## What is Visual Studio Code?

Visual Studio Code is a lightweight but powerful source code editor that runs on your desktop and is available for Windows, MacOS, and Linux. It comes with built-in support for JavaScript, TypeScript, and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C#, Java, Python, PHP, and Go) and runtimes (such as .Net and Unity).



Aside from the whole idea of being lightweight and starting quickly, VS Code has [IntelliSense](#) code completion for variables, methods, and imported modules; graphical debugging; linting, multi-cursor editing, parameter hints, and other powerful editing features; snazzy code navigation and refactoring; and built-in source code control including [Git](#) support. Much of this was adapted from Visual Studio technology.

VS Code proper is built using the [Electron shell](#), [Node.js](#), [TypeScript](#), and the [Language Server](#) protocol, and is updated on a monthly basis. The extensions are updated as often as needed. The richness of support varies across the different programming languages and their extensions, ranging from simple syntax highlighting and bracket matching to debugging and refactoring. You can add basic support for your favorite language through TextMate colorizers if no language server is available.

The code in the [Visual Studio Code repository](#) is open source under the MIT License. The VS Code product itself ships under a standard [Microsoft product license](#), as it has a small percentage of Microsoft-specific customizations. It's

free despite the commercial license.

## What is Visual Studio?

Visual Studio (current version Visual Studio 2019) is Microsoft's premier IDE for Windows and MacOS. With Visual Studio, you can develop, analyze, debug, test, collaborate, and deploy your software.

On Windows, Visual Studio 2019 currently has 17 workloads, which are consistent tool and component installation bundles for different development targets. Workloads are an important improvement to the Visual Studio installation process, because a full download and installation of Visual Studio 2019 can easily take hours and fill a disk, especially an SSD.

Visual Studio 2019 for Mac has a less complicated installer than the Windows version, mostly because it doesn't support as many targets. It allows you to develop for web, mobile, and desktop with .Net, with Unity, Azure, and Docker support included by default. The .Net Core, Android, iOS, and MacOS targets are optional; the latter three use Xamarin.

Visual Studio 2019 comes in three SKUs: Community (free, not supported for enterprise use), Professional (\$1,199 first year/\$799 renewal), and Enterprise (\$5,999 first year/\$2,569 renewal). The Enterprise Edition has features for architects, advanced debugging, and testing that the other two SKUs lack.

## Visual Studio or Visual Studio Code?

You would think that deciding between Visual Studio and Visual Studio Code for any given software development task would be as simple as deciding between an IDE and an editor.

It's not, mostly because VS Code can be configured to be quite close to an IDE for many programming languages. However, along this configurability come a number of trade-offs.

For example, if your development style is test-driven, Visual Studio will work right out of the box. On the other hand, there are some 15 test-driven development extensions for VS Code supporting Node.js, Go, .Net, and PHP. Similarly, Visual Studio does a good job working with databases, especially Microsoft SQL Server and its relatives, but VS Code has lots of database extensions. Visual Studio has great refactoring support, but Visual Studio Code implements the basic refactoring operations for half a dozen languages.

There are a few clear-cut cases. For instance, if you are a software architect and you have access to Visual Studio Enterprise, you'll want to use that. If you need to collaborate with team members on development or debugging, then Visual Studio is the better choice. If you need to do serious code analysis or performance profiling, or debug from a snapshot, then Visual Studio Enterprise will help you.

VS Code tends to be popular in the data science community. Nevertheless, Visual Studio 2019 has a data science workload that offers many features.

Visual Studio doesn't run on Linux; VS Code does. On the other hand, Visual Studio for Windows has a Linux/C++ workload and Azure support.

For daily bread-and-butter develop/test/debug cycles in the languages supported in both Visual Studio and VS Code, which you choose really does boil down to personal preference. If you tend to work on a development project for hours at a time, then Visual Studio might be a better fit. If you tend

to dip into development for brief periods and rotate between other tasks, then Visual Studio Code might make you happier.