

+ 25

What's New for Visual Studio Code

CMake Improvements and
GitHub Copilot Agents

ALEXANDRA KEMPER



20
25



Microsoft @ CppCon

Tues.	14:00	What's new in Visual Studio for C++ Developers in 2025	Augustin Popa & David Li
	14:00	Back to Basics: Code Review	Chandranath Bhattacharyya & Kathleen Baker
Wed.	14:00	LLMs in the Trenches: Boosting System Programming with AI	Ion Todirel
	15:15	C++ Performance Tips: Cutting Down on Unnecessary Objects	Kathleen Baker & Prithvi Okade
	15:50	Connecting C++ Tools to AI Agents Using the Model Context Protocol	Ben McMorran
		Welcome to v1.0 of the meta::[[verse]]!	Inbal Levi
Thurs.	14:00	MSVC C++ Dynamic Debugging: How We Enabled Full Debuggability of Optimized Code	Eric Brumer
	16:45	It's Dangerous to Go Alone: A Game Developer Tutorial	Michael Price
Fri	9:00	Reflection-based JSON in C++ at Gigabytes per Second	Daniel Lemire & Francisco Geiman Thiesen
	13:30	Duck-Tape Chronicles: Rust/C++ Interop	Victor Ciura

Agenda



1. Faster Performance for C/C++ Extension



2. C++ Tooling updates

Updating the build scripts in the VCMI repo



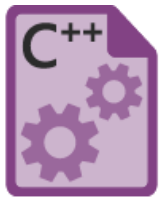
3. AI Agents

Building a C++ proof of concept from scratch



4. Other VS Code updates

Extensions for C++ Development in VS Code



Makefile Tools

pre-release

Preview

Microsoft microsoft.com | 7,907,736 | ★★☆☆☆ (39)

Provide makefile support in VS Code: C/C++ IntelliSense, build, debug/run.

Disable

Uninstall

Switch to Release Version



Auto Update



C/C++

pre-release

Microsoft microsoft.com | 86,268,727 | ★★★★★ (588)

C/C++ IntelliSense, debugging, and code browsing.

Disable

Uninstall

Switch to Release Version



Auto Update



CMake Tools

pre-release

Microsoft microsoft.com | 49,040,256 | ★★★★★ (74)

Extended CMake support in Visual Studio Code

Disable

Uninstall

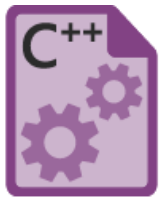
Switch to Release Version



Auto Update



Extensions for C++ Development in VS Code



Makefile Tools

pre-release

Preview

Microsoft [microsoft.com](#) | 7,907,736 | ★★☆☆☆ (39)

Provide makefile support in VS Code: C/C++ IntelliSense, build, debug/run.

Disable

Uninstall

Switch to Release Version



Auto Update



C/C++

pre-release

Microsoft [microsoft.com](#) | 86,268,727 | ★★★★★ (588)

C/C++ IntelliSense, debugging, and code browsing.

Disable

Uninstall

Switch to Release Version



Auto Update



CMake Tools

pre-release

Microsoft [microsoft.com](#) | 49,040,256 | ★★★★★ (74)

Extended CMake support in Visual Studio Code

Disable

Uninstall

Switch to Release Version



Auto Update



GitHub Copilot Chat

GitHub [github.com](#) | 38,524,558 | ★★★★★ (161)

AI chat features powered by Copilot

Disable

Uninstall



Auto Update



GitHub Copilot

pre-release

GitHub [github.com](#) | 48,044,345 | ★★★★★ (994)

Your AI pair programmer

Disable

Uninstall

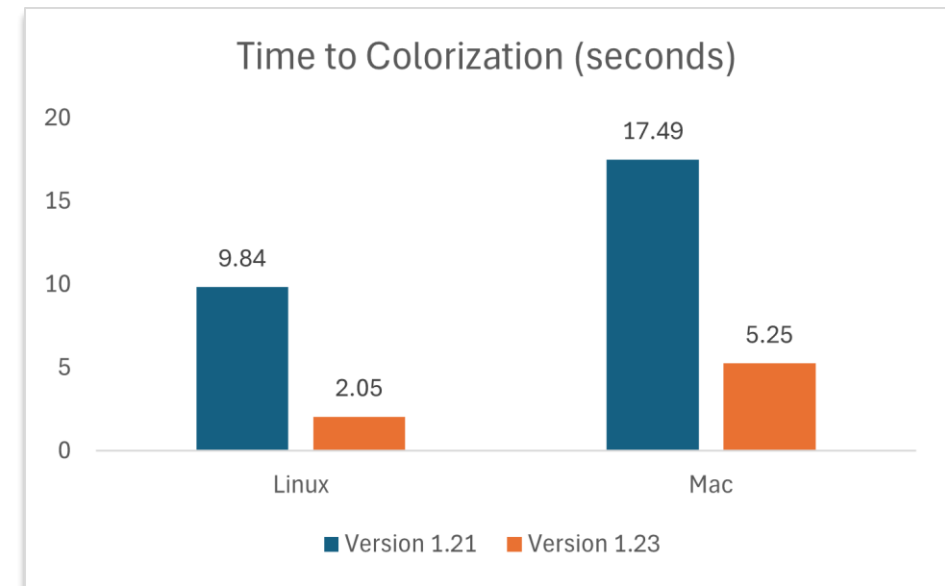
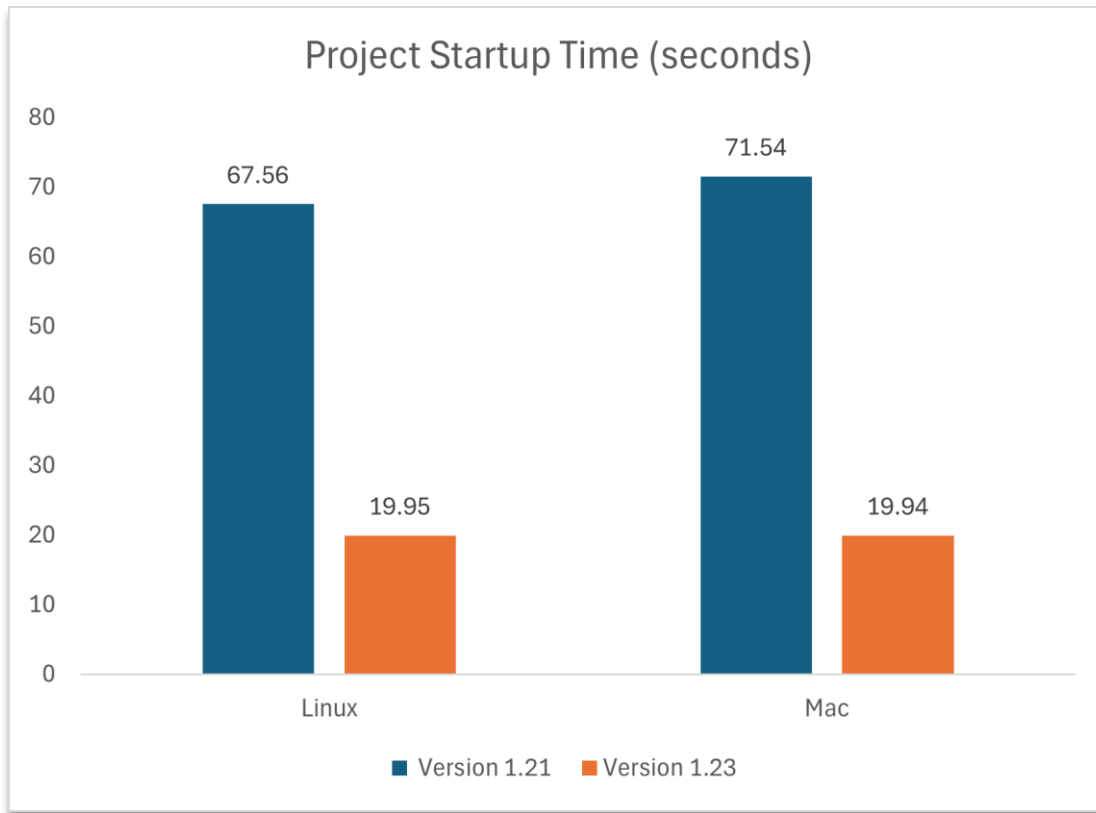


Auto Update



Performance Gains for large C++ codebases

Faster Colorization & Project Startup



But how?

- Configuration caching
- Better compile_commands.json support, custom config handling, file discovery...

Learn More at aka.ms/cpptools-perf

Performance: Customize recursive #includes

Default:

```
"includePath": [  
    "${workspaceFolder}/**"  
],
```

GameEngine.cpp

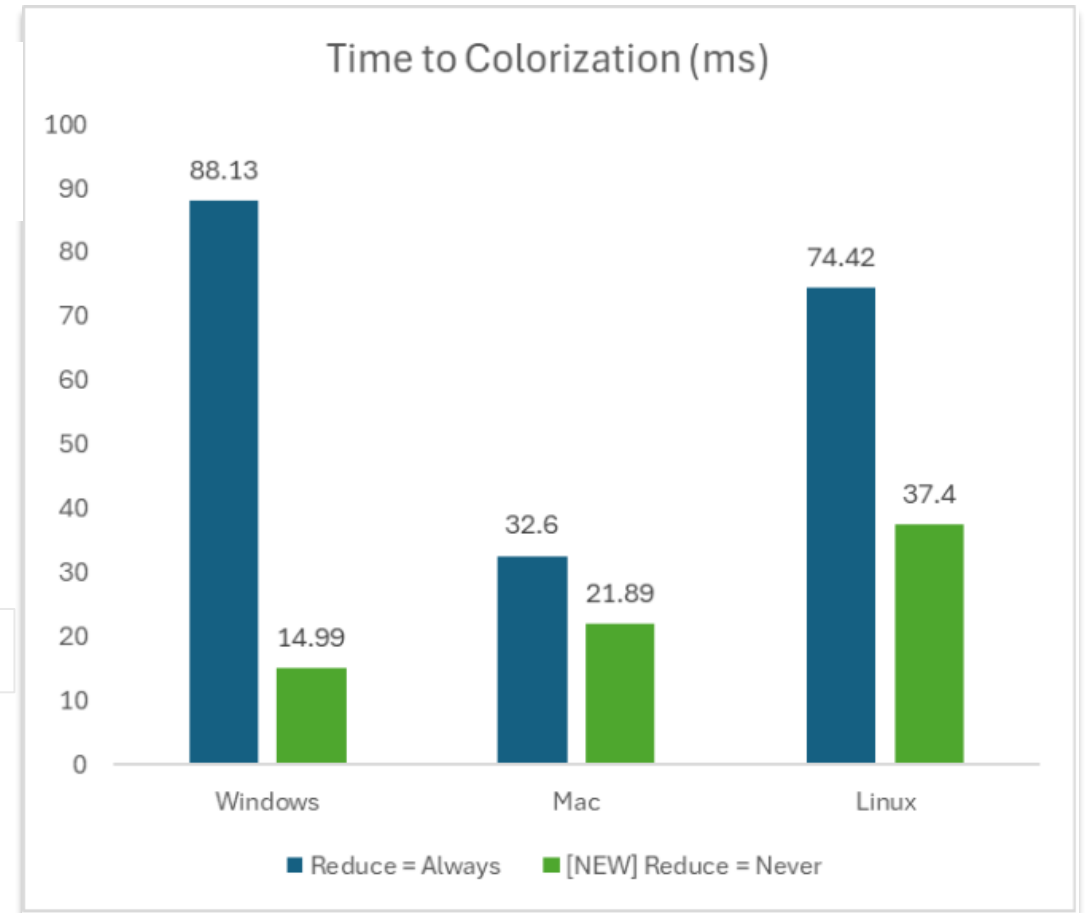
```
#include "render/Canvas.h"  
#include "render/Colors.h"  
#include "render/IFont.h"  
#include "render/EFont.h"  
#include "renderSDL/ScreenHandler.h"  
#include "renderSDL/RenderHandler.h"
```

```
#include "../gui/TextAlignment.h"  
#include "../../lib/Rect.h"  
#include "../../lib/Color.h"
```

```
#include "../../lib/Color.h"
```

```
#include "../../lib/Point.h"  
#include "../render/IScreenHandler.h"
```

```
#include "../render/IRenderHandler.h"
```

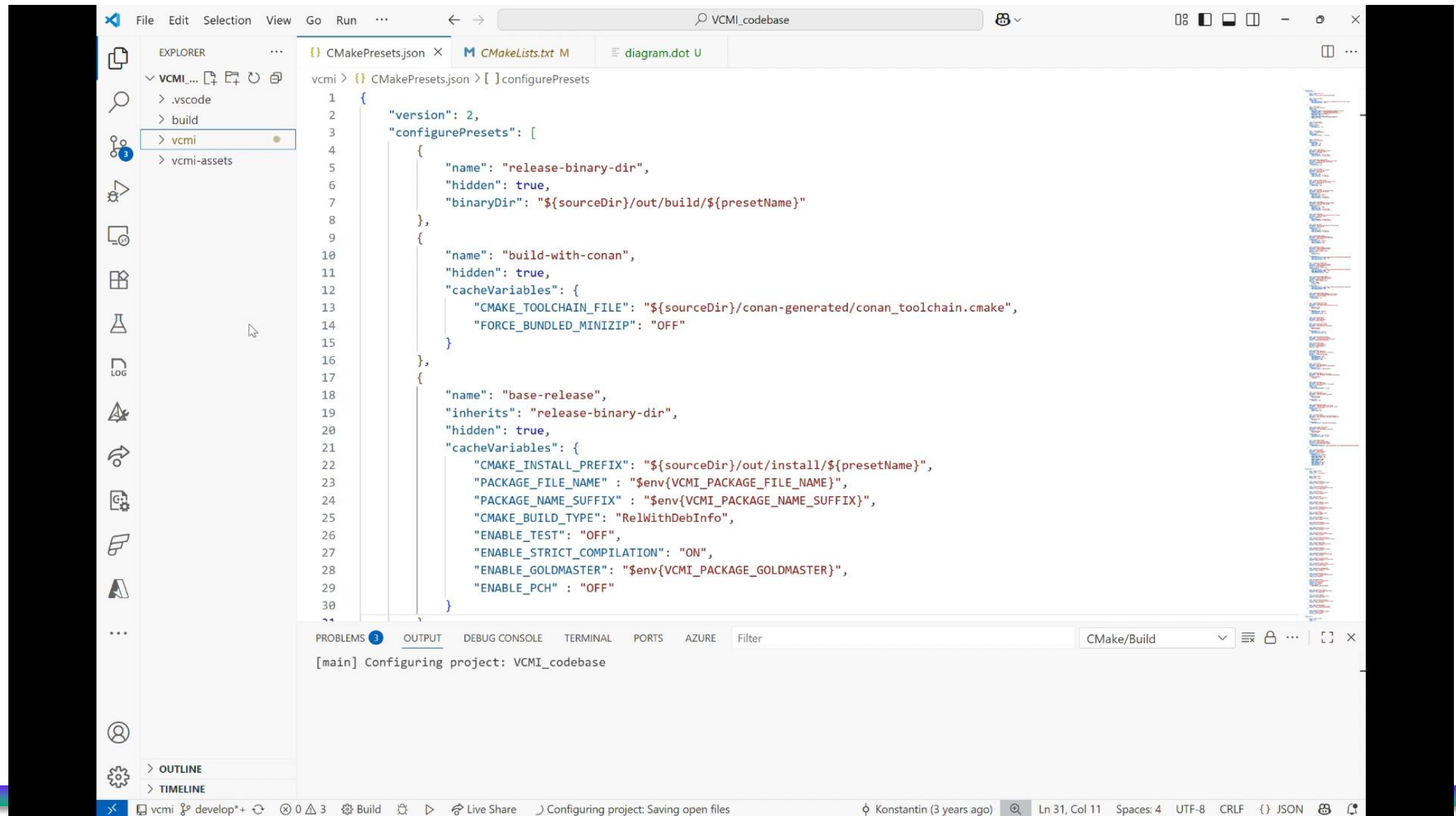


Learn More at aka.ms/cpptools-1.25

Demo: Updating your build process with the CMake Tools Extension

[vcmi/vcmi: Open-source engine for Heroes of Might and Magic III](#)

Backup Demo Recording



Recap: Modernizing your Cmake Scripts

CMake Language Services – now provided by CMake Tools extension

```
set(CMAKE_VS_GLOBALS
```

Add compile definitions to a target.

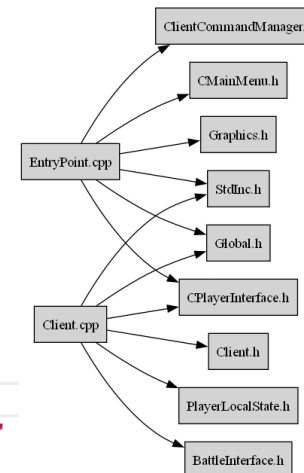
```
target_compile_definitions(<target> <INTERFACE|PUBLIC|PRIVATE> [items1...]  
[<INTERFACE|PUBLIC|PRIVATE> [items2...] ...])  
target_compile_definitions(foo PUBLIC FOO) target_compile_definitions(foo PUBLIC -DFOO)  
target_compile_definitions(foo PUBLIC "" FOO) target_compile_definitions(foo PUBLIC -D FOO)  
target_compile_definitions(foo PUBLIC FOO=1)  
target_compile_definitions(VCMI PRIVATE
```

Support through CMake Presets v10

```
{ } CMakePresets.json M ●
```

```
{ } CMakePresets.json > [ ] configurePresets
```

```
1 {  
2   "version": 10,  
3   "configurePresets": [  
4     {  
       "name": "macos-ninja-release",  
       "displayName": "Ninja release",  
       "description": "VCMI MacOS Ninja",  
       "inherits": "default-release",  
       "$comment": "MacOS build with Ninja for better performance"
```



Recap: Improvements to code navigation + understanding

Multi-root configurability for CMake



Cmake: Exclude

The extension will ignore the folders listed in this setting. The folders should be listed as absolute paths.

Copilot Hover

```
class CStack
```

Represents STACK_BATTLE nodes

✦ Copilot

CStack is a final class representing a stack of units in a battle, inheriting from CBonusSystemNode, battle::CUnitState, and battle::IUnitEnvironment. It encapsulates properties such as unique ID, unit type, owner, position, and state, and provides functionality for serialization, interaction with battle mechanics, and unit-specific behaviors like healing, terrain interaction, and spell effects.

```
ds(const BattleID & battleID, const CStack * stack, BattleHexArray hexes) const
```

Updates to C++ Completions: more **context-aware**

- Next-Edit Suggestions predict where and what you need to update next

GitHub > Copilot > Next Edit Suggestions: Enabled

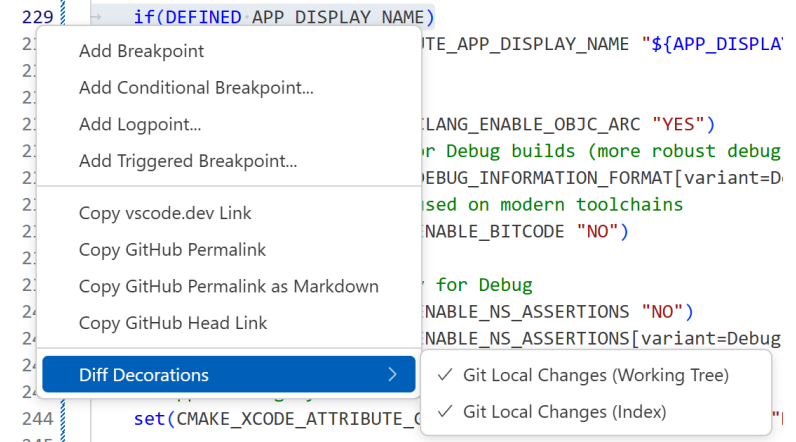


Whether to enable next edit suggestions (NES).

NES can propose a next edit based on your recent changes. [Learn more](#) about next edit suggestions.

- New, updated models trained on more C++
- Contextual Improvements tailored to C++ coding structure

View your Git staged commits from editor



But what about building from scratch?

CHAT

What steps do I need to take to build the game snake?

CMakePresets.json

Here are the steps to build the game Snake as a proof of concept in your CMake-based project (using C++ and SDL2, which you already have as a dependency):

- 1. Create a New Source File**
 - Add a new file, e.g., `snake_game.cpp`, in your project (e.g., in the `vcmi` or `client` directory).
- 2. Implement the Snake Game Logic**
 - In `snake_game.cpp`, implement a minimal Snake game using SDL2 for windowing, input, and rendering.
- 3. Add a New Executable Target in CMake**
 - In your `CMakeLists.txt`, add:

```
add_executable(snake_game snake_game.cpp)
target_link_libraries(snake_game PRIVATE SDL2::SDL2)
```
 - Place this after SDL2 is found in your CMake file.
- 4. Build the Project**
 - Use your existing CMake presets or build system:
 - Configure:

```
cmake -S . -B build -G "Ninja"
```
 - Build:

```
cmake --build build --target snake_game
```
- 5. Run the Game**
 - Run the resulting executable (e.g., `build/snake_game.exe` on Windows).

Summary of steps:

- Add `snake_game.cpp` with game logic.



Add Context...

CMakePresets.json

+

Add context (#), extensions (@), commands (/)

Ask GPT-4.1

Codebase wide changes, powered by Agents

Intelligent systems that

- Accept natural language instead of step-by-step instructions
- Automate complex workflows
- Can invoke tools & commands (compilers, debuggers, etc.)
- Can edit across multiple files



“Add a thread-safe logging utility to this C++ project”

Demo: Building from Scratch with custom instructions

Backup recording

File Edit Selection View Go ...

DependencyVisualizer

EXPLORER

- DEPENDENCYVISUALIZER
 - .github
 - prompts
 - copilot-instructions.md
 - build

CHAT

Build with agent mode.

AI responses may be inaccurate.

If handling customer data, [disable telemetry](#).

PROBLEMS COMMENTS ...

Filter (e.g. text, author)

There are no comments in this workspace yet.

LOG

OUTLINE

TIMELINE

Visualizer 0 0 Build Live Share

Copilot Coding Agent



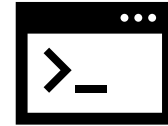
Async & in the cloud

- Integrated into GitHub
- Explores repo, writes code, runs tests, opens a pull request for review
- Works in the background

Example:

- Fix a bug in the codebase & create a relevant PR

Agent Mode in VS Code



Real-time & in VS Code

- Local & interactive for specific task that iterates until successful
- Agent edits your code & invokes tools
- Immediate feedback & control

Examples:

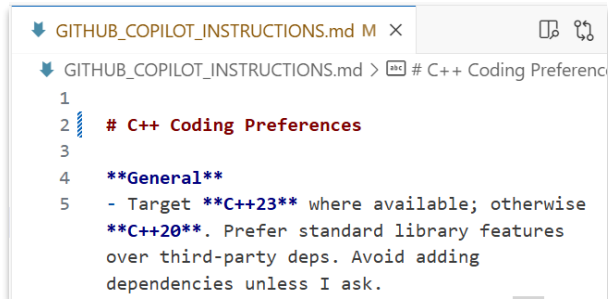
- Plan and implement new features
- Refactor parts of your codebase

Learn More at aka.ms/agent-mode & aka.ms/copilot-coding-agent

Recap from demo: Customize Copilot

Custom Instructions

Specify coding standards based on your preferences per workspace

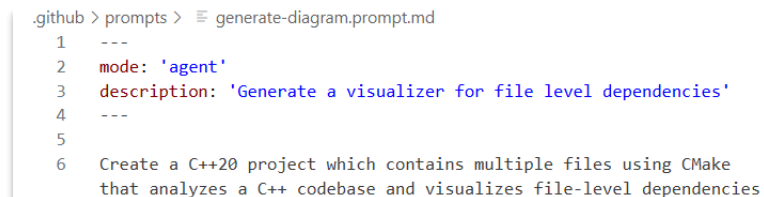


```
GITHUB_COPILOT_INSTRUCTIONS.md
# C++ Coding Preferences

**General**
- Target **C++23** where available; otherwise
  **C++20**. Prefer standard library features
  over third-party deps. Avoid adding
  dependencies unless I ask.
```

Prompt Files

define reusable prompts for common development tasks

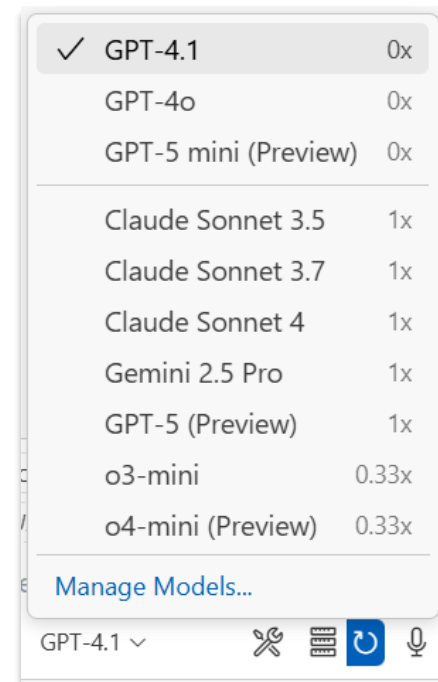


```
.github > prompts > generate-diagram.prompt.md
---
mode: 'agent'
description: 'Generate a visualizer for file level dependencies'
---

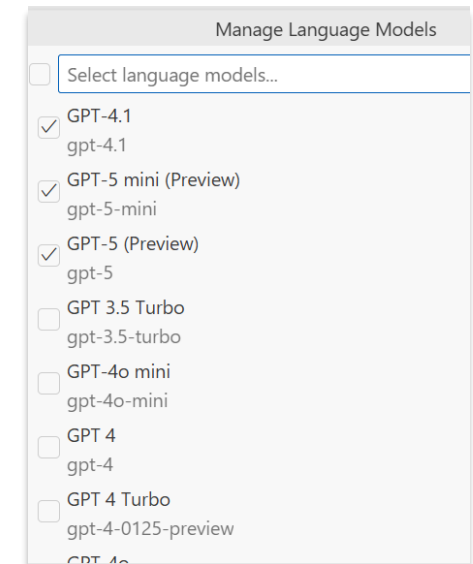
Create a C++20 project which contains multiple files using CMake
that analyzes a C++ codebase and visualizes file-level dependencies
```

Model Picker

Choose between a variety of different Models



Bring Your Own Key
Add additional models beyond built-in providers



Learn More at aka.ms/customize-copilot

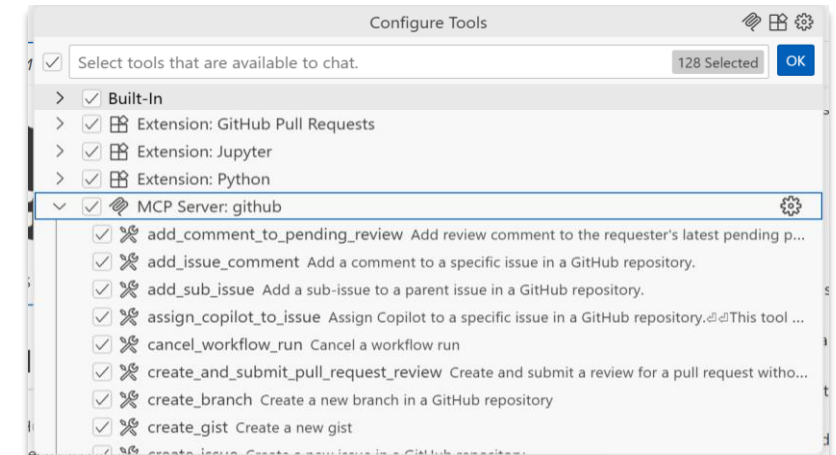
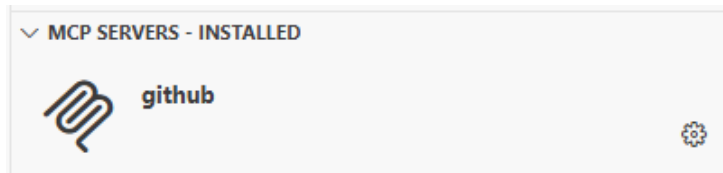
Extensibility for Agent: Tools + MCP

Tools extend the agent's capabilities beyond basic code generation, enabling it to interact with your environment, run commands, and more...

- Invoke manually and/or automatically by agent
- Agent supports tools provided by VS Code ecosystem (built-in and extensions) and MCP server tools
- Can enable/disable

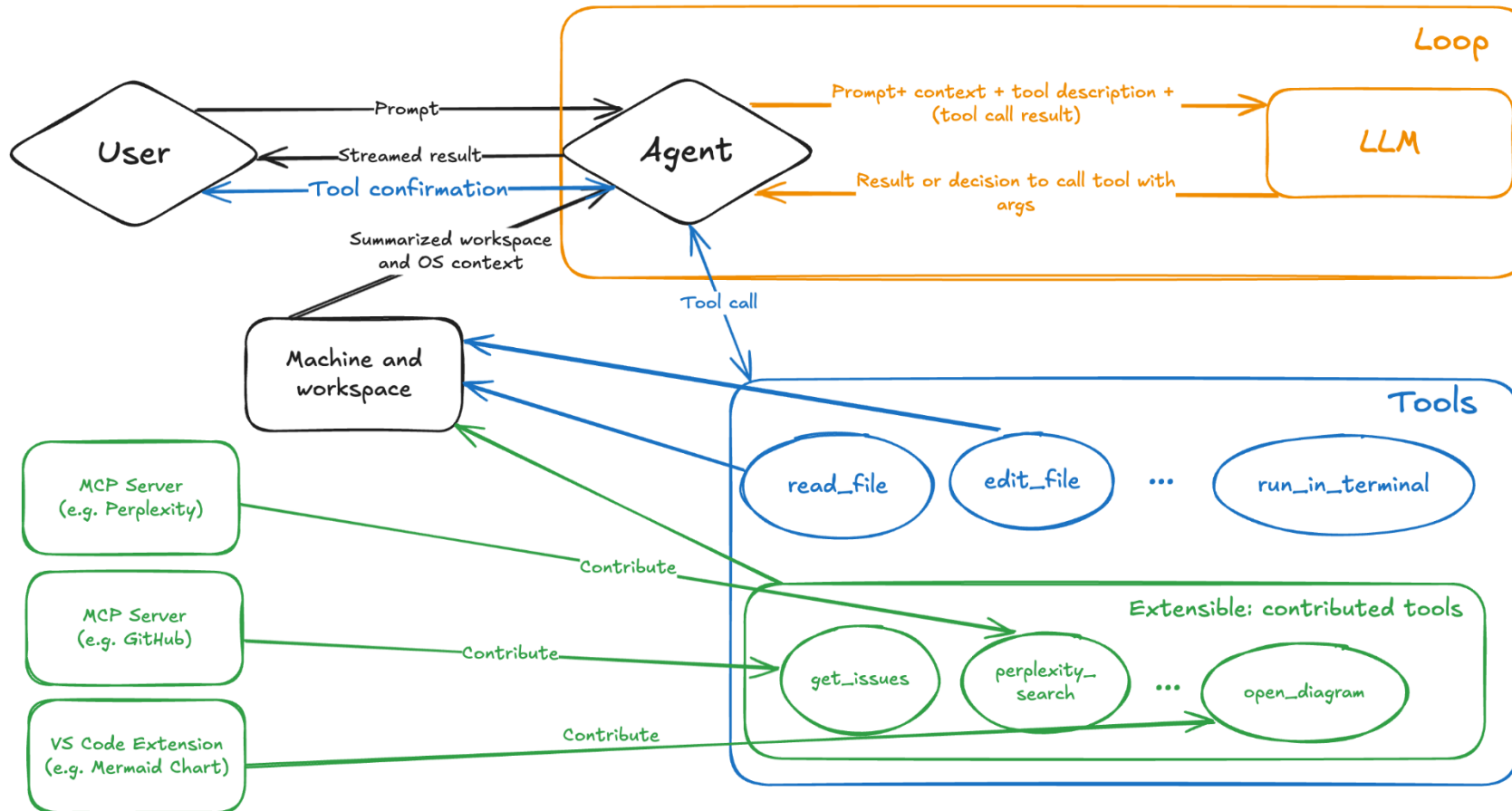
Model Context Protocol (MCP) enables GitHub Copilot to connect with wide range of external tools and data sources

- **MCP tools** are **server-based extensions** that Copilot can invoke in Agent Mode
- Configure enterprise-wide permission policies



Learn More at the MCP talk on Wed @15:50

Extensibility for Agents: Tools + MCP



GitHub Copilot: Free + Open-Source

- Copilot Chat Extension is now open source
 - Audit, extend, and customize chat

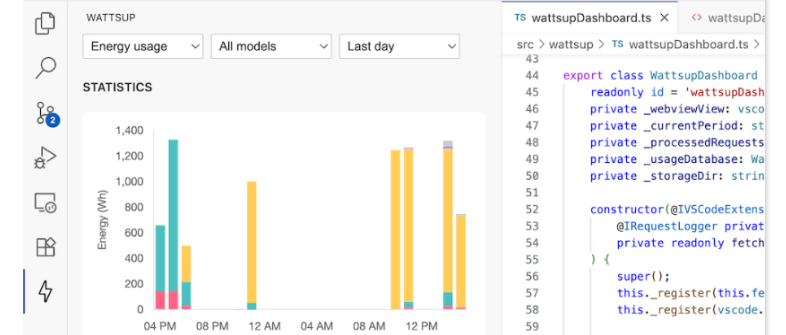
Enhance Prompt Feature Implementation


This implementation adds an "Enhance Prompt" feature to the GitHub Copilot Chat extension as requested in the issue.

Wattsup with Github Copilot

This is a fork of [vscode-copilot-chat](#), the official VS Code extension for Github Copilot Chat.

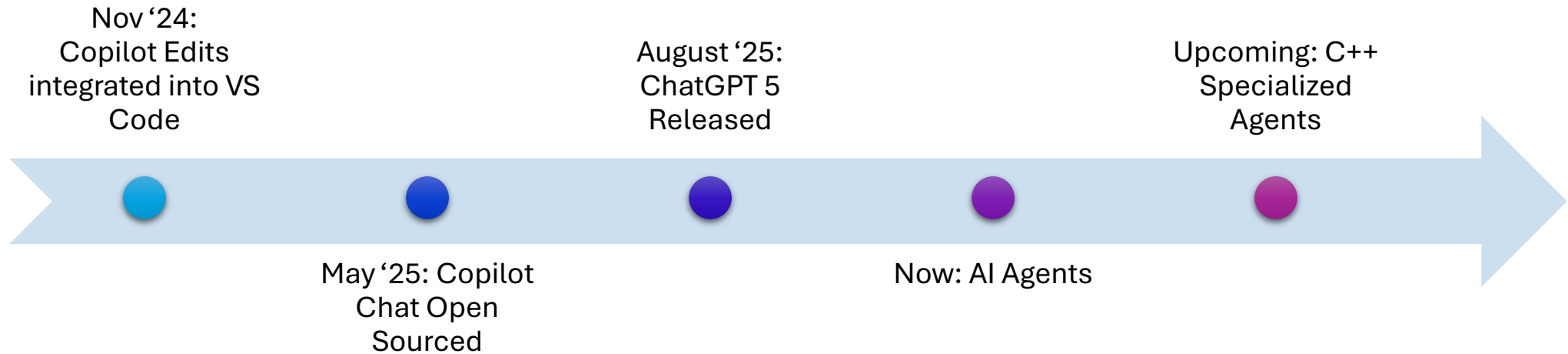
Wattsup with Github Copilot allows you to monitor AI requests usage with the integrated Wattsup dashboard, giving quick access to key metrics such as output token usage and CO2 emission equivalences over different time periods.



 **vscode-copilot-chat** Public
forked from [microsoft/vscode-copilot-chat](#)

Want to try copilot? GitHub Copilot Free Tier now available, sign up at <https://github.com/features/copilot>

Where we are now?



Questions?



@AlexandraKemperMS
@VisualC



visualcpp@microsoft.com