

Table of Contents

- [Reporting a Performance Issue](#)
- [Visual Studio Code is consuming a lot of CPU](#)
 - [Run with Extensions Disabled](#)
 - [Profile the Running Extensions](#)
 - [Collect Information about the running VS Code](#)
 - [The Renderer/Window process consumes a lot of CPU](#)
 - [The Shared process consumes a lot of CPU](#)
- [Visual Studio Code is sluggish](#)
- [Visual Studio Code starts up slowly](#)
 - [Read the Startup Timers](#)

Reporting a Performance Issue

To report a performance issue please use the 'Report Issue' command available in the 'Help' menu. In the Issue Reporter window that shows up, set the issue type to 'Performance Issue'.

Issue Reporter

This is a

Performance Issue

File on

Visual Studio Code

Try to reproduce the problem after [disabling all extensions and reloading the window](#). If the problem only reproduces when extensions are active, it is likely an issue with an extension.

Title *

Please enter a title.

Steps to Reproduce *

When did this performance issue happen? Does it occur on startup or after a specific series of actions? We support GitHub-flavored Markdown. You will be able to edit your issue and add screenshots when we preview it on GitHub.

Please enter details.

☒ Include my system information ([show](#))

☒ Include my currently running processes ([show](#))

☒ Include my workspace metadata ([show](#))

☒ Include my enabled extensions ([show](#))

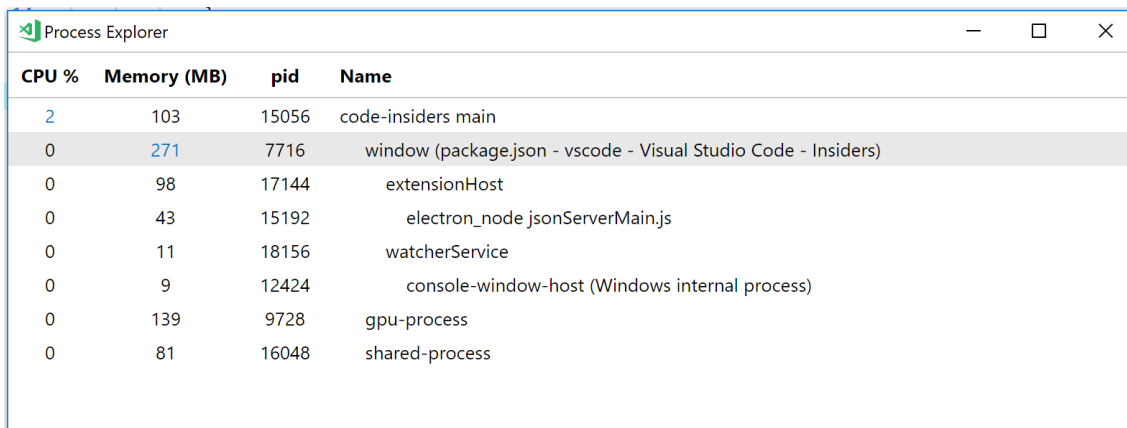
Preview on GitHub

The following sections describe how you can narrow down a performance issue.

Visual Studio Code is consuming a lot of CPU

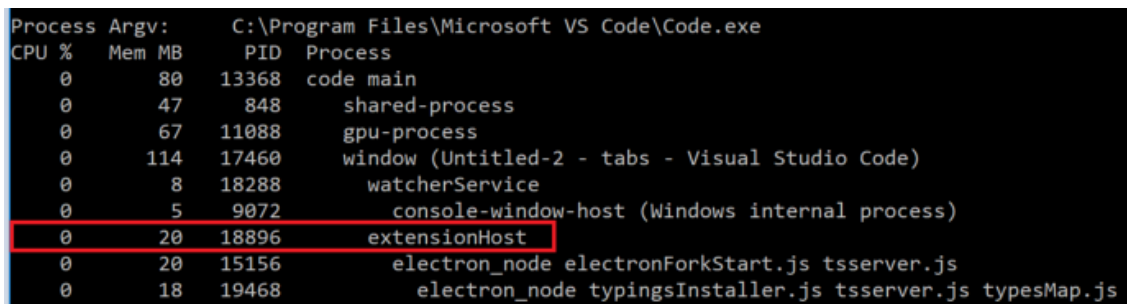
High CPU consumption is often caused by an issue in an extension. VS Code is running multiple processes and extensions are executed by the `Extension Host` process.

Find out whether an extension is causing the high CPU load. You can open the Process Explorer using `Help>Open Process Explorer` to open a new window with a continuously updating list of VS Code processes.



CPU %	Memory (MB)	pid	Name
2	103	15056	code-insiders main
0	271	7716	window (package.json - vscode - Visual Studio Code - Insiders)
0	98	17144	extensionHost
0	43	15192	electron_node jsonServerMain.js
0	11	18156	watcherService
0	9	12424	console-window-host (Windows internal process)
0	139	9728	gpu-process
0	81	16048	shared-process

Try to reproduce the issue with this window open, or take a snapshot of the processes by running `code --status` from the command line. For information on how to run `code` from the command line on OS X please see the [setup documentation](#). When the process listing in the Process Explorer or the output of `code --status` shows that the `extensionHost` is consuming a high percentage of CPU (see below) then this is likely caused by an extension.



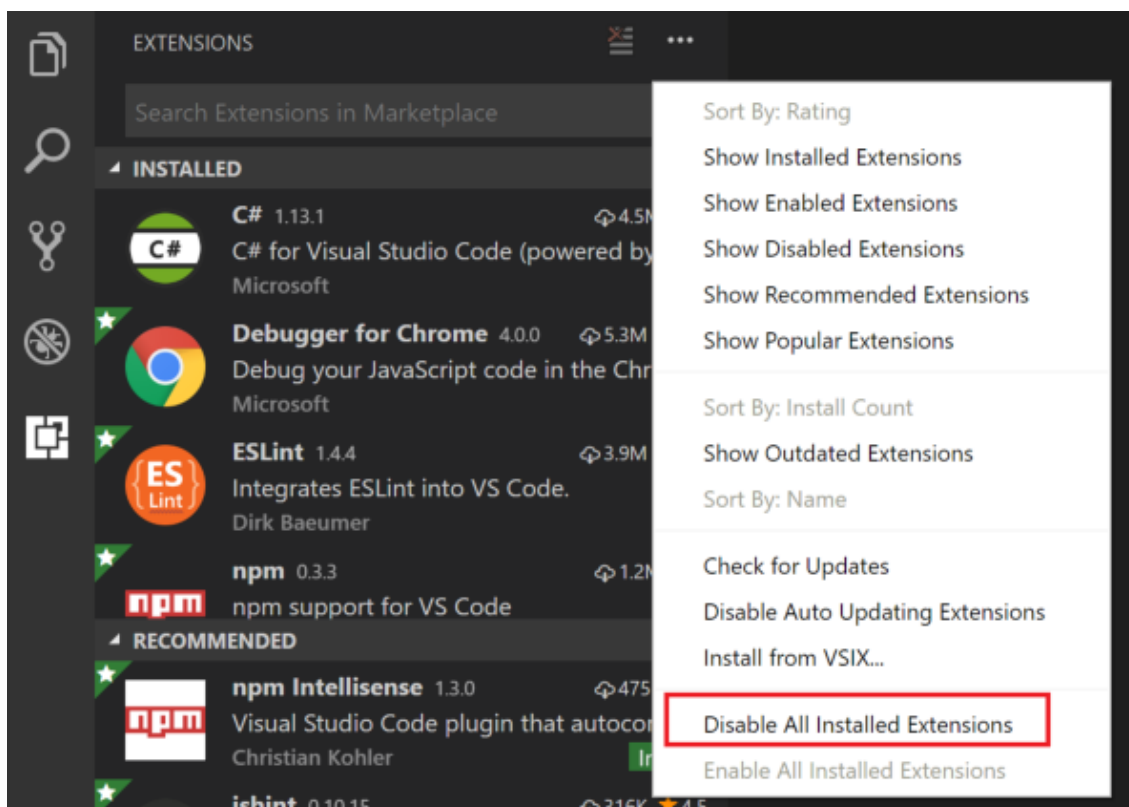
```

Process Argv: C:\Program Files\Microsoft VS Code\Code.exe
CPU %  Mem MB  PID Process
0      80     13368 code main
0      47     848  shared-process
0      67     11088 gpu-process
0      114    17460 window (Untitled-2 - tabs - Visual Studio Code)
0      8      18288 watcherService
0      5      9072  console-window-host (Windows internal process)
0      20     18896 extensionHost
0      20     15156 electron_node electronForkStart.js tsserver.js
0      18     19468 electron_node typingsInstaller.js tsserver.js typesMap.js

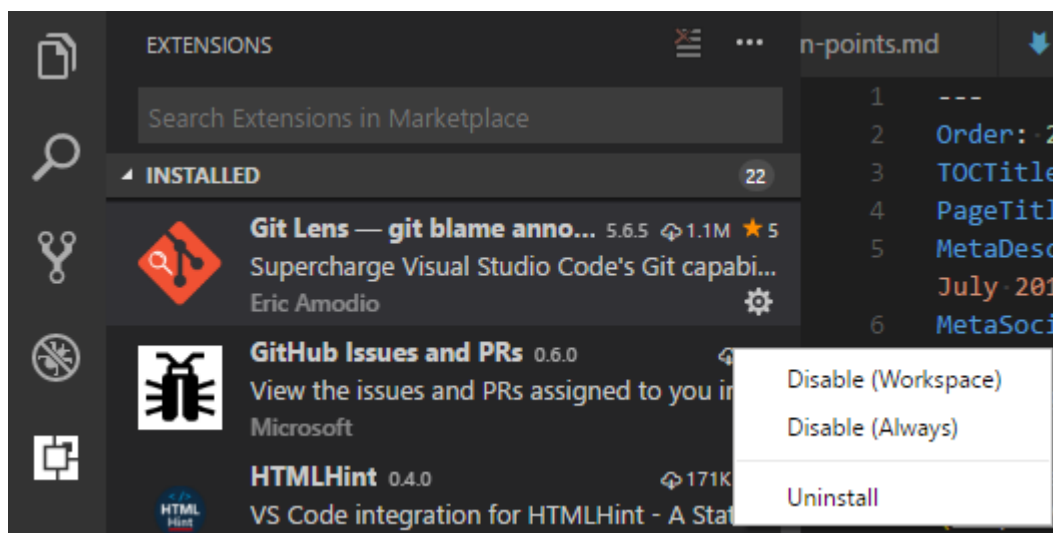
```

Run with Extensions Disabled

The first step is to try to reproduce the problem with all extensions disabled. You can disable the extensions either from the command line using the `--disable-extensions` command line argument, e.g., `code --disable-extensions` or from the `...` menu in the Extensions Viewlet:



If you cannot reproduce the issue when **all** extensions are disabled, then can you narrow down the extension by selectively disabling extensions? You can disable extensions in the Extensions Viewlet and restart/reload.

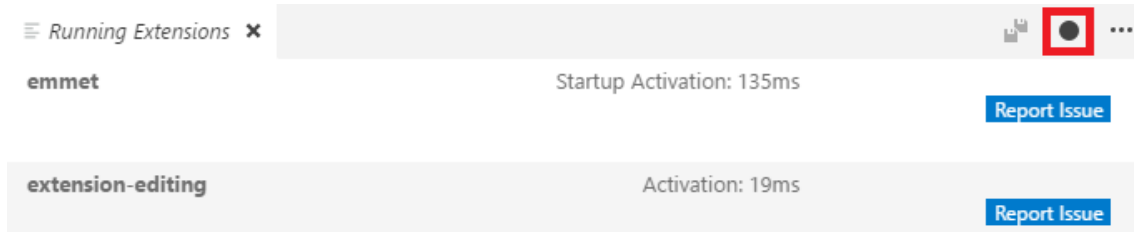


Once you found the extension, please file an issue against the extension. To file an issue against an extension open the Running Extensions editor by executing "**F1** > Developer: Show Running Extensions" and use the [Report Issue](#) button or find the extension in the marketplace, and use the [Issues](#) link in the [Resources](#) section.

Profile the Running Extensions

You can create a CPU profile and share it in the issue with the extension author or us. To create a CPU profile:

- Close all instances of VSCode and start with `code --inspect-extensions=9993` or any other port number.
- Execute "F1 > Developer: Show Running Extensions." This command opens an editor with all the running extensions.
- To start recording a profile click the run control in the editor's title bar:



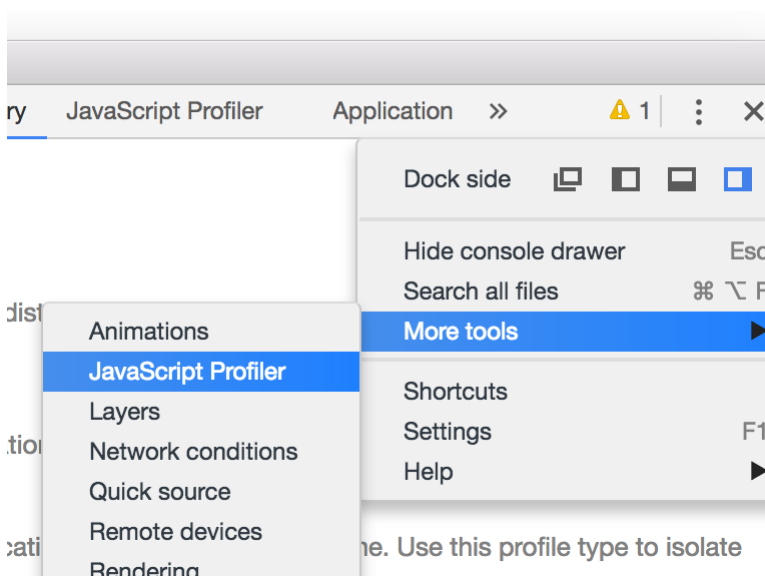
- Perform some steps in VS Code that expose the slowdown
- Stop the recording using the same control.
- Save the extension host profile:



- Attach the profile to a Github issue.

To analyze the performance yourself:

- remove the trailing `.txt` suffix from the profile name.
- Toggle the developer tools `Help > Toggle Developer Tools`
- Open the JavaScript profiler as shown below:



- Load the profile using the `Load` button.

Collect Information about the running VS Code

If you can reproduce the problem with extensions disabled, then use `code --status` from the command line.

Notice: this command requires that VS Code is already running

This command collects status about your running VS Code and the workspace you have opened. You can attach this information when reporting a performance issue.



The output includes information about the environment, all running VS Code processes, and the counts of some typical files in the workspace.

The Renderer/Window process consumes a lot of CPU

CPU %	Mem MB	Process
0	98	code-insiders main
0	82	gpu-process
0	82	shared-process
0	279	window (shell.ts - vscode (Workspace))
0	49	watcher-service
0	98	extensionHost
0	492	electron_node electronForkStart.js tsserver.js
0	49	electron_node typingsInstaller.js tsserver.js typesMap.js
0	82	electron_node server.js
0	49	electron_node jsonServerMain.js
0	49	searchService

Try to reproduce the problem on different folders; a folder with less files, a folder that is not under source control, or open a single file.

Try to reproduce the issue in the VS Code **Insider** version. This will run our latest code and use a different setup (settings, extensions). You can install the insider version here <https://code.visualstudio.com/insiders>.

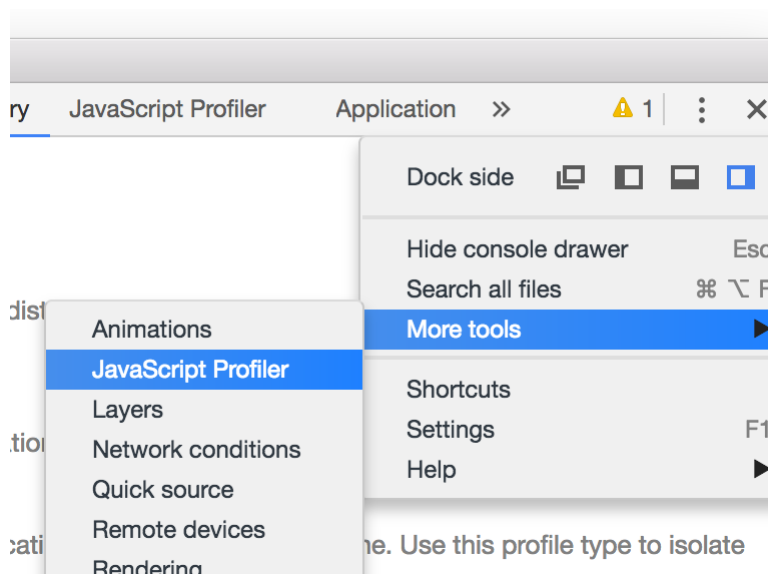
Run VS Code when your settings are reset/empty. Open your settings, copy your customizations and then empty the JSON contents so that only `{ }` is left.

Run VS Code in verbose mode and check whether there is any suspicious output in the console or the developer tools. You start VS Code in verbose mode with `code --verbose`. Also, open the DevTools ("Help>Toggle Developer Tools") and open the console there.

Profiling the Renderer Process

Very detailed information is captured in CPU profiles and that helps us best to narrow down on a slow-down. So, please create a CPU profile of the VS Code core (*renderer* process) and attach it to the issue. To create a profile:

- Execute "F1 > Toggle Developer Tools." In the overflow menu of the developer tools



- Select 'More Tools > JavaScript Profiler'. In there select start.
- Let it profile for 30 to 60 seconds, stop it.
- When the performance issue happens on startup, start the profiler and then reload the window using "F1 > Reload Window."
- Save the profile to a file and attach the file to your issue.

The Shared process consumes a lot of CPU

CPU %	Mem MB	Process
0	98	code-insiders main
0	82	apu-process
0	82	shared-process
0	279	window (shell.ts - vscode (Workspace))
0	49	watcherService
0	98	extensionHost
0	492	electron_node electronForkStart.js tsserver.js
0	49	electron_node typingsInstaller.js tsserver.js typesMap.js
0	82	electron_node server.js
0	49	electron_node jsonServerMain.js
0	49	searchService

Try to reproduce the problem on different folders; a folder with less files, or open a single file or open an empty window,

Try to reproduce the issue in the VS Code **Insider** version. This will run our latest code and use a different setup (settings, extensions). You can install the insider version here <https://code.visualstudio.com/insiders>.

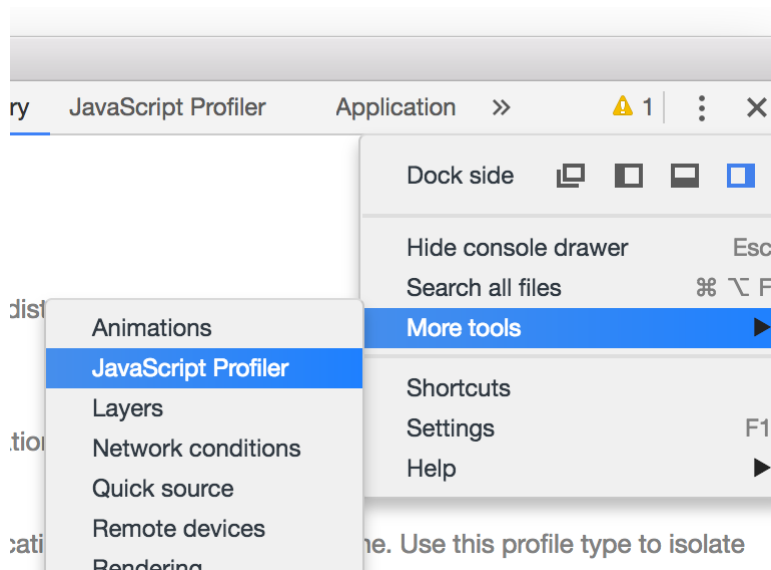
Run VS Code when your settings are reset/empty. Open your settings, copy your customizations and then empty the JSON contents so that only `{ }` is left.

Run VS Code in verbose mode and check whether there is any suspicious output in the console or the developer tools. You start VS Code in verbose mode with `code --verbose`. Also, open the DevTools ("Help>Toggle Developer Tools") and open the console there.

Profiling the Shared Process

Very detailed information is captured in CPU profiles and that helps us best to narrow down on a slow-down. So, please create a CPU profile of the VS Code core (*shared* process) and attach it to the issue. To create a profile:

- Execute "F1 > Toggle Shared Process." In the overflow menu of the developer tools



- Select 'More Tools > JavaScript Profiler'. In there select start.
- Let it profile for 30 to 60 seconds, stop it.
- Save the profile to a file and attach the file to your issue.

Visual Studio Code is sluggish

If VS Code is sluggish when scrolling, moving mouse or typing, please create a performance profile. Do the following

- Toggle the developer tools Help > Toggle Developer Tools
- Select 'Performance'. In there click the record button.
- Scroll, type or any operation leads to the slowness of the UI
- Let it profile for 30 to 60 seconds, stop it.
- Save the profile to a file and attach the file to your issue.

Visual Studio Code starts up slowly

If VS Code is slow to start then please create a startup CPU profile. Do the following

- Make sure to only have one window open
- Quit VS Code (CMD+Q for Mac, closing the last window on Linux/Windows)
- Start VS Code from the command line like so `code --prof-startup`
- VS Code will start and create two profile-files in your home-directory. Please attach these files to your issue or create a new issue with these two files

Read the Startup Timers

When VS Code feels slow to start, you can check the startup timers. Hit `F1` and select "Startup Performance." This will open developer tools and print some startup stats onto the "Console."

▼ Startup Performance Measurement		workbench.main.js:10
OS: win32 (10.0.14393)		workbench.main.js:10
CPUs: Intel(R) Core(TM) i7-3520M CPU @ 2.90GHz (4 x 2893)		workbench.main.js:10
Memory (System): 15.82GB (8.33GB free)		workbench.main.js:10
Memory (Process): 240.95MB working set (240.95MB peak, 219.84MB private, 45.58MB shared)		workbench.main.js:10
VM (likelihood): 0%		workbench.main.js:10
Initial Startup: false		workbench.main.js:10
Screen Reader Active: false		workbench.main.js:10
Empty Workspace: false		workbench.main.js:10
		workbench.main.js:10
(index)	Topic	Took (ms)
0	"[renderer] window.loadUrl() => begin to require(workbench.main.js)"	325
1	"[renderer] require(workbench.main.js)"	640
2	"[renderer] create extension host => extensions onReady()"	1014
3	"[renderer] restore viewlet"	911
4	"[renderer] restore editor view state"	1184
5	"[renderer] overall workbench load"	1402
6	"-----"	
7	"[main, renderer] start => extensions ready"	2062
8	"[main, renderer] start => workbench ready"	2424

Please share these numbers with us as they often allows us to understand what is slow.