

Opening VS Code From The Terminal Via Code

2025-11-30

Table of contents

Today I Learned added on 2025-11-30, learned on 2025-11-30; edited on 2026-06-14.

For my paid and unpaid programming work I've used [VS Code](#). The following has been routine for me for roughly 4 years: I open the target folder by pressing the `Open` button on VS Code's platform or, if I remember to, to `Cmd+O` on my MacOS to open a folder.

Today, I remarked internally to myself that VS Code probably has a method to open a new window from within the Terminal. Lo and behold, it does: `code .` to open the current directory in VS Code.

Of note, to get the `code` command working, I performed the following actions:

1. Open VS Code.
2. Open Command Palette via `Cmd+Shift+P`.
3. Type: "shell command"
4. Select: "Shell Command: Install 'code' command in PATH"
5. Gave permissions to VS Code.

Running `code --help` shows some other interesting options:

```
code --help
Visual Studio Code 1.105.1
```

```
Usage: code [options] [paths...]
```

To read from stdin, append `'-'` (e.g. `'ps aux | grep code | code -'`)

Options

<code>-d --diff <file> <file></code>	Compare two files with each other.
<code>-m --merge <path1> <path2> <base> <result></code>	Perform a three-way merge by providing paths for each file, the common origin of both modified versions, and the merge results.
<code>-a --add <folder></code>	Add folder(s) to the last active window.
<code>--remove <folder></code>	Remove folder(s) from the last active window.
<code>-g --goto <file:line[:character]></code>	Open a file at the path on the specified line and column.
<code>-n --new-window</code>	Force to open a new window.
<code>-r --reuse-window</code>	Force to open a file or folder in an already open window.
<code>-w --wait</code>	Wait for the files to be closed before returning.
<code>--locale <locale></code>	The locale to use (e.g. <code>en-US</code> or <code>zh-TW</code>).
<code>--user-data-dir <dir></code>	Specifies the directory that user data is kept for multiple distinct instances of Code.
<code>--profile <profileName></code>	Opens the provided folder or workspace with the specified profile. If the profile does not exist, a new one is created.

-h --help Print usage.

Extensions Management

--extensions-dir <dir> Set the root path for extensions.

--list-extensions List the installed extensions.

--show-versions Show versions of installed extensions, when using --list-extensions.

--category <category> Filters installed extensions by provided category, when using --list-extensions.

--install-extension <ext-id | path> Installs or updates an extension. The argument is either a path to a VSIX or an extension identifier. The identifier of an extension is '`{publisher}.{name}@{version}`'. To install a specific version, use '`{publisher}.{name}@{version}`'. For example: 'vscode.csharp@1.2.3'. To install the pre-release version of the extension, use '`{publisher}.{name}-pre-release@{version}`'. To force an update to the latest version, use '`{publisher}.{name}@{version}-force`'.

--pre-release Installs the pre-release version of the extension, when using --install-extension.

install-extension

--uninstall-extension <ext-id> Uninstalls an extension.

--update-extensions Update the installed extensions.

--enable-proposed-api <ext-id> Enables proposed API features for extensions. Can receive multiple arguments to enable individually.

Model Context Protocol

--add-mcp <jjson> Adds a Model Context Protocol server definition to the user profile. Accepts a JSON object: `'{"name":"server-name","command":...}'`

Troubleshooting

-v --version Print version.

--verbose Print verbose output (implies --wait).

--log <level> Log level to use. Default is 'info'. Allowed values are 'warn', 'info', 'debug', 'trace', 'off'. You can also filter log entries for a specific extension by passing extension id and log level: '`{publisher}.{name}:{logLevel}`'. For example: 'ms-vscode.csharp:debug'. You can receive one or more such entries.

-s --status Print process usage and diagnostics information.

--prof-startup Run CPU profiler during startup.

--disable-extensions Disable all installed extensions. This option is not available only when the command opens a new window.

--disable-extension <ext-id> Disable the provided extension. This option is not available only when the command opens a new window.

--sync <on | off> Turn sync on or off.

--inspect-extensions <port> Allow debugging and profiling of extensions. Check the connection URI.

--inspect-brk-extensions <port> Allow debugging and profiling of extensions with a breakpoint after start. Check the developer tools for the connection URI.

--disable-lcd-text Disable LCD font rendering.

--disable-gpu Disable GPU hardware acceleration.

--disable-chromium-sandbox Use this option only when there is requirement to run as a regular user on Linux or when running as an elevated user on Windows.

--locate-shell-integration-path <shell> Print the path to a terminal shell integration script.

'bash', 'pwsh', 'zsh' or 'fish'.
--telemetry Shows all telemetry events which VS code collects
--transient Run with temporary data and extension directories
time.

Subcommands

chat Pass in a prompt to run in a chat session in the current working directory.
serve-web Run a server that displays the editor UI in browsers.
tunnel Make the current machine accessible from vscode.dev or other machines through