GitHub Copilot CLI

A pocket-sized guide to where Copilot CLI keeps its setup, and which copy wins, as of October 2026.



My friend has a lot of questions.

What can you set up in Copilot CLI?

Copilot CLI reads its setup from files. There are five kinds, and each can sit in your home folder, in the repository you're working in, or come from your organization. When two copies disagree, Copilot CLI has a rule for which one wins.

The words agent, skill, tool and session are explained in An Accidental Guide to: Agents, Harnesses, Skills, and Tools.

Instructions file	Rules about your project, read in every session.
Skill	Steps for one kind of task, read when it comes up.
MCP server	A program that adds tools for another system.
Custom agent	A separate agent with its own job and tools.
Settings	How Copilot CLI runs, including trust and approvals.

Instructions file

In
01

Identify by `AGENTS.md` or `.github/copilot-instructions.md`.

What **Copilot CLI adds every instructions file that applies to the session.** An Accidental Guide to: Agents, Harnesses, Skills, and Tools explains them.

So what **Put rules for every session here.** No file outranks another, so keep each rule in one place.

Field marks

Loads	Check with
At the start of a session.	`/instructions`.

Confused with

Sk **Skill**

A skill's steps are read only when a task calls for them, so a long procedure belongs in a skill.

Skill

Sk
02

Identify by A folder in `.github/skills/` with a `SKILL.md` file.

What **Copilot CLI uses a skill when your request matches its description, or when you type / and its name.** An Accidental Guide to: Agent Skills explains skills.

So what A skill keeps the way you do a task in a file that can be reviewed and shared like code.

Field marks

Loads	Check with
Description first.	`/skills`.

Confused with

Ag **Custom agent**

The agent you're talking to follows a skill's steps itself. A custom agent is a separate agent with its own instructions and tools.

MCP server

Mc
03

Identify by An entry in `~/.copilot/mcp-config.json` or `.mcp.json`.

What **GitHub's own MCP server is built in and needs no setup.** An Accidental Guide to: Model Context Protocol explains MCP servers.

So what Add a server when the work needs a system Copilot CLI can't reach on its own, such as an issue tracker.

Field marks

Loads	Check with
When the session starts.	`/mcp`.

Confused with **Plugin**
A plugin is a package you install that can bring MCP servers, skills, custom agents and hooks with it.

Custom agent

Ag
04

Identify by A file such as `.github/agents/reviewer.agent.md`.

What **Copilot CLI reads custom agents from your Copilot folder, the repository and your organization.** An Accidental Guide to: Agents, Harnesses, Skills, and Tools explains agents.

So what You can pick one to run your session, or Copilot CLI can hand one part of a task.

Field marks

Loads	Check with
When it runs.	`/agent`.

Confused with **Subagent**
A subagent is any agent the main agent hands work to. A custom agent can be either the main agent or a subagent.

Settings

Se
05

Identify by	A `settings.json` file in `~/.copilot`, or in `.github/copilot` in the repository.
What	**Settings are the choices that control how Copilot CLI runs, such as the model.** Trusted folders are kept in `config.json`, and tools you've allowed to run without asking in `permissions-config.json`.
So what	Trust and approvals decide what Copilot CLI does without asking you first.

Field marks

Loads	Check with
When the session starts.	`/settings`.

Confused with	**Policy** A policy is a setting your organization or enterprise manages, such as which models you can use. Its value overrides yours.

Where things live

`~/.copilot` is your Copilot folder. If you set `COPILOT_HOME`, Copilot CLI uses that directory instead.

Your home folder

In **Instructions**	`~/.copilot/copilot-instructions.md` or `~/.copilot/instructions/`
Sk **Skills**	`~/.copilot/skills/` or `~/.agents/skills/`
Mc **MCP servers**	`~/.copilot/mcp-config.json`
Ag **Custom agents**	`~/.copilot/agents/`
Se **Settings**	`~/.copilot/settings.json`, with approvals in `permissions-config.json` and trusted folders in `config.json` beside it

This repository

In **Instructions**	`.github/copilot-instructions.md`, `.github/instructions/*.instructions.md`, `AGENTS.md`, `CLAUDE.md` or `GEMINI.md`
Sk **Skills**	`.github/skills/`, `.agents/skills/` or `.claude/skills/`
Mc **MCP servers**	`.github/mcp.json` to share, or `.mcp.json`
Ag **Custom agents**	`.github/agents/` or `.claude/agents/`
Se **Settings**	`.github/copilot/settings.json`, or `.github/copilot/settings.local.json` for you only
Hooks	`.github/hooks/*.json`

Your organization

Ag **Custom agents**	`agents/` in the organization's `.github` or `.github-private` repository
Policies	`copilot/managed-settings.json` in the `.github-private` repository, plus policies an administrator sets on GitHub

Note. As of October 2026, custom instructions that an organization sets on GitHub don't reach Copilot CLI. The CLI reads only the instruction files in your Copilot folder and in the repository.

How do the places combine?

Instructions and hooks add up. Skills, MCP servers, custom agents and settings have one winner for each name or setting.

Thing	When two places set it
In **Instructions**	Every file that applies is used, and none outranks another.
Sk **Skills**	The first found wins: the repository's, then yours, then a plugin's.
Mc **MCP servers**	From highest: `--additional-mcp-config`, a plugin, the repository, then `~/.copilot/mcp-config.json`.
Ag **Custom agents**	Yours wins, then the repository's, a plugin's, the organization's, then the enterprise's.
Se **Settings**	Later layers win: yours, the repository's, `settings.local.json`, environment variables, then flags. Managed settings win over most of them.
Hooks	Every hook runs.

Tip. Repository MCP servers, hooks and plugin settings load only after you trust the folder. `/env` shows everything the session loaded.

Same thing elsewhere

Copilot CLI	Claude Code	Claude Desktop	Codex	VS Code and Visual Studio
In **Instructions file**	`CLAUDE.md`	Instructions	`AGENTS.md`	Same file
Sk **Skill**	Skills	Skills	Skills	Same folder
Mc **MCP server**	`.mcp.json`	Connectors	`config.toml`	Same file
Ag **Custom agent**	Subagents	None documented	Custom agents	Same folder
Se **Settings**	`settings.json`	Settings	`config.toml`	IDE settings

Each of these tools has its own guide: An Accidental Guide to: Claude Code, An Accidental Guide to: Claude Desktop, An Accidental Guide to: OpenAI Codex, and An Accidental Guide to: VS Code and Visual Studio.

Sources

These public sources were checked on 22 October 2026.

GitHub Copilot CLI configuration directory
GitHub Docs · `docs.github.com/en/copilot/reference/copilot-cli-reference/cli-config-dir-reference`

GitHub Copilot CLI command reference
GitHub Docs · `docs.github.com/en/copilot/reference/copilot-cli-reference/cli-command-reference`

Adding custom instructions for GitHub Copilot CLI
GitHub Docs · `docs.github.com/en/copilot/how-tos/copilot-cli/customize-copilot/add-custom-instructions`

Adding MCP servers for GitHub Copilot CLI
GitHub Docs · `docs.github.com/en/copilot/how-tos/copilot-cli/customize-copilot/add-mcp-servers`

Invoking custom agents
GitHub Docs · `docs.github.com/en/copilot/how-tos/copilot-cli/use-copilot-cli/invoke-custom-agents`

Allowing and denying tool use
GitHub Docs · `docs.github.com/en/copilot/how-tos/copilot-cli/use-copilot-cli/allowing-tools`

copilot help
GitHub Copilot CLI 1.0.102 · copilot `--help`, copilot help config

In Sk Mc Ag Se

An Accidental Guide to…
GitHub Copilot CLI
A pocket-sized guide to where Copilot CLI keeps its setup, and which copy wins, as of October 2026.
Written and reviewed by a person. AI does the grunt work. Set in TeX Gyre Heros and Newsreader.
Leave a star: `accidental.info/star`

My friend has a lot of questions.