

The Claude Opus 5.5 Usage Limit Fix

The exact `claude.md` config that stops Claude Opus 5.5 from burning your heaviest model on tasks that never needed it.

Why the basic setup burns your usage limit

Claude Opus 5.5 dropped and it's genuinely a better model for running AI agents all day. But out of the box, it still defaults to doing the work itself, in its own heaviest mode, on every single task you hand it. A one-line typo fix and a full module refactor get treated the same way. That's how you hit your usage cap the day after it resets, run out mid-project, and sit locked out until the next window opens.

The fix isn't a different model or a paid upgrade. It's one file, two rules, sitting in your project root before you start working.

01

How it actually works

1. Claude reads a file called `claude.md` at the start of every session in that project, before it touches any task you give it.
2. The first rule tells it never to do the work itself: always dispatch a subagent instead. That keeps your main session light and hands the actual labor to a separate worker.
3. The second rule tells it not to always reach for the heaviest model: use Opus 5.5 itself for lighter tasks, and let the subagent pick a smaller model when the job doesn't need the top tier.

02

The one thing almost everyone gets wrong

People add the rules but leave them vague, and vague rules get ignored.

WATCH OUT

"Use lighter models when appropriate" is not a rule Claude can act on consistently. "Never do the work yourself, always dispatch a subagent" and "don't always use the heavy model, use Opus 5.5 for lighter tasks" are specific enough to change behavior every time. Write the rules as commands, not suggestions.

The exact config

CLAUDE.MD

Model & subagent routing

- Never do the work yourself. Always dispatch a subagent.
- Don't always use the heavy model. Use Opus 5.5 for lighter tasks.

SITES SHOWN IN THE VIDEO

- Prompting Claude Opus 5.5 (Anthropic docs)
<https://platform.claude.com/docs/en/build-with-claude/prompt-engineering/prompting-claude-opus-5-5>

Set it up in Claude Code or Codex

1. Open the root folder of the project you run Claude Code against, the top-level directory, not a subfolder.
2. Create a new file there named exactly CLAUDE.md.
3. Paste the two rules in above, either as-is or folded into your existing CLAUDE.md if you already have one for other project instructions.
4. Save it. Claude Code reads this file automatically at the start of every session in that project, so there's no extra command to run.
5. If you also run Codex on the same project, add the same two rules to AGENTS.md instead. Codex does not read CLAUDE.md, so the rules need to live in the file Codex actually checks.

QUICK TIPS

- Put the two rules near the top of the file. Claude weighs early instructions more heavily than ones buried under a long list of project notes.
- If you already have a CLAUDE.md with project context, add these as a short standalone section rather than mixing them into prose.

The commands I actually run to test it

TEST THE LIGHT-TASK ROUTING

Fix the typo in the README and update the changelog line. Don't touch anything else.

TEST THE SUBAGENT DISPATCH

Refactor [FILE OR MODULE] to remove the duplicated logic, then run the test suite and fix anything that breaks.

CHECK IT'S ACTUALLY ROUTING, NOT JUST CLAIMING TO

Before you start: tell me whether you're dispatching a subagent for this task and which model tier you're using, and why.

STRESS-TEST ON A REAL BACKLOG

Go through [LIST OF SMALL TICKETS/TASKS] one at a time. For each one, decide whether it needs a subagent and whether it needs the heavy model, then do it.

When it breaks

ERROR	FIX
Claude still does the task itself instead of dispatching a subagent	Confirm the file is named exactly CLAUDE.md and sits in the project root. A misnamed file or one buried in a subfolder is never read.
Claude ignores the "use lighter tasks" rule on small jobs	The rule is too vague. Add one or two concrete examples of what counts as light, like single-file edits or typo fixes, right in the file.
Usage cap still gets hit fast even with the rules in place	Check whether an old settings file or CLI flag is forcing high effort elsewhere in your config. Opus 5.5 defaults to medium effort; Opus 5 defaulted to high, and a leftover override can cancel out the claude.md rules.
Rules work in Claude Code but Codex ignores them on the same project	Codex reads AGENTS.md, not CLAUDE.md. Copy the same two rules into AGENTS.md for that project.
Subagent dispatch silently does nothing	Update Claude Code to the latest version. Subagent support is newer, and an outdated install can read the rule but have no subagent path to act on it.

Ready to go deeper?

This playbook is the short version. Inside the hub you get the full build walkthroughs, the prompt library, and the systems I use to ship this in a weekend.

Join the Build →

<https://hub.digicuratoragency.com/join>