Claude Code

# Make your Claude Code limit last longer

A Haiku helper for the chores: one subagent file and one note in CLAUDE.md. Checked against Claude Code's docs and run once on October 12, 2026.

**Opus 5.5 · planning, debugging** **Haiku 5.5 · chores**

Setup guide
October 2026

# Why your limit goes fast

## 1 Opus 5.5 is the default.

On Pro, Max, Team and Enterprise, Claude Code starts on Opus 5.5 (code.claude.com/docs/en/model-config).

## 2 Its helpers run on the same model.

The built-in Explore agent, which searches your codebase, uses "the main conversation's model" (code.claude.com/docs/en/sub-agents).

## 3 Every request counts.

Subagent requests "count toward the same usage limits as your main conversation" (same page). So a rename or a log read on Opus is paid at Opus rates.

**What this guide does:** it moves the chores (finding files, reading logs, renames, one-line edits) to a helper that runs on Haiku 5.5, and leaves planning and debugging to Opus.

**What it doesn't do:** make your limit unlimited. The chores still count, they just cost less. How much longer your limit lasts depends on how much of your work is chores.

# The math

Per token, **Haiku 5.5 costs a fortieth of Opus 5.5** on the API, for prompts up to 100,000 tokens.

| MODEL | INPUT / MTOK | OUTPUT / MTOK |
|---|---|---|
| **Claude Opus 5.5**<br>the default in Claude Code | $4 | $20 |
| **Claude Haiku 5.5**<br>prompts up to 100,000 tokens | $0.10 | $0.50 |
| **Claude Haiku 5.5**<br>prompts over 100,000 tokens | $0.50 | $2.50 |
| **Claude Haiku 4.5**<br>the last Haiku | $1 | $5 |

API list prices from platform.claude.com/docs/en/about-claude/pricing, October 12, 2026. Pro and Max plans don't publish per-model rates, so read these as the yardstick, not your exact limit.

> "For simple subagent tasks, specify `model: haiku` in your subagent configuration."
>
> Claude Code docs, Manage costs effectively (code.claude.com/docs/en/costs)

> "The subagent's own requests still draw on your usage. To spend less on them, choose a smaller model for a subagent."
>
> Same page, Delegate verbose operations to subagents

Step 1

# Paste this into Claude Code

Open Claude Code in your project folder and paste the whole block (it continues on the next page).

```
Set up a Haiku helper for this project. Do exactly this, then
show me both files:

1. Create the file .claude/agents/haiku-helper.md with this
content:

---
name: haiku-helper
description: Use proactively for quick, mechanical jobs -
finding files, searching code, reading logs or test output,
listing folders, renaming files, one-line edits. Not for
planning, debugging or changes across several files.
model: haiku
tools: Read, Grep, Glob, Bash, Edit
---

You do small, well-defined jobs fast. Do exactly the job you
were given and nothing more.
When you finish, reply in a few lines: what you did, and the
file paths or lines that matter.
If the job turns out to need real decisions (a bug to
diagnose, a design choice, edits across several files), stop
and say so instead of guessing.
```

Step 1

# Paste this into Claude Code (2/2)

```
2. Add this section to CLAUDE.md in the project root. Create
the file if it doesn't exist, and keep everything already in
it:

## Hand chores to haiku-helper

Send these to the haiku-helper subagent instead of doing them
yourself:
finding files, searching code, reading logs or test output,
listing folders, renaming files, one-line edits.

Keep planning, debugging and changes across several files for
yourself.

3. Don't change anything else.
```

# What each line does

The helper file lives at `.claude/agents/haiku-helper.md`.

| | |
|---|---|
| `name: haiku-helper` | Its name. You'll see it in the conversation, and you can call it with **@haiku-helper**. |
| `description: Use proactively for quick, mechanical jobs …` | How Claude decides to hand a job off. The docs say "use proactively" makes it delegate more readily. |
| `model: haiku` | Runs it on the latest Haiku, which is **Haiku 5.5**. Without this line a subagent uses your main model. |
| `tools: Read, Grep, Glob, Bash, Edit` | Enough to search, read, rename and make small edits. Nothing for the web. |
| `The three lines below` | Keep it to the job, keep the reply short, and make it stop and say so when a job needs real decisions. |

The note in **CLAUDE.md** is what makes Opus hand the chores off without being asked each time. Opus still decides, so it's a strong nudge, not a switch.

Step 2

# Claude writes both files

```
● I created both files. The project had no CLAUDE.md before, so I
  made a new one and there was nothing to keep. Nothing else was
  changed and nothing was committed.

  .claude/agents/haiku-helper.md
  ---
  name: haiku-helper
  description: Use proactively for quick, mechanical jobs -
  finding files, searching code, reading logs or test output,
  listing folders, renaming files, one-line edits. Not for
  planning, debugging or changes across several files.
  model: haiku
  tools: Read, Grep, Glob, Bash, Edit
  ---
```

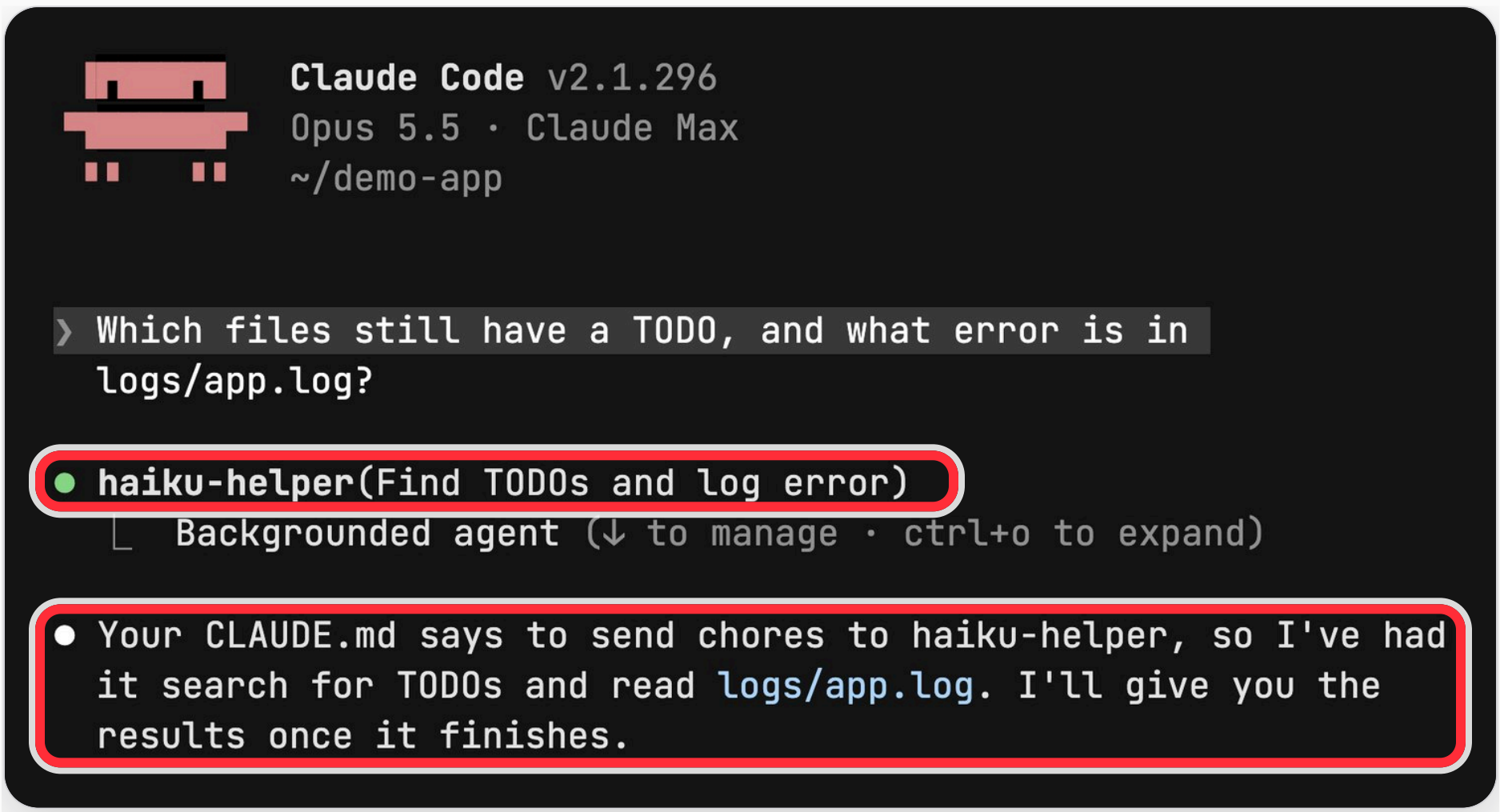
```
  Claude Code reads subagent files when a session starts, so
  restart it or open a new session before using haiku-helper.
```

**Then restart Claude Code once** if `.claude/agents` is new. It watches that folder only if it existed when the session started. After that, edits load within a few seconds.

Step 3

# Work as usual

We asked a normal question in a fresh session. Opus read the note and **handed the lookup to haiku-helper**.




A real run on October 12, 2026 (Claude Code 2.1.296, Max plan, Opus 5.5). Only the folder path is shortened.

Step 3

# The answer comes back

The helper did the searching and the reading. Opus wrote up what it found.

```
› Message from @haiku-helper (ctrl+o to expand)

● Two files still have a TODO:
  - src/greet.js:1: // TODO: i18n (the greeting is hardcoded as hi
    ${n})
  - src/math.js:2: // TODO: handle strings

  logs/app.log has one error, with no stack trace:
  2026-10-12 10:00:02 ERROR db timeout after 5000ms
  The database connection timed out after 5 seconds, and the next
  line (10:00:03 INFO retry ok) shows the retry worked a second
  later.

● Agent "Find TODOs and log error" finished · 7s
```

# Check it's working

Type `/usage`. Under Session, **Usage by model** lists **claude-haiku-5-5** next to Opus.

```
Session

Total cost:            $0.1302
Total duration (API):  21s
Total duration (wall): 2m 56s
Total code changes:    0 lines added, 0 lines removed
Usage by model:
    claude-haiku-5-5:  1.2k input, 1.1k output, 11.1k cache
read, 2.5k cache write ($0.0011)
     claude-opus-5-5:  3.8k input, 720 output, 208.0k cache
read, 7.2k cache write ($0.1291)
```

```
Each subagent runs its own requests. Be deliberate about
spawning them — and consider configuring a cheaper model for
simpler subagents.
```

**The numbers are small on purpose.** In this tiny demo Opus still cost more, because it reads its setup and your question either way. The bigger the chore (long logs, many files), the more of it stays on Haiku, and only a summary comes back to Opus. On Pro and Max, `/usage` also shows the tip at the bottom when subagents take a big share.

# If something goes wrong

### Claude doesn't use the helper

Restart Claude Code if **.claude/agents** was new. You can also send a job there yourself: type **@** and pick **haiku-helper (agent)**, or say "Use the haiku-helper subagent to ...".

### I want it in every project

Put the file in **~/.claude/agents/** and the note in **~/.claude/CLAUDE.md** instead.

### It hands off jobs it shouldn't

Narrow the description and the list in CLAUDE.md. Take **Edit** out of tools if you want it to only look, never touch.

### I use Bedrock, Vertex or Foundry

There, the **haiku** alias points to Haiku 4.5 (code.claude.com/docs/en/model-config), so the price gap is smaller.

### Will I never hit my limit again?

Nobody can promise that. The chores still count, they just cost less.

# This DM came from an automation

You commented HAIKU and Uplika sent this guide. Uplika lets Claude post to your channels, and sends a DM when someone comments the word you pick.

See how it works: uplika.com/en/integrations/claude

Connector URL: `https://api.uplika.com/mcp`