

The Claude Commands That Stop You Running Out

You saw me hit the wall twenty minutes into an evening. This is the list that fixed it: what actually burns your usage, and the commands that let you see it instead of guessing.

Two minutes to read. The first command takes five seconds.

Before anything else: type / into Claude and look at what comes up. That tells you in five seconds which of these you have. Different versions of Claude carry different sets, and your own menu is the only honest answer for your setup.

/context works in Cowork and in Claude Code. (Cowork is the agent mode inside the Claude desktop app; Claude Code is the version that lives in a terminal. Both need a paid plan.) The lists further down are Claude Code's, and they are the ones worth knowing rather than all of them. **Section 2 applies to you whatever you are using.**

1. Start here. Type this now

```
/context
```

That is the whole first step. It draws your context window as a coloured grid and tells you what is filling it. (Not in your  menu? Skip to section 2. Those four levers work everywhere.)

✓ **You'll know it worked when** you get a coloured grid plus a breakdown of tokens used against your total: messages, system prompt, tools, skills, free space.

Most people have never once looked. You cannot manage a number you have never seen, and almost everyone is guessing about the one thing that decides whether their session survives the evening.

How to read what comes back. Look at **Messages** first. If that is the biggest number, your chat is too long and section 3 is your fix.

Then look at the tools row. If it says **deferred**, Claude is only holding the *names* of your tools, not their full instructions, so that row is smaller than the number makes it look.

It is not free though. **Either way, a big tools row means more connectors are switched on than today's job needs, and that is section 2.**

2. The four things that actually burn usage

Not folklore. This is Anthropic's own list of what makes your usage go faster.

	The lever	What to do about it
1	How long and complex the conversation is	new task, new chat
2	Which features are switched on	turn off web search and connectors you are not using today
3	Which model you are using	small model for small jobs
4	The effort level you selected	drop it for routine work

Number 2 is the one nobody talks about. Every connector and tool you leave enabled is in play on every single message, whether you use it or not.

Switching off the ones today's work does not need is the cheapest win on this page and it takes about thirty seconds.

Where the switches are. In Cowork: **Customize > Connectors**, and turn off what today's job does not need. In Claude Code: type `/mcp`. In the Claude app: the tools button under the message box. The model and effort settings sit under the message box in Cowork and the app, and are `/model` and `/effort` in Claude Code.

Long chats cost you twice. The whole history rides along on every message, and once the conversation gets long enough that Claude has to summarise it just to keep going, that summarising costs you more again on top.

▶▶ **Skip for now:** you do not need to understand tokens, context windows, or how any of this works underneath. Run `/context`, look at which row is biggest, fix that one. Come back for the rest later.

3. The commands, in the order they will change your week

The five that matter

Command	What it does	When to reach for it
<code>/context</code>	Context usage as a coloured grid, with suggestions. <code>/context all</code> for everything	Before you start anything big
<code>/compact</code>	Frees space by summarising the conversation. Takes optional instructions	Mid-task, when clearing would cost too much. The one people skip
<code>/clear</code>	New conversation, empty context. Aliases <code>/reset</code> , <code>/new</code>	Between unrelated tasks. Never in the middle of one
<code>/usage</code>	A breakdown of what you have used and what it cost. Aliases <code>/cost</code> , <code>/stats</code>	When the wall arrived sooner than you expected
<code>/rewind</code>	Undo. Puts the conversation, and your files if it touched any, back to how they were	After the AI has confidently wrecked something

`/clear` **has a price and nobody mentions it.** The new chat knows nothing about the old one. Clear between tasks, never inside one. When you are mid-task and running low, `/compact` is the right tool: it keeps the thread and drops the dead weight.

Model and effort

Command	What it does
<code>/model</code>	Switch model for this session or all future ones
<code>/effort</code>	low, medium, high, xhigh, max, ultracode, or auto
<code>/autocompact</code>	Set the auto-compact threshold: <code>auto</code> or a token number
<code>/advisor</code>	Turn a secondary-model advisor on or off

Conversation shape

Command	What it does
<code>/resume</code>	Go back to an earlier conversation
<code>/branch</code>	Branch this conversation to try a different direction
<code>/fork</code>	Copy the conversation into a new background session
<code>/background</code> (<code>/bg</code>)	Detach the session to run as a background agent

Housekeeping

`/help` (everything available) • `/copy [N]` (last response to clipboard) • `/export [file]` (conversation as plain text) • `/cd <path>` (move the session) • `/exit` (`/quit`)

▶ **Skip for now:** everything under "conversation shape". Useful, but not why you are running out. Come back when the first five are habit.

There are 60+ built in. Type `/` and keep typing letters to filter the list down.

4. Make your own

Skills need a paid plan, Pro or above, with code execution switched on.

A command and a skill are the same thing now. A file at `.claude/skills/deploy/SKILL.md` gives you `/deploy`. In Cowork you never touch a file: finish a job the way you normally would, then ask Claude "**package what we just did into a skill**" and it writes it for you. You can see and manage them afterwards under **Customize > Skills**.

✓ **You'll know it worked when** the new skill shows up in that list and you can call it by name.

The test for whether something should be a skill: you have pasted the same instructions more than twice. That is it. If you keep explaining the same process, stop explaining it and save it.

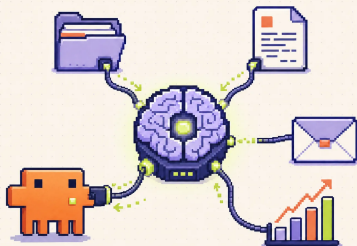
5. Do this now, in order

1. Run `/context`. Look at which row is biggest, ignoring anything marked deferred.
2. Turn off every connector and tool today's work does not need. ✓ You'll know it worked when you re-run `/context` and the tools number has dropped.
3. Drop the model and the effort level for whatever small job is next.
4. Next time you switch tasks, `/clear` instead of carrying on.

That is the whole thing. If you only do number 1 and number 2 you will get most of the benefit, and both together take under a minute.

Ready for the paid step?

The free guides get you started. These are the full systems I run my own work on. One-time prices, plain English, 14-day money-back.

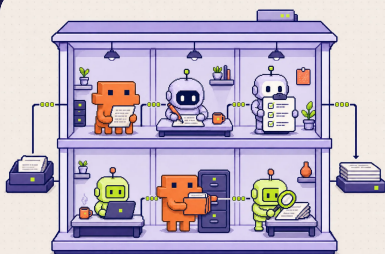


START HERE

The AI OS Kit

Give your AI a second brain: a folder that remembers your business, so every answer starts from you.

\$47 one time • see the AI OS Kit →



11 AGENTS

Your AI Team

Eleven AI agents with real jobs: brief you every morning, review your week, check work before it ships.

\$197 one time • see Your AI Team →

The AI OS Prompt Pack • \$17 • 58 ready-to-paste prompts written for the AI OS Kit. Add it at checkout, or grab it on its own →

buy once, every future update is free, 14 days to change your mind

The Front Row

I am building a community for people learning to use AI to do real work, not demos. Every new guide goes to that list first.

[Join the Front Row →](#)

Doors aren't open yet. You'll be the first to know when they open.

[Free guides](#)

[Products](#)

[Front Row](#)

[Privacy](#)

Made by Kojak. No code, no jargon, no gatekeeping.

I keep this updated for 12 months from the day you get it, and you'll always be able to grab the latest copy at the same link.