



Terminal Desk for Claude Code

The guide that goes with the mod from the reel. Install it on Windows or Mac, then read every figure on the bar and every panel in the pane, and know what to do about each one.

GraniteAI, Manchester, New Hampshire. Facts checked 5 October 2026.
The guide page and the mod download: graniteai.co/tools/terminal-desk

START HERE

What you are installing

You saw the desk in the reel. This guide gets it running in your own terminal and explains what each number is telling you.

- ☐ A one-line stats bar above your prompt in every Claude Code session
- ☐ A pane you open with `/desk`, holding five panels
- ☐ The last five judgment calls Claude reported, each with a Reject button
- ☐ A picture of what is filling your context window and how much room is left
- ☐ A countdown on your prompt cache, and an estimate of what the session would cost at API list prices

What a mod is

A mod is a plugin that changes Claude Code itself: it can draw in the interface, add commands, and watch what a session does. Terminal Desk is one folder with one code file, **hooks/register.tsx**. Mods require Claude Code v2.1.287 or later.

Check your version. Run in your terminal.

```
claude --version
```

Read this before you install any mod

This is Anthropic's own warning, and it applies to every mod, this one included:

"A mod is code that runs with your permissions. It can read and write your files, start processes, and make network requests. Install mods only from authors and marketplaces you trust."

What this mod does with that access: it reads session usage, tool call names and subagent activity. It adds one section to the system prompt and registers one tool. It makes no network calls and writes no files. The code is in the zip, so you can read it, or ask Claude Code to read it to you, before you load it.

What it adds to a session

The bar and the pane are drawn from numbers Claude Code already keeps. The assumptions panel is the one part that adds tokens: a system prompt section of roughly 200 tokens, plus one tool that Claude calls to report a judgment call.

Where it has been tested

The bar and the pane draw in the terminal CLI, editor terminals included, and in the Desktop app's Code tab. We tested one of those: the Windows terminal, on Claude Code v2.1.289. Mac and the Desktop app are untested.

INSTALL

Put the mod in your skills folder

Download the zip from the guide page: graniteai.co/downloads/terminal-desk-mod.zip

The zip holds one folder named **terminal-desk**. A plugin folder placed in your Claude Code skills folder loads in every session. Have Claude Code unzip it for you, or move the folder there by hand.

Windows

When you are done, this file exists: `C:\Users\<you>\.claude\skills\terminal-desk\.claude-plugin\plugin.json`

Paste into Claude Code

I downloaded terminal-desk-mod.zip to my Downloads folder. Unzip it into my personal Claude Code skills folder so this file exists: `C:\Users\<me>\.claude\skills\terminal-desk\.claude-plugin\plugin.json`. Create the skills folder if it does not exist. The zip already holds one top-level folder named terminal-desk, so do not nest it inside a second terminal-desk folder.

Then confirm that plugin.json exists at that exact path and list the files you unzipped. Do not change any of them.

Mac (untested)

When you are done, this file exists: `~/.claude/skills/terminal-desk/.claude-plugin/plugin.json`

Paste into Claude Code

I downloaded terminal-desk-mod.zip to my Downloads folder. Unzip it into my personal Claude Code skills folder so this file exists: `~/.claude/skills/terminal-desk/.claude-plugin/plugin.json`. Create the skills folder if it does not exist. The zip already holds one top-level folder named terminal-desk, so do not nest it inside a second terminal-desk folder.

Then confirm that plugin.json exists at that exact path and list the files you unzipped. Do not change any of them.

Your part: approve the file commands when Claude Code asks, and read the path it reports back. It should match the one above, character for character.

By hand: unzip the download, then move the **terminal-desk** folder into the skills folder named above. To remove the mod later, delete that folder.

FIRST RUN

Load it and open the desk

If a Claude Code session is already open, reload plugins in it. Otherwise the mod loads the next time you start Claude Code.

Type in an open Claude Code session

```
/reload-plugins
```

The bar appears above the prompt. Then open the pane: type in Claude Code

```
/desk
```

Or try the mod for one session without placing it in your skills folder. Run in your terminal

```
claude --plugin-dir <path to the unzipped terminal-desk folder>
```

REFERENCE

The five commands

Open the pane

```
/desk
```

Hide the bar

```
/desk hide
```

Show the bar again

```
/desk show
```

Estimate cache figures for a five minute cache (API key or cloud provider)

```
/desk cache 5m
```

Estimate cache figures for a one hour cache (the default)

```
/desk cache 1h
```

THE BAR

What each figure on the bar means

The bar is one line above your prompt. It reads left to right, with the cache on the right edge. On a narrow terminal it drops figures from the right end instead of wrapping.

Figure	What it means	What to do with it
Working or ready	Whether Claude is mid-turn or waiting for you.	Glance at it before you type. Ready means the last turn has finished.
Tokens	Tokens read and written since the mod loaded.	Use it to compare one task with another. It starts from zero when the mod loads, which is the start of the session unless you installed it partway through.
Cost	The session's cost as Claude Code reports it, at API list prices. It shows two hyphens until the first measurement.	On Pro or Max, usage is included in your plan, so read this as the size of the session and not as a charge.
Context	How full the context window is, on a 0 to 100 scale with a marker at the current position. Green under 60, amber from 60, red from 80.	When it turns amber, open <code>/desk</code> and see what is filling it. Panel 2 in the next section covers your options.
Tools/min	Tool calls in the last minute. It falls back to 0 when the session goes quiet.	A high number with no visible progress is a cue to read what Claude is doing.
Agents	Subagents running right now.	If it is above 0, work is still going on even when the main conversation looks idle.
Turns	Turns of the main conversation since the mod loaded.	A rough measure of how long this conversation has run.
Errors	Tool calls that failed or were denied.	If it climbs, scroll up and find the failing call before you send more work.
Cache	Hot with the minutes left, or cold. Beside it, the estimated cost to resend the conversation in each state.	Check it before you step away. Panel 3 in the next section explains why.

When you switch to a subagent's transcript, the bar becomes that subagent's: its status, run time, tool calls, tokens and model.

The five panels, and how to act on each

Type `/desk` to open the pane. The panels run top to bottom in this order.

1. What Claude is assuming

The last five judgment calls Claude reported, newest first. Each entry has an id such as A3, the assumption in one sentence, a "because" line with what led to it, and an "affects" line with what would differ if it is wrong. A short notice also pops up in the terminal when a new one is logged.

When to reject one. Read the "affects" line first. Reject when Claude settled on a scope you did not ask for, took the wrong reading of your request, treated a file as in or out of play wrongly, or chose to keep or delete something you would have chosen differently. Claude is asked to report an assumption before it acts on it, so an early rejection saves rework. Leave an entry alone when it is right.

How. Press the number beside the entry, or click its Reject button. The mod drafts the start of a correction in your prompt box, like this:

```
Assumption A3 is wrong ("the assumption, quoted"). Instead:
```

Your part: finish the sentence with what you want and press Enter. Nothing is sent until you do. Rejecting is always your call; the mod never rejects or corrects anything for you.

A rejected entry is marked "you flagged it". If Claude later overturns one of its own assumptions, the old entry is struck through. An empty panel on a small task is normal: Claude reports the judgment calls it notices, and a short task may have none.

2. Where your context went

A stacked bar of what fills the context window (messages, system tools, skills, memory files and so on), biggest first, with a legend, the percentage used and the free space left.

What to do when context is high. Look at which slice is largest before you do anything.

- If messages dominate and the task is still going, compact the conversation so a summary replaces the long history.
- If you are about to start unrelated work, clear the conversation and start fresh.
- If skills, memory files or tools take a large share before you have typed much, that is standing overhead in every session, and it is worth trimming at the source.

Type in Claude Code to compact the conversation

```
/compact
```

Type in Claude Code to start a fresh conversation

```
/clear
```

The context and cache figures describe the main conversation. Subagents get their own status but not their own context breakdown.

3. Prompt cache

Whether the cache is hot, how long until it goes cold, and the estimated cost of your next message in each state. Anthropic's docs call the two states warm and cold; the mod says hot and cold, as the reel does.

What hot and cold mean. Every message resends the conversation. While the cache is hot, that resend is read from the cache. Once it is cold, the whole conversation is written to the cache again, which is the larger of the two figures the panel shows.

What it means for walking away. On a Pro or Max subscription within the plan's included usage, the main conversation's cache lasts one hour after the last request, and each request resets it. So the countdown is how long you can be gone and come back to a hot cache. If you will be back inside it, pick up where you left off. If you will not, and the conversation is large, finish the thought now or compact first, so the cold resend is smaller.

On an API key or a cloud provider the default is five minutes. Claude Code does not tell a mod which one applies, so the mod assumes one hour until you switch its estimate.

Type in Claude Code on an API key or cloud provider

```
/desk cache 5m
```

To go back to the one hour estimate, type `/desk cache 1h`.

4. Subagents

Every subagent in the session with its status (running, done or stopped), type, model, run time and tool calls.

How to act on it. A row that shows stopped ended in an error or was aborted, so read its result before you trust it. A row that has run a long time with a climbing tool count is worth opening. When you switch to a subagent's transcript, the pane shows that subagent's details and its own assumptions.

5. Cost and tokens

Session cost, tokens read, the share of those served from cache, tokens written, and turns. Tokens and turns count from when the mod loaded.

How to read the dollar figure. It is an estimate of what the same tokens would cost at API list prices. On Pro or Max, usage is included in your plan, so nothing here is a charge. A high share served from cache means most of what you resent was read cheaply.

In the reel's demo, one run showed context up 7% after a subagent reported back, the cache hot with 59 minutes left, and an API-price estimate of \$1.28. Those are one demo run's figures. Yours will differ.

FIXES

When something goes wrong

What you see	What it means	The fix
No bar above the prompt	Your Claude Code is older than v2.1.287, or the session was open before you installed the mod.	Run <code>claude --version</code> and update if it is older. In an open session, type <code>/reload-plugins</code> , or start a new session.
Still no bar	The folder is nested one level too deep: skills/terminal-desk/terminal-desk.	Move the inner folder up so that plugin.json sits at skills/terminal-desk/.claude-plugin/plugin.json. Then reload.
Bar was there, now gone	The bar was hidden with a command.	Type <code>/desk show</code> .
Assumptions panel stays empty	Claude has reported none yet, which is normal on a small task. Or you are on a Team or Enterprise plan, or a managed machine, where Claude Code stops a user-installed mod from changing the system prompt.	Give it a task with real choices in it, such as a refactor. On a managed setup the other four panels and the bar still draw.
Cost says estimate	Every dollar figure is an estimate at API list prices, and your plan may bill differently. The cache figures use a table inside the mod, which goes stale when list prices change.	On Pro or Max, treat it as a measure of size. On an API key, your provider's usage page is the record.
Cache figures look wrong	The mod assumes a one hour cache. On an API key or a cloud provider the default is five minutes.	Type <code>/desk cache 5m</code> .
Removing it	The mod is one folder, and it writes no files of its own.	Delete the terminal-desk folder from your skills folder and start a new session.

KEEP GOING

The guide page

The mod download and the install steps for Windows and Mac live on the page: graniteai.co/tools/terminal-desk

We built Terminal Desk to watch our own sessions. See what else GraniteAI makes: graniteai.co/tools/terminal-desk#products