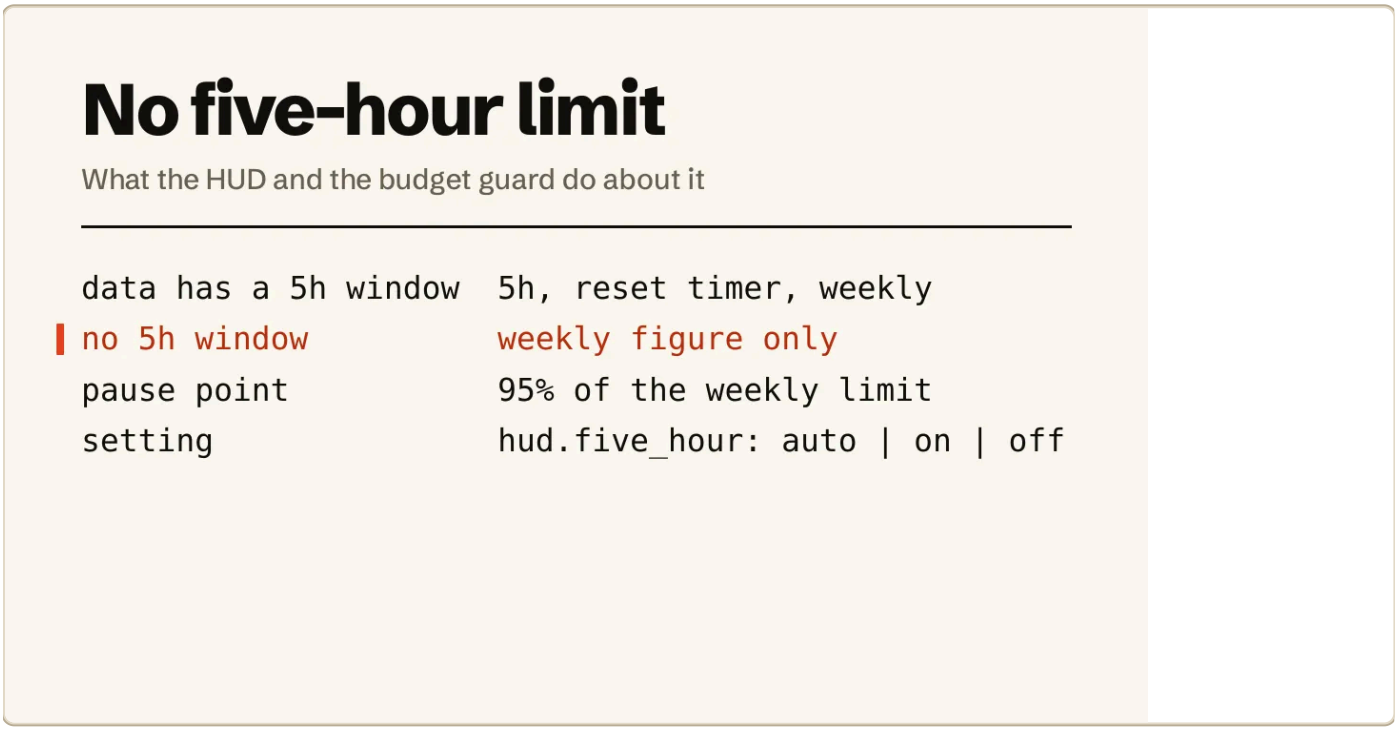
When your plan has no five-hour limit

Mullet 2.5.0 copes with a Claude plan whose usage data has no five-hour window: the HUD drops the five-hour figures and the budget guard paces on the weekly limit. A setting lets you override what it detects.



Published
1 November 2026

Reading time
2 minutes

About
Mullet

Tags
AI, Python, Testing

More on: Mullet

Mullet 2.5.0 copes with a Claude plan whose usage data has no five-hour window. The HUD drops the five-hour figures, and the budget guard paces and pauses on the weekly limit alone. A setting, `hud.five_hour`, overrides what Mullet detects. My notes say some plans send no five-hour window to Claude Code's status line. I have not checked which plans, so treat that as unverified.

What changed

Mullet shows your usage in a small HUD, and a budget guard reads the same numbers to decide when a job should slow down or stop. Both expect two limits: a five-hour window and a weekly one. In 2.5.0 they also cope when the five-hour window is absent.

If the usage data has no five-hour window:

- The HUD drops its five-hour parts: no `5h` figure, no reset timer and no five-hour pressure figure. The weekly figure stays.

- The guard paces on the weekly limit alone and pauses at 95% of it. The reset time it reports is then the weekly one.

If the data does have a five-hour window, nothing changes. The guard still pauses only when 95% of the five-hour limit and 90% of the weekly limit are both used.

When a single reading leaves the five-hour window out, the earlier figure is carried forward only while that window's reset time is still in the future, so a stale value cannot stick.

The setting

It is a config option, `hud.five_hour`, with three values:

- `auto`, the default, which follows the data.
- `on`, which always shows and paces against a five-hour limit.
- `off`, which never does.

The environment variable `MULLET_HUD_FIVE_HOUR` overrides it for one run, and `mullet config show hud` lists the current value.

It is a per-user setting because plans differ. I cannot tell you which plans have a five-hour window and which do not. That is whatever the usage data says, which is why the default follows the data and nothing is hard-coded to one account.

Tests

There are 24 new tests for this. A usage object that lacks the `has_five_hour` field is treated as having a five-hour window, as before. An existing test passes the guard a bare stand-in object with no such field, and I left that test as it was, because it covers exactly the case of usage data with a field missing.

Mullet's whole suite has about 18,700 tests, by pytest's collection count on 1 November. I have not run Mullet against every kind of plan, so if your HUD shows something odd, set `hud.five_hour` to `off` or `on` and tell me what the data looks like.

The code is at https://github.com/tomgould/mullet-public (https://github.com/tomgould/mullet-public).