

Claude Ultra drops its 5-hour cap; a malicious Claude Code extension circulates through Cursor's marketplace

Coding Agents Alpha Tracker

2026-10-31

Claude Ultra drops its 5-hour cap; a malicious Claude Code extension circulates through Cursor's marketplace

By Coding Agents Alpha Tracker • October 31, 2026

Anthropic removed the 5-hour limit on its top plan, and Cursor reported a tampered Claude Code extension in its marketplace. Also: a dispute over whether to compact long agent sessions or let agents retrieve from the full transcript, DHH on reading less agent-written code, and Theo on why the choice of harness matters less.

Claude Ultra: long sessions no longer stop at the 5-hour limit

Anthropic renamed its plans. Max 5x is now “Max” (\$100/month) and Max 20x is now “Ultra” (\$200/month). Verified Ultra accounts lose the 5-hour limit, starting with most existing subscribers. Newer accounts keep a temporary limit until they are verified, and you get an email when it is lifted. Weekly limits stay the same [1]. Claude’s product lead for subscriptions says the Ultra weekly limit stays at 10x Pro. The change is meant to let you “keep building without getting blocked mid-session,” and it rolls out in stages [2]. This matters most for long unattended or parallel agent runs. Before, those runs could hit the 5-hour limit partway through. Now only the weekly limit applies.

Check your Claude Code extension if you use Cursor

Cursor found a modified, malicious Claude Code VS Code extension in its IDE extension marketplace, affecting at most 5,000 users. Cursor took the marketplace down, worked with providers to neutralize the malware, then reinstated

the marketplace. It asks all customers to update to **Claude Code for VS Code 2.1.314**. A blog post explains how to find the malicious package [3]. Alex Albert says the extension was a tampered copy that Anthropic did not build or publish [4].

Compaction vs. retrieving from the transcript

The author of Mainplane (MIT-licensed) measured compaction on a real ~400k-token Opus 5.5 session. It took 38 s and cost \$1.99 with an expired prompt cache, and 29 s and \$0.17 with a cache hit. The summary also tends to drop exact error messages, file paths and numbers [5]. Mainplane takes a different approach. It saves every session as a file, and when the context fills, it starts a new session with one instruction: *“You are continuing session X. Retrieve key context from it as you go.”* The agent then copies exact lines from the old session as it needs them. The author reports 15 s and \$0.10 to resume, with no dependence on the cache [5].

Jediah Katz disagrees: “Done properly, compaction should be cheap and near instant.” The model summarizes itself, so the prompt cache still hits. You start compaction a little early and asynchronously, so the summary is ready when the window fills. His suggestion is to pair compaction with access to the past transcript, not to replace compaction with it. He points to his dynamic context discovery write-up [6].

DHH: agents improving existing apps, and less human code review

In his Kaigi on Rails keynote, DHH says he had an agent apply speed-ups learned from other Campfire implementations to the Rails version. It got about 20× faster in a few hours [7]. Agents also found and fixed security holes in code humans had written [7]. On his Rails agent benchmark, the best agent, Claude Opus 5.5, completes 65% of tasks [7]. He built a PDF-signing app from one prompt (“work roughly like... Apple’s preview app”), made one follow-up change, and shipped it [7].

His position on review: line-by-line reading can’t keep up with agent output. He still reads code that touches billing or data integrity [7]. He expects shared libraries to stay, with fewer abstractions and more inlined or repeated code (“Do repeat yourself”). Agents in his evals don’t know the whole framework and end up recreating features Rails already has [7].

Is the harness still worth choosing? Theo vs. Steinberger

In “Are Harnesses Dead?”, Theo argues that picking a harness barely matters now. Choose for the capability and price you need. He estimates that running Codex models in Claude Code versus in Codex differs by “three to 5% maybe” [8]. One practical detail: T3 Code’s model picker actually picks the harness.

Skills and plugins you set up for Claude Code don't carry over when you switch to Codex, so set them up in each harness [8]. The T3 Code MCP adds tools without replacing a harness's built-in ones. With it, an agent can search T3 Code history, message other threads, or start subagents in another harness [8]. He also says he set Claude to work after noticing unexpected failures. It set up alerting, monitoring and tracing to pinpoint when, where and in which regions failures occurred, and wrote a report he sent to Cloudflare [8].

Peter Steinberger argues the other side. He says to use your own harness, not one "where companies change what you can or can't do at will" [9]. He points to a report that around Oct 27, Claude Code picked up a server-side rule with no setting to turn it off. The rule permits original games, genre games, and games built from your own designs and assets, but not recreations of a specific existing game [10].

Worth studying

- **nesformer** (repo): Drew Breunig used Opus 5.5 to build a NES 6502 CPU emulator into the weights of Qwen3.5-4B, with ROMs running very slowly [11]. It took two prompts. The first was a feasibility study: read the qwenputer article and code, then estimate model size and clock speed. The second: "I think we should build it! Just for fun. It's okay if it's not fast". It asked for weights plus a CLI that runs a ROM [11]. Opus found and used a set of NES CPU conformance tests released 15 years ago [11].
- **Rust rewrite with no Rust developers:** Katelyn Lesse said on Latent Space that Anthropic rewrote its Messages API in Rust. Claude wrote the code, humans wrote the plan and checked that old and new matched on real traffic, and CPU use dropped 40–50% [12].
- **Test MCP connections before relying on them:** OpenAI's finance team found error rates of 40% or more on some big-name MCPs, so it brought the data in-house [13]. It says Codex got it "about 80% of the way" to a product, and it built the rest on ChatGPT's plugin primitives [13].

Short notes

- Theo switched Opus 5.5 from extra high to high effort and says it is "better almost all of the time" [14].
- Antigravity's agent features are now available in Android Studio [15].

Sources

1. X post by @claudeai
2. X post by @mboverell
3. X post by @cursor_ai

4. X post by @alexalbert____
5. X post by @Alezander9
6. X post by @jediahkatz
7. Keynote: Where the tracks go next / David Heinemeier Hansson - Kaigi on Rails 2026
8. Are Harnesses Dead?
9. X post by @steipete
10. X post by @chasmmmmmmmmmmmmmm
11. dbreunig/nesformer
12. X post by @somi_ai
13. Building ChatGPT for Finance — Joseph Kim, OpenAI's Head of Financial Services
14. X post by @theo
15. X post by @antigravity