

Beat Context Rot

The Claude Code Handoff Method for Sessions That Don't Get Dumber

01

The Problem: Your Session Is Rotting

The longer you run one Claude Code session, the dumber it gets. Every dead end, wrong turn, and abandoned idea stays in view, and that noise crowds out the signal Claude actually needs to do good work. We call this context rot. Running `/compact` doesn't fix it — it just squeezes the mess down and drags it forward, mistakes included.

QUICK TIPS

- Compact summarizes old context — it doesn't remove the bad assumptions baked into it.
- If Claude keeps repeating a broken fix, that's context rot talking, not a smarter retry.

02

The Fix: Do a Clean Handoff

Instead of compacting, have Claude write a short `handoff.md` before you stop — then wipe the slate and start fresh from that note. Three moves: tell Claude to write `handoff.md`, run `/clear` (or open a new session), then tell the fresh session to read `handoff.md` and continue. The new session gets the full picture with none of the rot.

QUICK TIPS

- Use this when you're stepping away for hours, or when Claude keeps looping on the same broken fix.
- Clearing without a handoff loses everything — the note is what makes `/clear` safe to run.

03

The Handoff Note: Six Things, No More

Tell Claude to put exactly six sections in `handoff.md`: the goal, the current state, the active files, the changes made, the failed attempts, and the next steps. The failed attempts section matters most — it's the only thing stopping Claude from circling back into a dead end you already ruled out.

QUICK TIPS

- Copy-paste prompt: “Before we stop, create `handoff.md` with: 1) Goal 2) Current state 3) Active files 4) Changes made 5) Failed attempts (and why each failed) 6) Next steps. Keep it short and factual.”
- Never let Claude drop the failed-attempts section to save space — that's the whole point of the note.

Run It in 3 Moves

Write the handoff before you walk away — Claude drops handoff.md into your project. Run `/clear` to wipe the context, or open a new session. Then tell the fresh session: “Read handoff.md and continue from the next steps.” That's it — no context rot, no re-explaining yourself from scratch.

💡 QUICK TIPS

- Commit handoff.md to git — your project now carries its own memory between sessions.
- Make the handoff your last move every session. Future you never has to reconstruct where things stood.

Ready to go deeper?

This handoff trick keeps one Claude Code session sharp. GrowthOS is the system that keeps your whole business sharp — one AI-powered app that replaces the tools you're renting (CRM, content, booking, email) so you stop paying monthly for six different logins. Own it once and run your business from it, or resell it and get paid to set it up for others.

Want to go deeper? →

<https://builds.digicuratoragency.com/growth-os/>