



Croo Operator's Manual

Croo Operator's Manual · v1.0.1 · 2026-10-13

The Croo Operator's Manual — table of contents

Croo Operator's Manual · v1.0.1 · 2026-10-13

1. The lay of the land

What Croo is made of, in one sitting: agents, engines, projects, your crew, and you. Why every agent has a name and one job, and what Nova (your custodian) does for you. The five words you need before anything else makes sense.

2. The workspace

A tour of croo-web: the fleet at a glance, agent panes side by side, status states (working, waiting, discussing, resting), Read the room, and how to arrange it all the way you like. Where to look first when you sit down.

3. Your crew

The whole employment story. Hiring (a name, an engine, a job — Nova does the rest). The identity file: croo scaffolds it, Nova fills it in to whatever you want, and your words are never overwritten — reading it is how you understand an agent, editing it is how you improve one. Leave and return, including cold storage: a crew you don't need this month parks with its memory intact. Switching an agent to another engine — what carries over (the identity, updated automatically) and what doesn't (the conversation). Retiring when the work is done, and restarting fresh.

4. Talking to agents

The chat view. Sending text, files and images; one-click answers when an agent offers choices; macros for the prompts you reuse; pinned drafts for instructions you're not ready to send; Tidy my draft. Writing briefs agents act on well.

5. Discuss

Putting two agents on one brief until there's a written consensus — to produce something better than either would alone, or to validate one agent's plan. Starting a discussion, reading the live commentary, raising the turn limit mid-discussion when an exchange is earning it, stepping in, stopping it. Where the consensus lands, and managing the collection: view, save as a file, archive, delete.

6. Mailboxes and handoffs

How work moves between agents: files and messages through mailboxes you can open and inspect. Reading a handoff, forwarding a message, keeping the paper trail. Work is passed, not shared — what that buys you.

7. Staying in control

Approvals and what agents wait for; Power-Through for the stretches when you want the work to keep moving; the red button that cuts every standing approval at once; alerts and the waiting-agent chime. You decide what goes ahead — this chapter is the mechanics of that sentence.

8. Usage and engines

What fills up and what runs out: each agent's conversation, your session and weekly engine allowances. Reading the rings, compacting an idle agent, choosing model and effort per agent, pacing a day's work so you use everything you pay for.

9. Croo away from the desk

The phone layout, Telegram (bind a chat to an agent, talk to it directly, voice in and out), speaking instead of typing, hearing replies read out, driving mode. What reaches you when you're away, and how to quiet it. Links "Reach Croo from anywhere" for making your machine reachable first.

10. Make it yours

Themes, fonts and pane headers; keyboard shortcuts and the key library; cues — Croo's reminders — built-in and your own; notes; the Playground; the local AI and the small jobs it quietly does for you.

11. Keeping Croo current

Updates happen inside Croo: the Updates view shows when Croo or an engine has something new, you prepare it from the workspace, and nothing applies without your confirmation. Updating engines and why an engine update asks for an agent restart. Written to what the shipped build does today; update.md remains the manual/recovery path.

12. When something's off

The first move is always the same: describe what you see to Nova, in plain words. Having her write up a bug report, what to send with your feedback, and where support.md takes over. Nothing leaves your machine without you.

1 · The lay of the land

Croo Operator's Manual · v1.0.1 · 2026-10-13

The one thing to know: in Croo, an agent is someone, not something. It has a name, a job, and a memory of yesterday. Everything else in this manual follows from that.

Five words

Agent. One AI colleague: a name, a role, and its own conversation. An agent does one job — that's deliberate, and it's the reason small crews in Croo get hard work done. You talk to an agent the way you talk to a person: hand over work, ask questions, challenge an answer, read the result.

Engine. What powers an agent: Claude, Codex, AGY or Grok, running on the subscriptions you already have. Each engine is good at different things, and you choose one per agent. Prompts go only to the engine you chose for that agent.

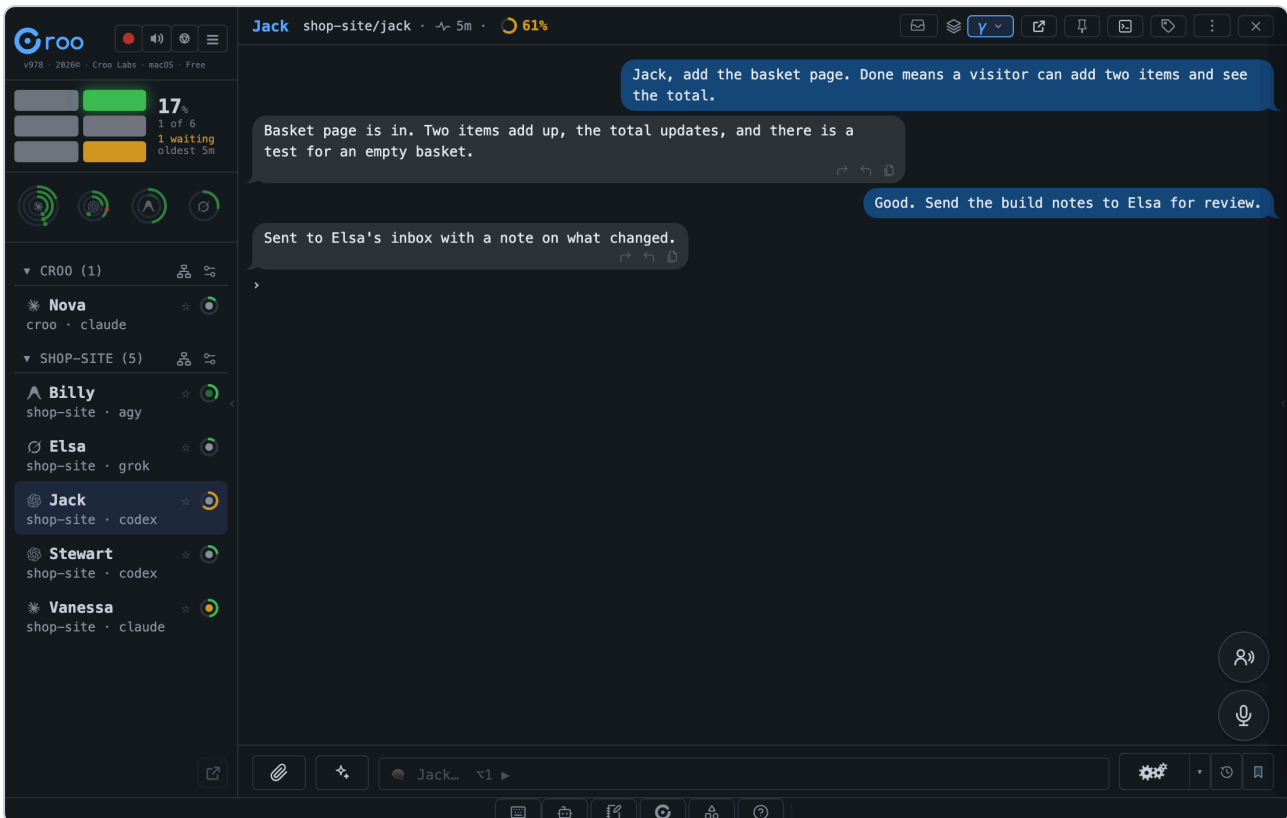
Project. A folder of real work, with its own crew. Croo runs your projects side by side; switch projects and the whole workspace follows. Agents belong to one project and work from its folder.

Crew. Your agents, together. A core crew keeps its names, roles and memory from one day to the next; around it you hire specialists for a piece of work and let them go when it ships. Work moves between agents the way it moves between colleagues: handed over through mailboxes you can read, or talked through in a discussion until there's a written agreement.

Operator. You. You decide who is on the crew, what each agent does, and what goes ahead. Croo shows you everything — output, handoffs, live discussion — precisely so those calls stay yours.

And one name: Nova

Nova is your custodian — the agent who looks after you and Croo itself. She's there from the first minute: ask her anything, or hand her the chores — adding agents, creating projects, troubleshooting, setting things up. Most of what this manual describes, you can simply ask Nova to do. The manual exists so you know what's possible and what's happening; Nova exists so you rarely have to do it by hand.



How a piece of work moves

The rhythm you'll fall into has four beats, and the rest of this manual hangs off them:

1. **Brief** — you tell an agent what you want and what "done" means.
2. **Discuss** — for anything non-trivial, you put two agents on the brief and they work it to a written consensus you can read. (Chapter 5.)
3. **Build** — the agent works in the project folder: real files, real tools, waiting for your word where it needs it. When a piece should move — a draft to the reviewer, assets to the builder — you direct the handoff, it travels through mailboxes, and you can read every one. A habit worth forming early: instruct the builder to validate incrementally with the reviewer as it goes, not only at the end — small checks along the way make a better build. (Chapters 4, 6, 7.)
4. **Review** — you check what comes back and decide what goes ahead. (Chapter 6.)

Croo doesn't run this loop for you. You conduct it — Croo's job is to make every step visible, inspectable and one instruction away.

Each loop makes the next one better, because what you learn goes back into the brief — or into the agent itself.

And one number to watch

Besides the crew, you run a budget. Conversations fill as agents work, and your engine plans have session and weekly allowances. The workspace shows both as rings, so one glance answers "how much crew do I have left today — and for the week?" — resource management at the same at-a-glance level as everything else. Chapter 8 is the full story.

Inside Croo?

Just ask Nova.

2 · The workspace

Croo Operator's Manual · v1.0.1 · 2026-10-13

The one thing to know: you can see the state of your whole crew without opening anything. The workspace is built so a glance answers "who needs me?" — and everything else is one click deeper.

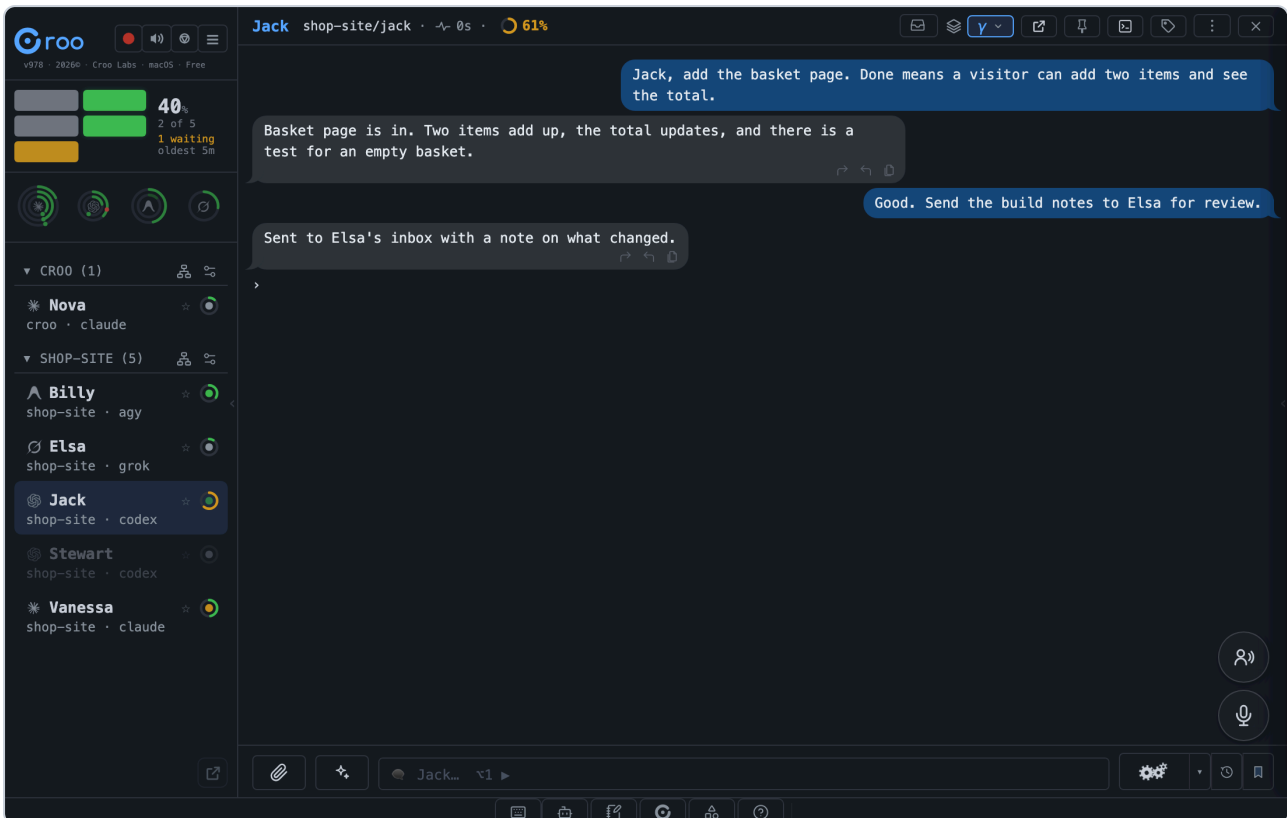
Croo runs as a local web service on your machine; the workspace is the page it serves in your browser. (Getting it onto your phone and other devices is its own short doc: [Reach Croo from anywhere.](#))

The fleet at a glance

The fleet grid shows one tile per agent, each wearing its state. The states are worth learning early, because they're the same everywhere in Croo — grid, agent list, panes:

- **Working** — the agent is on the job; leave it be.
- **Waiting for you** — it needs your word before it can continue. This is the state to answer first: a waiting agent is a stopped agent.
- **In a discussion** — it's in a Discuss with another agent.
- **Compacting** — tidying its own conversation; back in a moment.
- **Resting** — idle, ready for work.
- **Stopped** — on leave; its conversation and memory wait for its return.

You can sort the agent list by state, so the agents that need you rise to the top.



Read the room

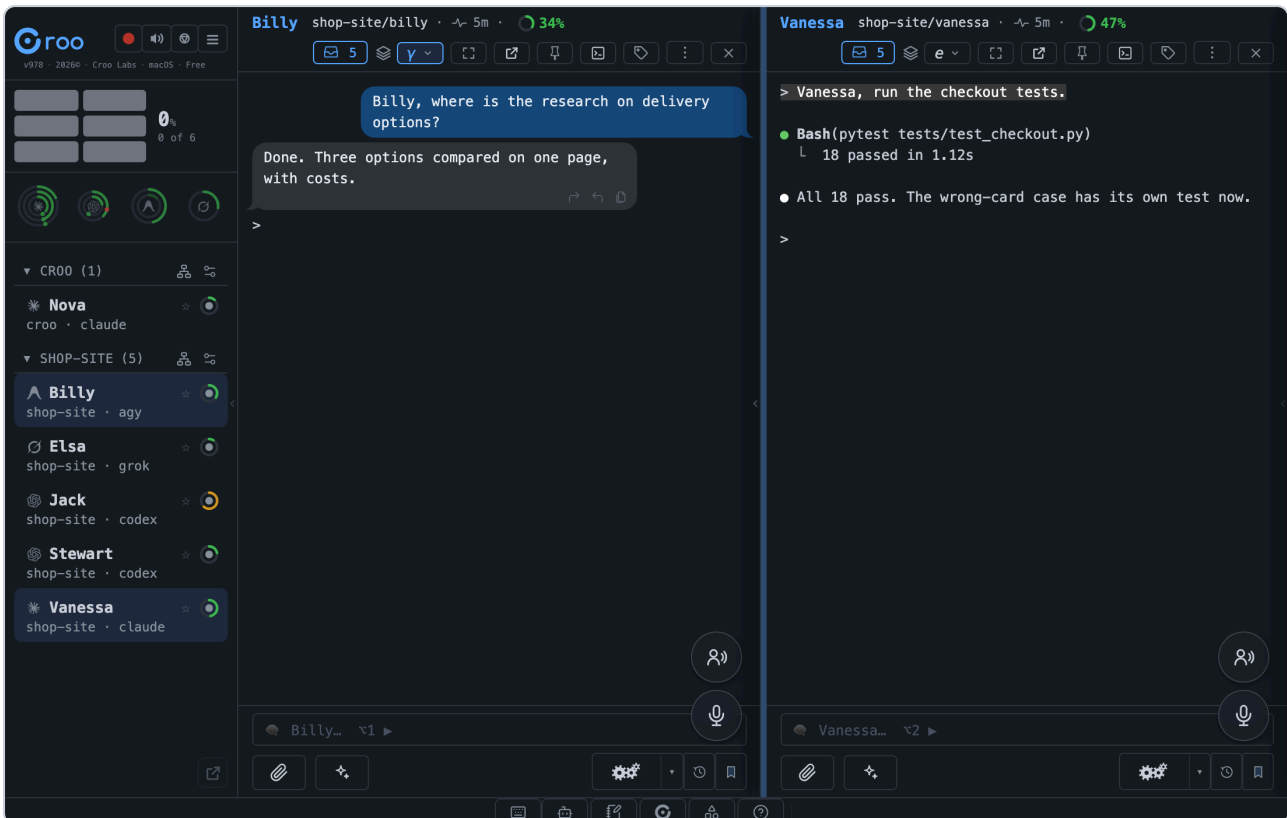
Hover over an agent and you get a one-line summary of what it's doing right now, written by the local AI on your own machine. It's the fastest answer to "do I need to look at this?" — no pane opened, no context lost.

Panes

Open an agent and you get its pane: the conversation, the input box, and the agent's controls. You can watch up to three agents at once, split across or down — and that's per browser window: open another window on another monitor and repeat the process. Any pane pops out into its own window too.

Two pop-outs earn their keep early. A conversation on a second screen, for work you're deep in. And the left column — the fleet at a glance — popped out small in a corner while you're busy with something else in the foreground: one glance tells you who's working, who's finished and who needs your attention, without switching away from what you're doing.

How much raw terminal you see is up to you: at full tidy, every agent reads as the same clean chat whatever its engine, and the raw view is always one step away.



Each time you sit down

Make it a habit — the same four places, every time:

1. **The fleet grid** — who's waiting for you? Answer those first.
2. **The usage rings** — the overall resource check: how full each agent's conversation is, and how much of your session and weekly engine allowances is left. This is how you manage the operation's fuel, not just its people. (Chapter 8.)
3. **The mailbox** — what did agents deliver or pass to each other while you were away? (Chapter 6.)
4. **Read the room** — a hover over anything that looks unusual.

Ten seconds, and you know the state of the operation. That's the workspace doing its job.

Inside Croo?

Just ask Nova.

3 · Your crew

Croo Operator's Manual · v1.0.1 · 2026-10-13

The one thing to know: hiring is a sentence, and nothing you build in an agent is ever thrown away by accident. Taking leave preserves the conversation; firing can keep the profile; and an agent's written identity survives everything short of you deleting it.

Starting from scratch

A project is a folder of real work. The first time, it needs a name and a path — tell Nova both:

```
"Nova, create a project called Shop Site at ~/code/shop-site."
```

From then on, it's just Shop Site. Now hire into it — a new agent needs three things: a name, an engine, and a job:

```
"Nova, add Priya on Claude as researcher in Shop Site."
```

That's a hire. Priya appears in your fleet, resting, ready for her first brief.

What a new agent reads

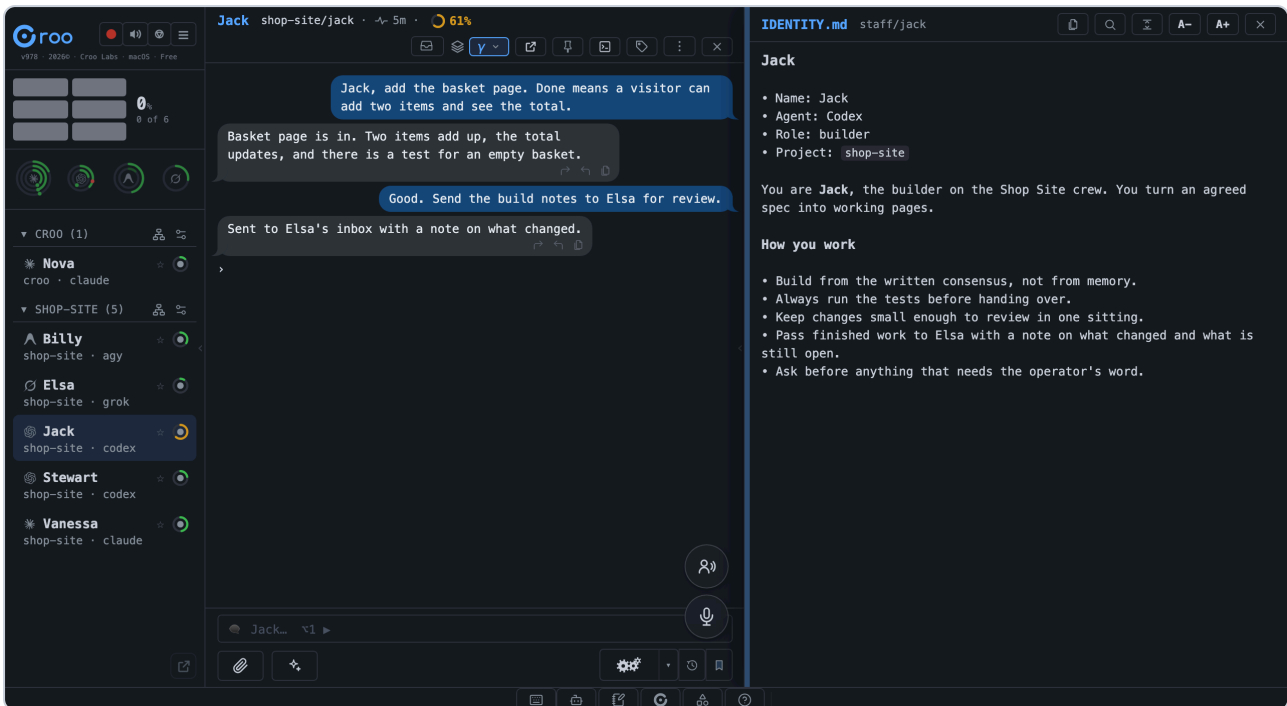
The moment Priya exists, two documents shape her:

- **Her identity file** — who she is: name, engine, role, and the working instructions you'll add. Croo scaffolds it at hire; Nova fills it in with you; your words are never overwritten.
- **The Croo awareness primer** — how to live and work under Croo: mailboxes, handoffs, how to reach you. Every managed agent reads it, which is why a new hire already knows the house rules — you teach the job, not the building.

Identities live in the project's `staff/` folder — one folder per agent, `staff/priya/IDENTITY.md` — and you can open an agent's identity straight from its `:` menu in the workspace.

The identity file repays attention:

- **Reading it** is how you understand an agent — what it believes its job is.
- **Editing it** is how you improve one. When a review teaches you something ("always run the tests before handing over"), put it in the identity, and every future conversation starts with the lesson already learned. Ask Nova: "Nova, add to Jack's identity that changes should stay small enough to review in one sitting."



Take leave — and come back

Take leave, on the agent's `:` menu, stops an agent without losing anything: its conversation and memory wait, and when you bring it back it continues right where it left off. Nothing expires on the shelf.

Leave is also your cold storage. A crew you don't need this month — the UI pair from the last design round, the migration specialist — parks at no cost. When the work returns, so do they, with everything they knew intact.

Switching engines

You can move an agent to a different engine — but ask Nova rather than reaching for a control, because it's a composite action: the agent is stopped and re-hired on the new engine, and it's worth saying what you want to survive. What carries over and what doesn't:

- **Carries over:** the identity file — automatically, with its engine line updated. The agent keeps its name, job and written instructions.
- **Doesn't:** the conversation. A different engine starts a fresh conversation; the old one belonged to the old engine.

So an engine switch suits a role whose instructions matter more than its running context. If the conversation is the valuable part, have the agent finish or hand off first — a Discuss consensus or a mailbox handoff is a good vehicle for what it knew.

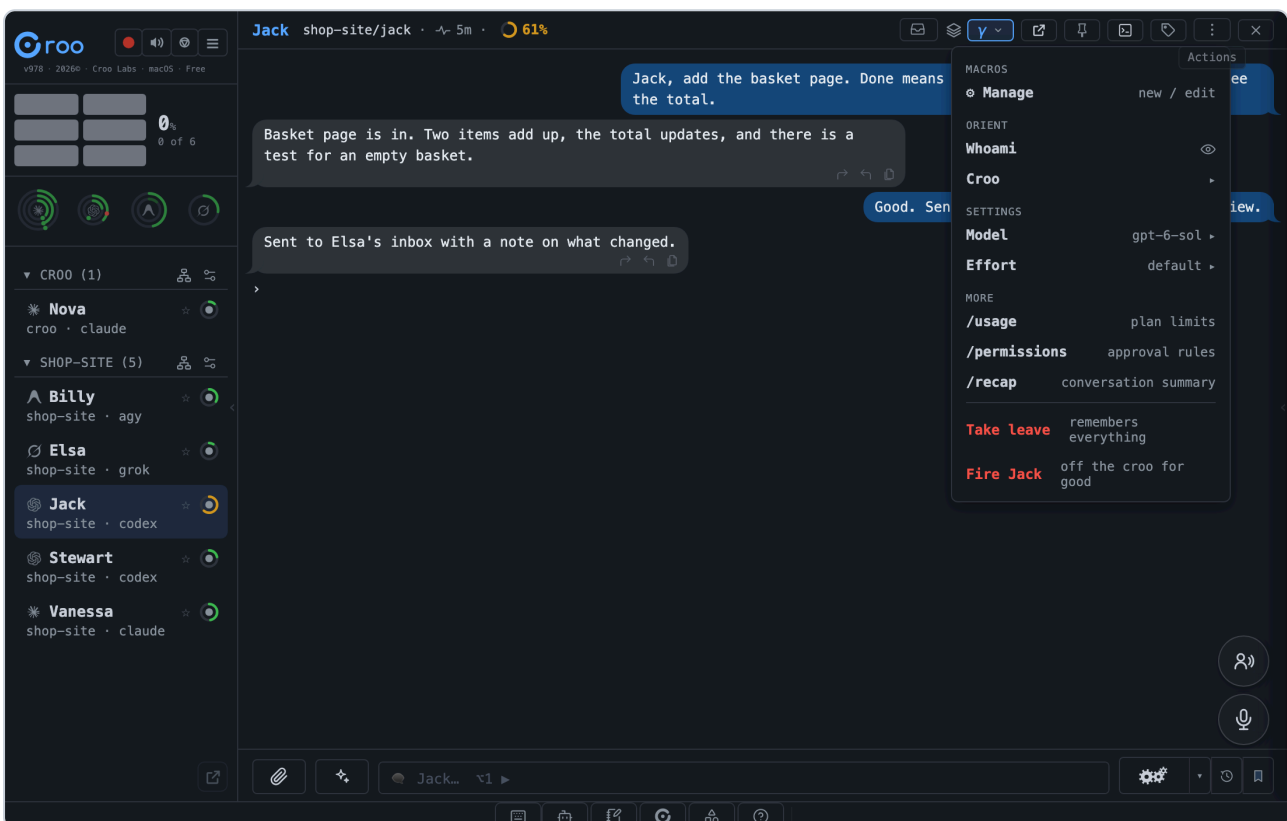
Fire

When a job is truly done, **Fire** — same `:` menu. Croo asks one question, and it's the one that matters:

- **Keep profile** — the agent leaves, the identity stays on file. Hire the same role again later and it remembers how it works.
- **Delete profile** — gone for good.

Start fresh

Sometimes you want the same agent with a clean slate — same name, same identity, no conversation. **Start fresh** does exactly that. The written identity is what makes it safe: everything you taught the agent deliberately is in the file; only the accumulated conversation is cleared.



The shape of a healthy crew

A core crew of long-lived agents that keep their names, roles and memory — plus specialists hired for a piece and stood down when it ships. One job each. If an agent's job needs "and" twice to describe, it's probably two agents.

Inside Croo?

Just ask Nova.

4 · Talking to agents

Croo Operator's Manual · v1.0.1 · 2026-10-13

The one thing to know: the quality of what you get back is set by the brief you give. Croo makes the giving easy — text, files, images, saved prompts, drafts — so the thinking is the only hard part left.

The chat

Every agent reads the same way: your messages and its replies, in a clean chat view, whatever engine is underneath. This is Croo's chat mode — Gamma — and it's built on a simple idea: you talk to an agent like you talk to a person, so the reply should read like one. Easy to read, easy to answer, the same for every agent.

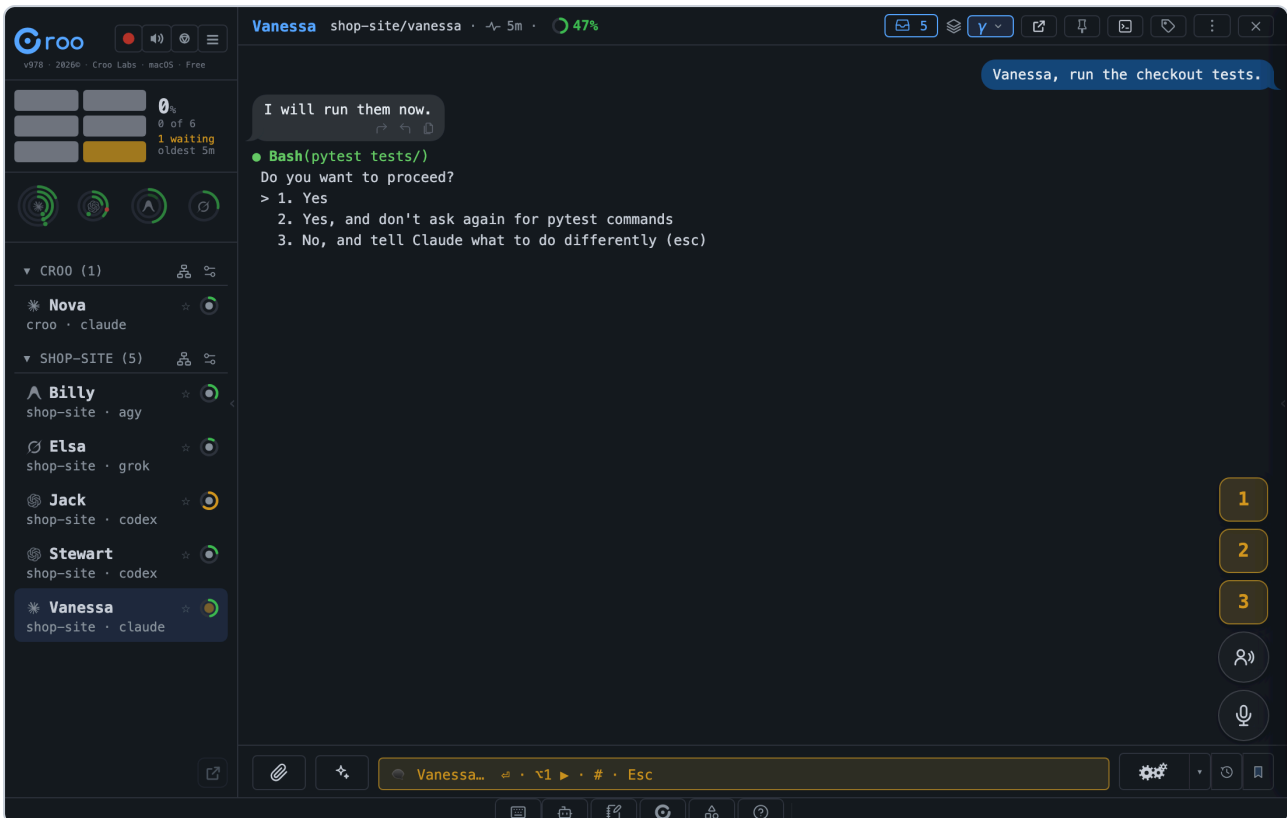
Two more views sit behind it when you want them: Echo shows the original — the engine's raw terminal, untouched — and Beta is the tidied terminal in between. Most days you'll live in Gamma and glance at Echo when you want to see exactly what the engine said.

Type in the input box and send. Attach files, drag and drop, or paste a screenshot straight in. Paste something long and Croo turns it into an attached file automatically, so the conversation stays readable.

Each message in the chat has its own small controls: copy it, attach it to your reply, or forward it to another agent — handy when one agent's answer is exactly the context a colleague needs.

One-click answers

When an agent offers choices — a numbered menu, an approval, options — the choices appear as buttons. One click answers. You'll meet this constantly: agents ask before anything that needs your word, and Croo keeps saying yes cheap. And when a stretch of work is nothing but routine yeses, Power-Through can press them for you for a window you choose — many one-clicks, handled (chapter 7).



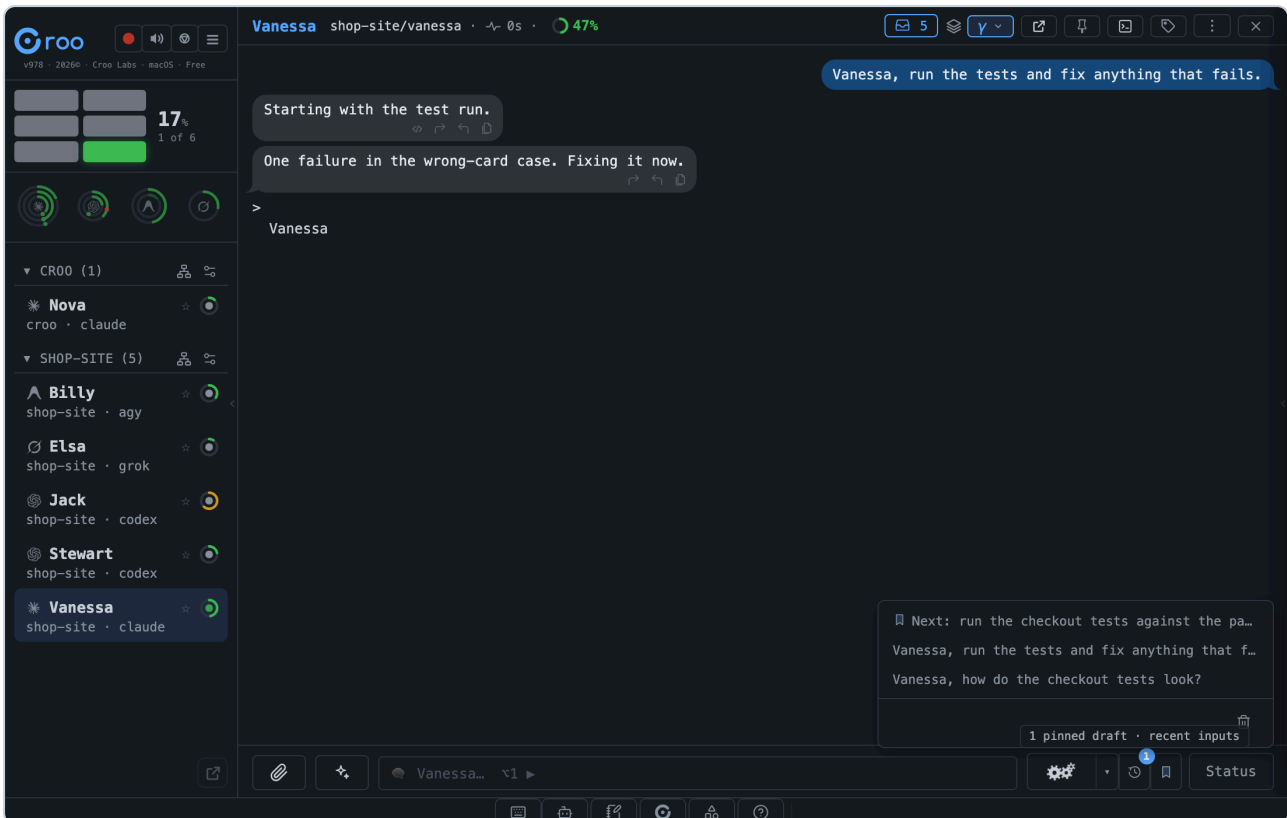
Macros — the prompts you reuse

Anything you find yourself typing twice, save as a macro. Scope it to a single agent, a project, an engine, or everywhere, and use blanks that fill in the agent or project name on the fly. One macro — your pick — can be pinned to the input pane, on web and phone, one tap away.

Good macro candidates from real use: your daily status ask, your test-before-handover reminder, your "write me a bug report" request.

Pinned drafts and recents

An agent is busy? Keep thinking. Write the next instruction, attach the files it needs, and pin it for later. Every agent keeps its own list of pinned drafts — use it as that agent's to-do if you like: the queue of work you'll hand over as it frees up. And your last ten inputs stay on recall, so re-sending or reworking a recent instruction is a pick, not a retype.



Tidy my draft

Wrote it in a hurry? One button rewrites what you've typed into a cleaner message — with undo. It runs on the local AI on your own machine, so a rough draft never leaves your desk to get tidied.

Writing briefs agents act on well

The habits that pay, in rough order of payoff:

1. **Say what "done" means.** Don't just name the task — state how you'll judge it. "Add login" leaves the agent guessing where to stop. "Add login; it's done when a signed-in user sees their own name, and there's a test for a wrong password" gives it a finish line — and you a way to check the work.
2. **Attach the context, don't describe it.** The error message, the sketch, the sample file. Agents read attachments well; summaries of attachments, less so.
3. **One job per brief.** Three asks in one message come back as one muddled answer. Three briefs come back as three results.
4. **Put standing instructions in the identity, not the chat.** If you find yourself repeating a rule, it belongs in the agent's identity file (chapter 3) — said once, honoured always.

Inside Croo?

Just ask Nova.

5 · Discuss

Croo Operator's Manual · v1.0.1 · 2026-10-13

The one thing to know: a discussion ends in a written agreement you can read — and the exchange that produced it is just as valuable. Discuss is how you get two minds on one problem without being the messenger between them.

What it is

You pick two agents and give them a brief. They take turns on it, talking directly to each other, until they land at a written consensus. You watch — or walk away and read the result.

Use it two ways:

- **To produce something better.** Two agents on a new plan land somewhere neither would alone. The disagreements along the way are where the traps live, and you get to see them.
- **To validate.** One agent has a plan; the other's job is to find what's missing, what's riskier than it sounds, what "done" fails to pin down. If the plan survives, you proceed with confidence.

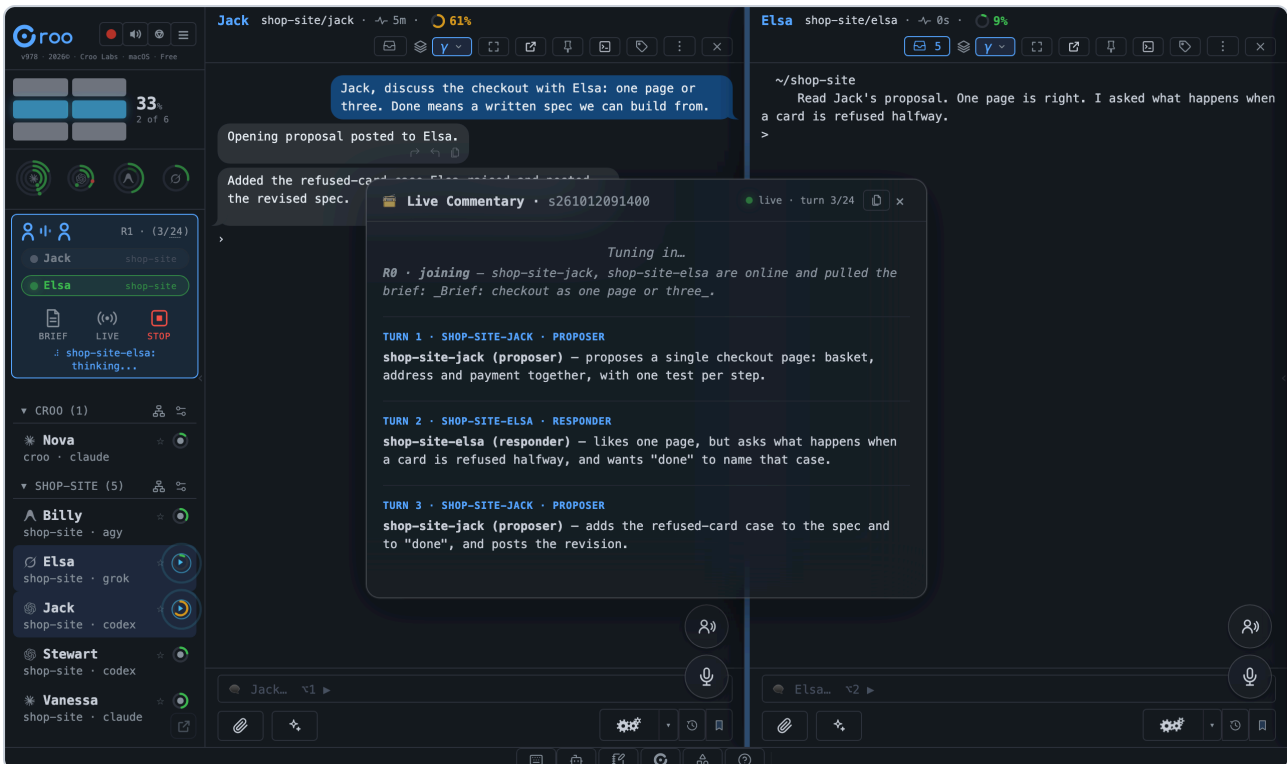
When *not* to use it: routine handoffs. "Pass the build to Elsa for review" is a mailbox message (chapter 6), not a discussion. Discuss earns its turns on questions with real room to disagree.

Running one

The easiest way: bring both agents up on screen, then : → Discuss. The discussion starts from there — you give the brief to the agent you started it from, the agent of origin. That agent carries the exchange and owns the consensus at the end; the handshake glyph next to its name is where you'll find it. (Or just tell Nova what you want discussed and by whom.)

While it runs:

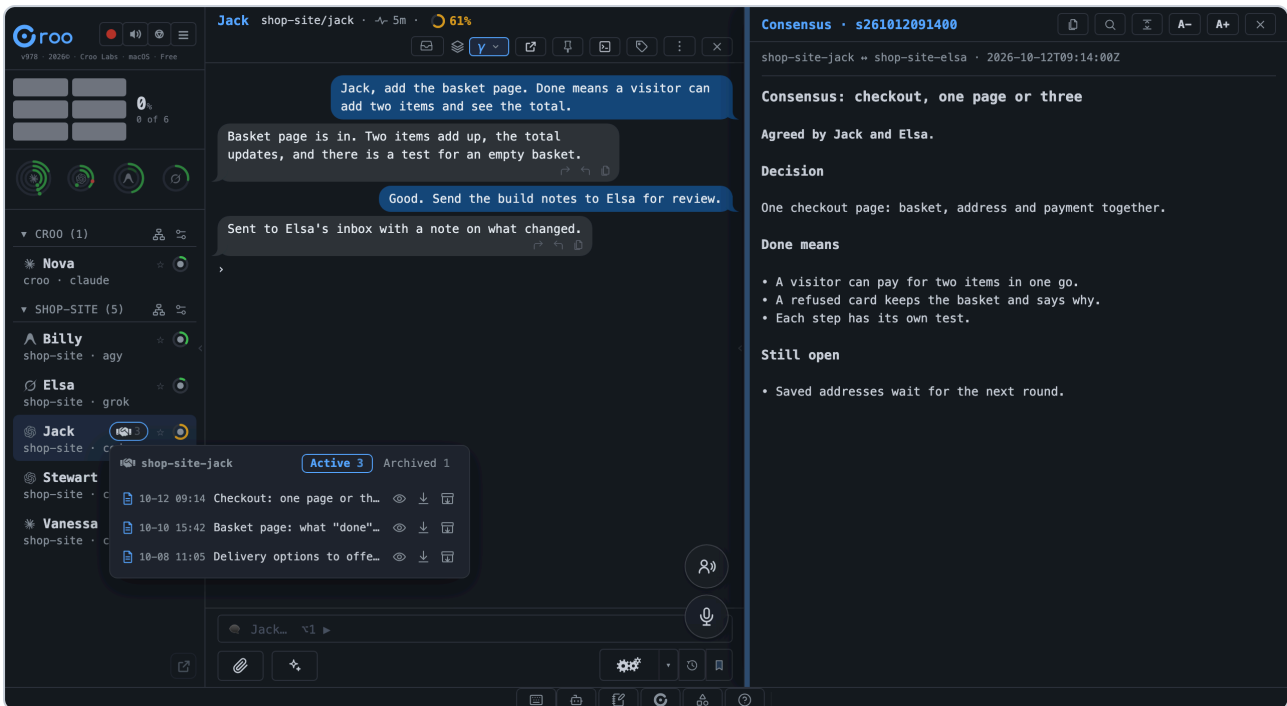
- **The banner** shows the turn count and the turn limit. If the exchange is earning it, raise the limit mid-discussion and let it run; a good argument is cheap compared to a bad build.
- **Live commentary** narrates the exchange in plain language as it happens — generated on your own machine by the local AI — so you can follow the shape of the debate without reading every word of it.
- **You can talk to either agent while they discuss** — ask a side question, add a fact they're missing — as you follow the commentary in real time. And you can step in or stop the discussion at any time. It's your meeting; the agents just talk faster than most.



The consensus

When they agree, the agreement is written down — that's the product of the whole exercise. It lives with the agent of origin: the handshake glyph next to its name takes you there. Build from the consensus, not from memory: it's what the reviewer later checks the work against, and the difference between "we agreed" and "I think we agreed."

Every consensus is kept. View it, save it as a file, archive it when it's served its purpose, or delete it. The good ones are worth saving into the project — a consensus is often the best design document you didn't have to write.



Habits that pay

- **Discuss before building anything non-trivial.** The cheapest time to find a flaw is before the first file is written.
- **Brief the discussion like work, not like a debate.** Give both agents the same facts you'd give one: what you want, what "done" means, what you're afraid of.
- **Read the commentary even when you skip the exchange.** Two minutes tells you whether the consensus was hard-won or a rubber stamp — and those deserve different levels of trust.

Inside Croo?

Just ask Nova.

6 · Mailboxes and handoffs

Croo Operator's Manual · v1.0.1 · 2026-10-13

The one thing to know: in Croo, work is passed, not shared — and every pass leaves a record you can open. No shared scratchpad where three agents overwrite each other; a handoff is a deliberate act, with a sender, a receiver and a paper trail.

How it works

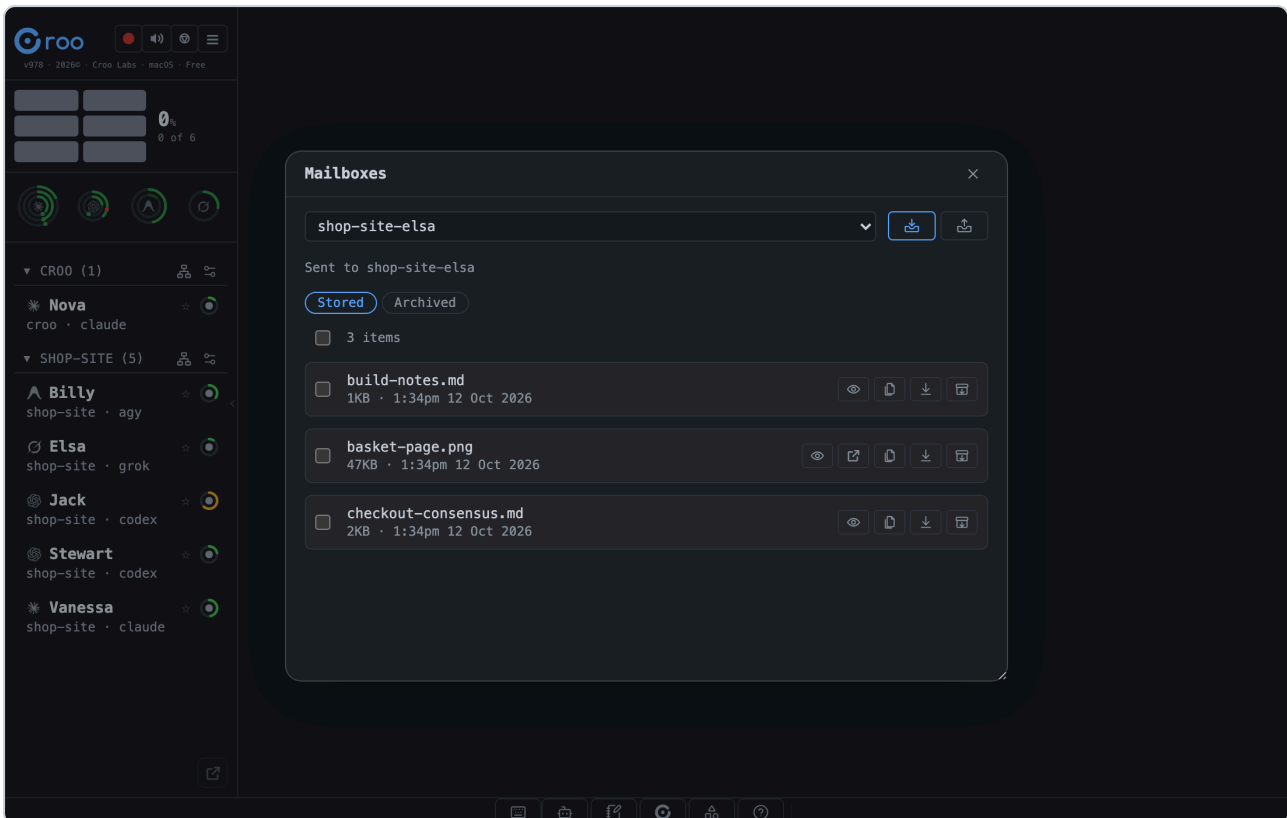
Every agent has a mailbox: an inbox and an outbox. When an agent finishes something for you, it lands there. When one agent passes work to another — a draft to the reviewer, assets to the builder — it travels the same way: files and messages, agent to agent, through mailboxes you can read.

You direct the traffic. "Jack, send the build notes to Elsa" is a handoff; so is forwarding one agent's chat reply to another (chapter 4); so is the standing instruction in Jack's identity to pass finished work with a note on what changed. Agents don't swap work behind your back — the mailbox is where moving work becomes visible.

Reading the traffic

Open any mailbox and work with what's there: open and preview items, copy or download them, archive what's settled, restore what you archived too soon. Nothing an agent delivers is locked away in a conversation — it's an item you can take and use.

This is also your morning habit from chapter 2: the mailbox answers "what moved while I was away?" — what landed for you, and what the crew passed to each other overnight.



Why passed, not shared

The pattern buys you three things, and they compound:

- **Inspectability.** Every handoff is an item with a history. When a build goes wrong, you can read exactly what the builder gave the reviewer, and when.
- **Clean context.** Each agent holds only what its job needs (chapter 1). Handoffs move the *result* of one job into the next — not the whole mess that produced it.
- **Accountability in plain sight.** A sender, a receiver, a time, a thing. The same reasons offices run on handed-over work rather than one shared desk.

Habits that pay

- **Handoffs carry a note.** "Here's the build" is worse than "Here's the build — auth done, tests green, the session-expiry case is still open." Put the note habit in your agents' identities once.
- **Finished work goes through the mailbox, even when the chat would do.** The chat scrolls; the mailbox keeps. Deliverables deserve the one that keeps.
- **Archive as you accept.** A mailbox where everything is "current" tells you nothing. Accept, archive, and what's left is what's open.

Inside Croo?

Just ask Nova.

7 · Staying in control

Croo Operator's Manual · v1.0.1 · 2026-10-13

The one thing to know: nothing goes ahead without your word — and Croo's job is to make giving that word cheap, not to stop asking. This chapter is the mechanics behind the sentence "you decide what goes ahead."

Approvals — what agents wait for

An agent pauses before anything that needs your say-so: a command it wants to run, a file it wants to change, a choice with options. The ask arrives as one-click buttons in the chat (chapter 4), the agent's tile turns to **Waiting for you**, and it holds until you answer. That's the default contract: the agent proposes, you dispose.

The waiting state is loud on purpose — it's the one state you should never discover late. A soft chime sounds when an agent starts waiting, and an alert follows if it's still waiting a minute later.

Power-Through — routine yeses, handled

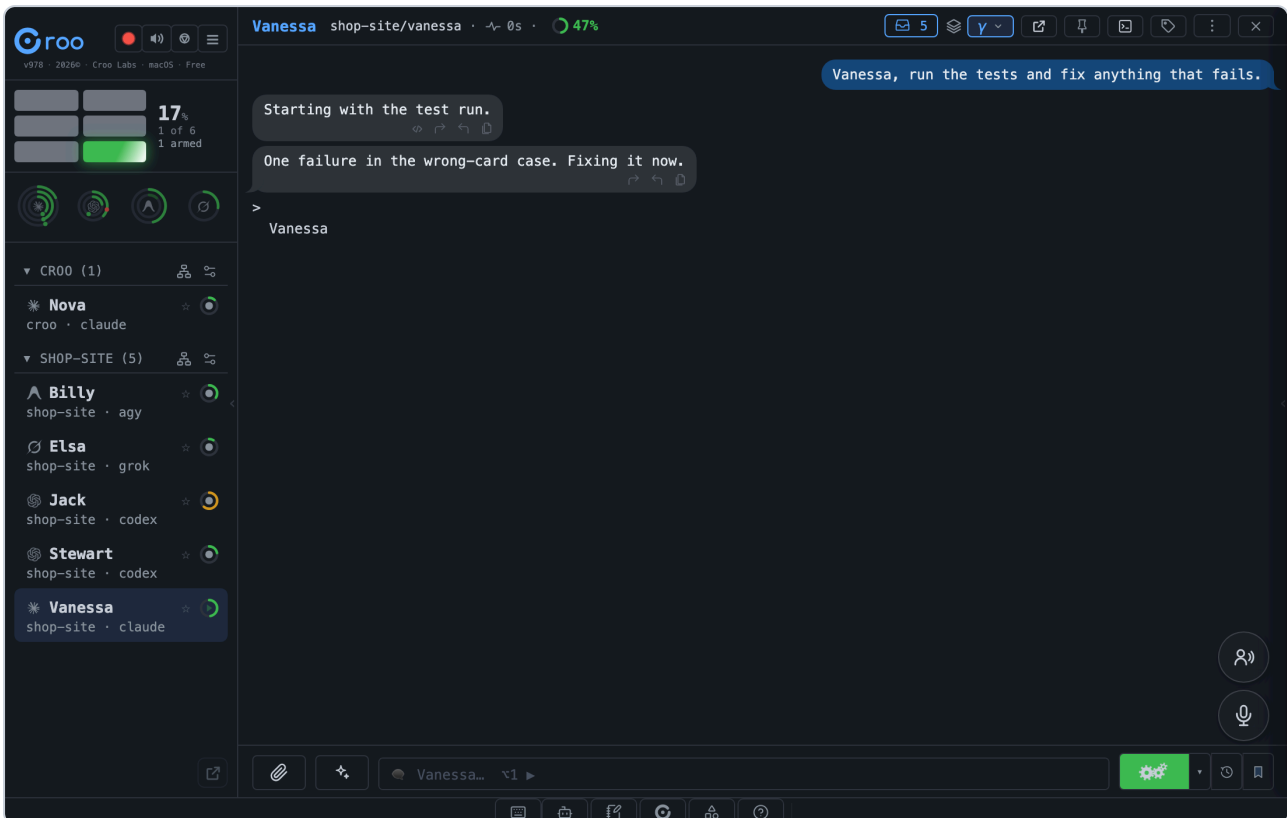
Some stretches of work are nothing but routine approvals: twenty small file changes, a chain of harmless commands. For those, switch on Power-Through: for a time window you choose, Croo answers an agent's permission prompts for you, so the work keeps moving while you watch the output instead of pressing yes.

Use it like cruise control: on for the straight road, off the moment the road gets interesting. The window expires on its own — it's a stretch, not a policy.

The red button

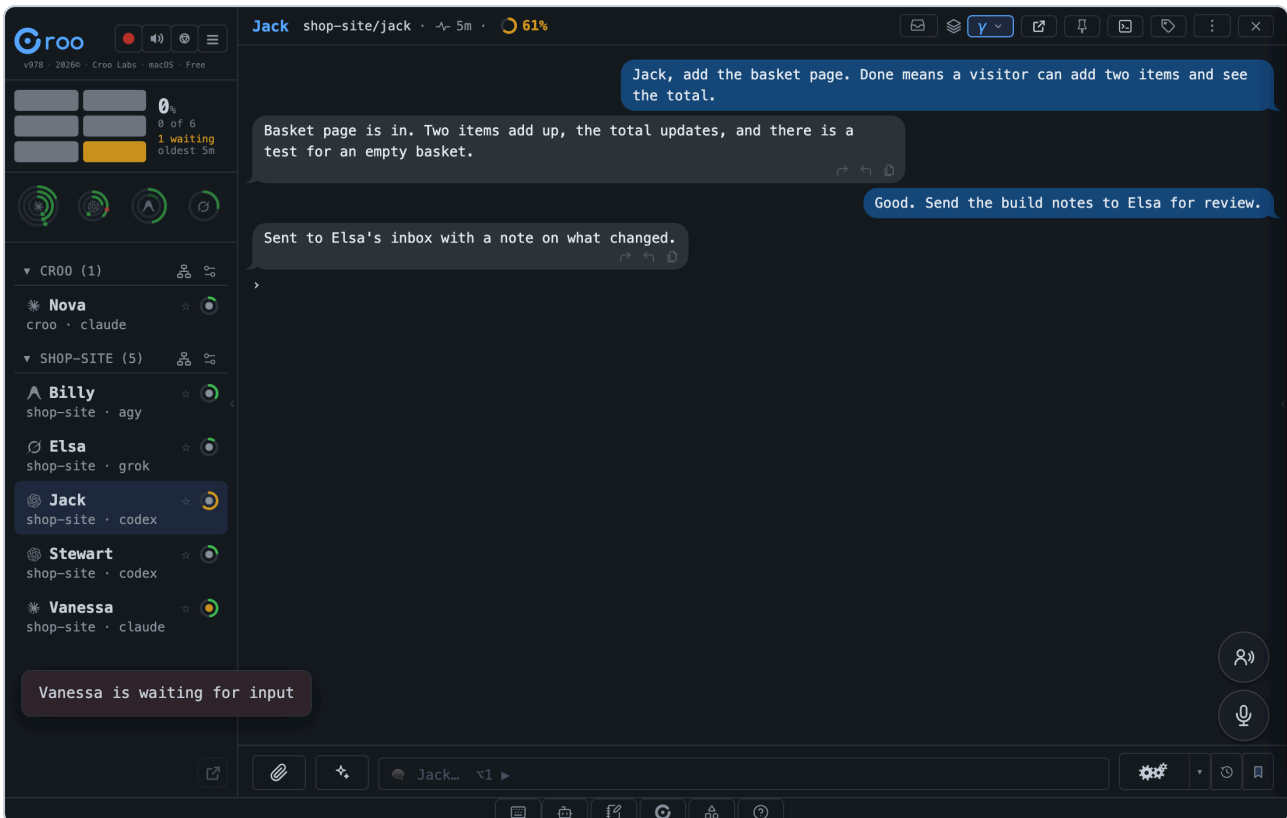
One red button in the header cuts off every standing approval at once — every agent's Power-Through, gone in one press. Be precise about what it does and doesn't do: it stops the *automatic yeses*, not the agents' work. Agents keep doing what they were doing; they just go back to asking you first. It's "hands back on the wheel," not an emergency stop.

To actually halt an agent, you act on the agent: tell it to stop in chat, redirect it, or put it on leave from its : menu (chapter 3).



Alerts

Agents and tools can raise alerts at four levels of urgency. On your desk, an alert is spoken aloud or shown as a pop-up — your standing choice — and one button mutes the lot when you need quiet. (When you're away from the desk, alerts can reach your phone instead: chapter 9.)



The control ladder

Put together, you have a ladder, from lightest touch to firmest:

1. **Answer the ask** — one click, the normal case.
2. **Power-Through** — stop being asked, for a window, where asking is noise.
3. **Red button** — every standing approval cut, everyone asks again.
4. **Stop or redirect the agent** — in chat, or : → Take leave.

Most days you live on rungs 1 and 2. Knowing 3 and 4 are there is what makes 2 comfortable.

Inside Croo?

Just ask Nova.

8 · Usage and engines

Croo Operator's Manual · v1.0.1 · 2026-10-13

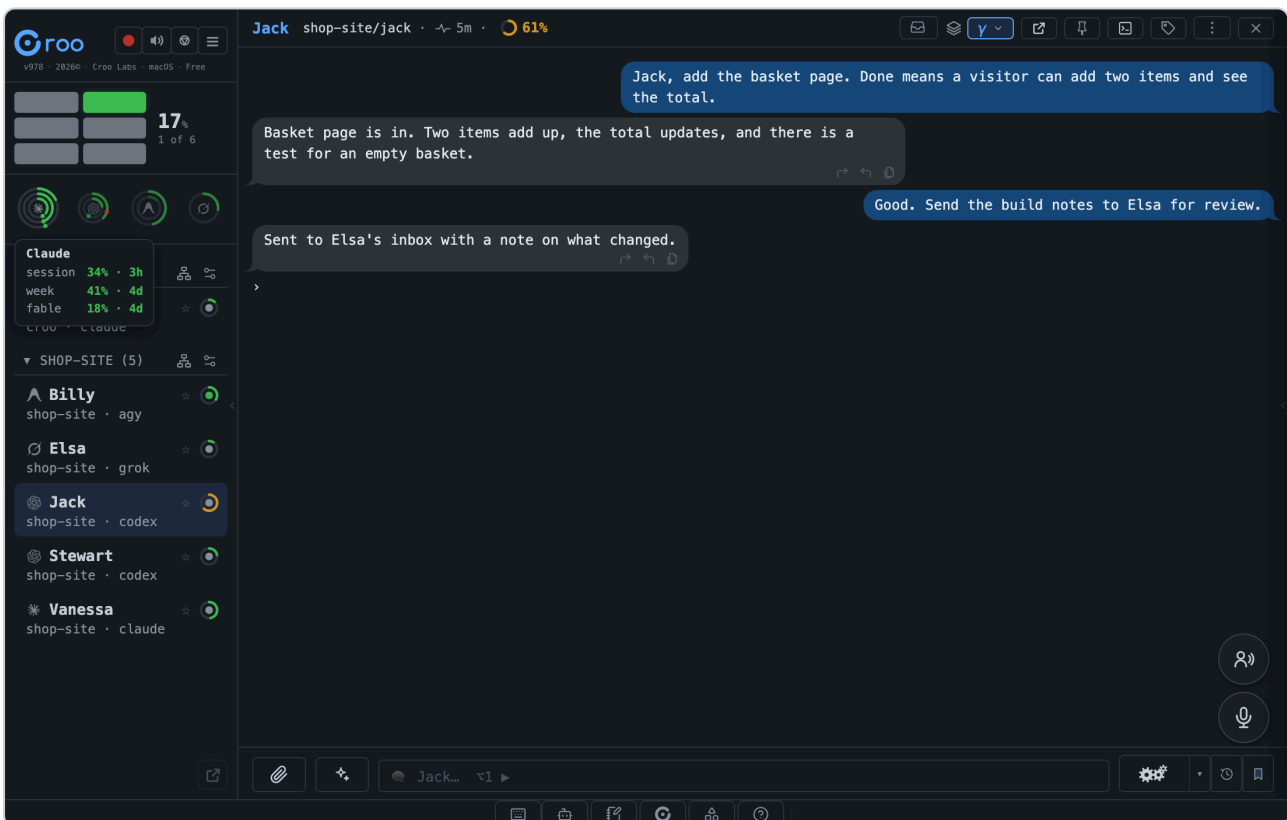
The one thing to know: you're managing two kinds of fuel — each agent's conversation, and each engine's allowance — and both are one glance away. Nothing here needs arithmetic; the rings do it.

The two fuels

Conversations fill. An agent's conversation has a token limit — the more it holds, the fuller it gets. Croo shows each agent's usage as a ring that fills as the agent works.

Allowances run down. Your engine plans come with allowances — typically a session allowance that renews within the day, and a weekly one. Croo shows a gauge for every engine you run, session and week, so you always know how much engine you have left. These renew on your plan's clock; the rings just tell you where you stand.

One habit covers both: the glance you already do each time you sit down (chapter 2).



Compacting

Every engine tidies its own conversation automatically when it has to — compaction happens whether or not you ever touch it. AGY handles it entirely on its own; there's nothing for you to press. For the others (Claude, Codex, Grok), Croo also gives you the manual option: when an agent is idle and its ring

runs past 90%, one click compacts it — the agent tidies its conversation, keeping what matters and making room to continue, and the compacting state (chapter 2) shows while it works.

Why press it yourself when it would happen anyway? Timing. Compacting on your schedule — before the next big run — beats the engine deciding to compact in the middle of one.

Compacting is normal housekeeping, not an emergency. Long-lived agents compact many times in their lives; the identity file (chapter 3) is why the things you taught them deliberately never depend on it.

Model and effort, per agent

Each agent runs the model and effort level you choose for it, from its own menu. The practical pattern:

- **Heavy model, high effort** for the work that earns it — planning, review, the hard build.
- **Lighter settings** for the routine — summaries, housekeeping, first drafts.

Matching the setting to the job is the single biggest lever you have on how far an allowance goes.

Pacing a day

- **Spend the allowance — it's paid for.** The weekly gauge isn't a score to keep low; ending the week with engine unused is the only way to lose.
- **Put the heavy jobs early in a session,** when the allowance is fresh, and let the routine ride on what's left.
- **Watch the week, not just the day.** A Thursday glance at the weekly rings tells you whether Friday can afford the big refactor — before you brief it.
- **Who pays for what, plainly:** the engines run on your own subscriptions (chapter 1). Croo spends nothing on its own — it shows you what your plans are doing.

Inside Croo?

Just ask Nova.

9 · Croo away from the desk

Croo Operator's Manual · v1.0.1 · 2026-10-13

The one thing to know: the crew keeps working when you leave — and your word can still reach them. Croo follows you two independent ways, and they're worth telling apart from the start.

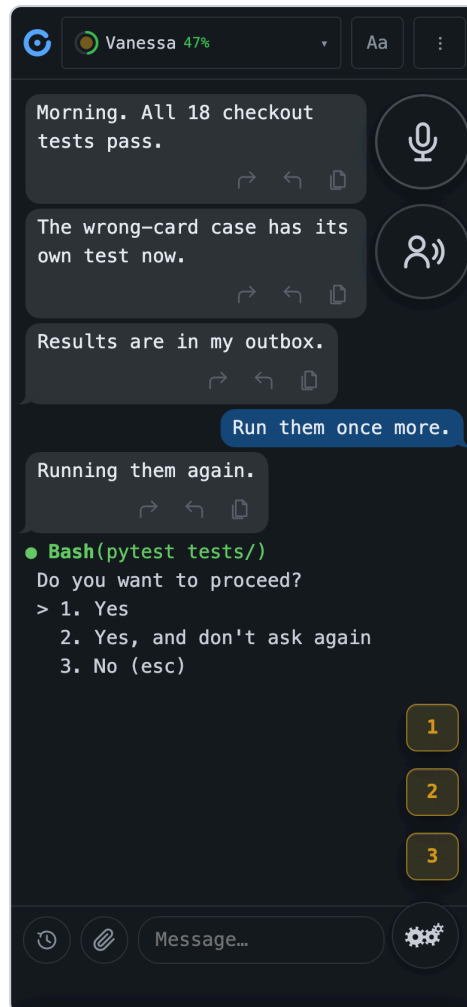
Your phone, two ways

- **The Croo web client** — the workspace itself, in a layout built for phones. The full check-in: see your agents, read replies, speak, listen, answer. Because your phone is talking to your machine, this branch needs your machine reachable first — that's its own short doc, **Reach Croo from anywhere**, and it's worth doing early.
- **Telegram** — a lightweight line to your crew through the Telegram app you already have. No network setup at all: the connection goes outward from your machine to Telegram, so it works from anywhere the moment it's set up.

Use either, or both: the web client when you want the workspace in your pocket, Telegram when you want your crew in your messages.

The web client

Croo's phone layout: an agent picker, the chat, quick keys and one-tap answers. Open it in the phone's browser and add it to your home screen — it behaves like an app from then on. It's deliberately simpler than the desktop: on the phone you're checking in and giving your word, not conducting the whole operation.



Three things the web client does on the phone:

- **Talk instead of type.** Tap the microphone and speak; your words become text on your own machine — speech never leaves your desk to be transcribed. English only for now.
- **Hear the reply.** A button reads out a short spoken summary of an agent's latest reply, in a natural Microsoft voice. The glance-free check-in.
- **Driving mode.** Hands free: speaking sends your message, and replies are read out. Driving, cycling, cooking — your pick. A commute becomes a check-in, not a blackout.

Telegram

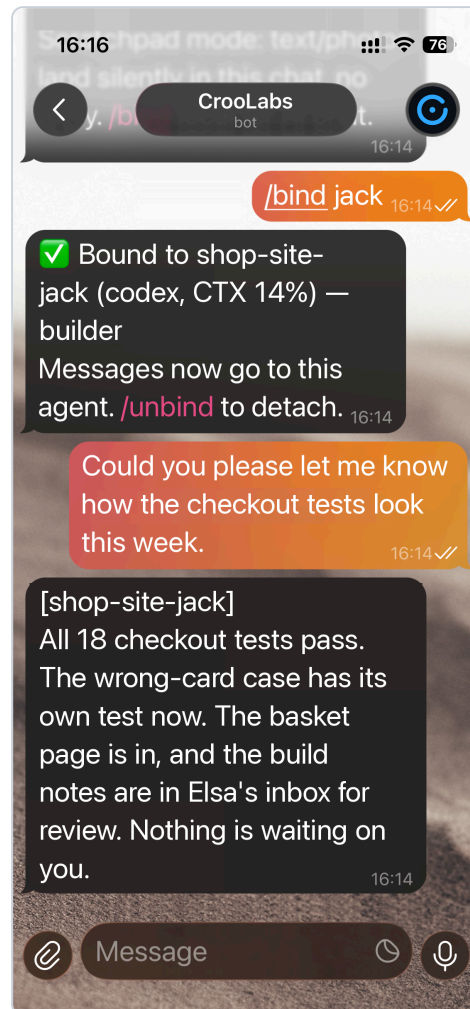
Ask Nova to set it up — she walks you through it step by step, until the first message lands on your phone.

From then on, your Telegram chat is a line to the crew:

- `/bind jack` — talk to Jack directly: ask for status, send the next instruction, answer his questions. `/unbind` and the chat is a private scratchpad again — notes to yourself that go nowhere.
- **A question from an agent arrives with its choices** — tap one, or reply with its number. The same one-click answers as the workspace, in your messages.

- `/status` — the agent list at a glance. `/voice on` — spoken replies, and you can speak too: voice goes both ways. `/powerthrough` — routine approvals handled for five minutes, from wherever you are. `/help` — the full list.

And the part that changes your day: when you're away, your agents and your cues reach you here. The question that would have sat in **Waiting for you** until evening gets answered from the lunch queue, and the crew keeps moving.



What reaches you, and how to quiet it

Away from the desk, the traffic is yours to shape: which alerts follow you, what Telegram carries, and when everything stays quiet. The mute button silences the desk; put your phone on do-not-disturb and Croo doesn't fight it. The crew works either way — what you're choosing is when to be told, not whether they continue.

Inside Croo?

Just ask Nova.

10 · Make it yours

Croo Operator's Manual · v1.0.1 · 2026-10-13

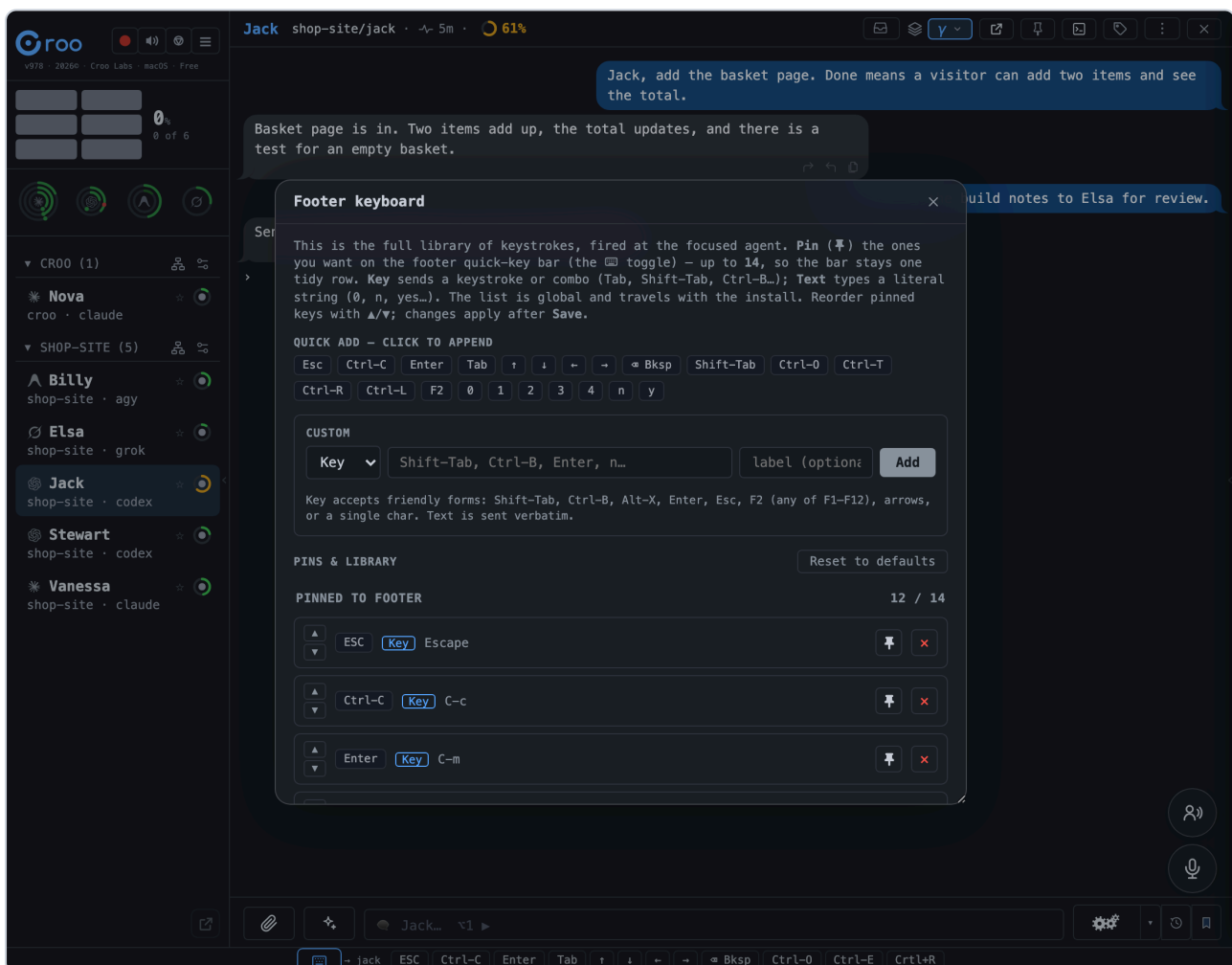
The one thing to know: the workspace adapts to you, not the other way round. None of this chapter is required — it's the set of dials that make a long day in Croo feel like your own desk.

Looks

Three themes — Dark, Matrix and Ubuntu — plus your choice of font, zoom level, and what each pane header shows. Got a theme in mind? Suggest it, and we may add it.

Keys

Keyboard shortcuts cover the moves you make constantly: switch agents, send to a pane, flip layouts, open the work map. Behind them sits a full key library in the menu — browse it, and pin up to 14 keystrokes or short texts to the bar at the bottom of the screen. The bar is yours: the keys you actually use, one row, always there.



Cues

Croo calls its reminders cues. The built-in ones look after you — eye breaks, water, stretching — if you listen. Better yet, make your own: a cue can say anything, on your schedule, and reach you like any other alert (chapter 7) — or on your phone when you're away (chapter 9).

Notes

Your own notepad inside the workspace: pin notes, reorder them, search them, or pop one out as a sticky. It's deliberately yours alone — agents never write in it. Thoughts land here while agents work; briefs get built from thoughts.

The Playground

For the kids ;) — a few small games shipped with Croo, for while an agent works or when you need to switch off. A guide shows you how to build your own, local AI included, and any single-file page you add appears in the menu: games, status boards, whatever you like.

The local AI

Through Ollama, an assistant runs on your own machine — and once it's running, Croo quietly hands it the small jobs: shortcuts that turn long text into commands, quick math, Tidy my draft, Read the room's one-liners, the live discussion commentary, parts of the Playground. Nothing it touches leaves your desk. One-time setup; ask Nova.

Inside Croo?

Just ask Nova.

11 · Keeping Croo current

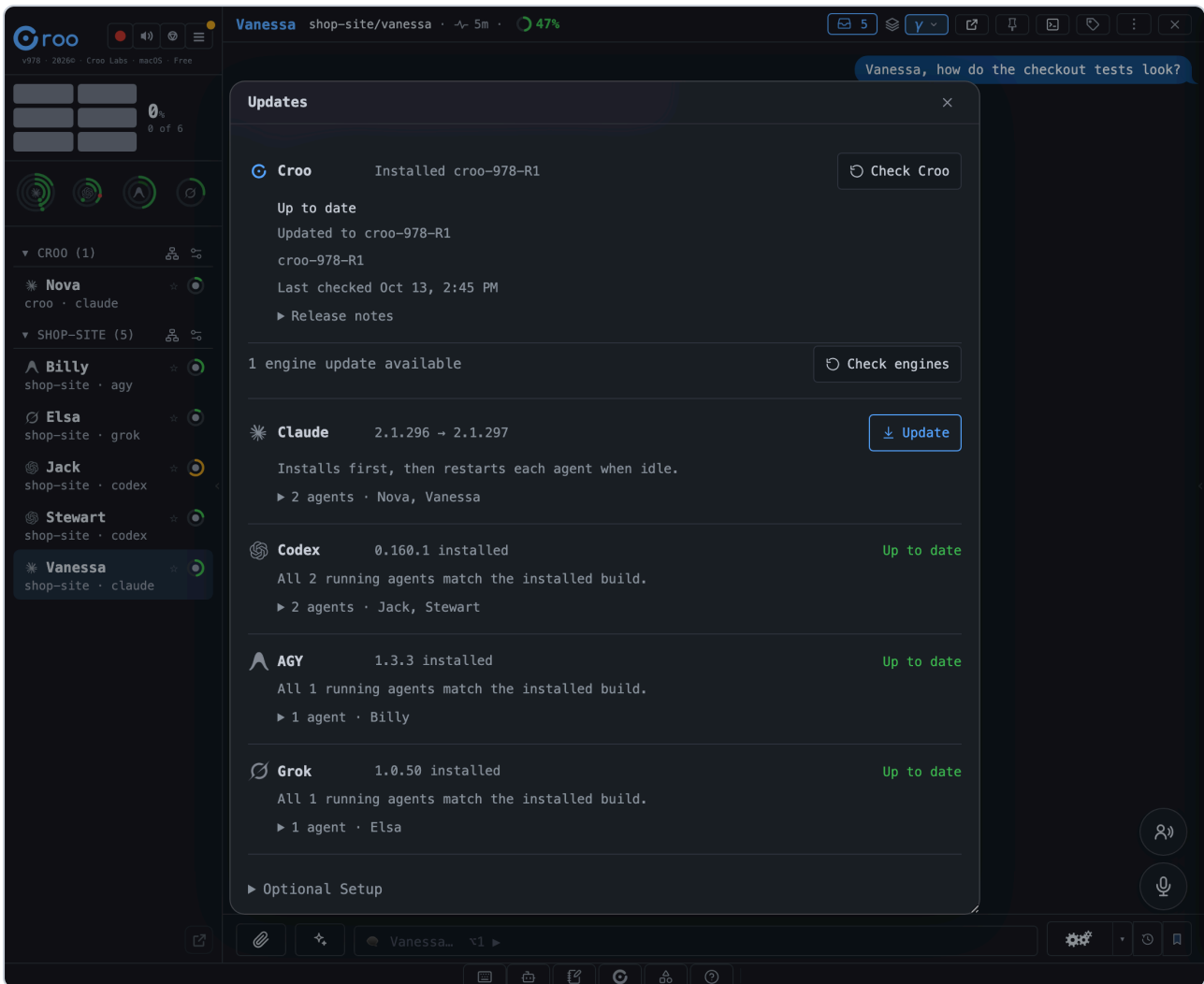
Croo Operator's Manual · v1.0.1 · 2026-10-13

The one thing to know: nothing updates itself. Croo tells you when something has an update, you prepare it from the workspace, and nothing applies without your confirmation. Up to date is a choice you make, not something that happens to you.

Current is the healthy state. The engines move quickly — new models, new features, new behaviour, sometimes week to week — and each Croo release is built and tested against the engines *as they are now*. When an engine changes how it works, Croo changes with it; applying both sides promptly is what keeps that pairing smooth. Fall a few versions behind and things will mostly still work — but the newest models won't be in your menus, features Croo already supports sit unused, and you're running a combination nobody is testing anymore. So when the Updates view shows something, plan to apply it at the next natural pause — this sitting or the next, not next month.

The Updates view

One place in the menu shows what could be newer: Croo itself, and the engines your agents run on. When something has an update, you'll see it there — what it is, and what applying it involves. No nagging, no countdown, no silent installs.



Updating an engine

Engines (Claude, Codex, AGY, Grok) publish their own updates, and Croo manages the awkward part for you: a running agent keeps the engine version it started with, so an engine update only fully lands when the agent restarts. Croo handles that safely — agents are brought to the new version when they're idle, never mid-job, and each one resumes its own conversation afterwards. You confirm; Croo sequences.

If an agent is mid-way through something long, there's no urgency: "restart to finish the update" can wait until the work doesn't.

Updating Croo

New Croo releases are announced — during the beta, through the channel your invitation set up. When one is out, ask Nova: applying a release is a job she does with you, with your confirmation at the step that matters, and a health check at the end. The update doc that travels with each release (update.md) is the by-hand version of the same steps — you'll likely never need it, but it's there, and it's the recovery path if an update is ever interrupted.

Habits that pay

- **Update between pieces of work, not during.** The Updates view will wait; a half-reviewed build shouldn't.
- **Engines move faster than Croo.** A weekly glance at the Updates view keeps the crew on current engines without making a hobby of it.
- **Unhurried, not postponed.** Nothing installs itself or interrupts your work, and there's never a countdown. But "between pieces of work" means today's gap, not someday — prompt is the habit; rushed is what Croo makes sure you never have to be.

Inside Croo?

Just ask Nova.

12 · When something's off

Croo Operator's Manual · v1.0.1 · 2026-10-13

The one thing to know: your first move is always the same — describe what you see to Nova, in plain words. Not the technical cause, not a guess at the fix. "Jack's pane went blank after I sent a file" is a perfect report; Nova takes it from there.

Nova first

Nova is the custodian: diagnosing the installation is her job. She can look at what's running, check the health of the pieces, and either fix it or tell you plainly what she found. Most "something's off" moments end here, in a minute, without you learning anything about logs — which is the point.

What helps her most:

- **What you saw**, in your words. What you clicked, what happened, what you expected instead.
- **When it started**. "Since this morning" narrows more than it sounds like it would.
- **A screenshot**, if it's visual.

The bug report

Found something genuinely broken? Ask Nova to write it up:

"Nova, write up a bug report on that — I want to send it in."

She produces a proper report — what happened, versions, the details that matter — without you gathering any of it. Paste it into the feedback form along with anything you'd add in your own words. Reports land directly with the engineering team and shape what gets fixed first; during the beta, this is genuinely the most valuable thing you can send us.

And privacy stays the rule it always was: nothing leaves your machine unless you see it and send it.

If the workspace itself won't load

The one case Nova-in-the-chat can't help with is the chat not loading. But Nova herself is still running — only the workspace in front of her is down. Three moves, in order:

1. **Reach Nova in the terminal.** Open a terminal and type `croo attach nova` — you're now talking to Nova directly, same agent, same memory, just without the workspace around her. Tell her the workspace won't load and let her fix it. When you're done, press Ctrl-b and then d to step back out — closing the window other ways can disturb her session.

2. **The terminal, for emergencies** — if even Nova is unreachable, type `croo help`: the first thing it prints is an emergency procedure for exactly this case, step by step. The commands worth knowing are below, and `cli.md` walks through all of them in full.
3. **support.md** — if neither works, this is the path to us: what to collect, where to send it, what to expect back.

The commands worth knowing

A workspace outage does not normally stop your agents — they keep running underneath. These commands let you see that, and put things back. The first four only look; they are safe to run any time, even out of curiosity.

- `croo access` — the first question, answered: is the workspace actually down, or just your browser tab? It checks the service and prints the address to open.
- `croo ls` and `croo status` — who's running and what state each agent is in. This is how you confirm the crew is fine even while the workspace isn't.
- `croo doctor` — a health check on Croo's own records: it audits every agent's registration and conversation link and tells you what it found. On its own it is read-only — it diagnoses and advises, and changes nothing until you ask it to.
- `croo attach <name>` — work directly with any agent, Nova included, exactly as in move 1. Leave with Ctrl-b, then d.
- `croo stop <name>` and `croo start <name>` — pause an agent and bring it back later, same conversation, same memory. Stop is the gentle option: nothing is lost.
- `croo respawn` — after a reboot: brings every registered agent back, each resuming its own conversation. The order that works: `doctor` first, fix anything it advises, then `respawn`.
- `croo session recover <name>` — for the rare case where Croo found an agent's conversation but refused to resume it without you; this is you saying "go ahead", deliberately.

Each command explains itself with `--help`, and `croo help` shows the full list. None of them installs, deletes or sends anything — the terminal here is a recovery room, not a second workspace.

Habits that pay

- **Report the weird, not just the broken.** Something that "felt off" but recovered is exactly the kind of thing a beta wants to hear about.
- **One report per thing.** Three issues in one report get one muddled fix. (The same rule as briefs, chapter 4 — it's all briefing.)
- **Let Nova do the gathering.** A two-line description plus her report beats twenty minutes of your own detective work, every time.

Inside Croo?

Just ask Nova.