

SELF-PACED COURSE COMPANION

Stop using Claude like a chatbot.

The Cowork *Operating System.*

The no-terminal playbook for running your business on Claude — persistent context, real app access, and work that runs on a schedule. Build the system once; every session starts fully briefed.

START HERE

What this is, and how to use it.

This is the companion to the Cowork Operating System course. It's built to be worked through with the app open, not read on the couch. By the last page you'll have Claude Cowork briefed on your business, your tools connected, and at least one workflow running without you.

Most people use AI at about ten percent of its leverage. Not because they prompt badly — because they treat Claude like a search engine with manners. They start every chat from zero, ask instead of delegate, and never connect it to the apps their business actually runs on. This guide fixes all three, in order.

How to work through it

Each part ends with an action. Do the action before moving on. The templates in Part 2 are yours to copy directly into your own workspace — they're written as fill-in-the-blank, not examples to admire.

Inside this guide

01	The shift — from chatbot to operator	p.3
02	Shift One — Cowork as operator, not chatbot	p.4
03	Shift Two — context is an asset you build (with templates)	p.6
04	Shift Three — connect your tools, schedule the work	p.10
05	Build-along — three workflows you can ship today	p.12
06	Your 30-day implementation plan	p.14
07	Quick reference — prompt patterns & the anti-AI cheat sheet	p.15

THE IDEA

You're not using AI *wrong*. You're using it at **10%**.

The 5–10x gain from AI doesn't come from a better model or a cleverer prompt. It comes from changing your relationship with the tool — from conversation partner to delegated operator with memory and access.

Cowork is the desktop app built for exactly this. It has a folder on your computer, a sandbox it can run code in, connectors into your real apps, skills that produce finished documents, and the ability to schedule work. Most people touch none of it. They open a chat, ask a question, copy the answer, and close the tab. Every drop of context they just built evaporates.

Three problems keep you at ten percent. The rest of this guide is the fix for each.

The chatbot habit

Re-explain your business every session · ask "how do I" · copy-paste answers by hand · tools never connected · nothing persists.

The operator habit

Brief Claude once, permanently · say "do this" · it writes the file and sends the email · apps connected · work runs on a schedule.

The whole method, in one line

Give Claude **persistent context**, **real access**, and **standing orders** — then stop watching it work.

→ Action

Open Cowork and connect a working folder for your main business (or make one). Everything in this guide lives in that folder.

Shift One

Cowork as *operator*, not chatbot.

Claude Cowork doesn't just answer — it acts. It reads and writes the files in your folder, runs code in its sandbox, uses your connected apps, and produces finished work. You delegate to it the way you'd delegate to a capable employee.

What it can actually do

- **Files** — read your documents and spreadsheets, then write new ones back into your folder. Real .docx, .xlsx, .pptx and .pdf, not text in a chat window.
- **Code & data** — run scripts in a sandbox to clean a CSV, crunch numbers, rename a thousand files, or scrape and reshape data.
- **Your apps** — through connectors it can read and act in Gmail, Calendar, Drive, Slack, Notion, your store and your CRM.
- **Skills** — pre-built capabilities that turn a request into a polished deliverable (a formatted proposal, a deck, a financial model).
- **Schedule** — standing jobs that run on their own: a morning brief, a weekly report, an inbox triage.

Stop asking. Start commanding.

The single highest-value habit change is the verb you use. Watch what happens when "how" becomes "do."

Asking

"How should I structure a proposal for a new client?"

You get advice. Then you go do the work.

Commanding

"Draft the proposal for [client] using my notes in this folder and my pricing. Save it as a .docx."

You get the finished file. You review and send.

Try it now

Point Cowork at a real file and give it a command with a verb and an output:

Conversation → command

Read the spreadsheet in this folder and build me a one-page summary of last month's numbers as a PDF.

→ Action

Give Cowork one real task today that ends in a file or a sent message — not a question. Notice that it finished the job, not just described it.

Shift Two

Context is an *asset* you build.

If you re-explain your business every session, you're paying to onboard Claude over and over. Build the context once, store it in your folder, and load it automatically. Every future session starts fully briefed and remembers where your projects stand.

This is the heart of the system. You create a small set of plain-text files — your **"About Me" folder** — and one instruction file that tells Cowork to read them at the start of every session. Four files do the work:

- `about-me.md` — who you are, your businesses, tools, audience and current projects.
- `writing-rules.md` — how you want things written, so drafts sound like you and never like a bot.
- `memory.md` — a running log of where every project stands, updated as you go.
- `CLAUDE.md` — the loader at the root of your folder that tells Cowork to read the other three, every time.

Naming convention

Keep every markdown file in **kebab-case** — lowercase words joined by hyphens (`about-me.md`, `writing-rules.md`). Consistent names are easier for both you and Claude to reference.

The templates below are yours to copy. Create a folder called `About Me` inside your business folder, make a file for each, and fill in the brackets. Then add the loader to the folder root.

Template 1 — your profile

```
About Me/about-me.md
```

```

# about-me.md
# Day-one briefing on me. Read at the start of every session.

## Who I am
- Name: [your name]
- Role: [what you do, in one line]
- How I operate: [direct? detail-oriented? delegate-heavy?]

## My business(es)
- [Business 1] – [what it does, who it serves]
- [Business 2] – [what it does, who it serves]

## My tools / stack
- [the apps and platforms I use daily]
- Treat me as fluent in these. No beginner explanations.

## My audience / customers
- [who buys from me, their type and size]

## Current projects
- [project] – [status, next step] (keep in sync with memory.md)

## How to work with me
- Lead with the recommendation, then brief tradeoffs.
- When I ask for something built, build it.
- Follow writing-rules.md for anything you draft.

```

Template 2 — how you write

This is the anti-AI filter. The condensed version is below; the full cheat sheet is on the last page.

About Me/writing-rules.md

```

# writing-rules.md
# Sound like a sharp human, never like an AI.

## Banned words (AI tells)
delve, leverage, tapestry, testament, realm, robust,
pivotal, crucial, vibrant, seamless, underscore, boast,
"stands as a testament", "plays a vital role",
"it's important to note", "in today's fast-paced world".

## Banned tics
- No "Moreover / Furthermore / Additionally" pile-ups.
- Kill the rule of three (not every list needs 3 items).
- Em dashes: rare. Straight quotes, not curly.
- No "In conclusion" wrap-ups. Stop when the point is made.

## Always
- Name the source; no "studies show".
- Open with the point. Vary sentence length.
- [your voice: e.g. direct, confident, no fluff]

```

Template 3 — the memory log

About Me/memory.md

```
# memory.md
# Running log so we never lose the thread. Newest at the bottom.

## How to use (rules for Claude)
1. Read this at the start of every session.
2. Append a dated entry when something changes.
3. Edit an existing entry if new info updates it.
4. Promote permanent facts into about-me.md.

## Standing facts
- [the things that rarely change]

## Log
### [YYYY-MM-DD] – [Project]
- What changed / decided. Next step + owner.
```

Template 4 — the loader

This file sits at the **root** of your business folder (not inside About Me). Cowork reads it automatically at the start of every session.

CLAUDE.md (folder root)

```
# Global instructions – read at the start of every session

Before doing anything else, read these three files in
`/About Me/` and operate accordingly for the whole session:

1. about-me.md – who I am, my business, tools, projects.
2. writing-rules.md – how I want things written.
3. memory.md – where projects stand. Update it as we go:
   append dated entries; edit entries when info changes.

## Defaults
- Direct, concise, action-oriented. Lead with the recommendation.
- When asked to build something, build it.
- Keep markdown filenames in kebab-case.
```

→ Action

Create the **About Me** folder and all four files now. Fill the brackets for your main business. Open a fresh session and confirm Claude greets you already knowing who you are.

Shift Three

Connect your tools. *Schedule* the work.

Context makes Claude smart about your business. Connectors make it able to act on it. Scheduling makes it run without you. This is where a tool becomes an operating layer.

Connect your apps

Connectors let Cowork read and act inside the tools you already use. Connect what's relevant; approve what it does. Common high-leverage ones:

- **Email & calendar** — triage the inbox, draft replies in your voice, prep your day.
- **Drive / storage** — pull source docs, write finished files back.
- **Chat (Slack, etc.)** — summarize channels, post updates.
- **Store & CRM** — read orders and contacts, generate reports, update records.

On safety

You choose what to connect and you approve actions. Scope access to what a job needs, and keep client data clean. Connect deliberately, not all at once.

Let it use skills

Skills turn a plain request into a real deliverable — a formatted Word proposal, a slide deck, a spreadsheet model, a designed PDF (this guide was produced with one). You don't invoke them by name; you just ask for the outcome and Cowork reaches for the right one.

Schedule the recurring work

Anything you do on a rhythm can run on its own. Describe the cadence in plain English and it becomes a standing job.

Standing orders that run without you

"Every weekday at 6am, give me a brief: today's calendar, overnight emails that need a reply, and anything overdue."

→ becomes a scheduled task. You wake up to the brief.

→ **Action**

Connect one app and schedule one recurring job — a daily brief is the best first one. You'll feel the shift from "tool I open" to "system that runs."

Three workflows you can *ship today*.

Pick the one closest to your business and build it now. Each is a starting prompt — adapt the brackets, then refine with Claude until it's right. Save the working version into your folder so you can reuse it.

1 · The proposal / quote generator

- 1 Drop your pricing, a past proposal, and your service list into a folder.
- 2 Tell Cowork: *"Using my pricing and the template in this folder, draft a proposal for [client] covering [scope]. Match my writing-rules. Output a .docx."*
- 3 Review, refine the bits it missed, and save the prompt as your reusable command.

2 · The weekly brief / inbox triage

- 1 Connect email and calendar.
- 2 Tell Cowork: *"Summarize this week — meetings, threads waiting on me, and anything overdue. Flag the three things that matter most."*
- 3 Once it's good, schedule it to run every Monday at 7am.

3 · The data clean-up

- 1 Drop a messy CSV or export into your folder.
- 2 Tell Cowork: *"Clean this file — fix the headers, de-dupe, standardize the dates — and give me back a tidy spreadsheet plus a short note on what you changed."*
- 3 It runs the work in its sandbox and writes the clean file back.

→ **Action**

Ship one of the three before you close this guide. A workflow running on your real data beats three pages of notes.

IMPLEMENTATION

Your *30-day* plan.

The system compounds. Don't build it all in a weekend — build the habit over a month so it sticks.

Week 1 — Foundation

- ✓ Connect a working folder and create your `About Me` files.
- ✓ Add the `CLAUDE.md` loader; confirm a fresh session starts briefed.
- ✓ Give Cowork three real "do this" commands instead of questions.

Week 2 — Access

- ✓ Connect email and calendar; draft replies in your voice.
- ✓ Connect one more core app (store, CRM, or storage).
- ✓ Produce one finished deliverable via a skill (proposal, deck, or sheet).

Week 3 — Automation

- ✓ Schedule your daily or weekly brief.
- ✓ Build and save one reusable workflow from the build-along.
- ✓ Start updating `memory.md` at the end of each work session.

Week 4 — Leverage

- ✓ Run two parts of your business through Cowork in the same day.
- ✓ Audit what still eats your time and turn the worst offender into a workflow.
- ✓ Decide what to delegate next — you're now the conductor, not the operator.

The test

By day 30 you should open a new session, type nothing about your business, and have Claude already know it — with at least one job running that you never have to start.

Keep this page open.

Prompt patterns that work

- **Command, not question** — start with a verb, end with an output. "Draft... save as .docx."
- **Point at the source** — "using the files in this folder," not "based on what you know."
- **Name the format** — .docx, .xlsx, PDF, "a table," "three bullets max."
- **Set the cadence** — "every weekday at 6am" turns a task into a standing job.
- **Invoke your rules** — "follow my writing-rules" so drafts sound like you.

The anti-AI cheat sheet

Paste this into [writing-rules.md](#). If a draft trips any of these, send it back.

Cut these words

delve · leverage · harness · unlock · elevate · foster ·
tapestry · testament · realm · landscape · vibrant ·
robust · pivotal · crucial · seamless · holistic ·
underscore · showcase · boast · navigate · "rich
history" · "stands as a testament" · "plays a vital role" ·
"it's important to note" · "in today's fast-paced world"

Break these habits

"Moreover / Furthermore / Additionally" pile-ups · the
rule of three in every list · em-dash overuse · curly
quotes · vague "studies show" claims · trailing "...,
highlighting its importance" clauses · symmetrical pros/
cons · "In conclusion" wrap-ups · chatbot residue
("Certainly!", "I hope this helps")

The five-second test before anything goes out

Banned words? Cut. More than two em dashes? Reduce. A list of three that could be one? Tighten. A claim with no named source? Name it. Read it aloud — does it sound like you, or like a bot?

NEXT STEP

You've got the *system*. Now build it *in a room*.

This guide is the self-paced course. The Live Workshop is the same method built on *your* real business in three hours, coached by Tim, with a group implementation call 14 days later.

Your \$250 credit

The full price of this course applies as credit toward a Live Workshop seat. You only pay the difference.