

Your download is a standing-instructions file. Here's how to put it to work.

A plain-text file the AI reads at the start of every session — the desk binder you'd hand a new hire on day one. Rename it, drop it in the right folder, and every session inherits your rules.

1 • Rename the file

Files on the download page are named `CLAUDE_[description].md` so you can tell them apart. The tool only recognizes the file when the description is gone. **Delete everything from the underscore through the closing bracket.** Keep `.md`.

`CLAUDE_[SingleFileApp].md`



`CLAUDE.md`

Windows tip: Explorer hides extensions by default, so a rename can silently become `CLAUDE.md.txt`. Turn on **View** → **Show** → **File name extensions** first. Keep the capitals as shown.

2 • Match the name to your tool

The content is identical; only the filename changes. Rename `CLAUDE` to whatever your tool expects:

TOOL	FILE IT READS AT THE TOP OF THE FOLDER
Claude Code · Claude Cowork	<code>CLAUDE.md</code> — use the download as-is after step 1
OpenAI Codex · Cursor · Cline	<code>AGENTS.md</code>
GitHub Copilot	<code>AGENTS.md</code> or <code>CLAUDE.md</code> — either works
Gemini CLI	<code>GEMINI.md</code>

Two tools on one project? Keep two copies with identical contents (`CLAUDE.md` and `AGENTS.md`) and edit both. Filenames as of September 2026 — your tool's own documentation wins if it says otherwise.

TWO RULES BEFORE THE FIRST PROMPT

Data rule first. Individual AI plans may train on what you paste unless you opt out; business plans don't by default. Decide the plan, set the opt-out, and decide what client material is allowed in — before anyone builds. Test on synthetic data.

Never paste a credential. The AI only needs the *name* of a setting, never the value. When it asks to install something or reach the internet, read what it wants — “the AI asked” is not a reason to click yes.

3 • Put it in the right place

The file goes at the **top level of the project folder** — one folder per tool, not the shared drive, not inside a job folder.

```
C:\Users\you\Coding Projects\
PointReformatter\      ← one folder per tool
  CLAUDE.md             ← the renamed download
  samples\              ← hand-checked sample
  DECISIONS.md          ← "we chose X because Y"
  field-import\
    CLAUDE.md           ← rules for this part
```

Subfolders work too. A second file inside a subfolder adds rules that apply only to work in that subfolder (Claude Code, Codex, and Cursor all read nested files; the nearest one adds to the top-level one). Keep firm-wide rules at the top; put task-specific rules — a DXF export routine, a plat checklist — next to the files they govern.

4 • Set up the project — five steps, once

- 1 Make the folder** and drop the renamed file in it. Add a `samples` folder: one real input and the answer you already computed by hand.
- 2 Open the tool in that folder.** Claude Code: open a terminal in the folder and type `claude`. Cowork: add the folder in the desktop app. Codex: run `codex` from the folder.
- 3 Confirm it's being read.** First message: “Read the instructions file and tell me, in plain English, the rules you'll follow in this project.” If it can't, the name or the location is wrong.
- 4 Make it yours.** Open it in Notepad and replace every bracketed placeholder — units, zone, code list, template columns — with your firm's truth. Keep it short; point at files (see `codes.csv`) rather than pasting them in.
- 5 Keep it current.** When the AI gets something wrong twice, add one line. Start each task in a fresh session — the file carries the memory.

Setup happens once. Every tool you build after that inherits it. — Questions, or want a file written for your firm?
info@apolloLens.io