

CRAFT for Claude

Built for how Claude receives instructions: an installable skill that loads itself, plus Projects, preferences, memory, and artifacts.

Version 1.0 · From How to Talk to AI, Session 01 · Agentix (goagentix.com)

In this folder: `craft.zip` (upload this), `craft.skill` (the same skill packaged for Claude's Save-skill button), the unzipped `craft/` folder (also at the root of the pack; read or edit it), this guide, `TEST-REPORT.md` (test prompts and their outputs), and three `PASTE-` text files.

Install A: let Claude do it (90 seconds)

1. Open a new chat. Attach `INSTALL-ME-Claude.md` and `craft.skill` from this folder.
2. Type: **Read the attached file and install CRAFT for me.**
3. Claude probes what it can do, hands the skill back as a file card with a one-click **Save skill** button, saves the core to memory, gives you the one preferences line to paste, applies CRAFT in that chat, and ends with a CRAFT INSTALL REPORT and the handshake.
4. Click **Save skill**. Paste the preferences line. Done.

Everything below is **Install B: by hand**, for people who prefer to see every step, and the reference for what Install A did.

Install B: by hand

How Claude receives instructions

Layer	Where	Applies to	Use it for
Skill	Customize → Skills (upload a ZIP)	Any chat, loaded automatically when it fits	The full CRAFT system with its reference library (this is the real install)
Preferences	Settings → Profile	Every chat	One line that tells Claude to use the skill on everything
Project instructions	Inside a Project	Chats in that Project	Your Context Card for one business or client, plus files
Memory	Settings → Capabilities → Memory	Every chat, where on	Backup reminder
Artifacts	Side panel, automatic	The current chat	Documents, pages, decks, and code you can select and edit part by part

What makes Claude different: a **skill** is a folder of instructions and reference files Claude reads on demand. Only the short description is always in view; the body loads when a request matches; the references (the ten prompt types, the Tune playbook, the commands, the full Tech Talk Dictionary) load only when needed. So CRAFT can be encyclopedic without crowding your chat.

One naming note: Claude requires skill names to be lowercase letters, numbers, and hyphens, so the folder and the technical name are `craft`. The system is CRAFT; the file is `craft`.

Skills work on Free, Pro, Max, Team, and Enterprise. The one requirement is **Code execution and file creation** turned on.

Install (about 2 minutes)

Step 1: Turn on code execution. Claude (web or desktop app) → **Settings** → **Capabilities** → switch on **Code execution and file creation**. On Team and Enterprise, an owner may control this under Organization settings → Skills.

Step 2: Upload the skill. 1. **Customize** → **Skills** (claude.ai/customize/skills). 2. + → **Create skill** → **Upload a skill**. 3. Choose

craft.zip. (If Claude shows you craft.skill as a file card with a **Save skill** button, that button does the same thing in one click.) 4. “craft” appears in your list. Make sure its toggle is **on**.

Step 3: Make it fire on everything. Skills load when Claude judges the request matches. CRAFT is meant to apply to everything, so add one line to your preferences: 1. **Settings** → **Profile** → the box asking what preferences Claude should consider in responses (on some versions: Settings → Personalization). 2. Paste PASTE-preferences-line.txt: Use my CRAFT skill on every request. If asked "what's the CRAFT rule?", answer from the skill. 3. Save.

Step 4: Confirm. New chat. Type **what’s the CRAFT rule?** (Your first ordinary request in a new chat, and every craft: command, will open with *Thank you for using the CRAFT skill by Agentix*. The handshake is the one reply that skips the greeting, so it verifies byte-for-byte.) Expected: *Brief it like a brilliant new hire: Context, Role, Ask, Format, then Tune. Structure beats length. (CRAFT Skill v1.0 by Agentix)*

Step 5: Build your Context Card. Type **craft: brief**. Answer the questions. Claude returns your Context Card in a code block. Save it in a Project (next section) or paste it into your preferences under the line from Step 3.

Projects: your context, saved once

For any business or client you work on repeatedly:

1. Left sidebar → **Projects** → **New project**. Name it after the business.
2. Open **Project instructions** (the gear or “Set project instructions”).
3. Paste PASTE-project-instructions.txt, then replace the placeholder with your Context Card from Step 5.
4. **Add files:** brand guide, price list, two or three writing samples, the Tech Talk Dictionary. Claude answers from them and says when something isn’t there.

Every chat inside the project starts with your C filled in. The skill handles the R, A, F, and T.

Memory (optional backup)

With memory on (Settings → Capabilities → Memory), say in any chat: “Remember that I use the CRAFT method and want it applied to every request.” The skill is the real install; memory reinforces it.

Also install in Claude Code and Cowork

- **Claude Code:** drop the unzipped craft/ folder into ~/.claude/skills/ (all projects) or .claude/skills/ (this project). It’s discovered automatically and /craft becomes a command. The code-related Tech Talk (section 19 of the dictionary) applies directly.
- **Cowork:** skills you enable in Customize → Skills are available in Cowork too. craft: talk works on documents and slides Cowork is editing.
- **Claude in Excel, PowerPoint, Word, Outlook:** enabled skills apply in the add-ins. Type / in the sidebar to see them.

Commands

Type this	What you get
craft: [your goal]	Interview (up to 3 questions), then the ideal prompt in a code block
craft: audit + a prompt	Scorecard out of 25, fixes, upgraded prompt
craft: brief	Interview (up to 8 questions), then your Context Card
craft: talk + an edit note	Plain English translated into part + action + result
craft: tune + feedback	KEEP / CHANGE / LEAVE applied; only what you named changes
craft: teach	The moves it just used, explained
craft: pack + a task	The closest Prompt Pack template, filled from your context
craft: help	The command list

<code>craft: quiet / craft: loud</code>	Greeting and tune lines off / on
<code>craft: off / craft: on</code>	Pause / resume CRAFT for the chat
<code>what's the CRAFT rule?</code>	The handshake line

`/craft` works too.

Using it day to day

Just ask. CRAFT runs quietly. You'll notice one-line assumptions, the deliverable first, and a tune line at the end.

Artifacts are your Tune surface. Anything long lands in the side panel. Click the exact paragraph, heading, or line and give the note: "condense this to two sentences," "make this heading a takeaway." That is pinpointing.

Extended thinking. For math, tradeoffs, and planning, turn on thinking (the model or mode selector) so reasoning comes before the answer.

Attachments and screenshots. Drop in the page, the slide, the error, the reference image. Claude will ask "match exactly" or "palette only" if you didn't say.

Research. Say "search first, then answer, and cite sources" when recency matters. For big questions, Research mode; brief it with CRAFT first (who it's for, what decision it feeds).

Long chats drift. When answers go generic, start a new chat, paste your best version, and say "This is the base. Change only X."

Troubleshooting

Problem	Fix
No Skills menu, or greyed out	Code execution is off. Settings → Capabilities. On Team/Enterprise, ask your owner.
Upload rejects the description as too long	Claude's help center lists a 200-character description limit while the platform docs allow 1,024; the pack ships the longer one. If your upload refuses it, open <code>craft/SKILL.md</code> and replace the description with only its first sentence (it is written to stand alone), re-zip, and upload.
Upload error	The ZIP must contain the folder <code>craft</code> with <code>SKILL.md</code> directly inside it (not <code>craft/craft/SKILL.md</code> , not loose files). The folder name must be exactly <code>craft</code> . Re-zip the folder.
Claude isn't using the skill	Check the toggle in Customize → Skills; add the preference line (Step 3); or say "use my CRAFT skill" in the request.
Handshake returns something else	The skill didn't load. New chat. If it persists, re-upload.
It asks too many questions	Say "assume sensible defaults and proceed," or give it your Context Card.
It rewrote what I didn't ask about	"Change only X; keep everything else identical." Or click the exact part in the artifact.

Editing the skill

Everything is plain text. Open `craft/SKILL.md` in any editor. Add your own defaults, your brand voice, or paste your Context Card under a heading "MY CONTEXT CARD." Keep the top block (the `---` lines with `name` and `description`) intact. Re-zip the `craft` folder, delete the old skill in Customize → Skills, upload the new ZIP.

The reference files in `craft/references/` can grow without limit; Claude reads them only when a request needs them.

Under the hood (for the technical reader)

The skill is a folder built to the open Agent Skills standard: `SKILL.md` (frontmatter `name` + `description`, then a ~290-line body) plus seven reference files that load only when a request needs them. At rest it costs about 150 tokens; on trigger about 3,500; the 900-term dictionary costs nothing until a design note calls for it. The same folder installs on Claude, ChatGPT (plans with Skills), Gemini (Spark), and most coding agents. Persistence layers, character budgets, the command grammar, the handshake protocol, idempotence rules, and the acceptance tests are documented in `ARCHITECTURE.md` at the root of the pack.

Acceptance checklist (run these after installing)

Type	Expect
what's the CRAFT rule?	The exact rule line, nothing before or after
write an email about our price increase (new chat)	Greeting line, one-line assumption, an email under 200 words with a subject, a tune line
craft: audit Write a blog post about HVAC maintenance.	A five-letter scorecard out of 25, a fix per weak letter, an upgraded prompt in a code block
Paste a draft, then shorter, keep the ending exactly as is	Only the length changes; the ending is byte-identical; no greeting on the follow-up
What's the capital of Tennessee?	One word. No greeting, no assumption, no tune line
craft: quiet then any request	No greeting, no tune line

If any row fails, the install is partial. Re-run Install A, or check the troubleshooting table above.

Questions, or want AI working in your business? Book a free call: cal.com/agentix · drake@goagentix.com · dale@goagentix.com