

Claude Skill Packs Setup Guide

Every download link for the five free Claude skill packs, the install commands, and the prompt that makes them write like your business instead of a generic one.

Why I stopped prompting and started installing skills

I ran my business solo for a long time by writing better prompts. It worked until it didn't. Every time I needed an ad brief, a contract read, or a month-end summary, I was rebuilding the same context from scratch in a fresh chat, and the output drifted every time.

A skill fixes that. It is a markdown file that sits on disk and hands Claude the whole procedure for one job: the steps, the output format, the checks, the things to avoid. You write it once. The agent reads it every time the task matches. Nothing to compile, nothing to pay for.

What changed for me in 2026 is that I stopped writing most of them. Five free packs now cover marketing, social media, design, finance and legal, and between them they hold over 80 skills. This guide is every download link, what is actually inside each pack, how to install it in Claude Code or Codex, and the one step that decides whether the output sounds like your business or like a stranger's.

01

How a skill pack actually works

1. A pack is a repo full of folders. Each folder holds one SKILL.md file, and that file is the skill. The folder name has to match the name: value written inside the file, or the agent will never load it.
2. You copy those folders into a place the agent reads: `~/ .claude/skills/` for Claude Code globally, `.claude/skills/` for one project, or `.agents/skills/` for Codex and other agents that follow the Agent Skills specification.
3. When you give the agent a task, it scans the skill descriptions, loads the one that matches, and follows that procedure instead of guessing. You do not call skills by name. You describe the job and the right one fires.

02

The one thing almost everyone gets wrong

People download five packs, run one prompt, get output that reads like every other AI post, and decide skills are overrated. The packs are fine. The problem is that a skill ships with its author's assumptions baked in, and yours are missing.

The pricing skill does not know your margins. The contract-review skill has no idea which clauses you refuse to sign. The social pack does not know how you talk. With nothing to go on, every one of them falls back to the average case, and average is exactly what lands back in your terminal.

WATCH OUT

Before you run a single skill, write a business context file and point the pack at it. In the social pack this is not optional: almost every skill reads the output of the voice-builder skill, which produces an about-me.md and a voice.md. Run voice-builder first or the other 16 skills write in nobody's voice. The legal plugin has its own version of this, a legal.local.md you write yourself. Both files are in the next section, ready to fill in.

03

The five packs and the two files you fill in

SOURCE CODE

The biggest of the five, 45 marketing skills when I filmed this and 60+ listed now:

<https://github.com/coreyhaines31/marketingskills>

EVERY PACK SHOWN IN THE VIDEO

- Marketing, 45 skills (now 60+): ad creative, copywriting, SEO audits, pricing, launches, retention
<https://github.com/coreyhaines31/marketingskills>
- Social media, 17 skills: Reels scripts, LinkedIn posts, hooks, content matrix, analytics
<https://github.com/charlie947/social-media-skills>
- Design, 2 skills: design systems, 192 palettes, 74 font pairings, 79 UI styles
<https://github.com/nextlevelbuilder/ui-ux-pro-max-skill>
- Finance, 8 skills: financial statements, reconciliation, variance analysis, month-end close
<https://github.com/anthropics/knowledge-work-plugins/tree/main/finance>
- Legal, 9 skills: contract review, NDA triage, GDPR and CCPA compliance, risk assessment
<https://github.com/anthropics/knowledge-work-plugins/tree/main/legal>

Inside the marketing pack the useful clusters are conversion (cro, signup, onboarding, popups, paywalls), search (seo-audit, ai-seo, programmatic-seo, site-architecture, schema), strategy (pricing, offers, launch, marketing-plan, content-strategy) and measurement (analytics, ab-testing, attribution). The finance plugin covers journal-entry-prep, reconciliation, financial-statements, variance-analysis, close-management and audit-support. The legal plugin covers contract-review, nda-triage, compliance, canned-responses, legal-risk-assessment and meeting-briefing.

Now the two files. Save the first one in your project root and reference it in every session. It is the single highest-leverage thing in this guide.

BUSINESS-CONTEXT.MD

```
# Business Context

## What I sell
[ONE LINE: the offer, who buys it, what it costs]

## Who I sell to
[WHO THEY ARE, what they already tried, what they are afraid of]

## Price points
[EACH OFFER AND ITS PRICE, plus the margin if the skill needs it]

## Channels I actually use
[e.g. Instagram Reels, YouTube Shorts, LinkedIn, email list of 2,400]

## Numbers that matter
[CONVERSION RATE, average order value, monthly revenue, churn]

## Competitors
[THREE NAMES and what each one does better or worse than me]

## Words I never use
[e.g. "unlock", "game changer", "supercharge", "elevate"]

## Proof I can cite
[RESULTS, case studies, screenshots, testimonials]
```

The second one is specific to the legal plugin, which reads it automatically. In Claude Code it goes in your project's `.claude/` directory. In Cowork, save it in any folder you have shared through the folder picker.

LEGAL.LOCAL.MD

```
# Legal Playbook

## Standard positions
- Governing law: [YOUR STATE OR COUNTRY]
- Liability cap: [e.g. fees paid in the prior 12 months]
- Payment terms: [e.g. net 14]
- IP: [who owns deliverables, and when it transfers]

## Terms I will not sign
- [e.g. uncapped indemnity]
- [e.g. exclusivity beyond 90 days]
- [e.g. auto-renewal without written notice]

## NDA defaults
- Term: [e.g. 2 years]
- Mutual or one-way: [WHICH]
- Carve-outs I always keep: [e.g. independently developed work]

## Escalate to a human lawyer when
- [DEAL SIZE THRESHOLD]
- [ANYTHING involving personal data transfer outside my region]
```

Set it up in Claude Code or Codex

1. Decide global or per project. Global skills live in `~/ .claude/skills/<name>/` and are available everywhere. Project skills live in `.claude/skills/<name>/` and travel with the repo. I keep marketing and social global, and finance and legal per client project.
2. Install the marketing pack. Run `npx skills add coreyhaines31/marketingskills`. It writes to `.claude/skills/` for Claude Code, or `.agents/skills/` for a universal agent setup. Prefer plugins? Run `/plugin marketplace add coreyhaines31/marketingskills` then `/plugin install marketing-skills` inside Claude Code.
3. Install the social pack. Clone and copy: `git clone https://github.com/charlie947/social-media-skills.git`, then `mkdir -p .agents/skills` and `cp -R social-media-skills/skills/* .agents/skills/`. Codex reads them from `.agents/skills/<name>/SKILL.md`. On Claude Code you can use the marketplace commands instead.
4. Install the design pack. It ships as a CLI: `npm install -g ui-ux-pro-max-cli`, then `cd` into your project and run `uipro init --ai claude`. Add `--global` to put it in `~/ .claude/skills/` instead. It needs Python 3 for its search scripts, which ships with macOS.
5. Install finance and legal together. They live in the same Anthropic marketplace: `claude plugin marketplace add anthropics/knowledge-work-plugins`, then `claude plugin install finance@knowledge-work-plugins` and `claude plugin install legal@knowledge-work-plugins`.
6. Check the folder names. On any manual copy, open the `SKILL.md` and confirm the `name:` value matches the folder it sits in. This is the most common reason a skill silently never fires.
7. Open a new terminal window. Claude Code reads the skills directory at startup, so a session that was already running will not see what you just installed.
8. Run voice-builder first if you installed the social pack. Ask Claude: run the voice-builder skill using my business-context.md. It produces `about-me.md` and `voice.md`, which the other 16 skills read.
9. Test with one small job. Ask for a hook, not a campaign. If the reply comes back in the pack's format with your details in it, the install worked and the context file is being read. If it comes back as a generic paragraph, the skill did not load.

QUICK TIPS

- For Codex, add a line to `AGENTS.md` pointing at the skills folder so the agent knows to look there.
 - If a pack needs an API key, ask Claude to add it to your shell profile for you, then open a new terminal before testing. A key exported in one window does not exist in the next one.
 - Install one department at a time. Five packs at once is over 80 skills and you will not remember what any of them do.
-

CUSTOMIZE A SKILL BEFORE USING IT

Read the SKILL.md in [SKILL FOLDER PATH]. Then read business-context.md. Rewrite every example and every default in that SKILL.md so it uses my offer, my price points, my audience and my channels. Keep the structure and the steps exactly as they are. Show me a diff before you write anything.

AUDIT WHICH SKILLS ARE ACTUALLY LOADING

List every skill folder under [~/claude/skills or .agents/skills]. For each one, print the folder name and the name: field from its SKILL.md, and flag any row where the two do not match.

RUN A MARKETING SKILL ON A REAL ASSET

Use the [SKILL NAME, e.g. seo-audit] skill from the marketing pack on [URL OR FILE]. Use business-context.md for who we sell to and what we charge. Give me the findings as a table sorted by impact, and skip anything I cannot fix myself this week.

WRITE THE SOCIAL VOICE FILES

Run the voice-builder skill. Interview me about [MY NICHE] and how I talk, then write about-me.md and voice.md. Pull the facts from business-context.md and ask me only about the things that file does not answer.

REVIEW A CONTRACT AGAINST MY OWN PLAYBOOK

Use the contract-review skill on [CONTRACT FILE PATH]. Check it against legal.local.md. List every deviation from my standard positions, classify each one as walk away, negotiate, or accept, and draft the redline language for the negotiate ones.

When it breaks

ERROR	FIX
Claude ignores the skill and answers like a normal chatbot	The folder name does not match the name: field inside SKILL.md. Rename the folder to match the file, then restart the session.
The skill does not appear after installing	Claude Code reads the skills directory at startup. Close the terminal and open a new one.
npx skills add fails or hangs	Clone it manually instead: git clone the repo, then copy the contents of its skills/ folder into .claude/skills/ or .agents/skills/.
Social skills write in a generic voice	voice-builder has not been run, so about-me.md and voice.md do not exist. Run it first. Every other skill in that pack reads its output.
Output is technically correct but says nothing about my business	The skill has no context file to read. Write business-context.md and name it in the prompt.
contract-review ignores my standard terms	legal.local.md is missing or in the wrong place. In Claude Code it goes in the project's .claude/ directory.
uipro init says command not found	The global npm install did not land on your PATH. Reopen the terminal, or run it with npx instead of installing globally.
Two packs both define a skill with the same name	The later install wins and the earlier one goes quiet. Rename one folder and its name: field, or keep the packs in separate projects.

Ready to go deeper?

This playbook is the short version. Inside the hub you get the full build walkthroughs, the prompt library, and the systems I use to ship this in a weekend.

Join the Build →

<https://hub.digicuratoragency.com/join>