

RECON11 SKILL ARMORY FIELD GUIDE

How to install, call and get the most from nine design and build skills for AI coding agents, with 35 more skills by other makers.

9	20	9	35
SKILLS COVERED	TITANIUM MOVES	DREAMSKILLZ ROLES	SKILLS BY OTHER MAKERS

START HERE

WHAT A SKILL IS

A skill is a folder with one instruction file, `SKILL.md`, and sometimes scripts and reference files beside it. Put the folder where your AI coding agent looks for skills, and the agent learns one job well. You call a skill by typing a slash and its name.

1
IT IS PLAIN TEXT

You can open any skill and read exactly what it tells the agent to do. Nothing is hidden. That is also why you should read one before you install it.

2
IT RUNS AS YOU

A skill works with your permissions. If it ships scripts, they can run on your machine. Every Recon11 package says what it runs and what it reaches.

3
IT OPENS WITH A CARD

Each Recon11 skill prints a field card first, every time: what it does, best use, commands and tips. You never have to memorize a command.

WHAT YOU SEE WHEN YOU TYPE /RECON11-TITANIUM

A FIELD CARD

```
RECON11 TITANIUM    /recon11-titanium <move> <target>
What it does : a design vocabulary. Each move is one named change to one target.
Best for     : building, grading and finishing pages when you want to steer in plain words.
Scout        : recon (graded review) | inspect (technical check) | sweep (tell hunt)
Build        : brief (plan first) | forge (build new) | rebuild (same content, new look)
Steer        : louder | quieter | strip | color | type | space | motion | charm
Finish       : temper (final pass) | armor (edge cases) | plain (copy) | fit (screens) | speed | welcome
Grade        : every recon ends in Scrap, Steel or Titanium, with the shortest path up.
Tips         : one move, one target. Run recon first. Finish with inspect, then temper.
```

THE ARMORY AT A GLANCE

SKILL	TYPE	REACH FOR IT WHEN
Titanium	`/recon11-titanium`	Steer or grade a page in plain words
DreamSkillz	`/recon11-dreamskillz`	Take an idea to a shipped page on your own
Forge	`/recon11-forge`	Build a new page from a clear brief
Overwatch	`/recon11-overwatch`	Review, finish or scan a page that exists
Intel	`/recon11-intel`	Look up a palette, font pairing, chart or UX rule
Quartermaster	`/recon11-quartermaster`	Find, check or install another skill
Artifact to Website	`/recon11-artifact-to-website`	Turn a ticket, menu or brochure into a page
Exhibit by Exhibit	`/exhibit-by-exhibit-design`	Build a themed site that argues one case
Hallmark	`/hallmark`	Build a new page and be surprised by the look

HOW THIS GUIDE IS LAID OUT

PAGES 3 TO 15

- Page 3: installing, three routes.
- Page 4: one build from first lookup to final pass.
- Pages 5 to 12: one skill at a time, with commands and tips.
- Pages 13 and 14: design and business skills by other makers.
- Page 15: fixes for common problems, a glossary, credits.

THREE HABITS THAT PAY OFF

BEFORE ANY COMMAND

- Name one target. A section beats a whole site.
- Bring real content. Skills will not invent facts, numbers or praise.
- Say what the page must not look like. It steers better than adjectives.

INSTALL

THREE ROUTES

Every Recon11 package is one zip with one folder inside, named for the skill. Pick the route that matches where you work.

ROUTE 1
CLAUDE CODE

1. Download the zip from the skill page and unzip it.
2. Move the folder into `.claude/skills/` in a project, or `~/.claude/skills/` for every project.
3. Start a new session and type the slash command.

ROUTE 2
THE CLAUDE APP

1. Download the zip. Leave it zipped.
2. Open Customize, then Skills.
3. Press the plus button, choose Create skill, then Upload a skill, and pick the zip.
4. Code execution has to be on.

ROUTE 3
OTHER AGENTS

1. Download the zip and unzip it.
2. Put the folder where your agent reads skills. The table below has the paths.
3. Call it the way your agent calls skills.

WHERE SKILLS LIVE

AGENT	IN ONE PROJECT	FOR EVERY PROJECT
Claude Code	`.claude/skills/`	`~/.claude/skills/`
Cursor	`.agents/skills/`	`~/.cursor/skills/`
GitHub Copilot	`.agents/skills/`	`~/.copilot/skills/`
Gemini CLI	`.agents/skills/`	`~/.gemini/skills/`
OpenCode	`.agents/skills/`	`~/.config/opencode/skills/`
Windsurf	`.windsurf/skills/`	`~/.codeium/windsurf/skills/`

WHAT EACH ONE NEEDS

SKILL	NEEDS
Titanium, DreamSkillz, Forge, Quartermaster, Artifact to Website, Exhibit by Exhibit	Nothing. Each is one instruction file.
Overwatch	A shell, and network access on the first run for one engine download. No Node, no Python.
Intel	Python 3. No network.
Quartermaster commands	Node.js 22.20 or later and Git, for the skills CLI and the vet step.
Hallmark	Node.js. Puppeteer and Chrome only for its rendered lint.

CHECK YOUR COPY
THE SHA-256 LINE

Each skill page prints the SHA-256 of its zip. Compare it with your download to confirm you have the copy that was tested. On macOS or Linux: `shasum -a 256 file.zip`.

PROJECT OR EVERYWHERE
START SMALL

Install into one project first. Move a skill to the every-project folder once it has earned a place. Fewer skills loaded means the agent picks the right one more often.

REMOVE OR UPDATE
IT IS A FOLDER

To remove a skill, delete its folder. To update, replace the folder with the new one. Overwatch and Intel are pinned to a reviewed upstream version, named on their pages.

Claude app steps are from Anthropic's help article "Use skills in Claude". Agent paths are from the open skills CLI's agent list, read Nov 27, 2026.

ONE BUILD, START TO FINISH

A WORKED EXAMPLE

Here is the whole chain on one job: a lead page for a roofing company. The company is made up, and "what you get back" describes the shape of each reply, not a recorded run.

	STEP	TYPE THIS	WHAT YOU GET BACK	YOUR CALL
1	**Pressure-test the idea**	`/recon11-dreamskillz founder a lead page for a roofing company`	Six questions, a verdict, and two other shapes of the idea.	Build it, shrink it, or stop.
2	**Look it up**	`/recon11-intel system roofing contractor storm damage homeowners`	A design system from the database: pattern, style, colors, type.	Keep what fits the subject. Drop stock palettes.
3	**Build it**	`/recon11-forge landing page for a roofing contractor, for homeowners after a storm, one job: book an inspection`	Art direction, what the page will avoid, the page plan, then the code.	Approve the direction before the build.
4	**Grade it**	`/recon11-titanium recon the home page`	Six scores, a grade, and the three moves that raise it most.	Which moves to make.
5	**Steer it**	`/recon11-titanium louder the hero`	One change to one target, reported in three lines.	Keep it or undo it.
6	**Scan it**	`/recon11-overwatch scan ./`	Every rule hit with file and line. Changes nothing.	What to fix now.
7	**Break it**	`/recon11-titanium armor the booking form`	Awkward input thrown at it, then fixes for what broke.	Nothing. Read the list.
8	**Finish it**	`/recon11-titanium temper the whole page`	The final pass, ending with a technical check.	Sign off.
9	**Ship it**	`/recon11-dreamskillz shipper`	The nine-line release checklist.	Say "ship it", or do not.

STEP 3 HAS FOUR DOORS

WHICH BUILDER

IF	USE
You have a clear brief and real content	Forge
You want a look you would not have picked	Hallmark
You want to steer every step in single words	Titanium, with brief then forge
The site argues one case, section by section	Exhibit by Exhibit
You have a ticket, menu or brochure with a look	Artifact to Website

WHY THIS ORDER

THE LOGIC

- **Think before you build.** An idea is cheaper to change than a page.
- **Look up before you design.** Data first, taste second.
- **Grade before you steer.** The grade tells you which moves matter.
- **Scan by rule after you sweep by eye.** Each catches what the other misses.
- **Break it before visitors do.**
- **Publish only on your word.** No skill here ships on its own.

SHORT ON TIME

THE THREE-COMMAND VERSION

Build with `/recon11-forge`, grade with `/recon11-titanium recon`, finish with `/recon11-titanium temper`. Add the rest as the stakes rise.

RECON11 TITANIUM

THE TWENTY MOVES

A design vocabulary of twenty named moves. Say what you want in one word, point it at one thing, and get a grade back: Scrap, Steel or Titanium.

```
/recon11-titanium <move> <target>
```

3 MOVES

SCOUT

`recon`	A graded review. Changes nothing.
`inspect`	A technical check with severities S1 to S4.
`sweep`	A fast hunt for the 20 tells.

3 MOVES

BUILD

`brief`	Plans the design in writing before any code.
`forge`	Builds a new page to the floor.
`rebuild`	Keeps the content, replaces the look.

8 MOVES

STEER

`louder`	Makes one thing lead on a timid page.
`quieter`	Calms a page that shouts.
`strip`	Removes what the page does not need.
`color`	Gives a gray page a palette from its subject.
`type`	Fixes fonts, scale and line length.
`space`	Fixes spacing and rhythm.
`motion`	Adds movement that answers the user.
`charm`	Adds one touch of personality.

6 MOVES

FINISH

`temper`	The final pass before shipping.
`armor`	Breaks it on purpose, then fixes it.
`plain`	Rewrites unclear copy.
`fit`	Adapts it to each screen size.
`speed`	Makes it load faster.
`welcome`	Designs first-run and empty states.

TYPE THIS	WHAT HAPPENS
`/recon11-titanium recon the home page`	A six-check score, a grade, and the shortest path up.
`/recon11-titanium sweep ./index.html`	Lists every tell it finds, with where it is. Changes nothing.
`/recon11-titanium louder the hero`	Makes one thing lead. Reports what changed in three lines.
`/recon11-titanium strip the pricing section`	Cuts what the section does not need, and lists the cuts.
`/recon11-titanium brief a booking page for a mobile dog groomer`	A written design plan, before any code.

RULES OF ENGAGEMENT

HOW EVERY MOVE BEHAVES

1. **One move, one target.** A second change needs a second ask.
2. **It reads before it changes,** and looks at the rendered page when it can.
3. **The brief wins.** A look you pin is followed, even one on the tell list.
4. **It stays inside the target.** Facts in your copy are never changed without asking.
5. **It reports in three lines:** the move, what changed, what to look at.
6. **Scout moves change nothing.**

FIELD NOTES

TIPS

1. **One move, one target.** Name the section, not the site.
2. **Open with recon.** It tells you which moves are worth making.
3. **Follow the shortest path up.** Three moves, in the order it gives them.
4. **Pin what you love.** If you name a look, a font or a color, it follows that even when it is on the tell list.
5. **Close with inspect, then temper.**
6. **Pair it with Overwatch.** Sweep by eye, then scan by rule.

TITANIUM: THE GRADE

SCRAP, STEEL OR TITANIUM

Every `recon` scores six checks from 0 to 2, adds them up out of 12, and names the three moves that would raise the grade most.

0 TO 5
SCRAP
No clear job, or it could belong to anyone. Start over with `brief`.

6 TO 9
STEEL
It works and it is honest, but it does not lead the eye or look like its subject yet. The steer moves live here.

10 TO 12, NO CHECK AT 0
TITANIUM
One job, one clear lead, a look only this subject could own, and it holds at every width. Ship it.

CHECK	A SCORE OF 2 MEANS
Purpose	One job, and a stranger can name it in five seconds.
Hierarchy	The eye lands on the main thing first, then the next.
Identity	It could only belong to this subject.
Craft	Spacing, type and alignment follow one system.
Function	Works at four widths, by keyboard, with readable contrast.
Honesty	Real content. No filler, no invented claims.

WHAT A RECON LOOKS LIKE
A SAMPLE REPORT

```
RECON  the home page          (sample)
Purpose    2  one job: book an inspection
Hierarchy  1  headline and photo compete
Identity   1  could be any contractor
Craft      2  one spacing scale, no strays
Function   1  form overflows at 390 wide
Honesty    2  no invented claims
TOTAL  9 of 12            GRADE: STEEL
Shortest path up
  1  louder  the hero
  2  color   the whole page
  3  fit     the booking form
```

THE TWENTY TELLS THAT `sweep` HUNTS

1. Blue to purple gradients, glowing blobs, glass panels.
2. Gradient text on headlines or numbers.
3. Identical rounded cards with the same soft shadow.
4. Cards inside cards.
5. A row of three icon cards, repeated.
6. Icons in colored circles or rounded squares.
7. A small all-caps label above every heading.
8. One word of the headline in italic or another color.
9. Every section centered.
10. The same padding on every section.
11. A fade and slide on every section.
12. Gray text on a colored ground.
13. Cream ground, serif headline, terracotta accent.
14. Near-black ground with one neon accent.
15. Numbers on things that are not a sequence.
16. Facts joined by middle dots, an arrow after every link.
17. A big number over a small label as the opener.
18. Filler copy such as "unlock the power of".
19. Testimonials, logos or statistics nobody supplied.
20. An overused default typeface chosen for no reason.

Each one is fine when your brief asks for it. On its own, it marks a page as generated.

BUILT INTO EVERY BUILD AND FINISH MOVE
THE FLOOR

- Colors and fonts go through named tokens.
- Works at 1440, 1024, 768 and 390 wide, with no sideways scroll.
- Text meets WCAG AA contrast, and keyboard focus is visible.
- No invented facts, metrics, testimonials, client names or logos.

RECON11 DREAMSKILLZ

NINE ROLES, ONE SPRINT

Nine specialists in one skill for a solo builder. Each role does one job, hands off to the next, and stops for your call at the gates.

```
/recon11-dreamskillz <role> <what you are working on>
```

ROLE	THE JOB	WHAT YOU GET BACK
`founder`	Pressure-tests the idea with six questions.	A verdict: build it, build smaller first, or not yet. Plus two other shapes of the idea.
`spec`	Turns the idea into a plan someone could build from.	Goal, scope, pages, data, what done looks like, build order.
`architect`	Decides how it gets built, sized to the job.	Stack, hosting, data, failure handling, secrets, tests, risky assumptions.
`designer`	Reviews the plan or page as a designer who codes.	Seven ratings from 0 to 10, what a 10 looks like, and the fixes in order.
`reviewer`	Reads the real changes for bugs that slip past a quick test.	Findings ranked must fix, should fix and note.
`tester`	Uses it like a customer, then tries to break it.	Steps, what happened, what should have happened, and what was not tested.
`shipper`	Runs the nine-line release checklist.	Each line marked done, failed or not applicable. Publishes only on "ship it".
`scribe`	Writes what changed in plain language.	Change notes and an updated README or help page.
`debrief`	Looks back at the sprint.	What to repeat, what went wrong and why, and one change for next time.

SPRINT

THE WHOLE RUN, WITH FOUR GATES

founder, your call, **spec**, **architect**, **designer**, your call, your call before the build, **build**, **reviewer**, **tester**, **shipper**, your call before publishing, **scribe**, **debrief**.

A sprint can stop at any handoff note and resume from it later.

EVERY ROLE ENDS WITH THIS

THE HANDOFF NOTE

```
HANDOFF from founder
Decided : build the one-page booking site first
Open    : which payment option, owner to choose
Next    : spec, with the three must-have pages
```

Sample wording, made up to show the shape.

TYPE THIS	WHAT HAPPENS
`/recon11-dreamskillz tester the checkout flow`	Walks it like a customer, then tries to break it.
`/recon11-dreamskillz shipper`	The nine-line release checklist. Publishes only on your word.
`/recon11-dreamskillz sprint a waitlist page for a new product`	Every role in order, stopping at the four gates.
`/recon11-dreamskillz careful on`	Asks before deleting, overwriting, publishing or spending.

FIELD NOTES

TIPS

1. **Start with founder.** It is cheaper to change an idea than a build.
2. **One role at a time.** Let it finish and read the handoff note.
3. **Keep the notes.** Ask for a `SPRINT.md` if the work will span sessions.
4. **Skip what does not fit.** A one-page site does not need an architect, and the skill will say so.

THE SIX FOUNDER QUESTIONS

ASKED ONE AT A TIME

1. Who has this problem today, and what do they do now?
2. What is the smallest version one person would use this week?
3. What would make them come back?
4. What are you assuming that you have not checked?
5. What does it cost you if you are wrong?
6. What are you deliberately not building?

CAREFUL ON

IT ASKS BEFORE IT

- deletes files or folders
- overwrites a file with no copy
- rewrites history
- drops or bulk-changes data
- publishes to a live site
- sends anything for you
- spends money

RECON11 FORGE

BUILD A NEW PAGE

Designs and builds a page that looks art directed instead of generated. Two design skills merged into one, with their contradictions settled.

```
/recon11-forge <what to build>
```

THE SHORT VERSION

WHAT IT DOES

- Writes an art direction before any code: tone, layout, four to six colors, type roles, and the one thing the page will be remembered for.
- Names three to five things this page will not do, chosen for this project.
- Builds to a floor: four screen widths, AA contrast, visible focus, reduced motion.
- Hands back four parts in order: direction, what it avoided, the page plan, the code.

REACH FOR IT WHEN

BEST USE

- A new landing page, marketing site, dashboard or app screen.
- A full rebuild of a page that looks like every other generated page.
- Any build where you can say who it is for and what its one job is.

USE SOMETHING ELSE WHEN

NOT THE RIGHT TOOL

- ☐ A small fix to a page that already works. Use Overwatch.
- ☐ You want a random fresh look every time. Hallmark deals themes.
- ☐ The page must match an existing system exactly. Say so in the brief and it will follow it.

TYPE THIS	WHAT HAPPENS
`/recon11-forge landing page for a Tampa roofing contractor, for homeowners after storm damage, one job: book an inspection`	Art direction, then the page.
`/recon11-forge dashboard for a five-truck fleet, used by a dispatcher on a laptop`	An Operate-mode screen: scanning and consistency first.
`/recon11-forge redesign ./index.html`	Says what makes the page look generated, sets a new direction, then rebuilds it.
`/recon11-forge`	Prints the field card and asks what the page is for and who it is for.

FIELD NOTES

TIPS

1. **Give it three things:** the subject, the audience, and the one job of the page.
2. **Say what it must not look like.** That steers it better than adjectives.
3. **Bring real copy and images.** It will write specific copy, and it will not invent facts, numbers or testimonials.
4. **One bold idea per page.** Ask it what the memorable thing is before it builds.
5. **The brief wins.** If you name a look, a font or a palette, it follows that exactly.
6. **Finish with Overwatch.** Run audit on what Forge built.

IN THE MERGE

WHAT WAS CORRECTED

1. One source carried two frontmatter blocks, one of them inside the body. Fixed.
2. Its fallback look, quiet Swiss minimal, is itself a generated default. The fallback is now to derive the look from the subject.
3. The two sources disagreed on motion. One rule now: motion that answers the user is fine, unprompted motion is used once.
4. The two disagreed on cards. Now: no cards by default, matched cards when they share a grid.
5. The brief now outranks every rule, so a banned look is allowed when you ask for it.
6. Added four surface modes and a build floor.

SEVEN LOOKS IT STAYS OUT OF UNLESS YOU ASK

THE USUAL SUSPECTS

Cream with a serif and terracotta. Near-black with one neon accent. Broadsheet rules and square corners. Identical rounded cards. Blue to purple gradients and glass. Stock chrome such as all-caps eyebrows and arrow links. Quiet Swiss minimal with one deep accent.

RECON11 OVERWATCH

REVIEW AND FINISH

Reviews and finishes UI that already exists, and scans code for the marks of generated design before it ships.

```
/recon11-overwatch <command> <target>
```

THE SHORT VERSION

WHAT IT DOES

- Twenty-four commands, each a playbook for one move: critique, audit, polish, bolder, quieter and the rest.
- A detector that scans your UI files against 62 fixed rules. No model and no API key are involved.
- Two project files it writes once and reads every time: PRODUCT.md for who the product is for, DESIGN.md for how it looks.

REACH FOR IT WHEN

BEST USE

- One page, one section or one component, right before it ships.
- A page that looks generated and you cannot say why.
- A last accessibility and responsive check before a deploy.

USE SOMETHING ELSE WHEN

NOT THE RIGHT TOOL

- ☐ You are starting from a blank page. Use Forge, then come back.
- ☐ The code has no interface in it.
- ☐ You want a whole site judged at a glance. Narrow targets get better results.

TYPE THIS	WHAT HAPPENS
`/recon11-overwatch critique the pricing section`	A scored UX review with the fixes ranked.
`/recon11-overwatch audit src/pages/checkout`	Accessibility, performance and responsive findings, ranked P0 (blocking) to P3 (polish).
`/recon11-overwatch scan ./src`	Runs the detector only and lists every hit with file and line. Changes nothing.
`/recon11-overwatch quieter the landing page`	Tones down a loud design. `bolder` does the reverse.
`/recon11-overwatch polish settings`	The final pass before shipping.
`/recon11-overwatch harden checkout`	Covers errors, overflow, long text and edge cases.

FIELD NOTES

TIPS

1. **Go narrow.** One section reviewed closely beats a whole page skimmed.
2. **Scan first.** It takes seconds and tells you where to point the other commands.
3. **Run init once per project** so every command knows who the page is for.
4. **End with audit, then polish.** In that order.
5. **Live mode needs a local dev server.** It is for a local checkout, never a deployed site.
6. **The first run needs the network once** to fetch the engine, which the launcher checks against a pinned checksum.

THIS EDITION

WHAT RECON11 CHANGED

1. Usage data is switched off on every call. Upstream ships with it on.
2. It will not sign in, install, update, pin or add hooks unless you ask for that by name.
3. Live mode keeps edits in your chat instead of starting a second agent with permission checks off.
4. Added `scan`, a plain detector run.
5. Nothing is called fixed until a scan or a rendered look confirms it.
6. Recon11 standards added: four screen widths, AA contrast, no invented facts or testimonials, commas instead of em dashes, transparent logos, matched cards.

BEFORE YOU START

WHAT IT NEEDS

- macOS, Linux or Windows on x64, or macOS or Linux on arm64.
- Network access on the first run, for one engine download.
- No Node and no Python.
- In the Claude app, the engine download may be blocked. The playbooks still work there, and the skill says so when a scan could not run.

RECON11 INTEL.

LOOK IT UP

A design database you can search. Ask for a whole design system or one rule, and it answers from data files you can open and check.

```
/recon11-intel <what you are building or fixing>
```

THE SHORT VERSION

WHAT IT DOES

- Holds 80 styles, 194 palettes, 74 font pairings, 141 UX guidelines, 105 icons, 20 GSAP presets, 25 chart types and guidance for 23 stacks.
- Builds a full design system from a short brief: pattern, style, colors, type and what to avoid.
- Exports that system as CSS variables or design-token JSON.
- Runs offline. It is Python reading CSV files.

REACH FOR IT WHEN

BEST USE

- The first ten minutes of a build, before any code.
- One precise question: which chart, which font pairing, what the accessibility rule says.
- Framework-specific answers for the stack you are in.

USE SOMETHING ELSE WHEN

NOT THE RIGHT TOOL

- ☐ You already have brand colors and fonts. Those win over the database.
- ☐ You want taste. It returns category defaults, so screen what comes back.
- ☐ You need the page reviewed. That is Overwatch.

TYPE THIS	WHAT HAPPENS
`/recon11-intel system veteran services nonprofit donation page`	A full design system for that product and audience.
`/recon11-intel ask ux focus not obscured`	The matching accessibility guideline, with do and do not.
`/recon11-intel ask chart real-time fleet status`	Chart types that fit, and what to avoid.
`/recon11-intel ask typography legal services serious`	Font pairings for that tone.
`/recon11-intel stack html-tailwind chip label overflow`	How to do it in that framework.

FIELD NOTES

TIPS

1. **Two to five keywords, one intent.** A long sentence searches worse than three good words.
2. **Name the product type and the audience.** That is what the database is indexed by.
3. **Say the domain** when a result is off topic: ux, color, typography, chart, icons, landing, style, product.
4. **Use the dials** on a system: variance, motion and density, each 1 to 10.
5. **Screen the palette.** In testing, a consulting brief came back blue and violet, and a trucking brief came back neon green on black.
6. **Nothing is saved** unless you ask it to write files.

THIS EDITION

WHAT RECON11 CHANGED

1. Calls `python3` first. Upstream examples use `python`, which many machines do not have.
2. Fixed the stack list. Upstream lists 22 and leaves out `pyside6-qss`, which ships in the data.
3. Added a screen for generated-looking results, and a hard rule against placeholder testimonials, logos and numbers.
4. Every recommendation names the file and row it came from.
5. Added four plain modes: system, ask, stack and tokens.
6. Recon11 standards added: four screen widths, AA contrast, no invented facts or testimonials, commas instead of em dashes, transparent logos, matched cards.

BEFORE YOU START

WHAT IT NEEDS

- Python 3. Nothing to install with it.
- No network. The search reads local files.

ASK TAKES ONE OF THESE

DOMAINS

`product` `style` `color` `typography` `google-fonts` `chart` `ux` `landing` `icons` `gsap` `react` `web`

RECON11 QUARTERMASTER

FIND IT, READ IT, INSTALL IT

Finds, checks and installs agent skills, and reads them before they run on your machine.

```
/recon11-quartermaster <command>
```

THE SHORT VERSION

WHAT IT DOES

- Drives the open `skills` CLI: find, add, list, check, update, remove and init.
- Runs a vet before any install: license, scripts, network calls, telemetry, and the commit it read.
- Scaffolds a new skill that follows the format.
- Carries a list of the 28 skills on Vercel's docs page, with where each installs from.

REACH FOR IT WHEN

BEST USE

- Adding a skill from GitHub or skills.sh.
- Someone sent you a skill and you want to know what it does before it runs.
- Turning your own instructions into a skill you can share.

USE SOMETHING ELSE WHEN

NOT THE RIGHT TOOL

- ☐ Installing plugins or MCP servers. Those are different things.
- ☐ You have no terminal. The CLI needs Node.js.
- ☐ A site hosted off Vercel does not need the Vercel deploy skills on the list.

TYPE THIS	WHAT HAPPENS
`/recon11-quartermaster find accessibility review`	Searches the skills.sh directory.
`/recon11-quartermaster vet nutlope/hallmark`	Clones it to a scratch folder, reads it, and reports license, scripts, network calls and commit.
`/recon11-quartermaster install vercel-labs/agent-skills web-design-guidelines`	Vets it, shows the report, then installs that one skill.
`/recon11-quartermaster list`	What is installed, and where.

THE CLI BEHIND IT	COMMAND
Search	`npx skills find <query>`
See what a repo offers	`npx skills add <owner/repo> --list`
Install one skill	`npx skills add <owner/repo> --skill <name> -a <agent>`
Install for every project	`add -g`
List, check, update, remove	`npx skills list, check, update, remove`
Scaffold a new skill	`npx skills init <name>`

ALWAYS BEFORE INSTALL

THE VET, STEP BY STEP

1. Clone the repo into its own empty scratch folder. Run nothing from inside it.
2. Record the commit hash and date.
3. Find the license file and name the license.
4. Read SKILL.md end to end. Flag anything that skips permission checks, sends data out, reads credentials or installs something else.
5. List every script. Search for network calls, shell execution and environment reads.
6. Note what it downloads at run time, what it sends as usage data, and any login it wants.
7. Report in five lines, with a plain recommendation.
8. Install only after you have seen the report.

FIELD NOTES

TIPS

1. **One skill at a time.** A repo can hold dozens, so name the one you want.
2. **Project first.** Install into the project, and go global only after it has earned it.
3. **Write down the commit you vetted.** An update is a new thing to read.
4. **Set `DISABLE_TELEMETRY=1`.** The CLI sends anonymous usage data unless you do.
5. **A search hit is not a recommendation.** Vet it like any other.

Why read first. A skill is instructions plus, sometimes, scripts that run with your permissions. The five-line report tells you what it does, its license, what it runs, what it reaches, and whether to install.

BEFORE YOU START

WHAT IT NEEDS

- Node.js 22.20 or later, for the `npx skills` commands.
- Git, for the vet step.
- Network access.

THREE MORE

ARTIFACT, EXHIBIT, HALLMARK

RECON11 ORIGINAL, FIRST SHOWCASE BUILD

ARTIFACT TO WEBSITE

Upload a concert ticket, a menu or an old brochure. Get a responsive page whose design comes from it, with three annotations tracing the choices back.

```
/recon11-artifact-to-website <photo or scan>
```

Best use

- A ticket, menu, brochure, label, poster or sign with a look worth keeping.
- An event, a restaurant or a small shop that wants a page with character.
- Showing a client where every design choice came from.

Tips

- **Shoot it flat and well lit.** Glare and angles hide the details it reads.
- **Say what matters.** If you love the border or the stamp, name it.
- **Bring the real facts.** The words on the artifact describe the original, not your page.

The three annotations

- **Lettering to headline.** The dominant lettering sets the headline style.
- **Structure to action.** A stub, a tear line or a price box becomes where the visitor acts.
- **Ink to palette.** The inks and the stock become the color tokens.

It borrows a look. It will not copy a logo, a mascot or a real person, or make something that passes as a real ticket.

RECON11 ORIGINAL, TAKEN FROM A LIVE SITE

EXHIBIT BY EXHIBIT

The design system behind the Die Hard Is a Christmas Movie site: a page that argues one case, where every block is an object from that world.

```
/exhibit-by-exhibit-design <subject and argument>
```

Best use

- A fan, holiday or opinion site that makes one argument and proves it section by section.
- A new page for the Die Hard site itself.
- Any themed site where plain cards would look wrong.

Tips

- **Name the argument first.** One sentence the whole site proves. It becomes the headline, the stamp and the footer line.
- **Pick the world before any CSS.** List eight to ten real objects from it.
- **One object per section.** No component repeats back to back.

Its twelve color tokens



Plus four typefaces, twenty components, seven motion rules and four breakpoint tiers, all read from the live site.

THIRD-PARTY SKILL, MIT, BY THE HALLMARK CONTRIBUTORS

HALLMARK

A design skill for new pages. It sets a hard floor against generated defaults and deals a direction, so two builds in a row do not come out alike.

```
/hallmark <prompt> | redesign <path> | audit <path>
```

Best use

- A brand-new page where you want a direction you would not have picked.
- A new build you want linted before you see it.
- A redesign or an audit, when you name Hallmark and give it a path.

Tips

- **Lock your brand.** A `design.md` at the project root makes Hallmark use your system instead of dealing a theme.
- **Pin what you reviewed.** Recon11 read version 2.0.0 at commit 04977e1.
- **Know what goes online.** Its scripts reach Wikimedia Commons for photos, and Together AI only when a Together API key is set.

Install from its own repo

```
npx skills add nutlope/hallmark -g
```

Recon11 does not repackage it. A `design.md` at your project root makes it use your system. Its floor bans small tracked-capital labels and holds a screen to about 80 words, so pick one rule set per build.

OUTSIDE SKILLS

20 BY OTHER MAKERS

20 design and build skills made by other people, with what each has built where its own repo shows examples. Numbers 12 to 20 are new in this edition. Full notes and links are on the site's Outside page.

	SKILL	REPO ON GITHUB	LICENSE	WHAT IT HAS BUILT
01	**Impeccable**	`pbakaus/impeccable`	Apache 2.0	The Neo Mirai case study on impeccable.style, and a with-and-without example in its README.
02	**frontend-design**	`anthropics/skills`	Apache 2.0	No example gallery ships with the skill.
03	**UI UX Pro Max**	`nextlevelbuilder/ui-ux-pro-max-skill`	MIT	Three sample projects in its repo: a healthcare dashboard, a dark portfolio, a SaaS landing page.
04	**web-design-guidelines**	`vercel-labs/agent-skills`	None at repo root	It reviews code, it does not build.
05	**Stitch design-md**	`google-labs-code/stitch-skills`	Apache 2.0	An example DESIGN.md in the skill folder, written from a furniture collections page.
06	**gstack**	`garrytan/gstack`	MIT	Its README points to "gstack + Capy: the software factory" and "Why gstack".
07	**Hallmark**	`nutlope/hallmark`	MIT	Its own site, usehallmark.com. No separate gallery found.
08	**Taste Skill**	`Leonxlnx/taste-skill`	MIT	An examples folder in its repo, and a launch film on its v2.0.0 release.
09	**Emil Kowalski's skills**	`emilkowalski/skill`	MIT	Example results on each skill page at aiforui.dev/skills, per its README.
10	**UI Skills**	`ibelick/ui-skills`	MIT	None found in its README. More at ui-skills.com.
11	**Superpowers**	`obra/superpowers`	MIT	A working method, not a builder. No gallery.
12	**Remotion Agent Skills**	`remotion-dev/skills`	None at repo root	No gallery. Its README gives an example prompt for most skills.
13	**shadcn**	`shadcn/ui`	MIT	It manages components in your project. No gallery.
14	**agent-browser**	`vercel-labs/agent-browser`	Apache 2.0	It tests pages, it does not build them.
15	**HyperFrames**	`heygen-com/hyperframes`	Apache 2.0	A showcase at hyperframes.heygen.com/examples, linked from its README.
16	**The Designer Skills Pack**	`Owl-Listener/designer-skills`	MIT	No gallery found. Ships an index that sorts skills by situation.
17	**Designer Skills**	`julianoczkowski/designer-skills`	Apache 2.0	A YouTube walkthrough linked from its README. No gallery found.
18	**Skills for Real Engineers**	`mattpocock/skills`	MIT	A working method, not a builder. No gallery.
19	**Karpathy guidelines**	`forrestchang/andrej-karpathy-skills`	MIT, per README	Not a builder. An EXAMPLES.md shows each principle applied.
20	**Anthropic example skills**	`anthropics/skills`	Apache 2.0	webapp-testing ships example test scripts. No design gallery found.

BEFORE YOU INSTALL ANY

READ IT FIRST

- A skill is instructions plus, sometimes, scripts that run with your permissions.
- Run `/recon11-quartermaster vet owner/repo`.
- Read the five-line report: what it does, its license, what it runs, what it reaches, and whether to install.

HOW THE LIST WAS MADE

THREE STEPS

1. **Found.** The skills named for the page, plus ones that kept coming up in public roundups and install lists.
2. **Opened.** Each repo was cloned, and its README, license and skill folders read.
3. **Examples.** Only what the authors publish themselves. No star counts, no reviews.

NOTHING IS RANKED

WHAT THE COLUMNS MEAN

License is what the license file in the repo says. "None at repo root" means no license file was there when checked, which is not the same as free to reuse.

What it has built lists only what the authors publish themselves. Where no examples were found, it says so.

BUSINESS SKILLS

15 FOR THE WORK AROUND THE BUILD

15 skills and plugins for selling, books, contracts, marketing, product decisions, documents and plain writing. Install commands for each are on the site's Business page.

	SKILL	BY	THE JOB	REPO ON GITHUB	LICENSE
01	**Small Business**	Anthropic	Run the whole shop	`anthropics/knowledge-work-plugins`	Apache 2.0
02	**Sales**	Anthropic	Prep a call or work a pipeline	`anthropics/knowledge-work-plugins`	Apache 2.0
03	**Finance**	Anthropic	Close the books	`anthropics/knowledge-work-plugins`	Apache 2.0
04	**Legal**	Anthropic	Get a first pass on a contract	`anthropics/knowledge-work-plugins`	Apache 2.0
05	**Marketing**	Anthropic	Plan and draft a campaign	`anthropics/knowledge-work-plugins`	Apache 2.0
06	**Operations, HR and Support**	Anthropic	Write the process down, hire, answer tickets	`anthropics/knowledge-work-plugins`	Apache 2.0
07	**Data**	Anthropic	Answer a question from your numbers	`anthropics/knowledge-work-plugins`	Apache 2.0
08	**Productivity**	Anthropic	Keep tasks and context in one place	`anthropics/knowledge-work-plugins`	Apache 2.0
09	**Claude for Financial Services**	Anthropic	Build a model, a memo or a pitch	`anthropics/financial-services`	Apache 2.0
10	**Marketing Skills**	Corey Haines	Fix a page that does not convert	`coreyhaines31/marketingskills`	MIT
11	**PM Skills Marketplace**	Pawel Huryn	Decide what to build	`phuryn/pm-skills`	MIT
12	**E-Commerce Skills**	Nexscope	Sell more on a marketplace	`nexscope-ai/eCommerce-Skills`	MIT
13	**Document skills**	Anthropic	Make the Word, Excel, PowerPoint or PDF file	`anthropics/skills`	Apache 2.0
14	**Humanizer**	Siqi Chen	Make a draft sound like a person	`blader/humanizer`	MIT
15	**Stop Slop**	Hardik Pandya	Cut the filler from a draft	`hardikpandya/stop-slop`	MIT

THREE LARGE LIBRARIES, TO BROWSE ONE SKILL AT A TIME

LIBRARY	REPO ON GITHUB	WHAT IT IS	LICENSE
claude-skills, Alireza Rezvani	`alirezarezvani/claude-skills`	Its README counts 434 skills covering engineering, product, marketing, compliance, finance and the C-suite, and publishes its own load-test results.	MIT
Claude Skills Library, OneWave AI	`OneWave-AI/claude-skills`	Its README counts 230 skills in three groups: business (sales, marketing, consulting, operations), everyday life, and coding.	MIT
Claude-Skills, borghei	`borghei/Claude-Skills`	A large library with board, C-level, sales, HR and data sets. 404 skill files when checked.	MIT with the Commons Clause

HABIT 1

TREAT IT AS A DRAFT

These skills draft work for a person to review. The finance, legal and small business plugins say so in their own READMEs. Check the numbers, and have a qualified professional review anything legal, tax or financial before you rely on it.

HABIT 2

READ IT FIRST

A skill runs with your permissions. Run `/recon11-quartermaster vet owner/repo`, read the report, then install.

Counts on this page are counts of skill files in the copy read on Nov 27, 2026, unless a README gave the number.

HABIT 3

CONNECT ONE TOOL AT A TIME

Several of these work better connected to a CRM, the books or an inbox. Those are live accounts. Start with one connection, try it on a copy or an export, and add the next when the first has earned it.

None of these skills was run for this guide.

FIXES, WORDS AND CREDITS

WHEN SOMETHING GOES WRONG

The problems people hit first, the words this guide uses, and whose work it stands on.

IF THIS HAPPENS	DO THIS
The slash command does nothing	Start a new session after installing. Check that the folder name matches the skill name and sits directly in the skills folder.
The field card did not print	The skill did not load. Ask the agent which skills it has, then reinstall.
Overwatch says the scan could not run	Its engine download was blocked. The playbooks still work. Run the scan somewhere with network access.
Intel says python was not found	Install Python 3. The skill tries python3, then python, then py -3.
Results look generic	Give the subject, the audience and the one job. Say what the page must not look like.
It asked me a question instead of building	That is by design when a fact is missing. Skills here do not invent content.
A plugin command is not recognized	Lines that start with claude or npx run in a terminal. Lines that start with a slash are typed inside Claude Code.
Two skills answer the same request	Remove the one you use less, or call the one you want by its slash name.

WORD	WHAT IT MEANS HERE
Skill	A folder with a SKILL.md that teaches an agent one job.
Field card	The short card a Recon11 skill prints first, every time.
Move	One named change to one target, in Titanium.
Role	One specialist with one job, in DreamSkillz.
Handoff note	Decided, open and next, written at the end of a role.
Tell	A habit that marks a page as generated.
The floor	The standards every build must meet.
Vet	Reading a skill, its license and its scripts before installing.
Plugin	A bundle of skills, commands and connectors installed together.
Connector	A link from the agent to one of your apps, such as a CRM.

CREDITS AND TERMS

WHOSE WORK THIS STANDS ON

Overwatch is a Recon11 edition of Impeccable by Paul Bakaus (Apache 2.0). Intel is a Recon11 edition of UI UX Pro Max by Next Level Builder (MIT). Forge merges frontend-design by Anthropic (Apache 2.0) with Human Frontend Direction by Recon11. Quartermaster describes the skills CLI by Vercel (MIT). Titanium credits Impeccable and frontend-design for ideas. DreamSkillz credits gstack by Garry Tan (MIT) for its idea. Hallmark is by the Hallmark contributors (MIT).

The skills on pages 13 and 14 belong to their authors. They are listed here, not repackaged, and Recon11 has no tie to any of them.

Recon11 editions are independent. They are not affiliated with or endorsed by any author named in this guide. Each download carries its license and a notice of what was changed. Skills run with your permissions, so read one before you install it. Provided as is, with no warranty.

Recon11 Global Systems, LLC. Edition 2, November 27, 2026.

NEW IN EDITION 2

WHAT CHANGED

- Outside skills grew from eleven to twenty.
- A business skills page, with fifteen skills and three large libraries.
- Fixes, the glossary and credits moved to this page.
- The guide went from 13 pages to 15.

RECON11SKILLARMORY.NETLIFY.APP

ON THE SITE

- **Every skill page:** full instructions, the skill file in full, a download and its SHA-256.
- **Titanium:** try six steer moves on a sample block.
- **Artifact to Website:** the 1994 ticket demo, with its toggle and three annotations.
- **Outside and Business:** links, licenses and install commands.
- **Showcase:** ten skill ideas, one built.

Next on the board: One Brief, Three Builds. Website Time Machine. Homepage First Impression. Mobile Recut. Brand Continuity Test. Real Content Stress Test. Motion With Purpose. Form Friction Review. Portfolio Story Editor.