

SKILLFORGE TOOLKIT

Autonomous Skill Creation System for AI Agents

Product Guide & Prompt

Give your AI the instinct to capture, package, and index every useful piece of knowledge it produces — **permanently and automatically.**

Table of Contents

1. What This Is
 2. What Problem It Solves
 3. How It Works
 4. Compatible Platforms
 5. The Prompt
 6. Tips & Tricks
 7. Customization Guide
 8. Frequently Asked Questions
-

1. What This Is

SkillForge is a standing directive that turns your AI agent into a self-improving system.

Every time your AI solves a problem, writes a prompt, builds a workflow, finds a workaround, configures an integration, or learns from a failure — it automatically packages that knowledge into a reusable "skill" that can be retrieved and executed again later.

Think of it as giving your AI the instinct to take notes, build SOPs, and organize its own playbook — permanently and automatically.

2. What Problem It Solves

- You solved something complex last week but your AI has no memory of it. You re-explain everything from scratch.
- You built a great workflow or prompt but it was never saved. It is gone.
- Your AI makes the same mistake twice because the lesson was never captured.
- Multiple agents or team members use AI but knowledge is siloed. One breakthrough never reaches the others.
- You keep doing repetitive setup because nobody documented the process the first time.

SkillForge eliminates all of this. Every useful piece of knowledge is automatically captured, structured, indexed, and available for future use.

3. How It Works

Step 1: Paste the SkillForge directive into your agent's system prompt or send as a standing instruction.

Step 2: Your AI now runs a background check after every interaction: "Did anything useful just happen?"

Step 3: If yes, it creates a skill file using a standardized template with triggers, steps, dependencies, and expiry dates.

Step 4: Every skill is logged in a master index (SKILL-INDEX.md) — the single source of truth.

Step 5: Weekly, the AI audits the library — updating stale skills, retiring obsolete ones, flagging gaps.

Step 6: Over time, your system builds a growing, self-maintaining knowledge base.

4. Compatible Platforms

- OpenClaw
- AutoGPT / CrewAI
- Custom agent frameworks
- Claude Projects
- ChatGPT with persistent memory
- Any multi-agent orchestration system

Single-agent setups work perfectly. Multi-agent setups get even more value.

5. The Prompt

Copy everything below and paste into your agent's system prompt or send as a directive.

SKILLFORGE — STANDING DIRECTIVE

Priority: ALWAYS ON — Permanent standing order

THE DIRECTIVE

Effective immediately, this is a permanent standing order. Every time you or any agent solves a problem, builds a workflow, creates a process, writes a prompt, finds a workaround, develops a strategy, figures out a fix, or learns something useful — turn it into a skill. This is automatic. If it was useful once, it will be useful again.

WHAT COUNTS AS A SKILL

1. A problem was solved — fix, root cause, exact steps
2. A prompt was written — full text, word for word, never summarized
3. A workflow was built — multi-step process, in order, with context
4. A workaround was found — limitation, workaround, risks, expiry
5. A template was created — any reusable document or framework
6. A decision framework was used — logic/criteria captured for reuse
7. An API or integration was configured — every step and gotcha
8. A research method worked — strategy that produced quality results
9. An agent protocol was established — rules, escalation paths, behaviors
10. Code or a script was written — full code, instructions, dependencies
11. A communication pattern worked — outreach/messaging that got results
12. Something failed and we learned why — what went wrong and how to prevent it

SKILL FILE STRUCTURE

SKILL NAME		SKILL ID		CREATED BY		DATE		VERSION		CATEGORY		TRIGGER
DESCRIPTION: 2-3 sentences												
CONTEXT / BACKSTORY: How this skill came to exist												
THE SKILL: The actual content (steps, prompt, code, framework, template)												
WHEN TO USE: Specific scenarios with examples												
WHEN NOT TO USE: Edge cases to prevent misuse												
DEPENDENCIES: APIs, tools, skills, versions needed												
RELATED SKILLS: Cross-references by Skill ID												
EXPIRY / REVIEW CONDITION: Shelf life or 'Evergreen'												
CHANGELOG: Version history												

CATEGORIES: troubleshooting, prompt-engineering, workflow, api-integration, code, template, research, decision-framework, agent-protocol, communication, failure-log, platform-knowledge, automation, data, security. Create new categories as needed.

SKILL INDEX: Maintain SKILL-INDEX.md — lists every skill by ID, name, category, creator, dates, status. Single source of truth. Not in the index = does not exist.

AUTOMATIC DETECTION:

- After every conversation/task: "Did anything produce reusable knowledge?" If yes — create skill.
- After every error: "What went wrong? Create a failure-log skill."
- After every research task: "Did the method work well enough to formalize?"
- Weekly: Audit full index. Update stale skills, retire obsolete ones, flag gaps. Deliver Skill Health Report.

QUALITY GATES: 1) Zero-context test 2) Trigger clarity 3) Step specificity 4) Completeness 5) Duplicate check 6) Prompt integrity (word for word)

MULTI-AGENT: Lead agent reviews all skills and maintains index. Technical agents own code/API skills. Research agents own research skills. Specialists own their domain. Everyone flags skill opportunities. Single agent handles everything.

RETROACTIVE SWEEP — DO THIS FIRST: Go back 7+ days. Capture every problem solved, prompt written, workaround found, config set up, process figured out, mistake fixed.

QUICK-START: 1) Paste directive 2) Create SKILL-INDEX.md 3) Retroactive sweep 4) Pick audit day 5) Assign ownership if multi-agent 6) Confirm agent understands 7) Work normally — skills auto-generate.

SkillForge is always on.

6. Tips & Tricks

Getting Started

- Test immediately — solve a simple problem and see if a skill is created automatically.
- The retroactive sweep is the most important first step. Your last 7 days probably contain 10-20 uncaptured skills.
- Create SKILL-INDEX.md first, even if empty. It signals the system is real.

Maximizing Quality

- The zero-context test is the top quality gate. Could a brand-new agent execute this cold?
- "WHEN NOT TO USE" prevents misapplication. A Python fix should not trigger for JavaScript.
- Expiry dates prevent stale workarounds from polluting your library.

Building Momentum

- First 2 weeks: you may need to remind the AI. After that, it becomes automatic.
- At 25 skills, do a full review. Merge overlapping ones. Clean indexing early prevents sprawl.

Multi-Agent Power Moves

- Cross-pollination: one agent's skill should be usable by all. The index enables this.
- Assign a skill champion — one agent that reviews everything before it enters the index.
- Route failure-logs to your most senior agent. Failures need the most judgment to document.

Organizing Your Library

- Name skills descriptively. "fix-auth-error" is bad. "openai-api-401-expired-key-fix" is good.
- Keep SKILL-INDEX.md sorted by category then by date.
- If one category grows large, split it (e.g., troubleshooting-api, troubleshooting-ui).

Advanced: Combining Systems

- Pair with the Conversation Summary Toolkit — summaries capture what happened, SkillForge captures reusable knowledge.
- Export skills periodically. They transfer to any platform — just text files.
- Use skills as onboarding. Point new agents at the index first. Instant knowledge transfer.

The Weekly Audit

- Pick a consistent day. Non-negotiable. This keeps the system healthy.
- Look for: skills needing updates, skills to retire, and gaps where repetitive work has no skill.
- Skill Health Report: skills added, updated, retired, gaps identified, total active. 5 lines max.

7. Customization Guide

Add categories: sales, design, legal, onboarding — whatever fits your work.

Modify the template: Add or remove sections. If you never use "RELATED SKILLS," remove it. If you need "ESTIMATED TIME," add it.

Add detection triggers: "After every client interaction..." "After every deployment..." Tailor to your workflow.

Change audit frequency: Weekly for most. Twice-weekly for high-volume (50+/day). Biweekly for low-volume.

Scale for teams: Each person's agent runs SkillForge independently. A lead agent merges the best skills into a shared index monthly.

8. FAQ

Q: Single or multi-agent?

A: Both. Single agents handle everything. Multi-agent setups distribute ownership by domain.

Q: How many skills in month one?

A: Active users: 30-60. Retroactive sweep alone: 10-20.

Q: Low-quality skills?

A: Quality gates catch most issues. Add: "Show me a draft before finalizing" for manual review.

Q: ChatGPT compatible?

A: Yes, with persistent memory or Projects. Periodically export skills to a separate document.

Q: Large library?

A: Category system and index handle scale. At 100+ skills, split index by category into sub-files.

Q: Slows down responses?

A: No. Detection is lightweight. Skill creation only triggers when something genuinely useful was produced.

Q: Share across platforms?

A: Yes. Skills are plain text. Transfer anywhere. Format is universal by design.