

ClaudeKit Engineer

v2.20.0

Updated Jun 2026

Skill Reference Guide

1,200+

Commits

88

Core Skills

1,600+

Files

90+

Hook Files

NEW IN v2.20.0:

ghpm • html-video • vibe • review-pr • chrome-profile • managed-hooks self-heal

ck:plan AgentWiki+HTML/GitHub • watzup cross-branch • Shopify GraphQL • brainstorm wiki mode

BREAKING: chrome-devtools removed • Human MCP off by default • bash hook runner dropped • mcp-management absorbed

Recommended Workflows

Standard Dev Flow

```
/ck:brainstorm → /ck:predict → /ck:plan → /ck:scenario → /ck:cook → /ck:test →  
/ck:code-review → /ck:ship → /ck:watzup
```

Large Features (Parallel)

```
/ck:brainstorm → /ck:predict → /ck:plan → /ck:team cook → /ck:team code-review →  
/ck:ship
```

Optimization Tasks

```
/ck:scout → /ck:loop → /ck:code-review → /ck:ship
```

Bugfixes (v2.0.0 Pipeline)

```
/ck:scout → /ck:debug → /ck:fix → /ck:test → /ck:code-review
```

Investigation

```
/ck:scout → /ck:debug → /ck:brainstorm → /ck:plan
```

What's New in v2.17.0 – v2.20.0

v2.20.0 — New Skills & Workflows

- **/ck:ghpm** — GitHub project management: issues, Projects boards, handoffs, gh/API automation
- **/ck:html-video** — HTML/CSS/JS templates → local MP4 via nexu-io/html-video
- **/ck:vibe** — full vibe-coding pipeline end-to-end
- **/ck:review-pr** — formal gh PR review (--reply) + anti-ai-slop reference; duplicate/product gates
- **/ck:chrome-profile** — real Chrome profile workflow (replaces deleted chrome-devtools)
- **/ck:brainstorm** — --html wiki + problem-first modes
- **/ck:plan** — AgentWiki publish flag, HTML/GitHub output modes, time estimates removed
- **/ck:watzup** — cross-branch handoff scanner; next-step ranking with plan-progress + roadmap scan
- **/ck:ship** — publishes journals to AgentWiki when available
- **/ck:git** — PR merge automation workflow
- **/ck:shopify** — modernized to current shopify.dev (2026-06), GraphQL-first API

v2.20.0 — Hooks & Platform

- **Managed-hooks manifest** — deterministic CLI self-heal of shipped hooks
- **Canonical hook wiring** — overhaul + bash runner dropped

- **Workflow artifact gate** — opt-in validation gate for fix/cook review
- **Human MCP** — disabled by default
- **code-review** — adversarial stage removed; policy-safe red-team wording

v2.17.0 – v2.19.0 — Skills & Catalog

- **v2.17.0 new skills:** agentize, ai-multimodal, cti-expert, excalidraw, graphify, show-off, xia, stitch (--project-name)
- **v2.17.0** — worktree --base; deep + TDD plan-cook workflows; skill schema contract + 82 descriptions hardened
- **v2.18.0 new skill:** tech-graph (vendored fireworks-tech-graph)
- **v2.18.0** — ck:predict --chain, ck:scenario saturation, ck:security red-team personas, ck:autoresearch family router
- **v2.18.0 removals:** chrome-devtools, template-skill, kanban, mcp-management (absorbed into use-mcp)
- **v2.19.0** — scout-first, exact-requirements, no-side-effects skill gates; generated context hooks disabled by default

v2.14.0 Highlights

- **/ck:ship** — One command: merge → test → review → commit → push → PR. Supports **official** and **beta** release flags.
- **/ck:team v2.1.0** — Event-driven hooks, cross-session agent memory, file ownership via glob patterns, parallel collaboration.
- **/ck:fix v2.0.0 rewrite** — Mandatory 3-skill pipeline: Scout → Diagnose → Assess → Implement → Verify. Auto-triggers /ck:problem-solving on multiple hypothesis failures.
- **/ck:code-review** — Always-on red-team: scope gating, context anchoring, attack coverage. 4 input modes.
- **/ck:security-scan** — Automated secret detection, vulnerability scanning, OWASP patterns, dependency auditing.
- **/ck:deploy** — Single skill for all platforms: Vercel, Netlify, Cloudflare, Railway, Fly.io, Render, Heroku, AWS, GCP.
- **/ck:design** — Consolidated: 55 logo styles, 50 deliverables, 22 banner styles, 15 icon styles.
- **/ck:preview** — Self-contained HTML with dark/light toggle. 7 modes + --html for browser output.
- **/ck:problem-solving** — 6 sub-techniques: collision-zone, inversion, meta-pattern, scale-game, simplification, when-stuck.
- **Hook system** — 90+ hook files: scout-block, session-state, diagnostics, privacy-block (managed-hooks self-heal v2.20.0).

Skill Catalog

Core Development Workflow

Skill	When to Use
/ck:brainstorm	Ideation, architecture decisions, technical debates; --html wiki + problem-first modes (v2.20.0)
/ck:vibe	v2.20.0 — full vibe-coding pipeline end-to-end
/ck:plan	Feature planning, system design, phase docs; HTML/GitHub output + AgentWiki publish (v2.20.0)
/ck:predict	Pre-implementation debate — 5 personas; --chain reason/--chain probe (v2.18.0)
/ck:scenario	Generate edge cases & test scenarios; iterative saturation (v2.18.0)
/ck:cook	Execute implementation — feature dev, plan execution; deep + TDD modes (v2.17.0)
/ck:test	Run unit/integration/e2e tests, coverage analysis, build verification
/ck:code-review	Code review — scope gating, 4 input modes (adversarial stage removed v2.20.0)
/ck:review-pr	v2.20.0 — formal gh PR review (--reply) + anti-ai-slop; duplicate/product gates
/ck:ship	Full ship pipeline: merge main → test → review → commit → push → PR; AgentWiki journals (v2.20.0)
/ck:fix	v2.0.0: Scout → Diagnose → Assess → Implement → Verify
/ck:debug	Systematic root cause analysis — logs, call stacks, multi-layer validation
/ck:loop	Autonomous optimization loop — N iterations against a measurable metric
/ck:security	STRIDE + OWASP security audit; red-team personas iterative discovery (v2.18.0)
/ck:security-scan	Automated secret detection, vulnerability scanning, OWASP patterns

Codebase Understanding & Navigation

Skill	When to Use
/ck:scout	Fast parallel codebase scouting — file discovery, context gathering
/ck:repomix	Pack repo into single AI-friendly file (XML/Markdown/JSON)
/ck:gkg	Semantic go-to-definition, find-usages via GitLab Knowledge Graph
/ck:graphify	v2.17.0 — build queryable knowledge graphs from code/docs/papers via tree-sitter
/ck:tech-graph	v2.18.0 — semantic code analysis: go-to-definition, find-usages, impact analysis

Multi-Agent & Orchestration

Skill	When to Use
<code>/ck:team</code>	v3.0.0: Event-driven hooks, cross-session memory, file ownership globs
<code>/ck:plans-kanban</code>	Plan progress dashboard with timeline visualization
<code>/ck:project-management</code>	Track progress, sync plan statuses, generate reports
<code>/ck:ghpm</code>	v2.20.0 — GitHub project management: issues, Projects boards, handoffs, gh automation
<code>/ck:agentize</code>	v2.17.0 — convert codebases into CLI + MCP server

Frontend & UI

Skill	When to Use
<code>/ck:frontend-design</code>	Replicate mockups/screenshots/videos into polished interfaces
<code>/ck:frontend-development</code>	React/TypeScript components with modern patterns
<code>/ck:ui-styling</code>	shadcn/ui + Tailwind CSS — themes, dark mode, responsive
<code>/ck:ui-ux-pro-max</code>	Design intelligence — 50+ styles, 161 palettes, 57 font pairings
<code>/ck:web-design-guidelines</code>	Audit UI for Web Interface Guidelines compliance
<code>/ck:react-best-practices</code>	React/Next.js performance optimization from Vercel Engineering
<code>/ck:stitch</code>	v2.14.0 — AI design generation via Google Stitch; --project-name isolation (v2.17.0)
<code>/ck:threejs</code>	3D web experiences — WebGL/WebGPU, GLTF, animations, physics, VR/XR
<code>/ck:shader</code>	GLSL fragment shaders — SDF, noise, fBm, generative art
<code>/ck:remotion</code>	Programmatic video creation in React

Backend & API

Skill	When to Use
<code>/ck:backend-development</code>	Node.js/Python/Go APIs (NestJS, FastAPI, Django), auth
<code>/ck:better-auth</code>	TypeScript auth — OAuth, 2FA, passkeys, RBAC, rate limiting
<code>/ck:databases</code>	Schema design, SQL/NoSQL queries, aggregation, indexes, migrations
<code>/ck:payment-integration</code>	Stripe, Polar, SePay, Paddle, Creem.io — checkout, webhooks

Frameworks

Skill	When to Use
<code>/ck:web-frameworks</code>	Next.js App Router, RSC, SSR, ISR, Turborepo monorepos
<code>/ck:tanstack</code>	TanStack Start (full-stack React), Form, AI streaming/chat
<code>/ck:mobile-development</code>	React Native, Flutter, SwiftUI, Kotlin/Jetpack Compose
<code>/ck:shopify</code>	Shopify apps/extensions/themes — GraphQL, Polaris, Liquid
<code>/ck:google-adk-python</code>	Google ADK AI agents — multi-agent, A2A protocol, MCP tools

Infrastructure & DevOps

Skill	When to Use
<code>/ck:deploy</code>	Single skill for all platforms: Vercel, Netlify, Cloudflare, Railway, etc.
<code>/ck:devops</code>	Cloudflare Workers, Docker, GCP, Kubernetes, CI/CD, GitOps
<code>/ck:worktree</code>	Git worktree isolation — <code>--no-prefix/--help</code> (v2.16.0), <code>--base</code> (v2.17.0)

Testing & Browser Automation

Skill	When to Use
<code>/ck:test</code>	Unit/integration/e2e tests, coverage, build verification
<code>/ck:web-testing</code>	Playwright, Vitest, k6 — e2e, load, security, visual, a11y
<code>/ck:chrome-profile</code>	v2.20.0 — real Chrome profile automation (replaces chrome-devtools)
<code>/ck:agent-browser</code>	AI-driven browser sessions, self-verifying workflows (discovery stub v2.18.0)

Documentation & Research

Skill	When to Use
<code>/ck:docs</code>	Analyze codebase and manage project documentation
<code>/ck:docs-seeker</code>	Search library/framework docs via context7/llms.txt
<code>/ck:research</code>	Deep technical research, architecture analysis, requirements
<code>/ck:llms</code>	Generate llms.txt files (LLM-friendly site indexes)
<code>/ck:mintlify</code>	Build docs sites with Mintlify — API docs, developer portals

MCP (Model Context Protocol)

Skill	When to Use
<code>/ck:mcp-builder</code>	Build new MCP servers (FastMCP Python, MCP SDK Node/TS)
<code>/ck:use-mcp</code>	Discover, select, execute MCP tools (absorbed mcp-management v2.18.0)

AI & Media

Skill	When to Use
<code>/ck:ai-multimodal</code>	v2.17.0 — multi-provider image routing; analyze images/audio/video (Gemini)
<code>/ck:ai-artist</code>	Generate images via Nano Banana — 129 curated prompts, style modes
<code>/ck:media-processing</code>	FFmpeg video/audio, ImageMagick images, RMBG background removal
<code>/ck:html-video</code>	v2.20.0 — HTML/CSS/JS templates → local MP4 via nexu-io/html-video

Document Processing

Skill	When to Use
<code>/ck:docx</code>	Create, edit, analyze .docx Word documents with tracked changes
<code>/ck:pdf</code>	Extract text/tables, create, merge, split, fill PDF forms
<code>/ck:pptx</code>	Create, edit, analyze .pptx presentations with layouts
<code>/ck:xlsx</code>	Create, edit spreadsheets with formulas, charts, data analysis

Utilities & Meta

Skill	When to Use
<code>/ck:ask</code>	Technical consultation, architecture questions, design decisions
<code>/ck:preview</code>	Visual explanations — --explain, --diagram, --slides, --ascii, --diff, --html
<code>/ck:show-off</code>	v2.17.0 — stunning self-contained HTML showcase pages for demos
<code>/ck:excalidraw</code>	v2.17.0 — Excalidraw diagrams — architecture, data flow, workflows
<code>/ck:mermaidjs-v11</code>	Mermaid.js v11 diagrams — flowcharts, sequence, ER, Gantt, state
<code>/ck:problem-solving</code>	6 techniques: collision-zone, inversion, meta-pattern, scale-game
<code>/ck:sequential-thinking</code>	Step-by-step analysis with revision capability
<code>/ck:context-engineering</code>	Monitor token usage, optimize context, debug context failures
<code>/ck:coding-level</code>	Set explanation depth (0=beginner to 5=expert)
<code>/ck:find-skills</code>	Discover and install new skills
<code>/ck:skill-creator</code>	Create/update skills with eval-driven iteration

Skill	When to Use
<code>/ck:copywriting</code>	Conversion copy — headlines, email, landing pages, CTAs
<code>/ck:design</code>	55 logo styles, 50 deliverables, 22 banner, 15 icon styles
<code>/ck:markdown-novel-viewer</code>	View markdown with calm, book-like reading experience
<code>/ck:cti-expert</code>	v2.17.0 — cyber threat intelligence / OSINT investigation

Session & Git Management

Skill	When to Use
<code>/ck:git</code>	Conventional commits, auto-split by type/scope, secret scanning; PR merge automation (v2.20.0)
<code>/ck:watzup</code>	Session wrap-up + cross-branch handoff scanner; next-step ranking (v2.20.0)
<code>/ck:journal</code>	Technical journal entries analyzing recent changes
<code>/ck:retro</code>	Data-driven sprint retrospectives from git metrics
<code>/ck:bootstrap</code>	Bootstrap new projects — research, tech stack, design, planning
<code>/ck:xia</code>	v2.17.0 — port/extract/compare features from another GitHub repo
<code>/ck:project-organization</code>	Organize files, directories, standardize project layout

Hook System (v2.20.0)

90+ hook files — event-driven backbone with managed-hooks manifest for deterministic CLI self-heal:

Hooks

Skill	When to Use
<code>scout-block</code>	Restrict access to sensitive directories. Scoped to git root. Windows support.
<code>session-state</code>	Global cross-session persistence with strategic compaction
<code>session-init</code>	Initialize session context on startup/resume/clear/compact
<code>diagnostics</code>	Structured logging with crash wrappers (hook-log.jsonl)
<code>privacy-block</code>	AskUserQuestion flow for sensitive file access (.env, credentials)
<code>dev-rules-reminder</code>	Scoped dedup for development rules on each prompt
<code>usage-quota-cache</code>	Quota monitoring with live countdown display
<code>descriptive-name</code>	Enforce descriptive file naming on Write operations

Skill	When to Use
cook-after-plan	Prompt to run <code>/ck:cook</code> after plan completion
post-edit-simplify	YAGNI/KISS reminder after code edits
plan-format-kanban	Auto-format plan files in kanban style
team-context-inject	Inject team context for subagent sessions
subagent-init	Initialize subagent context

Statusline System (v2.20.0)

Config-driven, modular statusline — overhauled in v2.16.0, quota palette refined v2.20.0:

- **Quota reset countdown** — live timer until API quota refreshes
- **Custom color themes** — configurable `sectionConfig.color` with bright variants
- **User-defined layout** — `statuslineLayout` config in `settings.json`
- **3 display modes** — full, compact, minimal (all respect user `configLines`)
- **Cache-first startup** — instant load from cache, background refresh
- **Native Claude gate** — quota display gated to native Claude subscriptions

Breaking Changes (v2.14.0 → v2.20.0)

1. **Namespace enforcement** — all skills require `ck:` prefix
2. **Command renames** — `debug` → `ck-debug`, `plan` → `ck-plan`
3. **chrome-devtools removed** (v2.18.0) — replaced by `/ck:chrome-profile` (v2.20.0)
4. **mcp-management absorbed** (v2.18.0) — capabilities folded into `/ck:use-mcp`
5. **kanban + template-skill deleted** (v2.18.0) — `plans-kanban` remains
6. **Bash hook runner dropped** (v2.20.0) — canonical hook wiring overhaul
7. **Human MCP disabled by default** (v2.20.0)
8. **useGemini default** — changed to `false` (requires explicit opt-in with pre-flight validation)
9. **Session-state storage** — moved to global storage (no longer in project directory)