

AGENTS.md × Claude Code Interop Handbook

One instruction file across Claude Code, Codex, Copilot,
Cursor, Gemini CLI, and more.

Edition: 1.0 (2026-06)

Scope: the #6235 AGENTS.md interop gap — 9-tool matrix, six paths, copy-paste templates

Tool conventions verified: 2026-06-02 against each tool's official docs

Author: yurukusa (github.com/yurukusa/cc-safe-setup)

Independent operator's handbook. Not affiliated with Anthropic, OpenAI, GitHub, Google,
Sourcegraph, or any tool vendor. Templates are free to use and adapt.

AGENTS.md × Claude Code Interop Handbook

One instruction file across Claude Code, Codex, Copilot, Cursor, Gemini CLI, and more

Practical patterns, a verified tool-by-tool matrix, and copy-paste templates for the single largest open feature request in Claude Code.

Who this is for

You run Claude Code, and you also touch at least one other AI coding tool — Codex, GitHub Copilot, Cursor, Gemini CLI, Windsurf, Cline, Amp, or Aider. You have written the same project rules twice: once in `CLAUDE.md`, once in `AGENTS.md` (or `.cursorrules`, or a config file). The two copies drift. One tool follows a rule the other has never seen.

This handbook removes the duplication. It gives you one source of truth that every tool reads, the exact setup for each tool, and a rollback-safe way to migrate an existing `CLAUDE.md`.

It is written for three situations:

- **Solo** — you alternate between Claude Code and one or two other tools on your own machine.
- **Mixed team** — some teammates use Claude Code, others use Cursor or Copilot, in the same repo.
- **Consultant / multi-repo** — you rotate across client codebases and hit this friction every time.

What's inside

1. **The gap** — what `#6235` is, why it is the largest open request in the repo, and why every other tool converged on `AGENTS.md`.
2. **Pick your path** — a three-question decision for which setup fits you.
3. **The tool matrix** — every major tool, the exact file it reads, and whether it needs setup. Verified against each tool's official docs (2026-06-02).
4. **The paths** — the official `@AGENTS.md` import, symlink, pre-commit sync, SessionStart hook, direnv, and CI drift check — with the trade-off of each.
5. **Templates** — copy-paste setup for each tool, ready to drop into a repo.
6. **Migration** — a rollback-safe runbook to move an existing `CLAUDE.md` to `AGENTS.md` without losing Claude-only behavior.
7. **Team drift guard** — pre-commit and CI checks that make divergence loud instead of silent.

8. **Official status** — where Anthropic stands, and what to watch.

How to use it

If you want the fastest correct setup, read **Pick your path** (one page), find your row in **The tool matrix**, and copy the matching block from **Templates**. Ten minutes.

If you are migrating an existing project or setting up a team standard, read it front to back once. The decisions compound: the path you pick changes which template you use and which drift guard you need.

A note on accuracy

Tool conventions change. Every tool claim in this handbook was checked against the tool's own documentation on **2026-06-02** and is dated where it matters. Before you rely on any single line, confirm it against the current docs — the matrix tells you exactly which file and setting to check. Where a tool's behavior differs by surface (for example, an agent mode versus inline completion), the handbook says so rather than flattening it.

This is an independent operator's handbook, MIT-spirited in its templates. It is not affiliated with Anthropic, OpenAI, GitHub, Google, or any tool vendor.

1. The gap, in numbers

`anthropics/claude-code` issue **#6235** asks for one thing: let Claude Code read `AGENTS.md`, alongside or instead of `CLAUDE.md`. It was opened in **August 2025**. It gathered **6,300+ reactions** — the largest feature request in the entire repository, and it was closed as completed on **2026-08-17** without changing the behaviour described below (see Chapter 8). The second-place issue (Pro/Max quota inflation, #16157) sits around 717 reactions. The gap is roughly **7×**.

About six months on, with no staff resolution, the community opened a follow-up, **#31005** — its title spells out the frustration: "the community has been asking since August 2025." It has gathered another **~240 reactions** on its own. More than **170 other issues** cross-reference #6235 — each one an operator describing the same friction from a slightly different angle.

The request is technically narrow. It is not a new feature, a model change, or a pricing ask. It is a file-reading convention. That is what makes the size of the signal remarkable: 6,300 reactions for a one-line behavior.

Why every other tool already agrees

The convergence on `AGENTS.md` was not coordinated. Several tool teams independently picked the same filename, for the same three reasons:

- **The name is generic.** `AGENTS.md`, not `CURSOR.md` or `CODEX.md`. Nothing about it ties to a single vendor, so adopting it costs no pride.
- **The format is plain Markdown.** No custom DSL, no schema, no front-matter requirement. Any tool can parse it; any human can read it.
- **The location is the repo root.** Not a hidden directory, not a dot-folder. It sits next to `README.md` where both humans and agents look first.

Once a tool team decides to interoperate at all, `AGENTS.md` is the obvious choice. Codex, Amp, GitHub Copilot's coding agent, and others read it directly today; Gemini CLI and Aider read it with one line of config. Claude Code is the notable holdout — it reads `CLAUDE.md` and does not read `AGENTS.md` on its own.

What the gap actually costs you

The cost is recurring duplication, and it scales with how you work:

- **Solo, two tools.** You edit `AGENTS.md` for the afternoon's Codex session, forget to mirror the change into `CLAUDE.md`, and the next morning Claude Code is working from stale rules. The failure is silent — nothing errors, the agent just behaves on yesterday's instructions.

- **Mixed team.** Half the team edits `CLAUDE.md` , half edits `AGENTS.md` . Every repo now carries two files that must be kept identical by social convention. Without a guard, they diverge within weeks, and code review starts catching "but my tool said to do it this way" disputes.
- **Consultant / multi-repo.** You hit the setup tax on every new codebase. Ten clients, ten times configuring the same sync.

None of these is catastrophic on any single day. That is exactly why it persists: the cost is a slow tax, invisible per-incident, expensive in aggregate. This handbook converts that tax into a one-time setup.

2. Pick your path

Three questions decide your setup. Answer them before you touch a file.

Q1 — Do you need any Claude-Code-only instructions?

Some rules apply to every tool ("use 2-space indent", "never touch `migrations/` without asking"). A few might be Claude-Code-specific (a note about a particular hook, a `/command` workflow, a sub-agent instruction).

- **No, everything is shared** → you can use the lightest setup (a symlink, or the native read). One file, nothing tool-specific.
- **Yes, a few Claude-only lines** → use the `@AGENTS.md import` path. It lets `CLAUDE.md` pull in the shared file and hold a short Claude-only section underneath.

Q2 — Are you on Windows (without WSL)?

Symlinks on Windows require Administrator rights or Developer Mode. If that is a problem in your environment:

- **Windows, no symlink privilege** → prefer the `@AGENTS.md import` (one real file, no symlink needed) or keep real copies plus a drift guard.
- **macOS / Linux / WSL** → symlinks work normally; all paths are open to you.

Q3 — Solo or a team sharing the repo?

- **Solo** → a symlink or import is enough. You control every clone.
- **Team** → symlinks do not always survive every clone or every tool's file-writing, so add the **team drift guard** (Chapter 7). It makes divergence fail a commit instead of leaking into the codebase.

The short answer

For most people, the recommended default is the `@AGENTS.md import`: it is the method documented in Claude Code's own memory docs, it is one real file (so it cannot drift), and on Windows it avoids the symlink-privilege problem entirely. Put your shared rules in `AGENTS.md`, point `CLAUDE.md` at it with a single line, and add any Claude-only notes beneath that line.

The rest of the handbook gives you the exact setup for every tool you run, the trade-offs of each alternative path, and the guards that keep a team honest.

This is the free sample

You have reached the end of the free sample of **Claude Code AGENTS.md Interop Handbook**. The full edition is 21 pages.

The remaining sections cover the six sync paths ranked by trade-off, copy-paste templates for every tool plus a complete repo layout, a rollback-safe migration runbook, and a team drift guard that makes divergence fail loudly.

The hooks referenced throughout are MIT-licensed and free at github.com/yurukusa/cc-safe-setup.