

# The Two-Agent Bridge Kit

GenAI Unplugged | 2026-06-11

## The Two-Agent Bridge Kit

This is the exact setup from the One Shot Show "Claude Code vs Codex" episode. It lets two AI coding agents, Claude Code and OpenAI Codex, work the same project and hand work back and forth without stepping on each other.

The whole idea is one shared file: `CHANGES.log`. Both agents read it before they start and append to it when they finish. That single file is the memory they share.

### What's in here

```
bridge-kit/           ← drop this into
any project
  CLAUDE.md           ← project brain +
the handoff rule (Claude reads this)
  AGENTS.md           ← identical copy
(Codex reads this)
  CHANGES.log        ← the shared
bridge, with a template + examples

compound-calculator-demo/ ← the finished
build from the show
  index.html, app.js, styles.css
  CLAUDE.md, AGENTS.md, CHANGES.log
```

### How to use the bridge kit (any project)

1. Copy `bridge-kit/CLAUDE.md` into your project. In Claude Code, this usually lives at `.claude/CLAUDE.md` (or `CLAUDE.md` at the root). Fill in your project name, files, and conventions. Keep the "Working with a second agent" section as-is.

1. Copy it to where Codex reads:

```
...
cp .claude/CLAUDE.md AGENTS.md
...
```

Now both agents share the same brain. Each reads its own file, the content is identical.

1. Start `CHANGES.log` (the template is included). Delete the example entries once you have real ones.
1. Work normally. Ask Claude for one task. When it finishes, it logs to `CHANGES.log`. Open the same folder in the Codex app (Codex will offer to import your Claude setup into `AGENTS.md` automatically), ask Codex for the next task, and it reads the log before acting. Hand it back to Claude the same way.

### The rules that make it reliable

The "Claude/Codex Handoff Bridge" section in `CLAUDE.md` / `AGENTS.md` says:

- Before any task: read the latest `CHANGES.log` entry.
- After EVERY task that touches a file (including a single-line CSS fix): append a `CHANGES.log` entry before ending your turn.
- A one-line fix still requires a log entry. No exceptions, no matter how small.
- Treat writing to the log as the final step of every task, like running tests or making a commit.

Each entry carries: timestamp and agent, a short task name, status, what is done, files modified, remaining work, and the exact next-agent pickup instruction.

Even with this, agents are not perfect. For a hard guarantee, back the rule with a hook so the log entry is enforced, not just requested.

### The demo

`compound-calculator-demo/` is the finished compound-interest calculator built live on the show. These are the real files from the broadcast. It was made in three passes: Claude added the year-by-year table, Codex replaced it with a live SVG growth chart and a responsive

card UI, then Claude rebranded it to the GenAI Unplugged colors (teal, yellow, cream) and added the footer. Read its `CHANGES.log` to see the exact round trip the two agents logged. Open `index.html` in a browser to run it.

---

Built on the One Shot Show by Dheeraj (GenAI Unplugged) + Wyndo (The AI Maker). Questions? DM me on Substack.