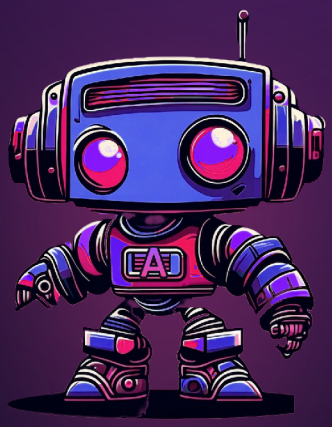
TOPIC 4 OF 6

Connecting Aperio to your world

Give Claude, Codex, or another AI tool you already use the same memory Aperio built for you.



The same memory that answers you inside Aperio can answer you inside Claude or Codex too.

WHAT'S INSIDE

PART 1

Connect Claude Desktop

> **After this page, Claude Desktop can see and use everything Aperio remembers about you.**

Getting started proved Aperio remembers things in its own chat. This page carries that same memory into an AI tool you already use. Once connected, ask Claude Desktop things it never learned from you directly:

- *"What's my Wi-Fi password?"* — the one you stored in Aperio, not Claude

1. Find your Aperio folder's exact path: run `pwd` in a terminal there (macOS/Linux), or check it in Explorer (Windows) — then add `mcp/index.js`.

2. In the system menu bar, open **Claude** → **Settings** → **Developer** → **Edit Config**, which opens or creates `claude_desktop_config.json`.

3. Paste this in, replacing the path with the one from step 1, and save the file:

```
{
  "mcpServers": {
    "aperio": {
      "command": "node",
      "args": ["/path/to/aperio/mcp/index.js"]
    }
  }
}
```

4. Quit Claude Desktop completely and reopen it — a running window doesn't pick up the change, only a fresh start does.

DONE LOOKS LIKE

Claude Desktop's tools list shows "aperio" as a connected server.

! **Aperio doesn't show up as connected.** Check the path points at `mcp/index.js` itself, not just the folder, and that you fully quit and reopened Claude Desktop, not just closed the window. An error mentioning `node` instead means Node.js isn't installed yet — run Aperio-lite's **START** once to fix that.

NEXT Connect from a terminal

PART 2

Connect from a terminal

> **After this page, Claude Code or Codex can reach the same memory too — one command, no file to edit by hand.**

If you already write code with Claude Code or Codex, you can hand them Aperio's memory the same way — so a coding session can, for example, recall a detail you only ever told Aperio's chat:

- *"What did I say about the staging database password?"*
- *"Remember that this project's release always ships on a Friday."*

1. Open a terminal.
2. Run the command for your tool, using the same `mcp/index.js` path from Part 1:

```
# Claude Code
claude mcp add aperio -- node /path/to/aperio/mcp/index.js

# Codex
codex mcp add aperio -- node /path/to/aperio/mcp/index.js
```

3. Start a new conversation in that tool — the connection applies from the next session on.

DONE LOOKS LIKE

Running `claude mcp list` (or `codex mcp list`) shows "aperio" in the list.

! **The `mcp` command isn't recognized.** Update Claude Code or Codex to its latest version — MCP support needs a reasonably recent release.

NEXT Prove the connection

PART 3

Prove the connection

> **After this page, you'll have real proof the same memory now follows you outside Aperio's own chat.**

This mirrors the proof from *Getting started* — only this time, ask somewhere new. Don't open Aperio's web app for this part at all.

1. Open Claude Desktop, Claude Code, or Codex — whichever you connected.
2. Start a new conversation there.
3. Ask the same question you proved in *Getting started*. Stored the Wi-Fi password? Ask: *"What's my Wi-Fi password?"* Stored the vet's number instead? Ask: *"What's the vet's phone number?"*

DONE LOOKS LIKE

The answer matches what you stored in Aperio — coming from a completely different tool, without you retyping it there.

! **It says it doesn't have a way to check.** Ask it to use the Aperio tool explicitly the first couple of times — for example, *"Use Aperio to recall my Wi-Fi password."* Most tools start calling it on their own after that.

! **It doesn't know, or answers something else entirely.** Double-check the path in your config points at the very same Aperio folder you've been using — a different copy of Aperio keeps its own separate, empty memory.

You've now connected Aperio's memory to Claude Desktop, Claude Code, or Codex, and proven the same memory follows you into all of them. Want to run Aperio from source, on a server, or with a cloud AI provider instead of the ready-made download? That's covered in the *Setup & configuration* topic.