AI Self-Improvement

An opt-in capture loop that lets Claude learn from your engineering sessions over time, with everything kept on your own machine.

AI Integration AI Self-Improvement

When you work with Claude through the Designer IDE or in Claude Code, you correct it, you point out invariants ("never do X"), you push back on wrong hypotheses, and occasionally you call out a particularly good catch. Each of those moments contains a tiny piece of know-how about your project, your platform, and your team.

AI Self-Improvement is an **opt-in capture loop** that turns those moments into a local knowledge base your future sessions can draw from. Default state is **off** — you opt in once per machine. Markers stay on your machine. Nothing is uploaded. Nothing is shared.

What It Does
When To Enable It
How To Use It
What Gets Captured
What Does NOT Get Captured
Where Data Lives
Privacy Posture
Frequently Asked
Related

What It Does

When enabled, Claude silently appends a marker to a local file every time one of seven specific signals fires — a correction you made, a surprise from a tool call, a stated invariant, a frustration, an explicit good catch, a repeated edit on the same object, or a closing-turn rule that crystallized after a wrong path.

Future sessions see a brief summary of recent learnings as part of their startup context, so the same wrong path is less likely to be retried tomorrow.

When To Enable It

Enable it on machines where you do recurring engineering work in the same project. The value compounds: a session that has a week of prior corrections to draw from is meaningfully more grounded than a session starting cold.

Skip it on machines you use only occasionally, or when working on highly confidential projects where you would prefer no local notes at all.

How To Use It

Turn it on

```
dotnet ai-selfimprove.dll enable
```

This creates `~/.claude/learnings/` and flips the flag for your machine. Claude in Designer and Claude Code sessions will start capturing markers from the next session onward.

Use Claude normally

Nothing changes in your workflow. Markers are dropped silently in the background; you will not see them in chat, and nothing is added to your code, your solution, or your project.

Automatic loop

Once enabled, every new session you start in Designer or via Claude Code automatically includes a brief, bounded summary of recent learnings as part of its startup context. You do not need to run consolidation manually for this to work — the platform calls it for you on each session start, and inlines a capped summary so the bootstrap stays lean.

The summary is presented to Claude as **soft context** — prior-session interpretations to consult, not invariants to obey. Your explicit *always X / never Y* statements carry more weight than first-pass guesses.

Review what was captured

From a terminal:

```
dotnet ai-selfimprove.dll consolidate
```

Writes a consolidated report to `~/.claude/learnings/report-<date>.md` — markers grouped by source surface (Designer vs Claude Code) and by trigger type, with the verbatim turn excerpts that fired each marker. The terminal path also archives the consumed marker files so the next consolidation only processes new content.

Turn it off

```
dotnet ai-selfimprove.dll disable
```

The flag flips off. Existing marker files and reports stay where they are — they are yours. Delete the `~/.claude/learnings/` directory yourself if you want a clean slate.

What Gets Captured

Seven specific triggers (full definitions in the platform's capture contract):

Trigger	Fires when
`correction`	You said "no", "actually", "that's wrong", or undid something.
`surprise`	A tool call returned something unexpected, or a hypothesis was falsified.
`invariant_stated`	You or Claude said "always X" or "never Y" about the system or workflow.
`repeated_object_editing`	An object was created and immediately re-edited (Designer only).
`user_frustration`	You said Claude was being slow, wrong, verbose, or missing context.
`positive_highlight`	You explicitly called out a good insight ("that's exactly right", "write that down").
`resolution_takeaway`	A wrong-then-right arc closed with a generalizable one-sentence rule.

Routine successful task completion produces zero markers. A session where nothing surprising happens leaves no trace.

What Does NOT Get Captured

- Routine task completions.
- Pleasantries or general thanks without explicit insight acknowledgment.
- Tool call results that matched expectations.
- Anything containing secrets, credentials, or customer-identifying solution data (tag names, plant names, IP addresses).
- Patterns seen only once that do not fit a trigger.

The capture contract enforces these exclusions on every marker before it is written.

Where Data Lives

Path	Contents
`~/.claude/learnings/ai-session-notes-YYYY-MM-DD.jsonl`	Raw markers, one per line, one file per calendar day.
`~/.claude/learnings/report-YYYY-MM-DD.md`	Consolidated reports.
`~/.claude/learnings/archive/`	Marker files already consumed by a consolidation run.
`MachineSettings/AIAssist/SelfImprovement.settings.json`	The on/off flag for this machine.

Everything is local. Sharing a report with a colleague is a separate, explicit action you take (copy/paste, attach, send). The platform never uploads.

Privacy Posture

- Default off out of the box. No capture happens until you explicitly enable.
- Single-machine. There is no cross-machine aggregation, no telemetry, no phone-home.
- The flag is read at session start; no mid-session toggling.
- The capture contract has an explicit "what NOT to capture" list that includes secrets and customer-identifying data.
- Sharing is always your call. Consolidation prints a privacy reminder.

Frequently Asked

Does this slow down my Claude sessions?

No. Capture is a local file append. Auto-consolidation on session start is a small text-processing pass with no network calls; the bootstrap latency hit is well under a tenth of a second on a typical week of markers.

What happens if I never run consolidate manually?

Auto-consolidation on session start runs it for you. The marker file accumulates during a calendar day; the next session that starts after that day picks it up automatically.

Can I edit a marker file by hand?

You own the files. Editing them is supported but not necessary. The consolidator ignores unknown fields, so adding metadata to existing entries does not break anything.

Will this affect my project files?

No. Capture writes only to `~/.claude/learnings/`. Your solution, your scripts, your displays, your tags — none of them are touched.

Can I see what triggered a specific marker?

Each marker carries the verbatim turn excerpt that fired it, so the report shows exactly what you and Claude said. Nothing is paraphrased at capture time.

Related

- AI Integration — parent page; full FrameworX AI capability set.
- The capture contract (full trigger definitions and marker format) ships with the platform under `FactoryStudio/bin/ai-self-improvement.md`.

In this section...
