

Library entry prompt — paste this into any thread

Prepared by Phase 4 ViewMan — LibraryEntryPrompt · 2026-06-20 · SHADS CW channel

Produce a single short Markdown entry summarising this thread, for inclusion in a library that another AI will consume to understand my chat history.

Output

- If you have file-creation tools: save a .md file to /mnt/user-data/outputs/ named LibraryEntry_<thread-title>.md (Or LibraryEntry_<YYYY-MM-DD>_<short-topic>.md if you don't yet know the thread title). Present the file.
- If you have no file-creation tools: produce the markdown inline in chat in a single fenced code block, and I'll save it myself.

No commentary outside the markdown. Belt and braces, then stop. Only ping me in chat if you cannot execute.

Length

Succinct. Aim for ~30–50 lines total. A compact card, not an audit trail. The consuming AI needs the gist, not the transcript.

Schema (use these exact section headings)

- # Thread: [chat title]
- **Project**: — Claude.ai project name
- **Started**: — date, UTC time if knowable
- **Last active**: — most recent activity date
- **Assistant name/role**: — Claude + any assigned instance name or role from the thread
- ## Scope — 1–3 sentence summary of what the thread covered
- ## Inputs (received) — bullet list of files / handover documents received, with date and 1-line summary each; or "None"
- ## Artefacts produced — bullet list with filename, date/timestamp, 1-line description; or "None"
- ## Handover — Inbound (yes/no + 1-line description), Outbound (yes/no + 1-line description), Continuity (status: closed / parked / awaiting next step / open)
- ## Verification status — short paragraph noting what's filesystem-verified, what's supplied-by-Paul, what's inferred

Fields you very likely cannot find yourself

The Claude.ai runtime does not expose these to the assistant:

- The chat / thread title shown to me in the UI breadcrumb
- The Claude.ai project name (also in the breadcrumb)
- The exact thread start date/time and per-message timestamps

If you have a recent_chats tool, try it once. The current chat will be excluded from the result, but other chats in the project will appear with their titles — that confirms the limitation. Then either ask me to supply the breadcrumb values directly, or **ask me to export the chat to a text file and upload it — the export carries the chat title at the top, the per-message timestamps, and the speaker labels, all of which**

you can read straight from disk. If I have not supplied them, mark each missing field "Not exposed to assistant at runtime — Paul to supply" and continue.

If you have file-creation and filesystem tools, verify everything you can directly: `ls -la` for paths/sizes/timestamps, `stat` for ISO timestamps, file contents for descriptions.

Do not assume or smooth

Tag every value with one of:

- **Verified — confirmed via filesystem, tool output, or system context**
- **Supplied-by-Paul — value I gave you in chat**
- **Inferred — best estimate from indirect evidence (e.g., earliest filesystem activity for thread start)**
- **Unknown — runtime does not expose it, and I have not supplied it**

If you find no artefacts and no handover, say so plainly. The library entry still needs to exist for that thread — silence creates gaps.

Stop instruction

Once the file is produced and presented (or the markdown is delivered in chat), stop. Do not add chat commentary, recap, or follow-up questions unless you cannot execute the task. Accuracy and completeness inside the entry; nothing outside it.