

SECTION A-A

BOX 03-05

Session log protocol

How I close every working session with an AI agent

REV A · ISSUED

TITLE	SHEET	REV	DRWN	DATE
Session log protocol	03-05	A	D.KRAVCHUK	2026-10-03

How I close every working session with an AI agent

01 /

The problem it solves

A chat with an AI agent has no memory worth the name. Close the window and the next session starts from a blank line. You re-explain the project, the agent re-discovers the same constraints, and the decision you made on Tuesday gets remade on Thursday, sometimes the other way.

I run several tracks at once: a product, a job search, a lab agent, media. Each has its own chat project and its own folder. Without a log the tracks bleed into each other and the agent in one of them has no idea what happened in the other three.

The fix is not a bigger context window. It is one database row per session, written at the end, read at the start. The row is short. The ritual is strict. The agent does the typing.

This guide is the protocol as it runs today, with the Notion schema and the prompts. It assumes the agent can read and write your Notion through a connector (MCP). If it cannot, the same row works as a markdown table in a file; the fields do not change.

02 /

The two rituals

START	CLOSE
read last 3 rows for this area	one row in Sessions
read last 7 digests, all areas	Digest ≤ 400 chars
answer in one line:	## Next in the body
date state → Next: 1-2 steps	## Bugs / Issues in the body
	a Decisions row, if one was made

Everything else is detail. If you take only this page, you have the protocol.

03 /

Start ritual

The first thing the agent does, before reading any file, is read the log. Two queries:

- The last three rows of Sessions filtered to the area of this chat.** Three, not ten. Three rows is a week of work and fits in the agent's head without crowding out the task.
- The last seven Digest values across all areas.** This is the cross-track awareness. The media chat learns that payments shipped. The product chat learns that a client call moved. Seven is roughly a week of sessions in my rhythm; tune it to yours.

Then the agent answers in exactly one line:

[2026-10-03] | vault PDFs built, build green → Next: update entries, run build

Date, current state in a few words, arrow, the next one or two steps taken from the `Next` field of the last row. No summary of what it read. No "I've reviewed your recent sessions". If the one-liner is wrong, I correct it in the first message and the session starts from the correction.

Why one line: it proves the agent read the log, and it costs nothing to check. A paragraph would hide a wrong read.

The query, in words

Both queries are "select these columns from Sessions, sort by date descending, take N". With the Notion connector that means selecting the columns you need and sorting client-side; the SQL layer behind the connector rejects `ORDER BY` on text columns and rejects `LIKE`. Ask for `Session`, `Date`, `Area`, `Next`, `Digest`, and let the agent sort.

04 /

Close ritual

The trigger is a word. I say "that's all", or "closing", or "we're done". The agent proposes the row, I say "ok", it writes. It does not write without the "ok". That one rule is what keeps the log clean: I read every row once before it exists.

The row goes into one database, `Sessions`. Not a page, not a per-project database. One base, filtered views per area.

The row

FIELD	TYPE	WHAT GOES IN
Session	Title	YYYY-MM-DD, Area: topic . Date first so the title sorts.
Date	Date	Today. A range if the session spanned two days.
Area	Relation to Areas	Required. The track this session belongs to. Never guessed.
Project	Relation to Projects	If the session was inside a project. The area can be derived from it.
Type	Select	Architecture, Research, Feature, Data, Bugfix, Strategy, Marketing, Life.
Status	Select	Done, or In Progress if the session was cut short.
Summary	Text	Two to four sentences. What was worked on.
Key Outcome	Text	One sentence. What exists now that did not this morning.
Decisions	Text	Decisions made in this session, one line each. Even small ones.
Next	Text	One to three concrete steps. This is what the next session's one-liner is built from.
Digest	Text	400 characters or fewer. Mandatory. See below.

The body

Two headings, always, even if one is empty:

```
## Next
- Update the two vault entries with file and page count
- Run the build; if it fails on another agent's files, report, do not fix

## Bugs / Issues
- Google Fonts did not load in the headless print on the first run; retry with network idle
```

Next repeats the Next field with more room. ## Bugs / Issues is the one place a half-fixed thing is allowed to live overnight. If it is still there after three sessions it becomes a task.

The Digest

The Digest is the field that makes the start ritual work. It is read by every other chat, so it has to stand alone.

Rules:

- **400 characters or fewer.** Hard limit. The agent counts.
- **Written for a stranger.** The media chat will read the product chat's digest. No pronouns without a noun, no "the bug" without saying which.
- **Past tense, concrete nouns, one or two numbers if there were any.** "Built two vault PDFs (7 and 8 pages), updated entries, build green. Next: home page copy." Not "worked on the vault".
- **Decisions are named in it.** If the session decided something, the digest says so in a clause. The reader should not need to open the row.
- **Not a copy of Summary.** Summary is for the row's own area. Digest is for everyone else.

A digest that fails: "Productive session on the site. Fixed several issues and made good progress on the guides. Next steps planned." Zero nouns. Nobody can act on it.

05 /

Decisions log

Some decisions outlive the session. The architecture choice between a CLI subprocess and an SDK. The rule that a certain kind of number never appears in public copy. The pricing model. Those get their own row in a second database, Decisions , linked back to the session where they were made.

The test for whether a decision earns a row: **will I need to explain it in a year?** If yes, row. If it is "used the blue one", it stays in the session's Decisions text field.

FIELD	TYPE	WHAT GOES IN
Decision	Title	The decision as a statement. "Agent runs as a CLI subprocess, not through the SDK."
Date	Date	When it was made.

FIELD	TYPE	WHAT GOES IN
Category	Select	Architecture, Hardware, Process, Product, Stack.
Impact	Select	High, Medium, Low. How much would have to change to reverse it.
Status	Select	Accepted, Under evaluation, Planned.
Why	Text	The reason, in the words of that day. This is the field you will read in a year.
Alternatives	Text	What else was on the table and why it lost.
Area	Relation to Areas	Same area as the session.
Session	Relation to Sessions	The session it came from. Two-way, so the session shows its decisions.
Notes	Text	Anything that does not fit above.

The **Why** field is the whole point. A decision without a reason is a rule nobody dares to break and nobody can defend. With the reason written down, the day the reason stops being true is the day you reverse it, cleanly.

06 /

Notion schema to copy

Two databases, two relations into the rest of the workspace. Build them in this order.

Sessions

PROPERTY	TYPE	OPTIONS / TARGET
Session	Title	
Date	Date	
Area	Relation	Areas database, two-way, show as "Sessions" on Areas
Project	Relation	Projects database, two-way, show as "Sessions" on Projects
Type	Select	Architecture · Research · Feature · Data · Bugfix · Strategy · Marketing · Life
Status	Select	Done · In Progress
Summary	Text	
Key Outcome	Text	
Decisions	Text	
Next	Text	
Digest	Text	agent enforces 400 chars

PROPERTY	TYPE	OPTIONS / TARGET
Decision Records	Relation	Decisions database, two-way (created from the Decisions side)

Decisions

PROPERTY	TYPE	OPTIONS / TARGET
Decision	Title	
Date	Date	
Category	Select	Architecture · Hardware · Process · Product · Stack
Impact	Select	High · Medium · Low
Status	Select	Accepted · Under evaluation · Planned
Why	Text	
Alternatives	Text	
Notes	Text	
Area	Relation	Areas database, two-way, show as "Decisions" on Areas
Session	Relation	Sessions database, two-way, show as "Decision Records" on Sessions

Views, not databases. Each area that needs its own log gets a linked view of Sessions filtered by `Area` , placed on that area's page. Do not create a second Sessions database for a second project. The cross-area digest query only works because there is one table to query.

No page template on Sessions. The agent writes the two body headings itself. A template would tempt someone to create rows by hand and skip the Digest.

If you do not use Notion: the same two tables as markdown files, one row per line, `Digest` as the last column. The start ritual becomes "read the last 3 lines matching this area and the last 7 lines overall".

07 /

Prompts to paste

These are the exact instructions I keep in the project-level instructions of each chat. Replace the words in angle brackets.

Start ritual (goes in the project instructions, runs on every first message):

```
At the start of every session, before anything else:
1. Read the last 3 rows of the Sessions database where Area = <this
   area>. Columns: Session, Date, Next, Digest. Sort by Date
   descending yourself; do not use ORDER BY.
2. Read the last 7 Digest values across all areas, same sort.
3. Reply with exactly one line and nothing else:
   [YYYY-MM-DD] | <state in a few words> → Next: <1-2 steps from
   the last row's Next field>
```

Do not summarise what you read. Do not greet. Then wait for my message.

Close ritual (triggered by "that's all" / "closing" / "we're done"):

When I say "that's all", "closing" or "we're done":

- Propose one row for the Sessions database. Show me every field before writing:
 - Session: "<YYYY-MM-DD>, <Area>: <topic>"
 - Date: today
 - Area: <this area> (required; if a Project is set, derive Area from it, never guess it)
 - Project: if we worked inside one
 - Type: one of Architecture, Research, Feature, Data, Bugfix, Strategy, Marketing, Life
 - Status: Done, or In Progress if we stopped mid-task
 - Summary: 2-4 sentences
 - Key Outcome: 1 sentence, what exists now that did not before
 - Decisions: one line each, including small ones
 - Next: 1-3 concrete steps
 - Digest: 400 characters or fewer, written for someone in a different project who has not read this chat. Past tense, concrete nouns, numbers if any, decisions named.

Body: "## Next" (the steps, as a list) and "## Bugs / Issues" (anything half-fixed; "none" if empty).
- If any decision made today would need explaining in a year, propose a row for the Decisions database too: Decision, Date, Category, Impact, Status, Why, Alternatives, Area, Session (link to the row from step 1).
- Wait for "ok". Write nothing before it. After writing, reply with the row title and the Digest, nothing else.

Mid-session decision (when I say "that's a decision"):

When I say "that's a decision" or "log this decision": propose a Decisions row with Decision, Category, Impact, Why, Alternatives. Leave Session empty for now; at close, link it to the session row. Wait for "ok".

Digest check (I run this when a digest reads vague):

Rewrite this Digest so that someone who has never seen this chat can act on it. Keep it under 400 characters. Replace every pronoun without a noun, every "the bug", "the issue", "the feature" with the thing itself. Name any decision in a clause. Past tense.

08 /

Common failure modes

I have hit each of these. The fix is in the protocol above; this is the list of what happens when a piece of it is skipped.

The agent writes the row without asking. Then the row is written in the agent's words, and the agent's words drift toward "productive session, good progress". Propose, confirm, write. Every time.

Area is inferred from the title. The agent reads "fixed the Stripe bug" and files it under the product area, except the session was about a client's Stripe setup, a different area entirely. Area is set from the chat's project instructions or derived from a linked Project. Never from the content.

Digest is a copy of Summary. Summary is written for the same area and can use shorthand. Digest is read cold by other chats. When they are the same text, the other chats get the shorthand and learn nothing.

Digest over 400 characters. It happens when the session was big. The limit is what forces a choice about what mattered. If it does not fit, the session had two topics and should have been two rows.

A second log database. "This project is special, it gets its own log." Three months later there are four logs, the cross-area query reads one of them, and the other three are invisible. One database, filtered views.

No Next . The next session's one-liner has nothing to build on, so the agent writes "ready to continue" and the session starts cold. `Next` is the most important field after `Digest` . Three concrete steps, written in the last five minutes while they are still obvious.

Start ritual skipped because "I remember". You remember the shape. You do not remember that Thursday's row said "do not retry writes". The read takes seconds. The re-discovery takes an hour.

Bugs / Issues as a dumping ground. Six sessions later it has twenty lines. Rule: a line that survives three sessions becomes a task in the task database and leaves the body.

Decisions row without Why . A year later it reads as a prohibition. "No SDK, use the subprocess." Nobody remembers that the reason was a specific limitation on a specific date, so nobody checks whether it still holds. `Why` is mandatory; `Alternatives` is what makes `Why` honest.

The log lives in the chat, not the database. "Here's a summary of what we did today" at the bottom of a chat is lost the moment the chat is. The row exists so that a different chat, a different model, a different week can read it.

Closing word never said. The session ends because the laptop closed. Nothing was written. This is the most common failure and the only cure is habit: the closing word is the last thing I type, and I type it before I stand up.

09 /

What it costs

About two minutes per session: thirty seconds at the start, ninety at the close, most of it reading the proposed row. The agent does the typing.

What it buys is a next session that starts with "[date] | state → Next: step" instead of "what were we doing?". Over a week across four tracks that is the difference between one person running four things and four things running one person.

END OF SHEET · DIMAK.APP/VAULT