
ELLM

How to Show the Audit Log and Debug Trace

Learn how to access and interpret audit logs and debug traces to track assistant activity and diagnose issues. This guide provides step-by-step instructions for viewing logs and traces stored in JSON format within your project folder.

AUTHOR

Paul Flanders

PUBLISHED

10 Sep 2026

DOCUMENT

InfoHub — Knowledge, news, and updates from
EssingtonITS

How to Show the Audit Log and Debug Trace

Paul Flanders · 10 Sep 2026

Estimated time: 3 minutes

Difficulty: Intermediate

Why you'd use this

In a regulated environment you need to show what the assistant was asked, what it read, what it changed and what it ran. The audit log is an append-only local record of exactly that, one line per event, written for every session. The debug trace is a deeper, optional record for diagnosing behaviour: the full prompt, each raw model reply, every parse and apply decision, and the reason behind every pause. Both stay inside the project folder.

Before you start

Permissions required:

- None to view. The files are plain JSON lines under `.project-ai/`.

You'll need:

- For the debug trace: **Debug logging** ticked in  before reproducing the behaviour you want to examine.

Steps

1. Press `/` and choose **Show audit log**, or run **eLLM: Show Audit Log**. The **eLLM Audit** output channel opens in the panel area.

2. Read the entries: each request with its endpoint, model, user identity and active skills; each file read, search and listing; each proposal with the Apply or Reject decision; each command with its exit code.
3. For the files themselves, open `.project-ai/audit/<date>.jsonl` . One JSON object per line makes them easy to grep or import.
4. To capture a debug trace, click , tick **Debug logging**, save, then reproduce the behaviour.
5. Choose **Show debug trace**. The **eLLM Debug** channel shows a readable rendering; the raw file is `.project-ai/debug/<date>.jsonl` .
6. Look for `decision` lines to see why a plan advanced or paused, `verify` lines for gate results, and `apply` lines with any edit failures and their reasons.

What you should see

Every controlled pause, every tool step and every gate verdict is on the record, so a "why did it stop" question is answered by the trace rather than by guesswork. The audit log is on by default and controlled by `eLLMCode.auditLog` .

Troubleshooting

- **The debug file is empty:** debug logging was off when the run happened. It has to be on beforehand.
- **The trace is large:** it records full prompts and replies. Turn it off when not diagnosing, and add `.project-ai/debug/` and `.project-ai/audit/` to `.gitignore` .

Related guides

- How to Download a Support Bundle and Report an Issue
- How to Send Identity and Intent Headers to the Gateway