

---

ELLM

# How to Set Up and Maintain Project Memory

*Learn how to set up and maintain project memory using markdown files to store essential project information and decisions. This guide covers initialisation, updating, and compacting project memory for efficient team collaboration.*

AUTHOR

Paul Flanders

PUBLISHED

10 Sep 2026

DOCUMENT

InfoHub — Knowledge, news, and updates from  
EssingtonITS

# How to Set Up and Maintain Project Memory

---

Paul Flanders · 10 Sep 2026

**Estimated time:** 5 minutes

**Difficulty:** Beginner

## Why you'd use this

Every conversation starts fresh, but a project has standing knowledge: coding standards, things never to change, architectural decisions, the current task list and a log of what was done. Project memory is a small set of markdown files in a `.project-ai/` folder that the assistant reads with every request and updates as it works. Commit the folder and the whole team's assistant shares the same context.

## Before you start

### Permissions required:

- Write access to the workspace root.

### You'll need:

- A few lines on the project's conventions and any rules the assistant must respect.

## Steps

1. Press `/` and choose **Initialize project memory**, or run **eLLM: Initialize Project Memory**. The folder is created with `memory.md`, `architecture.md`, `decisions.md`, `tasks.md` and `devlog.md`, and `memory.md` opens in the editor.

2. Fill in `memory.md` : the project's purpose, coding standards, and a "never change" list. Keep it short; it is sent with every request.
3. Work with the assistant as normal. It records decisions in `decisions.md` , keeps `tasks.md` current during Plan Mode, and appends a devlog entry when a request changes files.
4. To read or edit memory later, choose **Open project memory** from the / menu.
5. When the files have grown, choose **Compact project memory**. The assistant rewrites them shorter, keeping every rule and decision but dropping stale detail, and offers the result as diff cards for you to apply.
6. Commit `.project-ai/` to version control, except the `audit/` , `debug/` , `scratch/` and `checkpoints/` subfolders which are local working data.

## What you should see

A new project shows "Initialized project memory" with the file names; an existing one shows "Project memory already exists". Helper scripts the assistant writes for itself go under `scratch/` and are cleaned up when a build completes. The folder name is configurable with `ellmCode.workspaceMemory` .

## Troubleshooting

- **The assistant ignores a standard:** check it is in `memory.md` rather than only in a chat message; only the memory file persists across sessions.
- **Memory has become very large:** the prompt is fitted to the model's window and a note says which parts were trimmed. Compact the memory, or move long reference material out to normal docs the assistant can read on demand.

## Related guides

- How to Resume a Project and Catch Up
- How to Use the Standalone Planning Commands

- How to Check and Manage the Context Window