
ELLM

How to Use the Standalone Planning Commands

Learn how to use standalone planning commands to create specific planning artefacts without initiating a full build. This guide covers generating requirements, architecture plans, and task breakdowns, with tips on permissions and troubleshooting.

AUTHOR

Paul Flanders

PUBLISHED

10 Sep 2026

DOCUMENT

InfoHub — Knowledge, news, and updates from
EssingtonITS

How to Use the Standalone Planning Commands

Paul Flanders · 10 Sep 2026

Estimated time: 4 minutes

Difficulty: Intermediate

Why you'd use this

Sometimes you want one planning artefact without starting a full build: a requirements analysis to share with a stakeholder, an architecture plan for a review meeting, or a task breakdown to put into your tracker. Three commands run those stages on their own, in the normal chat, and write the results to the project memory files so they are there when a build starts later.

Before you start

Permissions required:

- Write access to the `.project-ai/` folder if you want the output saved there.

You'll need:

- Some description of the goal already in memory or in the chat. Run these after a conversation about the feature, or after filling in `.project-ai/memory.md`.

Steps

1. Press `/` and choose **Analyse requirements**. The assistant reads memory and the project and produces functional and non-functional requirements, open questions and

assumptions.

2. Choose **Generate architecture plan**. An input box asks for the feature or area to design; leave it blank for the whole project, then press **Enter**. The plan covers components, data flow and key decisions, and is offered as a change to `architecture.md` and `decisions.md`.
3. Choose **Break down tasks**. The assistant produces an ordered checklist with dependencies, proposed as a change to `tasks.md`.
4. Apply the proposed memory file cards, or leave **Auto-apply** on so they are written directly.
5. When you are ready to implement, run **Resume / catch up** to build from the saved task list, or start Plan Mode with the brief.

What you should see

Each command produces a structured markdown reply plus diff cards for the memory files. Because the reasoning model handles these intents when one is configured, the output is noticeably stronger than the everyday code model would give.

Troubleshooting

- **The output is generic:** the project memory is empty. Run **Initialize project memory** and fill in the standards and goals first, or describe the feature in the chat before running the command.
- **You cancelled the scope input:** pressing Escape on the architecture prompt cancels the command; run it again.

Related guides

- How to Set Up and Maintain Project Memory
- How to Plan and Build a Feature with Plan Mode