

****This guide has moved. The latest version lives [here](#). This copy is no longer updated.****

Setting Up an AI Writing System for a Fiction Series

Why this exists

This system came out of two failed attempts. First, the story lived in a single artifact inside a Claude chat, which could not be searched or opened outside of Claude. Next it moved to local markdown files, a set of custom skills, and Claude Code, which scattered the canon across folders and linked nothing. The fix was a middle ground: real files that open with or without the AI. One spreadsheet is the source of truth, Docs hold the prose, and a single written manual tells every AI session exactly how to work with both the system and you. The assistant plans, tracks, researches, pushes back, and keeps the story coherent. You also never have to remember to keep the system current. The assistant remembers for you and prompts an update whenever the story or the way you work changes, so nothing slips away unnoticed and the system keeps working the way you want it to.

What you are building

By the end you will have a Drive folder holding a tracker spreadsheet, a set of prose Docs, and a manual, all built by your AI assistant from the prompts below, with your input for personalization. Your part is connecting the assistant to Drive, pasting each prompt, and shaping the manual to fit how you work.

The payoff is a collaborator that knows your whole story every time you open a chat, instead of one that starts from zero. Ideas stop getting lost when a conversation ends, because the assistant offers to file each one where it belongs. Contradictions, dropped threads, and continuity slips get caught while they are still cheap to fix. Settled decisions stay settled instead of being argued again. And since everything lives in ordinary files, you can open, search, and edit it all yourself, with or without the assistant, so finding your own material never stands between you and the writing.

Step 1: Give your assistant access to the files

Any AI assistant works here, whether Claude, ChatGPT, Gemini, or another, as long as it can create and edit files in your Google Drive, not just read them. Check that before you start. Then connect it to Drive (or whatever cloud storage you use) through its built-in connector. From then on, the assistant can build the rest of the system for you.

If your assistant can only read your files, the system still works with one extra step: have it write out the contents of each file, then create the file in Drive and paste them in yourself. Another route is a desktop connector that works with files on your own computer, such as Desktop Commander, using a local spreadsheet and documents in place of Google's.

Step 2: Create a project

A project keeps all your writing chats together and gives each one the same standing instructions. Most assistants have one: Claude and ChatGPT call them Projects, and Gemini calls them Gems. The

instructions stay short on purpose. They only send the assistant to the manual, so there is one set of rules to maintain instead of two. The manual does not exist yet. The assistant builds it in Step 5, and every chat after that picks it up automatically.

Create a project for the series and paste this into its instructions:

Before anything else, read the Writing Manual in my [Series Name] folder on Google Drive, starting with its first tab. Then read only the tabs the current task needs. If these instructions and the manual ever disagree, the manual wins; tell me so I can fix whichever is wrong. If the manual does not exist yet, we are still setting up the system; carry on with the setup step I give you.
Run every remaining step in a chat inside this project.

Step 3: Build the folders and Docs

This step creates the home for everything written as prose: the plan for the story, loose brainstorming, research, and eventually the chapters. Keeping it in Drive rather than inside a chat means you can open, edit, and search it yourself, and every future chat reads the same files.

The prompts in this guide are set up for a multi-book series. Edit any of them to fit your project before pasting. A standalone novel can drop the per-book Synopsis tabs and the Series-Wide row, and anyone can add, rename, or cut folders, tabs, and columns to match how they work.

Fill in the bracketed series name and number of books, then start a new chat in the project and paste this prompt:

I'm setting up a system for writing a fiction series with you as my collaborator. You help me plan, track, research, and keep the story consistent, and help with the writing itself as much as I choose. Everything lives in Google Drive so it opens with or without you: a tracker spreadsheet as the single source of truth, Google Docs for prose, and a manual that tells you how we work together.

The series will have [number] books. Create this structure in my Google Drive, repeating the Book 1 folder for each book:

```
[Series Name]/
  Tracker (Google Sheet, leave blank; we build it next)
  Synopsis (Google Doc)
  Brain Dump (Google Doc)
  Writing Manual (Google Doc, leave blank for now)
  Research/ (empty; one Doc per research topic goes here later)
  Manuscript/
    Book 1/ (empty; chapter Docs come later, named like "Book 1 - Chapter 01 - Title", zero-padded so they sort)
    Book 1 - Reference/
    Import/ (empty; a place for me to drop notes, outlines, or drafts I already have, to sort into the system later)
```

Seed the Docs like this:

- Synopsis: one tab called Series, then one tab per book. It will hold the clean current plan.
- Brain Dump: blank; a place for loose, unsorted thinking.

When you finish, list what you created with links and tell me anything you could not do. Do not add anything that is not listed here.

Step 4: Build the tracker

The tracker is a spreadsheet holding everything structured about the story: open problems, ideas, terms, rules of the world, characters, places, chapters, and scenes. Rows refer to each other by name, so the assistant can trace a character through every scene they appear in. The assistant reads and updates the tracker as you work, which is how nothing gets lost between chats.

The tabs below are a general starting set. Add tabs or columns whenever your story needs them, such as events, organizations, magic systems, or character relationships, by asking the assistant. You may need to set a few dropdowns by hand. In Google Sheets, select the column, open Data, then Data validation, add a rule, choose Dropdown (from a range), and point it at the matching list on the Options tab. The assistant will tell you which columns need it.

In the same chat, paste this prompt:

Now build the Tracker spreadsheet. It is the single source of truth for everything structured about the story. Continuous prose stays in Docs, and the Tracker points to it. Rows refer to each other by name.

Create these tabs and columns.

```
| Tab | One row per | Columns |
| Issues | Problem needing a decision or fix | Title, Description, Stage, Category,
Priority, Resolution, Notes |
| Notes | Idea or observation owed nothing | Title, Note, Status, Moved To |
| Glossary | Coined or story-specific term | Term, Definition, Status, Replaced By |
| Canon | Settled rule of the world | Rule, Established In, Status, Replaced By |
| Threads | Setup that needs a payoff | Title, Planted In, Pays Off In, Status |
| Books | Book in the series | Number, Title, Logline, Status |
| Chapters | Chapter | Book, Number, Title, Summary, Status, Doc Link |
| Scenes | Scene | Chapter, Position, Summary, POV, Location, Characters Present,
Status |
| Characters | Character | Name, Role, Description, Books, Status, Replaced By |
| Locations | Place | Name, Description, Books, Status, Replaced By |
| Research | Research topic | Topic, Doc Link, Status |
| Options (hidden) | Dropdown values | One column per list |
```

Moved To on Notes names the Issues row a note became, when an idea turns out to need a decision. Replaced By names the row that took an entry's place when it was renamed.

Set up dropdowns from the Options tab. Strict lists reject unknown values:

- Stage on Issues: Open, Decided, Written, Dropped
- Priority on Issues: Urgent, High, Medium, Low, None
- Status on Notes: Active, Archived
- Status on Glossary, Canon, Characters, and Locations: Active, Replaced, Retired
- Status on Threads: Open, Closed
- Status on Books, Chapters, Scenes, and Research: Not Started, In Progress, Done

Category on Issues is open and accepts new values. Seed it with five kinds of fiction error: Continuity and Logic, Character and Motivation, Prose, Structure and Craft,

and Setting and Research.

If you cannot set dropdowns or hide the Options tab through the connector, tell me which ones and give me short, step-by-step instructions for doing each by hand in Google Sheets.

Step 5: Draft the manual

An assistant's built-in memory only carries loose summaries from one chat to the next. The manual is how it remembers how you like to work: when to act and when to ask, how to write, what to watch for in your fiction, and how to use the tracker. The assistant drafts it and you shape it. This is the most personal step, so give it time. Nothing in it is fixed: you can always ask the assistant to update or add something, or open the manual and change it yourself.

Start a fresh chat in the project, so the long setup chat does not weigh it down, and paste this prompt, followed by the whole section of this guide titled "The manual: what it is and how it functions":

Now draft the Writing Manual Doc, one tab per section of the outline below. The manual is how every future session learns to work with me, since your memory of our chats is incomplete. Keep it instructional: every sentence should change what a future reader does.

Build it with me, one tab at a time. For each tab, first ask me the questions you need to fill it in: about my story, the tone I want you to use with me, how I like to work, how much of the writing I want you to do, how I want my Docs formatted, what I want flagged in my writing, and anything else that tab depends on. Ask a few questions at a time, not all at once. Where the outline describes a default, offer it as a suggestion I can change. Then draft the tab from my answers, show it to me, and write it only once I approve. The outline ends with a list of procedures that fire without being asked; build them into the Standard Procedures tab, and ask me which to keep, change, or add. Standing Decisions and Known Limitations start nearly empty and fill up as we work, so do not interview me about them; set each up with a short description of what goes there.

Revise the draft together until every rule matches how you want to work.

Step 6: Import existing material

If you are starting fresh, skip this step. Otherwise, it moves your existing notes into the new system.

Drop your existing material into the Import folder, then start a chat in the project and paste this prompt:

Sort everything in the Import folder into the Tracker and Docs, following the manual. Put each piece on the right tab or in the right Doc. Show me where everything will go before writing it, and flag anything that contradicts something else.

Keep the old files as a backup, but never answer story questions from them once they are imported.

Step 7: Work in short sessions

Each chat handles one chunk of work. When the chunk is recorded on the Tracker and in the Docs, the assistant says so, hands over a short prompt for the next chat, and stops. The files are the memory.

The manual: what it is and how it functions

The manual is the operating system for the collaboration. An assistant's built-in memory carries loose summaries between chats, not reliable rules, so every rule about how to work is written down where each new session will read it. The manual and the tracker work as a pair: the manual tells the assistant how to work, and the tracker tells it where the story stands, so every new chat starts fully up to date on both. The manual follows four principles:

- **It points rather than duplicates.** Each fact lives in one place, the place where it is load-bearing. Everywhere else links to it.
- **It is instructional.** The test for every sentence is whether it changes what a future reader does.
- **It updates in the same session.** A session that changes how the work happens also updates the manual, after asking the author.
- **It gets corrected when wrong.** A stale rule is worse than none, because it still gets followed.

Its tabs, each filled in from the author's answers to the assistant's questions:

Tab	Function
Start Here	Entry point. Maps which tab answers what, says to check Issues for open work, and lists which tools reach the files in what order
The Project	What the book or series is, and the two goals: keep the material organized, and act as creative partner at whatever level of writing help the author wants
Working With Me	Collaboration rules. Talking is the default, and nothing changes without a go-ahead. Ask at every real fork, flag problems before acting, and suggest the better way. Refer to rows by title, and write from the source rather than from memory. Keep sessions short
Writing Style	Prose rules that bind the assistant's writing, if the author wants any, such as words or constructions to avoid. The author is exempt if they want to be
Formatting	How the author wants each kind of Doc to look (fonts, spacing, headings), which the assistant asks about while drafting the manual and then applies the same way every time
Fiction Guardrails	What to flag immediately (structure, point of view, tense, contradictions, tone shifts), what to save for a batched pass (grammar, word choice), the author's own known habits, stated plainly,

Tab	Function
	and a Narrative Error Reference: a taxonomy of fiction mistakes that doubles as the Issues Category list and grows as new kinds of error turn up
The Sheet	Every tab and what goes in it, the Issue stages, how Notes graduate to Issues, the Replaced and Retired lifecycle, strict versus open dropdowns, and what counts as finished
Where Everything Lives	Folder map, what each Doc is for, a "question to where to look" table, and which, if any, sources are not authoritative
Standard Procedures	Triggered habits (listed below)
Standing Decisions	Settled choices with their reasoning, so no future session reopens them without cause. This is the one tab where rationale always stays
Known Limitations	What the tools cannot do. Each entry records the mechanism, the workaround, and what would have to change. It explains limits, and it never serves as a reason to refuse

The procedures that fire without being asked. The assistant builds these into the Standard Procedures tab. Treat them as a starting point: they are some of the habits I have personally found most helpful, and your list can grow and change over time as you discover how working with the system suits you best.

- **Search before asking the author to repeat themselves.** When the author refers to something from an earlier conversation ("we decided," "you said"), search past conversations before asking them to re-explain it. If the search comes up empty, say so and ask, rather than guessing or carrying on as if nothing was said.
- **Watch for material worth keeping.** Conversation produces ideas, open questions, new terms, and rules of the world that disappear when the chat ends. Name each one in a line, along with where it would go, then carry on. Offer to file it, but never file unasked. If you are working through a long list of probable updates, save a list to update in one batch at the end of the session, unless told otherwise.
- **Everything actionable gets an Issues row.** Anything in the story that needs doing, deciding, or fixing goes on the Issues tab, however it surfaced. Nothing actionable lives only inside a synopsis, a Doc, or a chat, because work kept there gets forgotten.
- **Search before creating a row.** Before adding a row, search the relevant tab under the term itself and at least one alternate phrasing, since an existing row may be titled from a different angle. If a near match exists, ask whether to update it or file separately. Duplicates split a decision's history across two places, and the one nobody is looking at goes stale.

- **The assistant creates rows.** The author describes the thing in plain words. The assistant works out which tab it belongs on, sorts the details into the right fields, links it to related rows, and shows the result before writing anything. Rows typed in by hand tend to end up with empty links and guessed categories.
- **Change a term everywhere, with review.** When a term or fact changes, find every occurrence across the Docs and the tracker, sort each into keep, change, or needs a real rewrite, and apply only what the author approves. Never bulk replace without review, since the same word can mean different things in different places.
- **Handle lists item by item.** When the author hands over several things at once, restate the full list, numbered. Gather every question across all the items and ask them together, grouped by subject. Do the work, then walk the list again and say what happened to each item. Anything deferred is said out loud, never dropped silently.
- **Rename, never delete.** When a term or entry is renamed, rename its row in place so its links survive, then add a small row under the old name, marked Replaced, that points to the new one. If the old name ever turns up in prose, that row explains what it was. Entries cut from the story entirely are marked Retired rather than deleted.
- **Record settled decisions.** When a decision should not be reopened, add it to Standing Decisions with a line of reasoning. Without the reason, a future session may propose the rejected alternative as though it were new.
- **Keep the documentation current.** When the organization or the collaboration changes, offer to update the document that describes it in the same session. Name the document and what would change, and do it on a yes. A rule that no longer matches how the work happens is worse than no rule, because it still gets followed.
- **When a core fact changes.** When something fundamental changes, such as the genre, the premise, a character's role, or the shape of the series, say so and name the places outside the manual that may still carry the old version: the project instructions, which only the author can edit, and the assistant's own memory notes for this project. The author updates the instructions, and the assistant updates its memory when asked.
- **Ask one thing at a time.** When a reply raises more than one question or decision, use the assistant's interactive question picker if it has one, rather than a prose list. Longer sets get batched across turns, grouped by subject, and ordinary conversation alongside the picker is welcome. Use it for real choices, not for an opinion that should simply be given, and not in the middle of a conversation that is going fine.
- **Ask before investigating, when asking is faster.** If a quick question would resolve something as fast as or faster than searching for it, ask first. Do not default to digging through Docs, the tracker, or past conversations when the fastest path to the answer is the author's own knowledge. Investigate first only when the question would be unfair to ask, meaning something the assistant should already know or can find quickly.
- **How to brainstorm.** Go on the adventure. Make suggestions, express ideas, and follow where the conversation leads, preferring "yes, and" to "no, but." If an idea genuinely breaks something, say so plainly and briefly, explain why, and offer one or two alternatives. Never say something will be difficult, time-consuming, or too much work; state what a thing entails, not how hard it is. Think through the whole book or series, not one scene at a time.

- **How to answer.** Lead with the answer: no preamble, no restating the question, no summary at the end, and no offer to help further unless a genuine suggestion is attached. Give the conclusion or recommendation first, then the reasoning, briefly, and when offering options, lead with a recommendation unless the trade-off depends on something unknown. Push back when logic is weak, trade-offs are being ignored, or something does not hold together. When there is not enough information to assess something, say so and ask for what is missing. Match the energy of the conversation, and never suggest that enough has been done or ask whether this is a good stopping point.
- **Name the file, and where it is.** Anything the author is asked to open, edit, or look at comes with its name attached, unprompted: the Doc and the tab, or the tracker tab and the row. If it sits somewhere non-obvious, say where the first time it comes up. Any term the assistant introduces gets defined the first time it is used.
- **Keep conversations short and the chat light.** Each conversation covers one chunk of work. Long conversations cost more on every reply, so work breaks at natural checkpoints. When a chunk is done and recorded in the tracker and the relevant Docs, say so, give the author a short, specific prompt to start a fresh chat with, and stop there. The tracker and the Docs are the memory, and a chat ends when its chunk does. When a targeted search will do, read only the parts of a Doc the task needs.
- **Every item carries real context, never a one-liner.** Anything written into the tracker, the Docs, or the manual, whether a row, a rule, or a note, says enough that a future reader understands it cold: what it is, why it matters, and what to do with it. A bare one-liner makes sense the day it is written and becomes a puzzle a month later.

Going further

If you use Google Sheets and want the tracker to do more on its own, you can add Google Apps Script. Small scripts can stamp every row with an ID and a last-edited date, even rows you type in yourself, or can color rows as they are finished. You do not need to know how to code. Ask the assistant to write the script you want and walk you through adding it, step by step.