

Setting Up Projects in ChatGPT and Claude

A practical guide to giving AI the context it needs to work like a member of your team

A follow-up resource from the *Stop Prompting, Start Directing* webinar · Human-AI Systems with Third Circle

Most people use ChatGPT or Claude as a clever search box. They open a blank chat, type a question, get an answer, and start again from nothing the next time. It works, but it wastes most of what these tools can actually do. **A Project changes that.** It gives the AI a standing brief and a set of background documents, so every conversation starts with it already knowing who you are, what you are working on, and how you like things done.

This guide explains how Projects work in both ChatGPT and Claude, how to set one up well, what to put inside it, and how to prompt it once it is built. It uses a single worked example throughout, then shows how to adapt the same pattern to anything you do.

A note on the HSBC example

Throughout the guide we use one running example: a project built to prepare for business development meetings with HSBC. We use it because it was the live example in the webinar, where we built a project from scratch on screen. Not everyone reading this was on the call, so it is worth saying plainly what it is. It is an illustrative scenario, a consultant getting ready to sell into a large bank, chosen because it is concrete and recognisable. The bank is a stand-in. The point is the method, and the same steps work whatever your equivalent is: your client, your customer, your account, your area of work.

Projects exist in both tools. ChatGPT calls them Projects. Claude calls them Projects too. They work in much the same way: a custom instruction that always applies, plus a set of files the AI can draw on. The advice here applies to both. Where they differ, we say so.

1. What a Project actually is

A Project is a workspace that wraps three things together: the **conversations** you have inside it, a set of **instructions** that apply to every one of those conversations, and a collection of **files** the AI can read whenever it needs them. Instead of a stateless chatbot that forgets you between questions, you get a persistent assistant that already holds your context.

The mental shift is simple. A plain chat is a conversation with a stranger who happens to be well read. A Project is a conversation with a colleague who has read your briefing pack, knows the account, and remembers what you told them last week.

2. One project per focus area

The single most useful habit is to run a separate Project for each thing you work on. A project is not one big bucket for all your AI use. It is a focused space for a particular client, customer, account, theme, or piece of work.

Think of it the way you would think of folders, or of having a different notebook for each major piece of work. You would not keep your notes on three different clients on the same page. The same logic applies here.

Typical ways people split projects:

- **By client or customer.** One project per account. The HSBC project knows about HSBC. The Barclays project knows about Barclays. Neither bleeds into the other.
- **By area of work.** A project for marketing content, another for board reporting, another for recruitment.
- **By role or hat.** If you wear several hats, give each one a project with its own instructions and reference material.

The benefit is focus. Because the project only contains material relevant to that one area, the AI is not distracted by unrelated context, and its answers stay on topic. You also always know where to go: everything about HSBC lives in the HSBC project.

3. Project memory: keeping each project sealed

Both tools can remember things about you over time. The risk is obvious: you do not want a detail from a personal chat surfacing in a client project, or notes from one client leaking into another. This is where **project memory** matters.

ChatGPT now offers **project-only memory**. When you switch a project to project-only, it draws context purely from the conversations already inside that project. It does not pull in memories from your general chats, and nothing it learns inside the project leaks back out to the rest of ChatGPT. Each project becomes a sealed room. You set this when you create the project, under More options, by changing Memory from Default to Project-only.

Claude keeps each project's knowledge and conversations contained within that project by default, so material does not move between projects in the same way.

Why this matters

Sealed memory is what makes it safe to run a project per client. The HSBC project builds up useful context about HSBC over time, the conversations, the angles you have tried, the decisions you have made, without any of it mixing into your other work or your other clients. It is the difference between one shared notebook and a locked drawer per account.

4. Instructions: telling the AI how to behave

The instructions are a short standing brief that applies to every conversation in the project. You write them once. From then on, the AI follows them without you having to repeat yourself. Good instructions answer three questions: **who you are, how you want the AI to respond, and what to avoid.**

A few principles make instructions work better:

- **Front-load the context.** Open with who you are and what the project is for. The AI uses the first lines to frame everything that follows.
- **Be specific about the output.** Say what good looks like. "Give me background, sensible discovery questions, talking points, and things to watch for" beats "help me prepare".
- **Tell it what to avoid.** A short list of things not to do is often more effective than a long description of what to do. "Do not invent specifics you do not have. Say so instead."
- **Set the voice.** Spelling, tone, length. "Write in UK English. Keep it practical and concise."
- **Keep it tight.** Short, direct instructions work better than long paragraphs. You can always refine them once you see how the AI responds.

Example instructions (from the HSBC project)

"You are helping me, founder of a small AI transformation consultancy in the UK. Use the uploaded files to understand what we offer and my background. When I ask you to prepare for a meeting or a call, give me useful, well-organised preparation: a bit of background on the company, sensible discovery questions, talking points that connect their likely needs to what we offer, and a few things to watch out for. Keep it practical and professional. Write in UK English. If you do not have information about a specific company, say so and work from reasonable general assumptions rather than inventing specifics."

Notice the shape: who you are, what to produce, how to write it, and one clear guardrail against making things up.

5. Project files: giving the AI your context

This is where most of the value sits, and where most people do too little. The files are the AI's reference library for the project. Without them, even a well-instructed assistant is working from general knowledge and guesswork. With them, it answers from **your** reality.

The principle behind all of it is context. An AI is only as good as what it knows about your situation, and by default a project knows only what you give it. Out of the box it cannot see your email or your files.

You can change some of that with connectors, which both ChatGPT and Claude now offer. Connect your email and the AI gains that context. Connect your document storage and it can search your files. **Searching is the key word, though.** It gets a search window into your documents, not the structured, organised filing system that a full AI operating system would give it, so what it surfaces can be hit and miss.

That is why the project files still matter, even with connectors switched on. They are the context you have chosen, curated and consistent, sitting in front of the AI every time rather than left to a search to find. Files are how you give it your standing context at scale, once, for every future conversation.

What to include

- **Examples of your own writing.** A few pieces you have written, emails, a proposal, a report, a couple of posts. This is the single best way to get the AI to match your style and voice rather than producing generic prose. Show it, do not just describe it.
- **Information about you.** A short bio, your role, your background, what you are known for. So the AI frames things from your perspective.
- **Information about your business.** What the company does, your products or services, how you are positioned, who you serve. So recommendations and drafts are grounded in what you actually sell.
- **Frameworks, methods, and house rules.** Any model, process, or set of standards you work to. So the AI uses your language, not a textbook's.
- **Material specific to the project.** For a client project, that might be notes on the account, their priorities, prior correspondence, a summary of past meetings.

The context point, in one line

The quality of what you get out is set by the quality of what you put in. A blank-slate AI gives you blandly competent answers. A well-fed project gives you answers that sound like you, reference your offer, and fit your world. The files are not admin. They are the engine.

6. What file formats to use

Format matters more than people expect. The same information can be easy or hard for the AI to read depending on how it is packaged. Here is the order of preference, best first.

Format	Verdict	Why
Markdown (.md)	Best	Plain text with simple headings and lists. It is the format these models were trained on most, so structure comes through cleanly and nothing is wasted. The AI reads it almost exactly as you wrote it.
Word (.docx)	Good	The sensible choice if you do not know how to make a Markdown file. Headings, lists, and plain layout all read well. Use clear heading styles and avoid heavy formatting, text boxes, and complex tables.
PDF (.pdf)	Workable	Fine in a pinch, but weaker. PDFs are built for printing, not reading by machine. Columns, headers, and layout can scramble the running order, and a scanned PDF may be an image the AI cannot read as text at all. It also costs far more of the AI's working capacity to process than the same content as Markdown.
Images and diagrams	Avoid in files	Charts, screenshots, and diagrams embedded in project files are largely wasted.

Format	Verdict	Why
		The AI does not reliably pull meaning out of a picture sitting inside a document. If the diagram matters, do not rely on it being understood.

So the rule of thumb is simple: **Markdown if you can, Word if you cannot, PDF only if that is all you have, and never lean on images.**

Tip: turn important images into words

If a diagram or chart genuinely matters, do not bury it in a project file and hope. Instead, open a normal chat with ChatGPT or Claude, paste in the image on its own, and ask it to **describe the image in detail as words**. It will give you a clear written description of what the diagram shows. Take that text, tidy it, and add it to your project file. Now the information is in a form the AI can actually use, every time.

How to make a Markdown file, if you want to

Markdown is just plain text with a few light touches: a # at the start of a line makes a heading, a dash makes a bullet. You can write it in any plain text editor and save it with a .md ending. Or, easier still, ask ChatGPT or Claude to "put this into a Markdown file for me", paste your content, and save what it gives you. If that feels like a faff, a Word document is perfectly good.

7. Prompting a project well

Once a project holds your instructions and your files, the way you prompt it changes. You are no longer starting cold, so you do not have to explain yourself every time. Your prompts can be shorter, more direct, and more about the task in front of you. The project already carries the background.

Five habits get the most out of a project that already has context:

- **Do not re-explain what it already knows.** Your bio, your services, your frameworks, your style are in the files. You do not need to paste them again. Skip straight to the task. Repeating context just clutters the chat.
- **Give it the new information it cannot have.** The project knows about you, but not about today. Supply the variable detail it has no way of seeing: the specific person you are meeting, a fresh email thread, the live numbers, the actual deadline. The project supplies the standing context; you supply what just changed.
- **Be clear about the task and the output.** Say what you want and in what shape. "Draft a two-paragraph follow-up email" or "give me five discovery questions and three risks" beats "help with the HSBC meeting". Tell it the audience and the length.
- **Point it at the files when it helps.** A nudge like "using the frameworks in the project files" or "in the style of the writing samples" tells it which part of its context to lean on for this particular task.
- **Treat it as a first draft, then refine in the same chat.** You rarely get the perfect answer first time, and you should not expect to. Ask, read, then steer: "sharper", "too formal", "cut it by half", "more on the risk point". The back and forth is where the quality comes from.

A simple shape for a strong prompt

Task. What you want done. "Prepare me for a first call."

What's new. The detail the project cannot already know. "Here is the person I am meeting and their LinkedIn profile: [paste]."

Output. The form you want it in. "Give me background, five questions, and three things to watch, on one page."

Constraints. Anything to hold to. "UK English, plain, no jargon."

You will not need all four every time. But when an answer comes back thin, it is usually because one of these was missing.

The same prompt, in and out of a project

Without a project: *"I run a small AI consultancy. We help organisations adopt AI. Here is my bio... here is what we offer... now help me prepare for a call with a senior person at HSBC, give me background and some questions, and write in UK English."*

With the project: *"Prepare me for a first call with this HSBC contact: [paste their profile]. Background, five questions, three risks, one page."*

Same result, a fraction of the effort, because the project is already holding everything the first version had to spell out.

8. Worked example: the HSBC project

Here is the whole thing assembled, using the running example: preparing for business development conversations with HSBC. The aim is an assistant that helps prepare for meetings, in your voice, grounded in what your business actually offers.

Step by step

1. **Create the project.** In ChatGPT or Claude, create a new Project and name it for its focus. "HSBC", or "Business Development" if you would rather not name the account.
2. **Set memory to project-only** (ChatGPT), so anything the project learns about HSBC stays sealed inside it.
3. **Paste in the instructions.** The standing brief from section 4: who you are, what preparation to produce, UK English, do not invent specifics.
4. **Upload the files.** A few core documents that give the AI its context:
 - **A one-page overview of your business** (what you do, your services).
 - **A short summary of your frameworks or methods** (so it uses your language).
 - **A short bio** (so it frames things from your point of view).
5. **Prompt it.** Open a chat in the project and ask, in the shape from section 7: "Prepare me for a first call with this HSBC contact: [paste their profile]. Background, five questions, three risks, one page." Because the instructions and files are already in place, it produces tailored preparation without you setting the scene.

And it keeps paying off. The next time you have an HSBC conversation, the project is still there, still briefed, and now also remembers the ground you have already covered. You are not starting from a blank page. You are picking up where you left off.

Be honest about the ceiling

A project like this is a strong assistant, not a colleague who does the work for you. It can tailor to you, use your frameworks, and write decent preparation. It cannot look things up live, see a contact's profile unless you paste it in, or know anything that is not in its files. It hands you well-organised words. You still bring the judgement. Knowing where that line sits keeps your expectations right, and it points to where this goes next.

9. Adapting this to anything

The HSBC example is just one shape of a pattern that works for almost any focused area of work. The recipe is always the same four moves.

1. **Name the focus.** One project, one clear area: a client, a customer, a theme, a recurring task.
2. **Write the standing brief.** Who you are, what you want produced, how you want it written, what to avoid.
3. **Feed it context.** Examples of your writing, who you are, what your business does, plus anything specific to that focus, in Markdown or Word.
4. **Prompt it well and let it build.** Project-only memory on, short task-focused prompts, and the context compounds the more you use it.

Some other ways the same pattern lands:

- **Content and marketing.** Files: your best posts and articles, your positioning, your audience. Instruction: write in this voice, for this audience. Result: drafts that sound like you, not like an AI.
- **Recruitment.** Files: the job specs, your interview standards, the competencies you screen for. Instruction: assess against these. Result: consistent, structured candidate work.
- **Board and reporting.** Files: past board papers, the house format, the metrics that matter. Instruction: this structure, this tone. Result: papers that fit the room.
- **A specific customer account.** Files: account notes, their priorities, past correspondence. Instruction: prepare and draft for this account. Result: account work that stays on the account.

10. Where this goes next: from assistant to worker

It helps to see a project for what it is on the ladder of AI maturity. A plain chatbot is a **tool**: you operate it, one question at a time. A well-built project is an **assistant**: it knows you, holds your context, and helps you do the work faster. That is a real step up, and it is where most people should start. But it is not the top of the ladder.

The next rung is the **worker**. This is AI that does not just hand you words, it actually does the work. Rather than drafting an email for you to send, a worker can read the thread, pull the relevant numbers from your systems, draft the reply, and tee it up, taking multiple steps across your real tools while you supervise. The difference is the shift from **producing text to completing tasks**.

A few tools now operate at this level:

- **Claude Cework.** Works across your files and applications, taking multi-step actions to complete a piece of work rather than just answering a question.
- **ChatGPT Codex.** An autonomous agent that can pick up a task and work through it across steps, originally for software, now more broadly.
- **Microsoft 365 Copilot.** Works inside the tools you already use, your email, documents, meetings, and data, acting on the live information sitting in your Microsoft environment.

Why it is better again comes down to where the effort sits. With a tool, you do everything. With an assistant, you direct and it drafts, but you still assemble and act. With a worker, it connects to live information, takes the steps, and completes the task, and your job becomes setting direction and checking the output. You move from doing the work to **leading the work**. That is the real prize, and it is a bigger leap than the one from tool to assistant.

From Prompt to Autopilot

Getting to the worker rung is less about the tools and more about doing it safely and deliberately: choosing the right work to hand over, putting the right guardrails in place, and bringing your people with you. That is exactly the step **Mike and John** help organisations take, through the **From Prompt to Autopilot** programme. If this guide has

got you thinking about where AI could do more than draft, that is the conversation to have next.

Quick-start checklist

- Create one project per focus area. Do not pool everything in one place.
- Switch on project-only memory (ChatGPT) so each project stays sealed.
- Write a short standing brief: who you are, what to produce, how to write it, what to avoid.
- Upload context: examples of your writing, info on you, info on your business, anything specific to the focus.
- Use Markdown if you can, Word if you cannot, PDF only if you must, and keep images out.
- For any diagram that matters, ask the AI to describe it in words first, then add that text to your files.
- Prompt it for the task, not the background: give it what is new, say what output you want, then refine.
- Refine the instructions and files as you go. A project gets better the more you tend it.

Human-AI Systems with Third Circle. Helping organisations move from scattered AI experimentation to disciplined, value-creating adoption.

human-ai-systems.com · thirdcircle.co.uk