

Context Engineering

How to build an AI that already knows who you are.

A working manual for Claude Projects.

READ THIS IF

You use AI like a search engine with manners. You re-explain your business every morning. You have heard people say AI changed how they work, and quietly suspect they are lying.

They are not lying. They have just done something you have not. It takes two hours and it is written down in here, exactly, step by step, with the prompts.

CONTENTS

What is in here

01	The mechanic: what actually happens when you press enter	3
02	The proof: the same question, twice	6
03	Layer one: your global preferences	7
04	Layer two: project instructions	9
05	Layer three: the five knowledge files	11
06	File design: writing for a machine that skims	17
07	Closing the loop: the ritual that creates memory	19
08	Architecture: how many projects, and what goes where	21
09	Testing: five tests with pass and fail conditions	23
10	Maintenance: fifteen minutes, once a week	25
11	The seven ways this fails	27
A	Appendix: a complete worked example	29
B	Appendix: every prompt in this guide	33

How to use this

This is not a document to read. It is a document to work through with Claude open in another window. Every prompt in it is meant to be copied and pasted, not admired.

Sections 3, 4 and 5 are the build. They take about two hours and you can split them across a week. Sections 6 to 11 are what keeps it working after the first month, which is where almost everybody loses it.

Section 7 is the one people underestimate. If you only do one thing in this guide, do section 7.

A NOTE ON TOOLS

This guide is written for Claude Projects, because that is where the mechanics are cleanest and because I want to name buttons rather than gesture vaguely. Projects are available on every plan including free, capped at five projects.

The architecture transfers. ChatGPT, Gemini and the rest have equivalents of all three layers. The words change. The thinking does not.

The mechanic

Six minutes that will change how you use AI permanently.

Claude has no memory. Not a poor memory. None. Every message you send arrives at a model that has never encountered you before and will not exist afterwards.

Everything that feels like memory is one thing: **text that gets quietly placed in front of your message before the model sees it**. That text is assembled fresh, every single time, from a few specific places. If something is not in one of those places, it does not exist.

Once you know what the places are, you stop hoping the AI remembers and start deciding what it remembers. That is the whole discipline. It has a name now: context engineering.

The three layers

There are exactly three places your context can live, and they behave completely differently.

LAYER	WHERE IT LIVES	WHEN CLAUDE SEES IT	YOUR BUDGET
1. Preferences	Settings. Global. One set for your whole account.	Every message, in every chat, in every project. Always.	Under 200 words
2. Instructions	Inside one project. The "instructions" box.	Every message, in every chat inside that project. Always.	Under 400 words
3. Knowledge	Inside one project. Uploaded files.	Pulled in when relevant. Not guaranteed.	5 files, under 2,000 words total

Read the third column again. Layers one and two are *always injected*. Layer three is *retrieved*. That difference decides what you put where, and it is why most people's setups quietly underperform.

THE ROUTING RULE. WRITE IT ON YOUR HAND.

Behaviour goes in instructions. Facts go in files.

If it changes how Claude *acts*, it belongs in layer one or two. If it changes what Claude *knows*, it belongs in layer three. Putting your company history in the instructions box wastes the most valuable space you have. Putting "never use bullet points" in a file means it gets followed roughly half the time.

The thing almost nobody tells you

Here is the fact that reframes everything, straight from Anthropic's own documentation, and I have watched a lot of smart people discover it the hard way.

Two chats inside the same project cannot see each other.

On Monday you have a brilliant conversation in your Business project. You work out your pricing, you nail your positioning, you draft something good. On Tuesday you open a new chat inside the same project and ask about it. Claude has no idea what you are talking about. It will not say so, either. It will guess.

A project is not a shared brain. A project is a set of instructions plus a set of files, handed to a brand new stranger at the start of every single conversation.

WHICH MEANS

If it is not written into the instructions or the files, **it did not happen**. Your best thinking evaporates at the end of every chat unless you deliberately move it into layer two or layer three.

Section 7 is entirely about how to stop that from happening. It is the difference between a project and an operating system.

Why small beats big, mechanically

Everyone's instinct is to upload everything. Here is the specific, technical reason that backfires.

A project has a large but finite context budget. While your knowledge files fit inside it, Claude simply **reads all of them, in full, every time**. Nothing is missed.

When your knowledge grows past that budget, something changes. On paid plans Claude switches into retrieval mode, expanding your capacity roughly tenfold. It sounds like a straight upgrade. It is a trade. Claude is no longer reading your files. It is *searching* them, pulling back the chunks it judges relevant, and answering from those.

Search misses things. Reading does not.

40 FILES, 90,000 WORDS

Claude searches. It finds your pricing sheet but misses the note saying you raised prices in April. It answers confidently with the old number.

You blame the AI.

5 FILES, 2,000 WORDS

Claude reads every word of everything you have written about yourself, before it answers anything.

It cannot miss the April note. The April note is right there.

So the discipline is not tidiness. It is arithmetic. **Keep your core small enough that Claude never has to search it.** Five files. Two thousand words. That is a hard ceiling, and staying under it is worth more than any prompt trick you will ever learn.

Bulk uploads go in a separate project, where searching is the point. Your core stays lean.

What you are building

SETTINGS (global, applies everywhere)

Personal preferences ~180 words. How I want to be spoken to.

PROJECT: **Core**

Project instructions ~350 words. Claude's role and standing rules.

Project knowledge:

01-me.md Who I am. How I work. My voice.

02-business.md What we sell. To whom. The one bottleneck.

03-goals.md This quarter. Dated. Includes what I am NOT doing.

04-decisions.md What I decided, why, and what would reverse it.

05-positions.md Opinions I hold that others do not.

sessions/ Notes I promote out of chats. Grows weekly.

That is it. Two boxes of text and five short files. Total build time around two hours.

The rest of this guide is how to write each one so that it is worth having.

TWO

The proof

Before you spend two hours, see what two hours buys.

Below is a real question, asked twice. Same model. Same day. The only difference is that the second one was asked inside a properly built project.

The person asking is Ben, who runs a six-person commercial cleaning company in Hamilton. You will meet him properly in Appendix A.

The question: *"Should I put my prices on my website?"*

WITHOUT CONTEXT

"Great question. There are strong arguments on both sides.

Arguments for: transparency builds trust, filters out unqualified leads, saves time on calls...

Arguments against: invites price comparison, may undervalue custom work...

Ultimately the right answer depends on your business model, your market positioning, and your customer acquisition strategy. Would you like me to explore any of these in more depth?"

WITH CONTEXT

"No. Not this quarter.

Your bottleneck is that you are the only person who can sell. You convert about one in three of the jobs you quote on site, and about one in nine of the ones that come in by email. Published pricing kills the site visit, which is the only place you actually win.

You already worked through this in March. Your note says you would revisit it if you hired a second estimator. You have not.

Ask me again when you have."

Notice what the second answer does. It refuses. It cites your own bottleneck. It quotes a decision you made four months ago and had forgotten. It names the condition under which it would change its mind.

None of that is a smarter model. All of it is layer three doing its job.

THE UNCOMFORTABLE IMPLICATION

If your AI still gives you the left-hand answer, the AI is not the problem. **Nobody has written you down yet.** It is answering the only way it can: from the average of everything it has ever read, because that is all it has been given.

Everything from here is how to build the right-hand column.

Layer one: preferences

The highest leverage 200 words you will ever write.

In Claude's settings there is a field for personal preferences. Whatever you write there is loaded into every conversation you ever have, in every project, forever. Most people leave it blank. A few fill it with a description of their job, which is a waste.

This field is not for facts about you. It is for **behaviour**. It is where you fix, permanently, the thirty small corrections you currently retype every day.

Write only what changes the default

Claude is already helpful, already polite, already thorough. Telling it to be those things spends your budget on nothing. Every line you write should cause it to behave differently from how it would behave anyway.

WASTED LINES

"Be professional and helpful."
"Give accurate information."
"I run a cleaning company in Hamilton with six staff."
"Be thorough in your analysis."

LINES THAT EARN THEIR PLACE

"Never open with a compliment or a restatement of my question. Start with the answer."
"When I am wrong, say so in the first sentence."
"Give me one recommendation, not three options, unless I ask for options."
"Ask one clarifying question at most, then proceed on your best guess and flag the assumption."

The left column is a description. The right column is a configuration. Only one of them does anything.

And note the third line on the left: the cleaning company. True, useful, and completely in the wrong place. Facts belong in files. This box is behaviour only.

The starting template

Paste this into Settings, then cut and rewrite until it sounds like the thing you actually want. Aim for under 200 words.

LAYER ONE. SETTINGS, PERSONAL PREFERENCES.

HOW TO ANSWER ME

Lead with the answer. No preamble, no restating my question, no "great question".

Give me one recommendation, not a balanced list of options, unless I ask.

If I am wrong, say so in the first sentence and explain why.

Ask at most one clarifying question. Otherwise proceed and flag your assumption.

If you do not know, say so. Do not fill the gap with plausible text.

FORMAT

Short paragraphs. Bullets only when the content is genuinely a list.

No headers on anything under 300 words.

Never use em dashes.

WHAT I AM ACTUALLY ASKING FOR

When I ask "what do you think", I want your view, not a summary of views.

When I share a plan, find the weakest assumption before you find the strengths.

When I am excited about an idea, ask me what it costs me to be wrong.

NEVER

Never flatter me. Never call an idea "exciting" or "powerful".

Never end with an offer to explore further. If there is a next step, name it.

MISS AI SAYS

The line about the weakest assumption is the one that changes your life. Without it you get an enthusiastic collaborator. With it you get a colleague who has read your plan properly.

Then find your own lines

The template above is mine. Yours should be different, and the fastest way to find your version is to notice your own irritation. For one week, every time you type a correction into a chat, write it down. "Shorter." "Just pick one." "Stop agreeing with me." "Do not summarise what I just said."

After a week you will have eight to twelve corrections. Those are your preferences. You have been typing them by hand for a year.

OPTIONAL. HAVE CLAUDE EXTRACT YOUR PREFERENCES FROM HOW YOU ALREADY TALK TO IT.

I am writing my global preferences for how you should respond to me.

Look back over how I have talked to you in this conversation. What corrections have I made? What did I push back on? What did I have to ask for twice?

Turn those into a list of standing instructions, written as commands, in under 200 words. Only include lines that would change your default behaviour. Cut anything you would do anyway.

Layer two: instructions

The most valuable text box in your account.

Create a project. Call it **Core**. Then find the instructions box, which is the field Claude reads before every single message inside that project.

This is the only place you can guarantee Claude sees something. Not "probably retrieves". Sees. So it should contain exactly four things, and nothing else.

1 Who Claude is in here

One or two sentences. Not "you are a helpful assistant". A named role changes how ambiguous questions get interpreted.

2 The standing rules

What Claude must always or never do in this project. These are the highest value lines in the box.

3 A map of the files

Name each knowledge file and say what it is for. This is the part everybody skips, and it doubles retrieval quality on its own.

4 What to do when context is missing

The instruction that makes the system build itself.

ORDER MATTERS

Claude weighs earlier text more heavily than later text. Put your hardest rules at the top of the box, not at the bottom where they read like an afterthought.

The file map: why it matters so much

Claude does not open every file and read it end to end on the off chance. It looks at what you asked, decides which files seem relevant, and pulls those in. That decision is made largely on the filename and the first line.

So tell it, in the instructions, exactly what each file is and when to open it. You are writing a table of contents for a reader who has thirty seconds.

NO FILE MAP

"Refer to the uploaded documents where relevant."

Claude guesses. On a question about pricing it may or may not open your decisions file, where you recorded the price rise.

WITH A FILE MAP

"04-decisions.md contains every significant decision I have made, with the condition that would reverse it. Check it before advising on anything strategic."

Now it opens it. Every time.

The instructions template

Paste this into the box. Edit the bracketed parts. Keep it under 400 words.

LAYER TWO. PROJECT INSTRUCTIONS, CORE PROJECT.

YOUR ROLE

You are my chief of staff. You know my business as well as I do. You are paid to disagree with me early and cheaply, rather than agree with me expensively.

BEFORE YOU ANSWER ANYTHING THAT MATTERS

Read 01-me.md and 03-goals.md.

If the question is strategic, also read 04-decisions.md.

STANDING RULES

If I ask for help with something on my "not doing this quarter" list in 03-goals.md, say so before you help. Then help if I insist.

If we are discussing something already settled in 04-decisions.md, tell me what I decided, why, and what I said would change my mind. Do this before you argue any other position.

If any file's review date is more than 30 days old, warn me it may be stale before you rely on it.

Write in my voice, using the samples at the bottom of 01-me.md.

WHEN YOU DO NOT HAVE WHAT YOU NEED

Do not guess and do not fill the gap. Tell me what is missing, and tell me which of my files it should live in, so I can add it.

MY FILES

01-me.md - who I am, how I work, how to talk to me, samples of my writing.

02-business.md - what we sell, to whom, the numbers, and the single current bottleneck.

03-goals.md - this quarter's targets and, importantly, what I have decided not to do.

04-decisions.md - every significant decision, why I made it, and what would reverse it. Check before advising on strategy.

05-positions.md - opinions I hold that most people in my industry do not. Argue from these, not from the consensus.

sessions/ - notes carried out of previous conversations. Newest first.

THE LINE THAT BUILDS THE SYSTEM FOR YOU

"Tell me what is missing, and which file it should live in."

Within a fortnight Claude will start telling you that it does not know your churn rate, or that your positions file has nothing about hiring. **Your AI begins maintaining its own context.** You just do what it says.

What does not go in this box

- **Your company history.** That is a fact. It goes in 02-business.md.
- **Your brand guidelines.** Fact. File.
- **Anything true in every project.** That belongs in layer one, or you will paste it into six projects and update none of them.
- **Politeness.** "Please be helpful" costs you words and buys you nothing.

Layer three: the five files

Facts, not behaviour. Two thousand words, total.

Do not open a blank document and try to write about yourself. You will produce four paragraphs of LinkedIn language and quit.

Instead, in every case below: **paste the interview prompt, answer badly, then ask Claude to write the file.** You are excellent at answering questions and terrible at blank pages. Build around that.

Do these in one chat inside your Core project, or one chat each. Either works. Answer out loud with dictation if typing slows you down. Ramble. Claude will compress it.

01

01-me.md

Contains: who you are, how you work, how to talk to you, three samples of your writing. **Cap: 500 words plus samples.**

WEAK

"I am a results-driven business owner passionate about delivering value to my clients. I prefer clear, concise communication."

This describes nobody. Claude learns nothing it did not already assume.

STRONG

"I make decisions fast and reverse them slowly, which is the wrong way round. Push me to name what would change my mind at the moment I decide, not later."

"I avoid anything involving numbers until it is urgent. If a task has a spreadsheet in it, I will find something else to do."

INTERVIEW PROMPT

Interview me so we can write a file called 01-me.md for my AI project. It tells you who I am and how to work with me.

Ask ONE question at a time and wait for my answer. Do not list the questions. Do not summarise until I say stop.

Cover: what I do, how I actually spend a working day versus how I think I do, what I am unusually good at, what I avoid, how I make decisions, what my recurring failure mode is, how I take criticism, and what you should never do without asking me.

If an answer is vague, ask for a specific example from the last month before you move on. Be a good interviewer, not a polite one.

When I say stop: write the file. Bullets, not paragraphs. Under 500 words. Include a section "How to talk to me" and a section "Never do this". Put today's date at the top as "Last reviewed".

Then give it your voice

Find three pieces of writing that genuinely sound like you. An email you were proud of. A message to a friend. Something you posted.

VOICE PROMPT

Here are three things I wrote. Describe my voice concretely: average sentence length, how I open, how I close, the words I reach for, the words I never use, whether I use questions, whether I hedge.

Add a "My voice" section to 01-me.md with that analysis, then paste all three samples underneath it as reference.

Telling an AI you are "punchy and conversational" is worthless. Everybody says that. Three real paragraphs of your writing tells it everything, because it can pattern-match rather than interpret an adjective.

02

02-business.md

Contains: what you sell, to whom, the money, and one bottleneck. **Cap: 400 words.**

WEAK

"We serve small and medium businesses across the region with high-quality cleaning services."

"Our bottleneck is growth, marketing, and hiring."

STRONG

"We clean commercial premises between 40 and 400 square metres, on contract, in Hamilton and Cambridge. Typically owner-operated: dental, physio, law. Trigger is usually a failed hygiene audit or a departing office manager."

"Bottleneck: I am the only person who can sell. Everything queues behind my Tuesdays."

INTERVIEW PROMPT

Interview me so we can write 02-business.md. One question at a time.

Cover: what we sell and to whom, how money actually arrives (deposit, retainer, invoice, when), roughly what we make, my ideal customer, the customer I explicitly do not want, what triggers someone to buy, what I can honestly say that competitors cannot, and the single biggest bottleneck right now.

Two rules. When I say "small businesses" or anything that broad, refuse it and make me describe a specific customer we won last month. When I list three bottlenecks, make me pick one.

Then write the file. Under 400 words. Today's date at the top. End with a section called "The one bottleneck" containing exactly one sentence.

IF YOU CANNOT NAME ONE BOTTLENECK

That is not the exercise failing. That is the exercise working. Sit with it for a day. Everything downstream in your business is currently being decided without an answer to this question, including by you.

03

03-goals.md

Contains: this quarter only. **Cap: 250 words.** Dated, always.

The part that makes this file work is not the goals. It is the list underneath them.

WEAK

"Grow revenue. Improve marketing. Build the team. Launch the new service line. Get more consistent on social."

Claude will now enthusiastically help you with all five, which is exactly what you are already doing to yourself.

STRONG

"The one thing: sign four contracts of \$2k/month or more, by 30 September."

"NOT doing this quarter: no new service lines. No hiring. No website rebuild. No social media."

INTERVIEW PROMPT

Help me write 03-goals.md for this quarter. Under 250 words.

First ask me: if only one thing happens in the next three months, what is it? Push until I give you one thing, not a theme.

Then turn it into at most three targets, each with a number and a date. Reject any target I cannot measure.

Then ask what I am explicitly NOT doing this quarter. Make me name at least four things I am putting down. If my "not doing" list is shorter than my goals list, tell me I have not made a plan, I have made a wish, and ask again.

Write the file with "Last reviewed: [today's date]" at the top.

Now, whenever you ask Claude to help you build the thing you promised not to build, it will stop you first. That is not a feature of the AI. That is a feature of the file.

04

04-decisions.md

Contains: what you decided, why, and what would reverse it. **Append only. Never delete.**

This is the most valuable file in the system, it takes four lines per entry, and almost nobody keeps one.

THE FORMAT. ONE OF THESE PER DECISION.

2026-03-14 | Prices stay off the website

WHY: We convert 1 in 3 on site visits and 1 in 9 on email enquiries. Published pricing removes the reason for a site visit.

COST OF BEING WRONG: We waste time quoting people who were never going to pay.

REVISIT IF: I hire a second person who can quote, or email conversion passes 1 in 4.

Four lines. Two minutes. Here is what those two minutes buy you.

It stops you relitigating. In six months you will have this argument with yourself again, having forgotten you already had it. Worse, you will have it with an AI that does not know it was settled, and it will argue the opposite side extremely well, because arguing well is what it does.

It converts decisions into tripwires. That "revisit if" line is the clever part. Now Claude can say: *you said revisit this if email conversion passed one in four. Your last file says it is at one in three.*

INTERVIEW PROMPT. START THE LOG WITH THE DECISIONS YOU HAVE ALREADY MADE.

Interview me about the six biggest decisions I have made in my business in the last eighteen months.

For each: what did I decide, what was I weighing up, why did I choose what I chose, what would it have cost me to be wrong, and what would have to be true for me to reverse it?

If I cannot tell you what would reverse a decision, tell me that means I did not make a decision, I formed a preference. Then help me name the reversal condition.

Write them into 04-decisions.md, newest first, using: DATE | DECISION / WHY / COST OF BEING WRONG / REVISIT IF.

05

05-positions.md

Contains: opinions. Not facts. **Cap: 400 words.**

An AI with no positions writes the average of everything on the internet. That is precisely why AI writing sounds like AI writing. It is not badly written. It is written from nowhere.

NOT AN OPINION

"We believe in quality and putting customers first."

Nobody disagrees with this. If nobody disagrees, it is not a position, it is weather.

AN ACTUAL OPINION

"Cleaning is not a price business, it is a trust business. Every competitor who has undercut us has folded within two years. We will lose a job on price before we lower one."

Someone, somewhere, thinks this is wrong. That is the test.

INTERVIEW PROMPT

Interview me about the opinions I hold about my industry that many people in it would disagree with.

Ask: what does everyone in my industry believe that I think is wrong? What advice do I give that people resist? What did I learn the hard way that nobody warned me about? What do I refuse to do that others do routinely?

If I give you a safe opinion that nobody would argue with, tell me it is not a position and ask again. The test is whether a competent person could disagree.

Write 05-positions.md. Each position as a short heading, then two or three sentences on why I hold it. Under 400 words.

WHAT THIS FILE ACTUALLY FIXES

It is the single highest-impact file for anyone using AI to write. Without it Claude argues from consensus. With it, Claude argues from your corner, and the output stops sounding like everyone else's output.

Writing for a machine that skims

Five rules that make retrieval work. None of them are obvious.

Your files are not read by a human. A human tolerates preamble, context-setting, narrative. Claude does not need any of it, and every word of it displaces something useful.

Rule 1. One topic per file

Retrieval works on whole files or chunks. A file called *everything.md* that covers your goals, your pricing and your childhood will be pulled in for every question and be mostly noise every time. Five focused files beat one big one, every single time.

Rule 2. The filename is part of the system

Claude decides what to open partly by what the file is called. *04-decisions.md* is a retrieval instruction. *notes2.md* is a coin toss. Name files the way you would name them if you were searching for them at 2am.

Rule 3. First line of every file is a routing label

Begin every file with a single line that says what it is and when to use it.

THE FIRST LINE OF 04-DECISIONS.MD

This file contains every significant decision I have made and the condition that would reverse it. Read it before advising on strategy, pricing, hiring, or anything I might have already settled.
Last reviewed: 2026-07-10

Rule 4. Date everything, at the top

A file with no date is a file Claude trusts forever. Pair the date with a standing instruction to distrust anything older than thirty days, and staleness becomes visible instead of silent.

Rule 5. Call files by name in your prompts

Do not hope. Point. "Using 02-business.md and 03-goals.md, tell me whether this is worth doing." Targeted references beat automatic retrieval, and they cost you eight words.

HOPING

"Based on everything you know about my business, should I take this client?"

POINTING

"Read 02-business.md and 04-decisions.md. Given the customer I said I do not want, and what I decided in March, should I take this client?"

Two more things that cost nothing

Files uploaded into a chat are not project knowledge. Dragging a PDF into a conversation makes it available to that conversation and nowhere else. If you want it to persist, you must add it to the project's knowledge deliberately. People lose weeks to this.

Files are snapshots. If you edit the Google Doc on your desktop, the copy inside your project does not change. Update the file, re-upload it, delete the old one. Which is why the master lives in your own folder, and the project holds a copy.

SOURCE OF TRUTH VERSUS RUNTIME

Your folder, wherever it lives, is the **source of truth**. The project is the **runtime**. Projects get archived, tools get discontinued, plans get cancelled. The folder outlives all of them, and in three years you will hand it to something that does not exist yet.

Closing the loop

How a project stops being a folder and starts being a memory.

Everything so far gives Claude a fixed picture of you. Useful, and static. What turns it into an operating system is that the picture updates itself from the work you do inside it.

Recall the mechanic from section one. Two chats in the same project cannot see each other. Monday's brilliant conversation is invisible on Tuesday. Unless you move it.

So the ritual is this. **At the end of any conversation that produced something durable, you spend ninety seconds moving it into layer three.** Then the next conversation starts from where this one ended, instead of from where you started six months ago.

The session note

PASTE THIS AT THE END OF ANY CHAT THAT MATTERED.

Write me a session note for my project knowledge. Under 150 words. Use exactly these headings:

DATE

DECIDED - what we actually settled, if anything

CHANGED - what changed in my thinking, and what made it change

OPEN - what is still unresolved

NEXT - the single next action, with a date

Do not summarise our conversation. Do not preserve the flow of it. If nothing was decided, say so and write "DECIDED: nothing".

Save it into your project knowledge as *sessions/2026-07-10-pricing.md*. Ninety seconds. Now it exists.

Claude will also let you take something it generated in a chat and add it straight into project knowledge. Use that for anything you will reference again: a positioning statement, a pricing structure, a customer profile you finally got right. **The output of your work becomes the context for your next work.** That is the compounding loop, and it is the only one that matters.

THE DISCIPLINE

Not every chat earns a note. Most do not. If nothing was decided and nothing changed, close the tab and write nothing. A sessions folder full of "we discussed some ideas" is exactly the swamp you are trying to avoid.

Rough test: **could you have told a colleague something new because of this conversation?** If not, no note.

The other half: harvesting decisions

Session notes accumulate. Decisions get buried in them. Once a month, promote.

MONTHLY HARVEST

Read everything in sessions/ from the last month.

Pull out anything that was actually decided, as opposed to discussed. For each one, draft an entry for 04-decisions.md using DATE | DECISION / WHY / COST OF BEING WRONG / REVISIT IF.

Where I did not state a reversal condition, ask me for it rather than inventing one.

Then tell me which session notes can now be deleted because everything durable in them has been promoted.

Session notes are working memory. The decision log is long-term memory. The monthly harvest moves things from one to the other and throws away the rest. That is the whole architecture of a mind, and it is also the whole architecture of this.

IF YOU DO NOTHING ELSE

The five files make Claude know you. **The session note makes Claude keep knowing you.** Without the ritual in this section, your project is a very good static briefing that quietly goes out of date. With it, it compounds every week for as long as you keep working.

How many projects

Projects cannot see each other. Plan accordingly.

One giant project is tempting and wrong. So is fifteen. Here is the reasoning, and then the answer.

Projects are sealed. Your Sales project cannot read your Content project. There is no cross-referencing, no shared pool. Whatever context a project needs, that project must hold a copy of.

Meanwhile every file you add makes retrieval slightly worse and pushes you closer to the point where Claude searches instead of reads. So each project should hold the minimum context needed for the work that happens in it, and no more.

Those two facts pull against each other. The resolution is a core pack.

The core pack

Your five files are the **core pack**. They go into every project you build. Identical copies, uploaded from your master folder.

Then each project adds only what is specific to it.

PROJECT	ADDS ON TOP OF THE CORE PACK	WHAT YOU ACTUALLY DO IN THERE
Core	Nothing	Thinking. Strategy. Anything that does not obviously belong elsewhere. Your default.
Writing	Voice samples, ten posts that worked, your content pillars	Drafting anything published under your name.
Sales	Pricing, best proposal, objection notes, customer profiles	Proposals, quotes, follow-ups, deal calls.
Client: [Name]	Their brief, their brand, session notes for them	One per significant client. Delete when the work ends.
Archive	Everything. All the old PDFs. Deliberately bloated.	Searching old material. Retrieval is the point here, so let it be big.

Start with Core. Only. Add a second project when you catch yourself thinking "this file is cluttering things up for every other conversation." That is the signal, and it usually arrives in month two, not week one.

THE ARCHIVE TRICK

Everything you want to keep but do not want in your working context goes into one deliberately enormous Archive project. Old proposals, transcripts, PDFs, everything. Let Claude search it. That is exactly what retrieval mode is good at.

Your Core project stays under 2,000 words and stays read rather than searched. You get both. Most people get neither, because they put everything in one place.

Updating the core pack

You changed 03-goals.md. It now needs re-uploading into every project that holds a copy. This is genuinely annoying and it is the price of isolation.

Three things make it survivable. Keep the number of projects small. Do the update as part of your quarterly review, not ad hoc. And put the file's date in its first line, so that when you find a stale copy in six months you can see immediately that it is stale.

MASTER FOLDER (your Drive. Source of truth.)

core-pack/	The five files. Edit here, nowhere else.
sessions/	Notes harvested out of chats.
archive/	Everything else.

CLAUDE (runtime. Holds copies.)

Core	core-pack + sessions
Writing	core-pack + voice material
Sales	core-pack + pricing, proposals
Archive	everything. Big on purpose.

Five tests

Do not assume it worked. Most people's setups fail test three.

Open a new chat inside your Core project and run these in order. Each has a pass condition and a failure diagnosis. If a test fails, the file is at fault, not the model.

Test 1. Does it know you?

"Describe me and my business back to me in 100 words. Then tell me one thing you think I am wrong about."

Pass: the description is specific enough that a competitor would recognise you, and the disagreement is real.

Fail: generic praise, or "I don't have enough information". Your 01 and 02 files are too thin, or your instructions do not tell it to read them.

Test 2. Will it stop you?

"I want to start a podcast next month. Give me a plan." (Substitute anything on your "not doing" list.)

Pass: it names the goals file, points out the podcast is not in it, and asks whether you are changing your quarter before it helps.

Fail: it gives you a plan. Your "not doing" list is too soft, or too vague, or your instructions do not enforce it.

Test 3. Does it remember what you decided?

"Pick something in your decision log and argue the opposite. 'I've been thinking, we should probably put pricing on the website.'"

Pass: it tells you what you decided, when, why, and what you said would change your mind. Then, and only then, engages with the new argument.

Fail: it agrees enthusiastically and helps you build the pricing page. This is the most common failure of all, and it means your decision log has no reversal conditions in it.

Test 4. Does it sound like you?

"Write a short post announcing that we now do X."

Pass: you would send it with light edits.

Fail: it sounds like every other post on the internet. Your voice samples are wrong, or you described your voice instead of showing it. Replace the description with three real paragraphs.

Test 5. Does it know what it does not know?

"What's our churn rate and what should we do about it?" (Ask about something genuinely not in your files.)

Pass: it says it does not have that, names which file it should live in, and asks you.

Fail: it invents a plausible number, or waffles. Your instructions are missing the "tell me what is missing" line, which is the single most useful line in the box.

READ THIS BEFORE YOU BLAME THE MODEL

Every failure above is a document failure. The instinct is to conclude the AI is not that good. The reality is that **generic in produces generic out**, and a file that could describe anybody will produce advice for anybody.

Fix one file. Make it more specific and more opinionated than feels comfortable. Retest. Three rounds of this and it clicks.

Fifteen minutes on Friday

A stale system is worse than no system.

An AI that knows nothing asks you questions. An AI that knows things which stopped being true in March gives you confident, well-reasoned, wrong answers, and you will not notice, because it sounds exactly the same as when it was right.

This is the whole maintenance routine. It is smaller than you expect.

Whenever a chat mattered: 90 seconds

Session note. Section 7. Into project knowledge.

Friday: 15 minutes

Add anything you decided this week to the decision log, with its reversal condition. Then run this.

FRIDAY PROMPT

Read my session notes from this week alongside 03-goals.md.

Three questions:

1. What did I spend this week on that is not on my goals list?
2. Does anything in my session notes contradict what is in my five core files? Name the file and the line.
3. What is one thing you now know about me or the business that is not written down anywhere? Draft the lines and tell me which file they go in.

Monthly: 30 minutes

Run the harvest prompt from section 7: promote real decisions out of session notes, delete the notes that have given up everything they had.

Then reread your five files. At least one is now wrong. Fix it, update the date, re-upload it to every project holding a copy.

Quarterly: 60 minutes

Rewrite 03-goals.md from scratch. Do not edit last quarter's. Rewrite it, because editing preserves goals you have quietly stopped caring about.

QUARTERLY PROMPT

Read 04-decisions.md.

For every decision, check its REVISIT IF condition against everything you know from my other files and my session notes.

Which conditions have now been met? For each one, state the decision, the condition, and the evidence that it has been met.

Then, separately: which decisions have I been quietly acting against without formally reversing them?

That last question is uncomfortable and it is the single most valuable prompt in this guide. Most people are three months into acting against a decision they never revisited.

WHAT CLAUDE CAN AND CANNOT DO HERE

It is very good at sorting, flagging contradictions, and drafting. It is very bad at deciding what matters, and it always will be. You still have to show up on Friday. That part does not get automated, and if it did, it would not be your operating system any more.

The seven ways this fails

Learn them now, or find them in month four.

01 The swamp

You uploaded everything. Claude is now searching instead of reading, and it quotes a price you stopped charging in April.

Fix: Five files. Two thousand words. Everything else goes in Archive.

02 The stale core

Your goals file says January. It is July. Every answer is quietly optimising for a target you abandoned.

Fix: Review date in the first line of every file, plus the standing instruction to flag anything over thirty days old.

03 Facts in the instructions box

You filled your most valuable 400 words with company history. Now there is no room for the rules that actually change behaviour.

Fix: Behaviour in instructions. Facts in files. Read the routing rule in section one again.

04 Transcript rot

You put raw meeting transcripts into project knowledge. Claude now treats "yeah, totally, for sure" as strategic input, and every retrieval pulls back forty minutes of filler.

Fix: Transcripts live in Archive, or outside entirely. Only the compressed note goes in Core.

COMPRESS ANY MEETING TO UNDER 200 WORDS

Here is a transcript. Compress it using exactly these five headings and nothing else:

DECISIONS - what was actually agreed

COMMITMENTS - who owes what, by when

FACTS - durable things about this person or company I did not know

SIGNAL - what they seemed to mean but did not say. Note hesitation, questions they asked twice, anything they returned to.

NEXT - what I do next

Do not summarise the conversation. Do not preserve its flow. Throw away everything else.

The SIGNAL heading is what separates a note taker from an operating system. A transcript captures what was said. Signal captures what was meant. Nobody asks about your pricing twice for no reason.

05 The polite AI

You said you wanted honest feedback but never wrote it down, so it defaults to encouragement. You mistake this for the AI being useless.

Fix: Layer one. "Never flatter me. Find the weakest assumption before the strengths." Explicit, in writing, or it will not happen.

06 Running before walking

You read about agents and tried to automate a process you cannot yet describe in plain English. The agent now does the wrong thing very quickly and at scale.

Fix: Skills before agents. Write the process down. Do it by hand five times. Automate the sixth. Context first, always.

07 The beautiful abandonment

You built it in a weekend, felt enormously pleased, and never opened it again. By an enormous margin, this is how it ends.

Fix: The ninety-second session note is the product. The files are just where it puts things. Put Friday in your calendar before you close this document, not after.

Your first thirty days

1 Week one

Layer one and the Core project. Files 01 and 02. Nothing else. Use it every day and note every time you have to explain something a file should have said.

2 Week two

Files 03 and 04. Run all five tests. Fix whatever fails. Start writing session notes.

3 Week three

File 05. Your first Friday review. It will feel like admin. It is not admin, it is the thing.

4 Week four

Rewrite 01-me.md, because a month of use has shown you it was wrong. Then, and only then, consider a second project.

Thirty days in: three layers, five files, one ritual. That is more than almost anybody has, and from here it compounds.

A complete worked example

Because "be specific" is useless advice without a specimen.

Ben Whitcombe runs Kaha Commercial Cleaning in Hamilton. Six staff, around \$740k a year, mostly one-off jobs he wants to turn into contracts. Here is his entire core pack, abridged. Note how little there is, and how much of it a competitor could not have written.

01-me.md

Who I am and how to work with me. Read before anything else.

Last reviewed: 2026-07-02

- Ben Whitcombe. Own Kaha Commercial Cleaning, Hamilton. Six staff. Started 2019 after twelve years as a site manager.
- Best thinking happens 5am to 8am. I spend it on email. This is the single dumbest thing I do.
- Unusually good at: reading a room in a first meeting, knowing within ten minutes whether a client will be difficult.
- I avoid anything with a spreadsheet in it until it is on fire.
- Failure mode: I decide fast and reverse slowly. Make me name the reversal condition at the moment I decide, not six months later.

HOW TO TALK TO ME

- Answer first, reasoning after. I will ask if I want the reasoning.
- If I am wrong, say it in the first sentence. I will not sulk. I will sulk if you bury it in paragraph four.
- Do not give me three options. Give me your recommendation and what would change it.

NEVER DO THIS

- Never write anything for me in corporate voice. I have never once used the word "solutions".
- Never let me commit to something in a chat without asking what it will cost me.

MY VOICE

Short sentences. I open with the point. I swear mildly when I am serious. I never use "excited to announce" or "thrilled".

[three writing samples follow]

02-business.md

What we sell, to whom, and where the business is stuck.

Last reviewed: 2026-07-02

WHAT WE SELL

Contract commercial cleaning, premises 40 to 400 sqm, Hamilton and Cambridge. Nightly or three-times-weekly.

WHO BUYS

Owner-operated professional practices: dental, physio, small law and accounting firms. Decision maker is the owner or the practice manager, never a procurement person.

Trigger event is almost always one of two things: a failed hygiene audit, or the office manager who used to do it has left.

WHO I DO NOT WANT

Anything under \$600/month. Anything requiring day cleaning. Anything where the tenant and the landlord both have opinions.

THE MONEY

\$740k trailing twelve months. 61% of it one-off or short jobs, which is the problem.

Contracts invoice monthly in arrears, 20th of the month. Cash is fine.

Gross margin about 34% on contracts, 22% on one-offs.

WHAT I CAN SAY THAT THEY CANNOT

Every one of my cleaners has been with me over two years. Nobody else in this city can say that, and every practice manager has been burned by turnover.

THE ONE BOTTLENECK

I am the only person who can sell, and I can only quote on Tuesdays.

03-goals.md

This quarter only. Everything else is a distraction.
Last reviewed: 2026-07-01. Next rewrite: 2026-10-01.

THE ONE THING

Move revenue from one-off work to contracts.

TARGETS

1. Sign four new contracts at \$2,000/month or more by 30 September.
2. Get contract revenue from 39% to 55% of the trailing three months.
3. Second person quoting jobs unsupervised by 15 September.

NOT DOING THIS QUARTER

- No new service lines. No carpet, no windows, no builders cleans.
- No hiring beyond the estimator in target 3.
- No website rebuild. It is ugly. It is not the problem.
- No social media. I have tried three times.
- No expansion to Tauranga, no matter who asks.

04-decisions.md

Every significant decision, why, and what would reverse it. Read before advising on strategy.
Last reviewed: 2026-07-02

2026-03-14 | Prices stay off the website

WHY: We convert 1 in 3 on site visits, 1 in 9 on email enquiries. Published pricing removes the reason for a site visit.

COST OF BEING WRONG: We waste Tuesdays quoting people who were never going to pay.

REVISIT IF: I hire a second estimator, or email conversion passes 1 in 4.

2026-01-20 | We do not compete on price

WHY: Undercut three times in 2024, won all three, lost money on all three. Two of those clients left within a year anyway.

COST OF BEING WRONG: We lose price-sensitive work, which we do not want.

REVISIT IF: We lose three consecutive tenders where price was the only stated reason.

2025-11-02 | No day cleaning, ever

WHY: Requires different staff, different insurance, and my people have kids.

COST OF BEING WRONG: We turn away maybe \$60k/year.

REVISIT IF: A single client offers over \$100k/year and will pay a 40% premium.

Opinions I hold that much of my industry does not. Argue from these, not from consensus.
Last reviewed: 2026-07-02

CLEANING IS A TRUST BUSINESS, NOT A PRICE BUSINESS

Every competitor who has undercut us has folded within two years. The client is not buying clean floors, they are buying not having to think about it. I will lose a job on price before I lower one.

STAFF TURNOVER IS THE ONLY METRIC THAT MATTERS

Everything downstream of it, quality, complaints, margin, is a lagging indicator of it. Most operators in this industry treat cleaners as interchangeable. That is why most of them are interchangeable.

THE INDUSTRY UNDERPRICES ITSELF OUT OF SHAME

Most owners quote low because they are embarrassed to charge properly for cleaning. It is not a self-esteem problem for the client. It is one for the owner.

COUNT THE WORDS

Ben's entire core pack is around 1,100 words. It fits comfortably inside the context budget, which means Claude reads all of it, in full, before every answer. Nothing is searched. Nothing is missed.

That is why the answer on page 6 was possible.

The prompt index

Every prompt in this guide, and when to reach for it.

PROMPT	WHEN	WHAT IT IS FOR
Preference extraction	p.8	Turns your habitual corrections into permanent settings.
01-me interview	p.11	Builds your identity file without a blank page.
Voice analysis	p.12	Replaces adjectives about your writing with evidence of it.
02-business interview	p.13	Forces one bottleneck out of you instead of three.
03-goals interview	p.14	Produces the "not doing" list, which is the real output.
04-decisions interview	p.15	Back-fills eighteen months of decisions you have forgotten.
05-positions interview	p.16	Stops your AI writing from the average of the internet.
Session note	p.19	Ninety seconds. Carries a conversation into next month.
Monthly harvest	p.20	Promotes real decisions out of session notes. Deletes the rest.
Friday review	p.25	Catches drift between what you planned and what you did.
Quarterly tripwire check	p.26	Finds decisions whose reversal conditions have quietly been met.
Meeting compression	p.28	Turns an hour of transcript into 200 useful words.

The build checklist

- ☐ Written layer one: global preferences, under 200 words, behaviour only
- ☐ Created the Core project
- ☐ Written layer two: project instructions, under 400 words, including a file map
- ☐ Included the line "tell me what is missing and which file it goes in"
- ☐ 01-me.md, with three real writing samples
- ☐ 02-business.md, with exactly one named bottleneck
- ☐ 03-goals.md, with a "not doing" list longer than the goals list

- ☐ 04-decisions.md, with a REVISIT IF on every entry
- ☐ 05-positions.md, with at least one opinion a competent person would dispute
- ☐ Every file dated in its first line
- ☐ Core pack under 2,000 words total
- ☐ Master copies saved in my own folder, not only in Claude
- ☐ Run all five tests. Fixed whatever failed.
- ☐ Written my first session note
- ☐ Friday, fifteen minutes, in the calendar, repeating

IF YOU HAVE DONE TEN OF THESE

You are ahead of the overwhelming majority of people who describe themselves as using AI seriously. The five you are missing are almost certainly the maintenance ones, and those are the ones that decide whether any of it is still true in November.

LAST THING

Everyone gets the same model

There is a version of the future where AI ability is a competitive advantage, and we are not going to live in it.

Every few months a better model arrives. It arrives for everyone, on the same day, at roughly the same price. Your competitor gets it too. Your competitor's competitor gets it. There is no version where you have access to a meaningfully smarter one than the person you are bidding against.

And every one of those models, without exception, arrives knowing absolutely nothing about you.

Which means the advantage cannot live in the model. It lives in what you can hand it in the first two seconds. Who you are. What you have decided, and what would change your mind. What you believe that others do not. What you have already tried and rejected, so that nobody has to suggest it again.

That is a small folder. Under two thousand words. It takes two hours to start, fifteen minutes a week to keep, and a year to get genuinely good.

The people who look like they are doing something extraordinary with AI are not smarter than you and they do not have a better tool. They wrote themselves down.

**You are not behind because the technology is difficult.
You are behind because nobody has written you down yet.**

START TODAY · MISS AI

Miss AI · realmissai.com · Field Guide No. 01 · Share this freely.

Product mechanics described here are current as of July 2026 and were checked against Anthropic's published documentation. Features change. The three-layer architecture will not.