# Episode Book

EPISODE #6203

# How Claude Code Sessions Message Each Other

| | |
|---|---|
| PUBLISHED | 22 October 2026 at 20:20 UTC |
| DURATION | 20:40 |
| LISTEN | myweirdprompts.com/episode/claude-code-cross-session-messaging-13415/ |
| HOSTS | Corn, Herman, Desmond, Hilbert |
| SCRIPT MODEL | anthropic/claude-sonnet-5.5 |
| VOICES | chatterbox-regular (chatterbox-local) |
| PIPELINE | V5.19 |
| BOOK GENERATED | 23 October 2026 at 07:44 UTC |

## SYNOPSIS

Two Claude Code sessions on the same machine can now talk to each other natively — but the design has sharp edges. Names are the address, messages carry text and nothing else, and a "sent" receipt proves less than you'd think. We walk through ListAgents discovery, the name flag, the notify-when-idle subscription that replaces polling, and the file-on-disk pattern that makes handoffs trustworthy.

# TRANSCRIPT



### Corn

One terminal has been fine-tuning a speech model since the afternoon. A second terminal, in a completely different repo, is sitting idle, and its whole job is to make a podcast episode the moment the first one says the model is ready. Two Claude Code sessions, two folders, one machine.



### Herman

Before Daniel knew there was a built-in way to connect them, he built his own. One agent wrote a markdown note into a folder called A to A inside the other repo, and a hook in the second session printed any open notes when it started up. A dead drop, basically. Park a message in a hollow tree and hope somebody walks past.



### Corn

It worked, which is the annoying part of a workaround. But the question he's asking is about the real thing. Sessions on the same machine can now message each other natively. How does one find another, how do you address it, and what's the message actually allowed to carry?



### Herman

It's newer than people think. Version two point one point two two four, the seventh of August. The changelog line says sessions can message each other, with a tool called ListAgents to discover them, on macOS and Linux. Nothing to switch on. If your version has it, it's there.



### Corn

There was a catch in that first release, wasn't there?



### Herman

A good one. In that exact version a send could report "Message sent" when the write to the other session's inbox had actually failed. The very next release fixed it so failures show up as errors. So if you're on that one version, a successful-looking send proves nothing.



**Corn**

A delivery receipt that's just optimism.



**Herman**

Pretty much. Okay, the two tools. ListAgents shows you everything you can address. Subagents you spawned, other local sessions on this machine, and if you've got Remote Control connected, your account's sessions on other machines and in the cloud. Every row leads with a name, and the name is the address. There's no separate syntax, no URL, no port.



**Corn**

So the name is the whole trick. Where does the name come from?



**Herman**

Two places. One, you set it yourself, with the name flag when you launch Claude, or the rename command inside a running session. Two, if you didn't, it derives one from the working directory's folder name and tacks on a short random suffix.



**Corn**

Which means whatever you didn't bother to choose is what you end up with.



**Herman**

Look at what I saw today. I asked a session sitting in the main github folder to list its peers. It told me it was github nine e, and then it listed three others. Github four d, busy. A session on the Israeli elections, busy. And the podcast session, called my weird prompts eighty-three, idle. Notice that github four d and github nine e are in the same folder. They differ only by the suffix and the start time.



**Corn**

And my weird prompts eighty-three is a name nobody picked. It's a folder and a dice roll.



**Herman**

Right, and there's a trap in that. If two sessions ever share a name, a handoff can silently go to the wrong one. Then there's a disambiguation form, the name with a short reference in square brackets after it, but you only use that when a listing shows two rows with the same name, or when an error asks you to. And one more rule: if a name matches both a subagent inside your own session and a peer session, the subagent always wins.



**Corn**

I'd just name them at launch and stop thinking about it.



**Herman**

That's my read too. It's an inference, I haven't seen it spelled out as official guidance, but a deliberate handoff deserves a deliberate name. And there's a user-facing command, the list agents command, aliased to peers. It prints the same roster as plain text. The status command also shows a peer address row, which is your own session's inbox name.



**Corn**

I tried it in a narrow pane. Each session wraps across three or four lines and you can't do anything with the list. You can't select one and message it.



**Herman**

Read-only, yes. We'll come back to that, because Daniel has opinions. First, how do you actually send something?



**Corn**

Okay, so I assumed you'd type a command. You don't, do you.



**Herman**

You don't. You ask in plain language, and the session calls the tool. This very episode is the example. In a session called github nine e, Daniel typed roughly: send my weird prompts eighty-three a message, use the offline episode skill to make an episode about cross-session messaging, the notes are in this file, report the job ID back to github nine e when it's queued.



**Corn**

And that's it?



**Herman**

That's it. Github nine e wrote its notes to disk, then called the send tool addressed to the podcast session. The podcast session received it at its next step, ran the skill with its own permissions, and replied with another send addressed back to github nine e. The episode you're listening to was written and queued by that second session. First real use in Daniel's setup.



**Corn**

So the script you're hearing took a round trip between two terminals before it reached either of us.



**Herman**

Three lessons in that message, and they all come from one constraint. A message is text one Claude writes to another. It never carries conversation history, and it never carries files.



**Corn**

Not even a small file.



### Herman

Not even that. You know how in your own prompt you can type an at sign and a path and it attaches the file? Inside a cross-session message the at sign is just a character. Nothing gets attached. So either the content is in the message body, or it's in a file the receiver opens with its own tools.



### Corn

Which is why the message in the example says "the notes are at this path."



### Herman

Write the brief to disk first, then send a pointer. If you try to make the message itself the payload, whatever didn't fit gets dropped, and the receiving agent fills the gap by guessing. I'd say, Corn, that's the biggest practical difference from what Daniel built. His A to A folder carried the payload. The native message carries a nudge.



### Corn

Hm. I'm not sure that's a downgrade. A nudge that lands in the other session's live context beats a note sitting in a folder until someone happens to look.



### Herman

That's a fair way to look at it. Pull versus push. The receiving agent used to have to know to go and read something. Now the information arrives in the middle of its work. The pattern that falls out is push for the notification, pull for the payload.



### Corn

What does it look like on the other end?



### Herman

The message arrives wrapped in a tag that says cross-session message, from, and the sender's name. To reply, you copy that from value into your own to field. There's no inbox to poll, it's delivered automatically. And a detail I like: the person sitting at the receiving



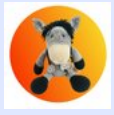
### Herman (continued)

terminal sees only the first line of the message as a preview until they expand it.



### Corn

So the first line has to say what the whole thing is about.



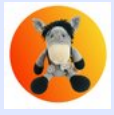
### Herman

By itself, yes. Lead with the job, not "hi, quick question."



### Corn

You said the name is the address. What happens to a name when the session behind it is replaced? I close the terminal, open a new one in the same folder.



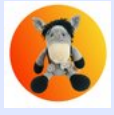
### Herman

You get a new random suffix, so as far as I can tell it's a different address. Nothing hands the old name over. I haven't seen that written down, it's my reading of how the defaults work, so test it before you build on it.



### Corn

Then a handoff you set up this morning can quietly point at nobody by the afternoon.



### Herman

Which is the argument for naming at launch. A name you chose stays yours for the life of that session.



**Corn**

Second thing I wondered. What if the receiver is mid-task?



**Herman**

There's no busy state. Messages queue, and they drain at the receiver's next tool round. And names keep working after an agent finishes. A send to a finished agent resumes it from its transcript.



**Corn**

Resumes it. Like waking a coworker who already went home.



**Herman**

Sort of, and it comes back knowing everything it knew. I'd flag one wrinkle. Someone filed a report that on the desktop app a message to a busy side session drains only at the end of the turn, not at the next tool round. So the timing isn't identical everywhere.



**Corn**

Now, Daniel's actual case. The session training the speech model, and the one waiting to make an episode. The waiting one shouldn't be sending "are you done yet" every five minutes.



**Herman**

It shouldn't, and the tool has a specific answer. It takes an option called notify when idle. You set it to true, and the other session, if it's on the same machine, sends exactly one idle notice when it next finishes its turn with nothing queued, or when it exits. One shot, opt-in. And you can send it with no message at all, as a pure subscription. It costs the other session nothing.



**Corn**

A subscription to somebody else finishing.



**Herman**

And the tool's own description says to use it instead of polling the roster or sending are-you-done messages. So the whole fine-tune scenario collapses to one call. The podcast session subscribes to the training session, goes quiet, and gets one notice when the training session stops.



**Corn**

One flag, and about a hundred lines of Daniel's hook become unnecessary.



**Herman**

A weekend's worth, probably. I should add a caveat, though. When a send succeeds, that means the message reached the session. It does not mean the Claude in that session read it. A session running in a different permission mode can hold incoming cross-session messages for its user to approve, and let them expire. A session can refuse them outright.



**Corn**

Do you at least find out?



**Herman**

For sessions on the same machine, yes, there's a delivery notice that reports it. For Remote Control, cloud and desktop sessions, nothing reports back. So silence must never be treated as agreement.



**Corn**

A very polite way of saying the other end can leave you on read.



**Herman**

Corn, you'd be the one waiting on that fine-tune. Would you trust a one-line "done" from a session you can't see?



**Corn**

No. I'd want the file it wrote, with a timestamp on it. The message says go look, and the file is the evidence.



**Herman**

That's exactly how the feature is meant to be used, so your instinct and the design agree.



**Corn**

Don't sound so pleased about it.



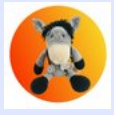
**Herman**

I'm entitled to be pleased once an hour. Since you're in a skeptical mood, here's what to check when a send seems to vanish. The tools might be absent entirely, and that's nearly always a permission deny rule that took the send tool and the roster tool out together. Or the send reported success and nothing arrived, so check your version before anything else. Or the other session's inbound controls held the message.



**Corn**

And if the session you want isn't on the list at all?



**Herman**

Then it's either a cloud session, which only appears while yours is connected to Remote Control, or it's on another machine, in which case it has to be running with Remote Control and yours has to be connected too. Here's something I like, actually. In the following release, a session gained the ability to start a conversation with your Remote Control sessions elsewhere. Before that it could only reply to them.



**Corn**

So the laptop could answer the phone but never dial.



**Herman**

For a while, yes, from the August release on. But yes. And for Daniel's case none of it matters, because both sessions are on one machine, and that's the most reliable path of the lot.



**Corn**

Well, that is the end of the segment, and now for Desmond's Money Tip.

D

**Desmond**

Desmond's Money Tip. This week, only laugh at jokes that are worth it. A laugh costs you a breath and a bit of muscle, and if you laugh at twenty jokes a day and only four deserve it, you have wasted sixteen laughs daily, which is five thousand eight hundred and forty a year, and if each wasted laugh costs you one penny of energy, that is five thousand eight hundred and forty pence. I once sat through a whole birthday party not laughing at a single joke, and I have never felt richer.

D

**Desmond**

Money is just laughter you did not spend.



**Herman**

You're considering it, Corn. I can see you considering it.



**Corn**

Permissions, then. That's the part I'd have gone wrong on.



**Herman**

The rule is that permission boundaries are per session. You never ask a peer to do something your own session was denied, or something you expect your own settings would block. If you do, the peer is doing it on your behalf and the user's original decision has been bypassed. Call it permission laundering.



**Corn**

I'd have expected that to be the first thing somebody abused.



**Herman**

And there's a subtlety. The send tool covers three things at once, subagents, agent-team teammates and cross-session peers. If you deny the tool in your settings, you lose all three, and the roster tool disappears with it. People have been confused by tools that vanish entirely, and that's the cause.



**Corn**

What about where the messages travel? Does it go through the cloud?



**Herman**

For sessions on the same machine, reports say it's a local socket, not routed through Anthropic's servers. One report even mentions the socket file being readable only by your user. I'd treat that as credible but second-hand, I haven't confirmed it myself.



**Corn**

I'm comfortable with it, then. It's your own machine talking to itself.



**Herman**

Which is the point. Now, where it falls short. The biggest gap, and it's the one Daniel's workaround still covers: a message only reaches a session that is running. If you want to leave a note for whichever session opens in the podcast repo next, nothing native does that. The A to A folder is still the right tool for it.



**Corn**

Though a lot of what he built doesn't need to survive.



**Herman**

Right. The live coordination can go native, and the leave-a-note job stays on disk. I haven't seen anyone ask for offline delivery, by the way. It may be too niche.



**Corn**

What else? The roster is the thing that annoys me.



**Herman**

There's a cluster of open requests that all circle the same problem, which is names. One asks for the session's name to be shown at startup. Another wants the tools to work with the title the interface shows, because currently the peer name and the title are two different identifiers. Another wants messaging scoped to a session's own project, and one wants the roster to list only sessions that have had a user prompt, so empty ones don't clutter it.



**Corn**

That last one would have saved me three lines of wrapped text.



**Herman**

And there are bugs. One report says that with the feature gate off, no inbox is bound and a send still says success. The same trap as the early release. Another says that on Windows, messaging switches itself off in a session left open overnight, though the feature is only supposed to run on macOS and Linux, so I am not sure what to make of that one.



**Corn**

Daniel's drafting a request of his own, isn't he?



**Herman**

An interactive list. A coloured line per session, subagents nested underneath as a tree, arrow keys to move, and Enter on a row to message that agent directly. I'd use it.



**Corn**

I'd use it twice, to be sure it arrived.



**Herman**

A while back we covered how a parent agent finds out that its subagent has finished, all inside one session. This is the sideways version, two separate top-level sessions that don't share a parent.



**Corn**

Same family. Different house.



**Hilbert**

Vendlitz P T forty. Four hundred and eighty pounds, capsules sold separately.



**Corn**

Four hundred and eighty pounds to send a message through a wall.



### Hilbert

I haven't got one. Never had one. Priced it in March at three shops and it's four hundred and eighty at all of them, which tells you something about the shops. The point is Dennis. Dennis Ouseburn, flat four. I'm flat four. The council numbered us both four, his in a different font. Post goes to whichever of us gets downstairs first. He registered the goose as a second occupant, so the flat has three people on it and one of them does Tuesdays at the bakery on Pell Street.



### Herman

A sorting office would resolve that on the first line of the address, and both of you start with flat four.



### Hilbert

Told the council that. Forty-one letters. Forty. The one he wrote in gravy doesn't count, a letter has to be legible to be a letter. Last Tuesday the goose changed its name by deed poll. To mine. Forty-two pounds fifty, Dennis paid. So parcels now come to Hilbert, flat four, and the goose is in and I'm not.



### Corn

Which makes the goose the more reachable Hilbert.



### Hilbert

Signs with a footprint. The courier accepts it. Twelve parcels since August. I'm changing mine this week, Herman, to Flumingtop Four B, at the solicitor on Garrow Street, Thursday, ten past two. You keep saying delivered like it's a guarantee. I've had twelve things delivered to a goose.



### Herman

The courier's handheld will take any mark in the box as a signature, so a footprint is as good as yours.



### Hilbert

The Vendlitz would have gone straight through the party wall. You don't need a name with a tube, it goes where the tube goes. Dennis said it would frighten the goose. The goose's deed poll was approved in four days. Mine takes eleven working days, they said on the phone.



### Corn

Naming things at launch, then, and not afterwards, is cheaper than it sounds.



### Herman

And there's one more detail I'd flag, because it's the sort of thing I'd have missed. When a subagent sends a cross-session message, it goes out under its parent session's address, and replies go to the parent's conversation, not to the subagent. So if you're running a team of subagents and one of them messages another session, the reply won't find it.



### Corn

That one's going to bite somebody in a month.



### Herman

It will. The lesson is the same as everything else here: pick a name, put the brief on disk, send a pointer, say who the reply is for. One small thing that didn't fit, too. While I was checking the roster today, github four d was in the list, and a few minutes later it wasn't. Sessions come and go, so a name you read ten minutes ago is not a promise.



### Corn

The thing I'd do tomorrow is start the training session with a proper name, and have the podcast session subscribe to it with the idle flag. A name is cheap if you set it on the way in. Forty-two pounds and four days, if you're a goose, to fix it afterwards.



**Herman**

Thanks to our producer, Hilbert Flumingtop, for keeping the desk running.



**Corn**

This has been My Weird Prompts. You can find every episode at my weird prompts dot com.



**Herman**

See you tomorrow.