

● PLAYBOOK

The Mac Dictation Playbook

45 concrete voice workflow recipes, the habits that make push-to-talk stick, and a ten minute setup checklist. Works with any dictation tool on your Mac.

By The Rambleproof team. Free for everyone, no signup wall.

● SETUP

Ten minutes, once.

01

Pick a hotkey you can hold without looking

Push-to-talk lives or dies by the key. Choose a chord your left hand finds blind, so your eyes never leave the text field.

02

Do one 15 second test ramble

Open Notes, hold the key, describe your day badly on purpose. You are checking the microphone level and learning how release feels, not producing anything.

03

Teach it your words

Names, products, acronyms and project jargon are where every dictation tool stumbles. Add them to your tool's dictionary, or to macOS text replacements as a fallback, before you need them.

04

Check the input device

System Settings, then Sound, then Input. A headset mic beats a laptop mic in a noisy room; the laptop mic is fine in a quiet one. Watch the level meter while you speak.

05

Decide where you will dictate first

Pick two apps, usually mail and your AI tool, and use voice only there for a week. Spreading thin across every app at once is how the habit dies.

06

Learn your tool's hands-free option

Holding a key suits short thoughts. For meeting notes or long drafts, find the lock or toggle mode so your hands are free.

07

Put the status somewhere visible

Know at a glance whether the mic is live. Menu bar icon, HUD, or overlay: pick one and trust it.

● HABITS

Push-to-talk that sticks.

Speak the thought, not the sentence

Trying to compose polished sentences out loud defeats the point. Say it messy and let the cleanup pass, or a quick edit, do the polishing.

Restart out loud

Saying "no wait, Friday" and continuing is faster than stopping, deleting and starting over. Correct yourself in the stream.

Hold until the thought ends

Releasing mid-thought splits one idea across two insertions. Keep the key down through the pause while you find the next word.

Dictate the draft, type the edit

Voice is for getting the material out. The keyboard is for the five percent of surgery afterwards. Do not fight small word swaps by voice.

The two sentence rule

If a reply will take more than two sentences, it is faster to say it. Under that, just type.

Slow down for the brittle bits

Numbers, email addresses and URLs deserve a slower, clearer beat. Everything else can be full speed.

One idea per dictation

Notes apps reward small atomic captures. Release, breathe, hold again for the next idea instead of one giant blob.

Warm up once a day

The first dictation of the day always feels strange. Burn a throwaway sentence in Notes and the rest of the day flows.

● RECIPES

Prompting AI tools

Voice shines with AI assistants because good prompts are long and typing them is a tax. These patterns work in any chat or agent tool.

The context dump

"I am trying to get the login form to validate on blur. I already tried adding an onBlur handler and it fires twice. Look at the form component first and tell me why before changing anything."

Goal, what you tried, where to look, in that order. Thirty seconds of talking replaces five minutes of typing the same context.

The constraint sandwich

"Write a product update for the beta group. Constraints: under 150 words, no exclamation marks, plain language. Output as a single paragraph I can paste."

Goal, constraints, output format. Saying constraints out loud is where dictation beats typing: you would have trimmed them to save keystrokes.

Rubber duck debugging, recorded

"Here is what confuses me. The request succeeds in curl but fails from the app with the same headers, which should be impossible, unless the"

client rewrites something."

Explaining a bug aloud to the AI often solves it before the answer comes back. The transcript doubles as your notes.

Structure my ramble

"I am going to think out loud about our onboarding problem. Turn what I say into a short list of the distinct problems I mention, ignore the repetition."

Tell the model up front that a ramble is coming and what to do with it. Then genuinely ramble.

Meeting to action items

"From what I am about to say, extract action items with an owner and a date each. We agreed Sara ships the pricing page Friday, I review the copy Thursday night, and legal sees it before it goes live."

Dictate from memory right after the call while it is fresh. The model formats; you just talk.

Steelman both sides

"Give me the three strongest arguments for rewriting in Swift and the three strongest against, then say which side you find stronger and why."

A decision prompt you will use weekly. Say the options; do not prejudge them in your phrasing.

Read the error aloud

"The build fails with: cannot find type Publisher in scope, in AudioEngine line 40. It started after I removed the Combine import. What is the minimal fix?"

Reading an error message is faster than copying it when you are in a terminal or a screenshot. Include where and when it started.

The spoken spec

"New setting: retention. As a user I want dictation history deleted automatically after a period I choose, including audio, so old recordings never pile up. Default 30 days. Write acceptance criteria."

User story plus one default. The model expands it; you stay at the level of intent.

Steer without retyping

"Keep the structure and the facts, change the tone to something a tired engineer would actually read, and cut it by a third."

Follow-up steering is the highest value per word you will ever dictate. Name what stays, then what changes.

The background briefing

"Before you answer, background: we are a two person team, the app is a paid Mac utility, we do not want a server. Now, how should we do license validation?"

Say the context you always retype. Better: dictate it once into a note and reuse, appending the fresh question by voice.

● RECIPES

Email

Most email time is not thinking, it is typing what you already decided. Say the decision instead.

The three beat reply

"Thanks for flagging this. Yes, the invoice amount is wrong, our mistake. I will send a corrected one by tomorrow morning."

Acknowledge, answer, next step. Nearly every reply fits three beats, and three beats fit one breath.

The scheduling email

"Happy to meet. I can do Tuesday after 2 or Thursday morning. If neither works, send me two slots and I will make one fit."

Two concrete options plus a fallback ends scheduling ping-pong in one message.

The polite no

"Thank you for thinking of me. I have to pass this time because the timeline conflicts with our release. If it moves to next month I would genuinely like to help."

Appreciation, clear no with a reason, honest alternative. Saying it out loud keeps the warmth typing tends to lose.

The nudge

"Quick nudge on the contract I sent last Monday. One signature is all we need to start the pilot. Is anything blocking it on your side?"

Reference, one line ask, an opening for the real obstacle. Twenty seconds of voice, no guilt spiral about phrasing.

The cold intro

"I build a small dictation tool for Mac developers. Your post about typing pain after surgery made me think you might want to try it. One line back and I will send a build, no list, no follow-ups."

Who you are, why specifically them, one tiny ask. Dictating keeps it human; over-edited cold email smells like cold email.

The bad news email

"The launch slips one week. Cause: the notarization issue we found Monday needs a fix from upstream. Plan: fix lands Wednesday, we test Thursday, new date is the 14th. Questions welcome."

Fact, cause, plan, date. Saying it plainly beats a typed paragraph of cushioning.

The after-call summary

"Writing down what we agreed: we drop the annual plan for launch, Sam owns the pricing page, I own the announcement, both due Friday. Reply if I misremembered anything."

Dictate it within five minutes of hanging up. The last line turns your memory into a shared record.

The specific thank you

"Thank you for the review, specifically for catching the retain cycle in the recorder. That one would have shipped. I owe you a coffee."

Name the specific thing and its effect. A generic thanks is typed; a real one is easier said.

● RECIPES

Slack and chat

Chat rewards short, frequent, low-ceremony messages. That is exactly what a held key produces.

The async status

"Status: shipped the settings screen, now on the importer, blocked on the API key from finance, need it by Thursday to stay on track."

Done, doing, blocked, need-by. Four beats, one hold of the key, no meeting.

The question with homework shown

"Has anyone seen the build cache ignore changed assets? I already cleared derived data and bumped the version, no luck. Next I would try disabling the cache entirely unless someone knows better."

Showing what you tried gets answers instead of guesses, and saying it takes seconds.

The decision request

"Need a call on the trial length: A is 7 days, B is 14. I lean B because setup takes a weekend. Objections by tomorrow noon, otherwise I go with B."

Options, your lean, a deadline that defaults to action. Dictate it and the decision makes itself.

The tactful disagree

"I am with you on the goal. My worry is that moving the button will break muscle memory for existing users. Could we A B it for a week before committing?"

Agree with the goal, name the concrete concern, propose a test. Tone survives better spoken than typed.

The handoff

"Handing off the importer: parsing works, progress UI works, error states are stubbed. Next step is wiring retry. Everything lives in the importer branch, notes in the PR description."

State, next step, where things live. The person picking it up starts moving instead of asking.

The end of day brain dump

"Open loops before I log off: waiting on the icon export, the crash from this morning is not reproduced yet, and I promised Ana a review that I will do first thing tomorrow."

Say your open loops into the channel or a note. Tomorrow-you starts with a map instead of a fog.

The long thread reply

"My take after reading the thread: option two, because it fails loudly instead of silently. Happy to disagree and commit if the group prefers one."

One line stance plus one reason. Resist summarizing the thread back; everyone read it.

● RECIPES

Docs and notes

Voice is a first-draft machine. The trick is separating the talking pass from the shaping pass.

Headers first, then fill

"Problem. Proposal. Risks. Rollout. Open questions."

Dictate just the section headers of the doc, then hold the key again inside each one. Structure first stops the ramble from sprawling.

Live decision minutes

"Decision: we keep the beta free until the importer ships. Decided by the three of us, today, revisit if support load doubles."

Use your tool's hands-free mode in meetings you run, and speak decisions as they happen. Nobody minds; everybody benefits.

The two minute braindump

"Everything in my head about the pricing page, in no order: the hero number feels high, we have no annual story, the FAQ promises a trial we removed, screenshots are from two versions ago."

Announce the topic, ramble for two minutes, stop. Reorder the mess afterwards by keyboard. Do not edit while talking.

The daily three lines

"Went well: shipped the fix before lunch. Learned: the flaky test was a timezone. Tomorrow: start the migration before checking messages."

Three fixed prompts make a journal you will actually keep, because it costs 20 seconds of speech.

The idea inbox

"Idea: the onboarding checklist could live in the menu bar until completed, then remove itself."

One idea, one dictation, one note titled Inbox. Capture beats organizing; sort weekly.

Margin notes while reading

"Reacting to chapter 4: the claim that habits need 21 days contradicts the study he cited earlier, check that footnote."

Speak reactions while the book or article is still open. Typed notes wait until you have a keyboard; spoken ones happen now.

The verbatim quote catch

"Quote from the user call, as close as I can recall: I do not trust tools that fix my words when I cannot see what changed. That is the landing page headline somewhere."

Right after a call, capture the phrasing you never want to lose. Memory rots in minutes; speech is instant.

The decision log entry

"Decision log. Context: two crash reports from the same Mac model. Options: block the model, or ship the workaround. Chose the workaround because blocking punishes working machines. Revisit if reports pass ten."

Context, options, choice, revisit condition. Your future self, six months out, will thank the you who said this into a note.

● RECIPES

Around code

You should not dictate code. Everything around code, though, is prose that you currently type.

Commit messages that explain why

"Guard the audio session against device changes. Unplugging a headset mid-recording used to kill the session; now we rebuild the tap and keep going."

Say what changed and why it matters. Spoken commit messages come out in sentences instead of grunts.

The PR description

"Problem: silence was inserted as text on some machines. Approach: gate insertion on a confidence floor and fall back to the raw transcript. Testing: 40 silence samples, zero insertions, plus the normal corpus."

Problem, approach, testing. Dictate it while the diff is fresh, before the context evaporates.

The review comment

"Observation: this retries forever on a 401. Question: is that intentional for token refresh? Suggestion: cap at three attempts and surface the error, happy to be wrong."

Observation, question, suggestion. The spoken version is naturally less prickly than the typed one.

TODO with a future

"TODO: this timeout is a guess. Replace with the measured p95 from the beta telemetry once we have two weeks of data."

A TODO without context is a trap for whoever finds it. Saying the context costs nothing.

The bug report

"Steps: start a recording, unplug the mic, plug it back in. Expected: recording continues. Actual: the session ends silently. Every time, on macOS 26, MacBook Air M4."

Steps, expected, actual, environment. Dictated while reproducing, it is accurate; typed later, it is fiction.

Name it out loud

"Candidates: CleanupGuard, FidelityGate, MeaningLock, TranscriptShield. Hearing them, FidelityGate says what it does without drama."

Names are heard in code review anyway. Saying five candidates aloud picks a winner faster than staring.

● RECIPES

Planning and life

The lowest friction wins for anything you were never going to sit down and type.

The weekly review, spoken

"Week review. Biggest win: the demo works. Biggest drag: two days lost to signing issues. Next week matters most: ship the beta invite email."

Three fixed questions, answered out loud on Friday. Fifteen minutes of typing becomes ninety seconds.

Tomorrow's top three

"Tomorrow: finish the importer error states, reply to the App Store rejection, book the dentist."

Last dictation of the day, straight into tomorrow's note. Closing the laptop on a spoken plan beats a vague dread.

The walking capture

"Capture: the onboarding video should show a real ramble, not a script. Also, check if the domain renewal hit the card."

With a headset, thoughts on a walk stop being lost. One hold per thought, sorted when you are back.

The list by room

"Hardware store: sandpaper 120 grit, wood filler, two brushes. Kitchen: descaler, bin bags, the good coffee."

Group by where you will stand when you need it. Say the group name first and the list sorts itself.

The trip dump

"Trip constraints: five days in October, direct flights only, budget around a thousand, one museum day maximum, my partner vetoes hostels."

Constraints first, wishes second. Paste the transcript into any planning tool, or hand it to an AI with recipe one.

The pre-meeting minute

"Before this call: I want a yes or no on the co-marketing idea, I will offer the newsletter swap first, and I will not commit engineering time."

Sixty seconds before any meeting, say what you want out of it. Meetings obey the person who arrives with a sentence.

The recipes work with any tool.

We wrote them while building Rambleproof, a push-to-talk dictation app for Mac that cleans up the ramble and never changes your meaning. Free while in beta.

[See how Rambleproof works](#)

[Join the waitlist](#)

Rambleproof

Ramble in. Clean text out. Push-to-talk dictation for Mac, processed on your Mac. Made by Zena Labs.

Requires macOS 26 on Apple Silicon.

Site

[How it works](#)

[Proof](#)

[FAQ](#)

[Join the waitlist](#)

[The Mac Dictation Playbook](#)

Compare

[Wispr Flow vs Rambleproof](#)

[Superwhisper vs Rambleproof](#)

[Best dictation app for Mac](#)

[All guides](#)

Company

[Contact](#)

[Privacy](#)

