How to combine Claude Code with Obsidian CLI

If you've been using Obsidian for note-taking for over a year, your archives are probably a mess. Try giving the problem to Claude Code, combine it with a few Obsidian CLI commands, and watch it accomplish what you haven't been able to do in years, in about 90 minutes.

After years of users having to live with a chaotic repository, Claude Code has done something no other plugin has been able to do.

If you've been using Obsidian for note-taking for over a year, your archives are probably a mess. Try giving the problem to Claude Code, combine it with a few Obsidian CLI commands, and watch it accomplish what you haven't been able to do in years, in about 90 minutes.

What is the Claude Code? Why is it important for note-taking?

An AI that operates in the terminal, not the chat box.

If you didn't know, Claude Code is Anthropic's command-line AI tool. Unlike the traditional Claude interface, Claude Code runs directly in the terminal, reading files from the local system, writing and editing code, and performing multi-step tasks automatically without you having to manage each prompt.

That final point is what sets Claude Code apart. Most AI tools require your constant interaction and information exchange. Claude Code can be given a goal, granted access to a directory, and trusted to systematically solve the problem on its own. It views your file system like a developer, not like a chatbot.

For a task like reorganizing a notes archive, that distinction is incredibly important.

How to connect Claude Code to the Obsidian vault

Why is notesmd-cli the right tool for this job?

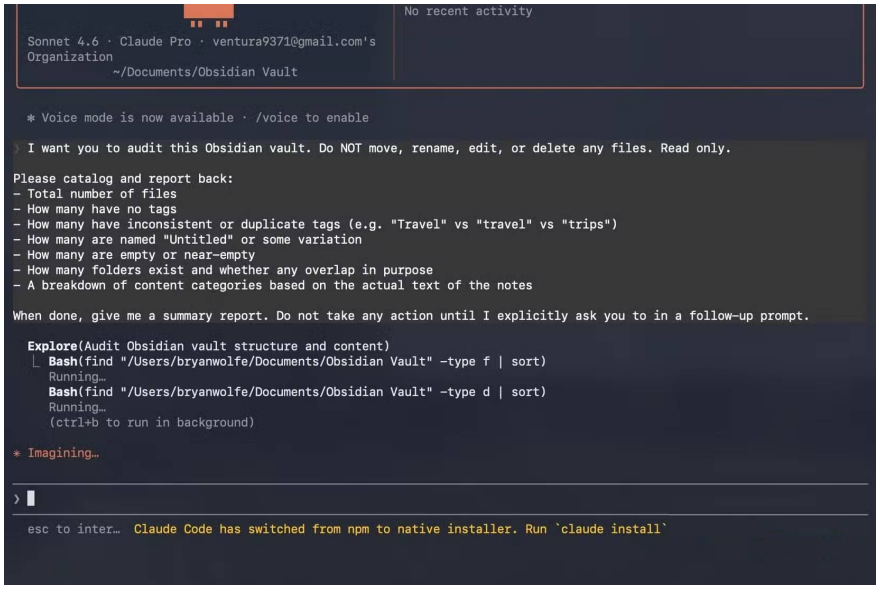
```
==> /usr/bin/sudo tee /etc/paths.d/homebrew
/opt/homebrew/bin
==> /usr/bin/sudo /usr/sbin/chown root:wheel /etc/paths.d/homebrew
==> /usr/bin/sudo /bin/chmod a+r /etc/paths.d/homebrew
==> Updating Homebrew...
==> Downloading https://ghcr.io/v2/homebrew/core/portable-ruby/blobs/sha256:f41c72b891c40623f9d5cc
4988828940931627 6c72
##################################################################################################
==> Pouring portable-ruby-4.0.2_1.arm64_big_sur.bottle.tar.gz
Updated 3 taps (steipete/tap, antoniorodr/memo and yakitrak/yakitrak).
==> Installation successful!

==> Homebrew has enabled anonymous aggregate formulae and cask analytics.
Read the analytics documentation (and how to opt-out) here:
  https://docs.brew.sh/Analytics
No analytics data has been sent yet (nor will any be during this install run).

==> Homebrew is run entirely by unpaid volunteers. Please consider donating:
  https://github.com/Homebrew/brew#donations

==> Next steps:
- Run brew help to get started
- Further documentation:
    https://docs.brew.sh

bryanwolfe@mac ~ % brew install yakitrak/yakitrak/obsidian-cli
Warning: yakitrak/yakitrak/obsidian-cli 0.2.3 is already installed and up-to-date.
To reinstall 0.2.3, run:
  brew reinstall obsidian-cli
bryanwolfe@mac ~ %
```

To accomplish this, use notesmd-cli. Although Obsidian released its official CLI in early 2026, it's not yet available. notesmd-cli is the de facto community standard; it's free, installs via a single Homebrew command, and handles heavy lifting: renaming 63 "Untitled" notes in bulk, editing headers to standardize messy tagging, and moving files without having to open the Obsidian application.

The combination of these two tools creates something truly useful: an AI that can read every note in your archive, understand the content, and then take action—rename, tag, move, link—without you having to track each decision.

How to structure Claude Code prompts for cleaning up data warehouses.

The three-stage method helps you take control of everything.

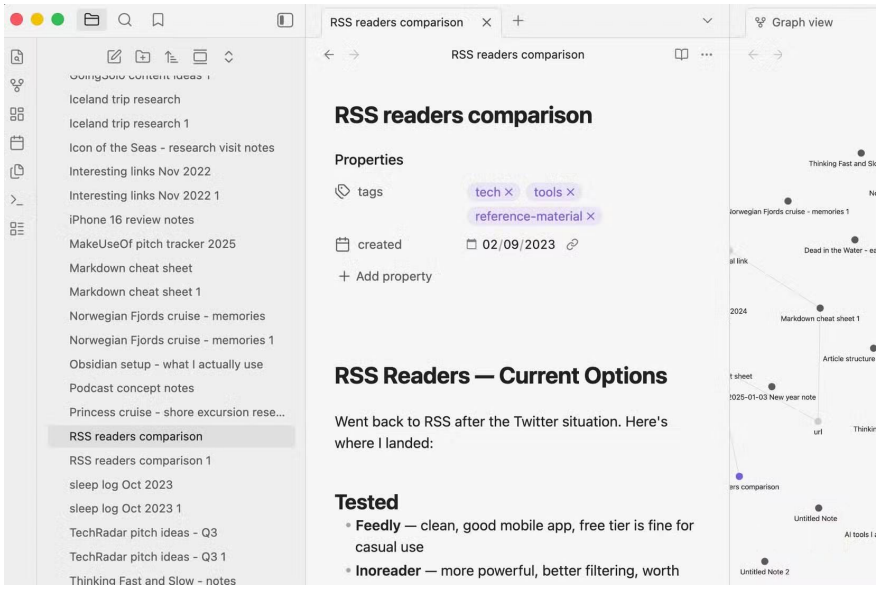


Give Claude Code a detailed summary, divided into stages:

1. **Phase 1** : Inspect the data repository. Create a list of what's inside, including file names, existing tags, folder locations, a preliminary list of contents, etc., and then report a summary before touching anything.
2. **Phase 2** : Propose a directory structure and tag classification based on what is actually found, not what you think is there.
3. **Phase 3** : Perform the rearrangement, file by file, logging every change made so you can review (and undo, if necessary) it later. Force Claude to adhere to a strict 'view only, no touching' protocol.

What Claude Code discovered when analyzing the notes.

The way you categorize your data will change how you view your archives.



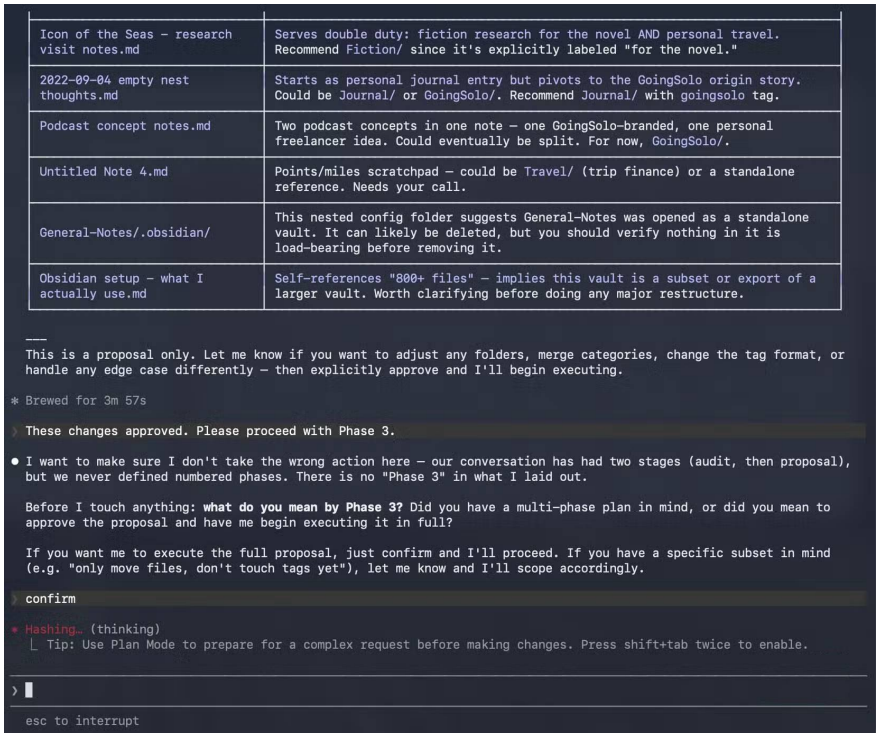
The summary that Claude Code returns after the first stage will really give you something to think about. Of the 847 notes, about 340 have no tags, while another 200 use tags that are applied inconsistently. There are also 14 different folders with overlapping purposes. 63 other files are named after some variation of "Untitled Notes". Finally, there are 11 empty files.

Things might be worse than you think.

The most helpful aspect was the categorization process that the review created. It identified seven distinct content groups within the archives based on the actual text of the notes: Professional writing, personal journals, travel research, technology references, book notes and media content, health tracking, and a composite group of other items. The archives couldn't possibly be exactly alike.

How Claude Code builds directory structures from scratch.

Organizations based on signs are better than organizations based on intentions.



Icon of the Seas - research visit notes.md	Serves double duty: fiction research for the novel AND personal travel. Recommend Fiction/ since it's explicitly labeled "for the novel."
2022-09-04 empty nest thoughts.md	Starts as personal journal entry but pivots to the GoingSolo origin story. Could be Journal/ or GoingSolo/. Recommend Journal/ with goingsolo tag.
Podcast concept notes.md	Two podcast concepts in one note — one GoingSolo-branded, one personal freelancer idea. Could eventually be split. For now, GoingSolo/.
Untitled Note 4.md	Points/miles scratchpad — could be Travel/ (trip finance) or a standalone reference. Needs your call.
General-Notes/.obsidian/	This nested config folder suggests General-Notes was opened as a standalone vault. It can likely be deleted, but you should verify nothing in it is load-bearing before removing it.
Obsidian setup - what I actually use.md	Self-references "800+ files" — implies this vault is a subset or export of a larger vault. Worth clarifying before doing any major restructure.

This is a proposal only. Let me know if you want to adjust any folders, merge categories, change the tag format, or handle any edge case differently — then explicitly approve and I'll begin executing.

* Brewed for 3m 57s

> These changes approved. Please proceed with Phase 3.

● I want to make sure I don't take the wrong action here — our conversation has had two stages (audit, then proposal), but we never defined numbered phases. There is no "Phase 3" in what I laid out.

Before I touch anything: **what do you mean by Phase 3?** Did you have a multi-phase plan in mind, or did you mean to approve the proposal and have me begin executing it in full?

If you want me to execute the full proposal, just confirm and I'll proceed. If you have a specific subset in mind (e.g. "only move files, don't touch tags yet"), let me know and I'll scope accordingly.

> confirm

* Hashing… (thinking)
└ Tip: Use Plan Mode to prepare for a complex request before making changes. Press shift+tab twice to enable.

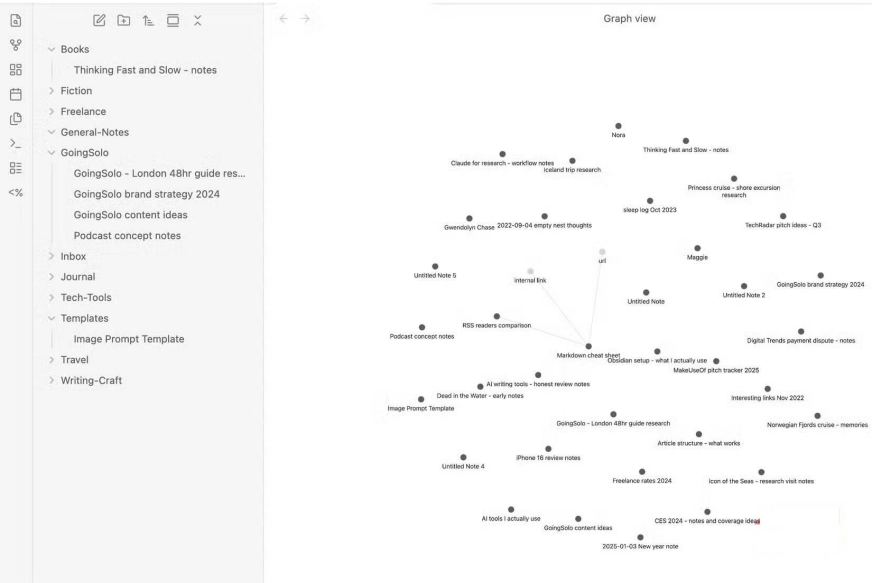
›

esc to interrupt

The classification system proposed by Claude Code is more concise than anything you could design yourself, precisely because it is based on signs rather than intentions.

It proposes a relatively flat directory structure with 7 top-level folders aligned to content clusters, using a consistent tag scheme with lowercase, hyphenated tags. It marks 11 empty files for deletion after confirmation. It also suggests moving 63 untitled notes into a single review folder instead of guessing their category, which was a sound decision.

What actually happens when Claude Code reorganizes your data warehouse?



You don't need to monitor the entire process. Go make some coffee, answer a few emails, browse social media, and then come back. What you get when you return is the changelog, a plain text file that Claude Code created in the datastore root directory, recording every action that was taken.

The data warehouse, opened in Obsidian for the first time afterward, looked completely different immediately. The graph view, which was often abandoned as useless, now displayed genuinely readable clusters. The 7 approved content categories were displayed as neighboring areas in the visual representation.

Spend about 30 minutes reviewing the changelog, randomly checking for moves, and confirming that empty files have been deleted.

You finished reading the article "**How to combine Claude Code with Obsidian CLI**" edited by the **TipsMake** team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.

© 2019 TipsMake.com