

# Manga Cleaner - App Summary

## What it is

Manga Cleaner is a Rust app that reorganizes one manga series folder into numbered batch folders, normalizes volume filenames, and creates batch cover images.

The primary interface is a native desktop app (`manga_cleaner_native``) with a companion CLI (`process_manga_rs``).

## Who it's for

Primary user/persona: Not found in repo.

Inferred from repository behavior: a manga collector or librarian managing local volume archives (`.cbz``, `.cbr``, `.cb7``, `.zip``).

## What it does

- Scans a series folder for volume archives and natural-sorts them.
- Builds a full execution plan that groups volumes into folders of 20 files each.
- Normalizes names by removing suffix noise, collapsing patterns like `v71_1_1``, and zero-padding to `v001`` format.
- Resolves series cover art in this order: first image from first `.cbz/.zip``, local cover files, then MangaDex -> AniList -> Kitsu.
- Generates per-batch `cover.jpg`` with centered batch number while preserving base art as `cover_old.jpg``.
- Archives any existing `cover.jpg`` to collision-safe `cover_old_*.jpg`` before writing a new cover.
- Executes file moves with collision-safe naming and cross-device fallback (`rename`` then `copy/remove``).

## How it works (repo-evidence architecture)

- Desktop UI (`src/main.rs``, `Iced`) handles folder selection, staged status, and progress display.
- UI launches worker threads and receives events through `std::sync::mpsc`` for analysis and processing updates.
- Shared core library (`src/lib.rs``) performs scanning, cover resolution, plan generation, renaming, moving, and cover rendering.
- CLI entrypoint (`src/bin/process_manga.rs``, `clap`) invokes the same core library for dry-run and commit flows.
- Data flow: series folder -> resolve cover -> `build_plan`` (`Vec<BatchPlan>``) -> preview -> `execute`` moves + numbered cover output.
- Remote integration: `reqwest`` calls MangaDex, AniList, and Kitsu only when local cover sources are unavailable.

## How to run (minimal)

- Install Rust toolchain (README provides macOS command).
- `cargo build --release``
- `cargo run --release --bin manga_cleaner_native -- "/path/to/Series Folder"``
- Optional CLI preview: `cargo run --release --bin process_manga_rs -- --dry-run "/path/to/Series Folder"``
- Linux/Windows setup details: Not found in repo.

Evidence used: `README.md`, `src/lib.rs`, `src/main.rs`, `src/bin/process_manga.rs`