CONNECTOR PATTERN

Intelligrit Labs | 22 commits | 1 primary builder

Take a GUI-bound system and make it programmable.

mail-app-cli gives Mail.app structured, scriptable access to accounts, mailboxes, messages, and attachments so automation and higher-level interfaces can treat email like infrastructure instead of a hand-operated GUI.

22	JSON	24h	MIT
COMMITS	DEFAULT OUTPUT	CACHE TTL	LICENSE

The Short Version

mail-app-cli demonstrates one of Intelligrit's most relevant patterns: wrapping a real existing system in a structured interface that other tools can safely build on. It is not just an email convenience tool. It is a bridge layer.

> *This is the same shape as future agent work: take the ugly real system, expose the useful parts, make the output structured, and keep the behavior bounded.*

What The History Shows

The commit history shows practical hardening, not just feature sprawl. Inputs were escaped to close JXA and AppleScript injection risks. Success messages were moved to stderr to keep stdout machine-parseable. Concurrency improvements were added for performance. A Makefile and performance notes reinforced the operational story.

- Accounts, mailboxes, messages, send, search, and attachment handling.
- JSON-first output for scripting and downstream tooling.
- Security-conscious input handling.
- Performance work and caching instead of naive repeated calls.

Why It Matters

It is strong Labs evidence for secure connectors, low-drag integrations, and agent-friendly system wrapping. If Intelligrit says it can connect real systems to useful AI, this repo shows the team already thinks in that shape.

Where It Fits

mail-app-cli belongs in Labs, but it should be referenced whenever Intelligrit needs to show a concrete “connector and structured interface” pattern without exposing client systems.