WIZARD CENTRAL

Wizard Central Toolkit (Apache 2.0) & Wizard Central Pro AI Modes Toolkit
Local-first CLI tooling & gated AI orchestration — sovereign, checkable, yours
"A command is what you type. The name is what the project is."

Canonical Name vs. Executable Command — **Canonical Name:** who the wizard *is* (e.g. *Code Guard Wizard*). **Executable Command:** what you *type* in the terminal (e.g. **`codeguardwiz`** — the full form **`codeguard_wizard`** runs too; both spellings work).

■ WIZARD CENTRAL TOOLKIT — FREE

Available now · v0.2.8 · Apache 2.0 · PyPI & GitHub

- **13 commands**: the Wizarding Router + 12 wizards, all local, all checkable
- **147 subcommands** + 39 REPL slash command forms
- Static code audit, secret redaction by default, consent-gated writes
- Standard library only, plus two small display libraries — no heavy stack
- Built for developers: fast local workflows you can read, test, and trust

■ WIZARD CENTRAL PRO AI MODES TOOLKIT

In development · the full gated system · proprietary

- **39 of 41 wizards built** — the fleet closes with Bonus Wizard & Keep Wizard
- **70 AI modes** advise; **guard modes** bind — server code that refuses (9 of 23 built)
- MCP bridge **off by default**: 349 commands, only **75** ever offered to a model
- Hash-chained receipts for every crossing; destructive actions ask twice
- Coming: seat cards, a creation system of **nine kinds**, 17 model providers, the REPL through the gate — then a 100-tester beta

■ COMPLETE COMMAND REGISTRY — FREE TOOLKIT (13)

Canonical Name	Executable Command (type this)	Primary Capabilities
Wizarding Router	`wiz`	Routes what you type to the right wizard · *--json* output · *--version*
Godfather Wizard	`godfatherwiz / godfather_wizard`	Fleet dispatcher & interactive REPL cockpit (39 slash command forms): census, status, routing, health. Its consent wall now reads verb + object — composing (a poem, a draft) passes free, real file writes still ask twice — and generation runs on a lightweight call built for CPU-only machines
Banner Wizard	`bannerwiz / banner_wizard`	ASCII banners — 571+ fonts found at runtime, 18 style presets, 6 display modes
Code Guard Wizard	`codeguardwiz / codeguard_wizard`	Static code audit with plain verdicts & severities; optional local-model seat (audit, scan, fix, explain)
Debug Wizard	`debugwiz / debug_wizard`	Traceback autopsy, error census & hotspots, fix suggestions, CI-ready JSON reports (22 commands)
Diff Wizard	`diffwiz / diff_wizard`	The honest differ: file & directory diffs, word-level diffs, same/different verdicts
JSON Wizard	`jsonwiz / json_wizard`	JSON doctor: validate, pretty, compact, inspect, keys, dotted-path get, structural diff, merge
Python Wizard	`pythonwiz / python_wizard`	Parse-only Python checks — syntax, imports, complexity, health score 0–100; *secrets* scan redacts values by default (*--show-secrets* is the opt-in)
Read Wizard	`readwiz / read_wizard`	Read-only policy engine: ALLOW / DENIED verdicts on raw shell commands
Rust Wizard	`rustwiz / rust_wizard`	Rust file & project checker, crate builder
Tmux Wizard	`tmuxwiz / tmux_wizard`	tmux sessions: new, attach, kill, list, census, rename
Todo Wizard	`todowiz / todo_wizard`	The personal work ledger: add, list, done, notes, oldest, overdue, archive, export
Write Wizard	`writewiz / write_wizard`	The consent wall for file writes: propose → preview → commit, quarantine, backups, rollback, protected paths

PyPI: `pip install wizard-central-toolkit` | **GitHub:** github.com/rewardsbycode-eng/wizard-central-toolkit

Try the free toolkit today. If you like it, Pro is what's coming next — and its closed 100-tester beta opens once the build is complete.