

Introducing NeoZip CLI

If you have ever built an automated workflow or a CI/CD pipeline, you already know the drill: package files, compress them, and deploy. It is simple and universal. What has changed is how much we depend on those automated bundles—and the rise of autonomous AI agents that need to interact with the file system securely.

That is why we built the **NeoZip CLI**, a full-featured command-line ZIP application that keeps standard ZIP compatibility while adding robust AI agent support and a built-in Data Wallet.

Command-line archiving, rethought

NeoZip CLI is designed for everyday command-line work across macOS, Linux, and Windows. You can create, extract, and list ZIP archives using modern Zstd compression by default, or use Deflate and Store methods. We have included AES-256 encryption as the default for strong security, with legacy PKZIP encryption available when you need to support older tools. In fact, with the `--legacy` flag, NeoZip CLI provides ~99% compatibility with standard InfoZip utilities. Where NeoZip CLI goes further is in what happens *around* the archive. You can tokenize archives to create on-chain tokens and verify archive integrity directly from your terminal.

Built for AI Agents and Automation

One of the biggest additions in version 0.95.1 is first-class support for AI agents. Every command exposes a stable, machine-readable surface alongside its human interface. Agents can integrate via direct shell invocation, bundled Skills, or MCP-native agents.

To make NeoZip safe and predictable for agents, we have included:

- **Machine-readable JSON I/O:** The `--format json` flag writes a single structured JSON object to standard output.
- **Separation of output:** Human hints and progress stay on stderr, while agents can parse the JSON on stdout.
- **Schema introspection:** The `neozip schema` command allows agents to learn accepted fields, types, defaults, and exit codes dynamically.
- **Dry-run previews:** Agents can use `--dry-run` to resolve and print the full execution plan as JSON without writing anything to the filesystem.
- **Bundled Agent Skills:** Six bundled skills ship directly inside the npm package.

- **Skill installation:** These skills can be easily installed into Claude Code or Cursor using the `neozip skills install` command.

Deep Token Service & Data Wallet Integration

At the heart of the CLI is the NeoZip Token Service and your Data Wallet. All wallet identity and Token Service credentials live in a shared connection store. This is exactly the same profile used by NeoZip Desktop and the MCP server.

This shared store allows you to seamlessly tokenize archives and request testnet gas grants using `neozip connect fund`. Version 0.95.1 introduces a silent token refresh feature. The CLI can automatically renew an expired access token via a silent wallet-signature login before it fails, making timestamping and on-chain proofs "just work" without manual intervention.

What you might use it for

Here are a few plain-language examples of how you can put NeoZip CLI to work:

- **Empowering AI Assistants.** Install the bundled skills so your AI coding assistant can safely package projects, examine archive contents, or extract files.
- **Securing sensitive files.** Compress and encrypt your sensitive files with AES-256 before sharing them.
- **Keeping a paper trail.** Automatically timestamp your documents on the Ethereum or Bitcoin blockchain to prove when a package was created.
- **Verifying integrity.** Extract and verify the on-chain integrity of your tokenized archives.

Available now

NeoZip CLI is in active development, and version 0.95.1 is available now. You can install it globally via npm using `npm install -g neozip-cli`. It requires Node.js 16.0.0 or higher. ZIP files are not going away—they are too useful. NeoZip CLI is our answer for a world that needs more than compression alone: packages you can stand behind, verify, and automate with AI.

Visit <https://neozip.io/cli> to learn more or dive into the command reference.