Node Stop Utility (Src/Main/stop.py)

Status: Alpha documentation snapshot (2026-03-12).

This utility stops Atho node processes even when they were started from another terminal/background session.

It targets:
- fullnode.py
- minernode.py
- walletnode.py
- runminer.py
- runnode.py (launcher, optional/available)

Why this exists

runnode.py launches background processes but does not include a dedicated stop command.
stop.py gives you one place to:
- stop one node role (full, miner, wallet)
- stop everything
- stop specific PIDs
- integrate the same logic from API/GUI code

CLI Usage

From project root:

No flags opens an interactive menu where you can choose:
- stop all
- stop full
- stop miner
- stop wallet
- stop launcher
- stop by PID

Non-interactive examples:

API-Friendly Functions

You can import and reuse these directly:

Main callable functions:
- discover_node_processes(include_launcher=True)

- list_node_processes(include_launcher=True)
- active_nodes_snapshot(include_launcher=True)
- stop_nodes(target="all", include_launcher=True, timeout_seconds=5.0, force_kill=True, dry_run=False)
- stop_pids(pids, timeout_seconds=5.0, force_kill=True, dry_run=False)

Signal Behavior

For each matched process:
- send SIGTERM
- wait up to timeout (default 5s)
- if still running and force-kill enabled, send SIGKILL

This is designed to cleanly stop nodes first, then force-stop only when needed.

GUI/API Integration

The API node stop routes now call this stop logic, so GUI stop actions can terminate:
- processes started by API controller
- processes started from other terminals/background launchers

The API also exposes active node monitoring:
- GET /nodes/active -> returns active node list, roles, and PIDs (counts_by_role, processes, by_role).

Safety Scope

stop.py only matches Atho node/launcher command paths listed above.
It does **not** blanket-kill all Python processes.