

# Coinbase TradeBot — Version 1.0.8

---

 *Naval Command Edition — Captain, Skipper, Commodore, Navigator, and Crew Coordination*

## Overview

Version 1.0.8 (Naval Command Edition) presents the most stable and feature-rich release of the Coinbase TradeBot. This version refines operational hierarchy — each trading subsystem now carries a naval rank, reflecting its strategic importance and role in coordinated market maneuvers. Together, these officers ensure balanced, adaptive, and profitable operations under all sea conditions.

## Crew Hierarchy and Responsibilities

- **Captain** (EMA): Determines overall market direction using short and long exponential moving averages.
- **Navigator** (Autotune): Adjusts tactical parameters in response to changing seas (market regimes).
- **Commodore** (MACD): Oversees momentum and signal crossovers, confirming the fleet's directional intent.
- **Skipper** (RSI): Monitors overbought/oversold waters to identify favorable entry or exit zones.
- **Quartermaster** (Take-Profit & Stagnation Officer): Manages exits, profit captures, and stale position clearance.
- **Deckhand** (24 h, ±2%): Handles daily maintenance — pruning, log management, and light telemetry duties.
- **Swab**: Keeps the decks clean by pruning processed fills and maintaining state efficiency.

## Navigator (Autotune) Enhancements — v1.0.8

The Navigator steers the fleet based on changing market currents. It fine-tunes each officer's parameters to ensure the Captain, Skipper, and Commodore remain coordinated during volatile or choppy conditions. Key enhancements:

- Adaptive alpha curve for smoother yet more decisive BLEND transitions.
- Per-knob learning rates for nuanced tuning across all officers.
- Quantized, weighted, and bounded adjustments (0.5 bps steps, 2 bps max per vote).
- BLEND gently shifts settings toward the winning regime, while SNAP enforces full preset alignment.
- Optional drift telemetry showing deviation from the golden CHOPPY preset baseline.

- Preserves the CHOPPY regime as the calm-sea anchor baseline.

## **Navigator Modes of Operation**

The Navigator determines operational mode based on regime vote share:

- SNAP:  $\geq 70\%$  vote share — immediately adopts the winning regime preset.
- BLEND: 55–69% vote share — gradually aligns tactical knobs toward the winner using smoothing and quantization.
- CHOPPY:  $< 55\%$  vote share — maintains the golden CHOPPY preset without deviation.

## **Parameters Adjusted by the Navigator (Autotune)**

The Navigator dynamically adjusts the following operational knobs:

- `ema_deadband_bps` — Captain's sensitivity to short/long EMA convergence.
- `rsi_buy_max / rsi_sell_min` — Skipper's buy/sell confirmation thresholds.
- `macd_buy_min / macd_sell_max` — Commodore's trend validation parameters.
- `confirm_candles` — Number of confirmations before acting on signals.
- `per_product_cooldown_s` — Time buffer before the next maneuver in a given pair.

EMA periods (Captain's `short_ema=40, long_ema=120`) remain fixed unless optional tuning is enabled (`autotune_tune_ema_periods=True` in `config.py`).

## **Reconcile Cadence**

Reconcile synchronizes past fills, positions, and P&L. At startup, a full reconcile runs before Autotune for accurate telemetry. During the session, automatic sweep reconcilers run every 60 minutes by default (configurable). When enabled, SELL operations perform a quick validation reconcile just before placing the order.

## **Architecture Overview**

TradeBot operates as a modular command system, with clear separation between strategy, persistence, telemetry, and tuning. Its runtime flow is designed for stability, transparency, and autonomous adjustment:

1. Load API keys and configuration — initializes credentials, product list, and all runtime parameters.
2. Reconcile portfolio and backfill recent fills — ensures local state matches exchange data before trading begins.
3. Perform initial Autotune (Navigator) — classifies market regime and calibrates EMA deadband, RSI, MACD, and cooldown knobs.
4. Launch live WebSocket session (candles + tickers) — streams market data for all configured products.
5. Monitor Captain, Skipper, and Commodore signals — interprets EMA, RSI, and MACD crossovers to execute trades.
6. Periodic maintenance — performs mid-session reconciles (every 60 min by default) and periodic Autotune refreshes.
7. Logging and telemetry — maintains detailed log output, CSV trade records, and rolling daily P&L summaries for analysis.

### Key Settings (config.py)

Setting	Default (example)	Purpose
<code>dry_run</code>	True	Simulation mode (no live orders).
<code>autotune_enabled</code>	True	Enable Autotune at startup.
<code>autotune_lookback_hours</code>	18	Candle window hours for regime voting.
<code>autotune_vote_interval</code>	15m	Dedicated timeframe for regime voting.
<code>autotune_vote_min_candles</code>	72	Minimum number of vote candles (~18h at 15m).
<code>lookback_hours</code>	48	Reconcile horizon for fills and KPI sync.
<code>mid_reconcile_enabled</code>	True	Enables periodic mid-session reconcile sweeps.
<code>mid_reconcile_interval_minutes</code>	90	Time interval for mid-session reconcile (minutes).
<code>autotune_preview_only</code>	True	Preview tuning suggestions without applying changes.
<code>autotune_elapsed_refresh_hours</code>	4	Hours between optional elapsed Autotune refreshes.

## Telemetry and Logbook Entries

The Commodore's telemetry records each decision in real-time — including regime votes, parameter changes, and fleet readiness. An optional drift summary shows how far the ship's settings have deviated from the golden CHOPPY baseline:

*[AUTOTUNE DRIFT] ema\_deadband\_bps:-0.50, rsi\_buy\_max:+1.00, ...*

## Quartermaster Module (Take-Profit & Stagnation Officer)

The Quartermaster safeguards profits and eliminates idle trades.

- Take-Profit (8%): Automatically executes a market SELL when unrealized profit reaches 8% ( $\approx 800$  bps) or higher. This ensures strong moves are captured without relying solely on EMA crossovers.
- Stagnation (48 h,  $\pm 2\%$ ): Detects positions held for more than 48 hours with less than  $\pm 2\%$  movement and a flat MACD histogram. When triggered, the bot performs a stagnation exit to recycle capital into more active opportunities.

Quartermaster logic runs before EMA-based signals each candle close. It will not interfere with active EMA trades and respects cooldown timers to prevent rapid re-entry.

All Quartermaster actions are logged to trades.csv with the exit\_reason field set to 'take\_profit' or 'stagnation'. In dry\_run=True, no trades are written — only simulated.

## Quartermaster Settings (config.py)

Setting	Default (example)	Purpose
<code>enable_quartermaster</code>	True	Enables take-profit and stagnation logic.
<code>take_profit_bps</code>	800	Trigger threshold (6%) for Quartermaster take-profit exits.
<code>max_hold_hours</code>	48	Hours before stagnation check activates.
<code>stagnation_close_bps</code>	200	$\pm 2\%$ band for stagnant positions.
<code>flat_macd_abs_max</code>	0.40	Defines 'flat' MACD for stagnation detection.

## Threads in This Program (v1.0.8)

TradeBot runs a compact multi-threaded command loop, designed for stability and low-latency market reaction:

- MainThread — Oversees the startup sequence (*Reconcile* → *AutoTune* → *WebSocket*) and runtime management, acting as the ship's command bridge.
- WebSocket Thread — Streams live candle and ticker data from Coinbase, feeding signal updates to the Captain (EMA), Skipper (RSI), and Commodore (MACD).
- Mid-Session Reconcile Thread — Performs scheduled portfolio synchronization sweeps (every 60 minutes by default) to ensure telemetry accuracy.
- Elapsed AutoTune Thread — Triggers one-shot retuning after a fixed interval (4 hours by default) to re-evaluate market regime alignment.

## Fleet Stability Notes

Version 1.0.8 introduces optimizations to minimize I/O turbulence, avoid redundant pruning operations, and improve the Swab's cleaning routines. ProcessedFills management has been refactored to ensure smooth sailing, and synchronization routines run efficiently without clogging the bilge (disk).

## Run Settings

Example configurations for running TradeBot in different modes:

```
# Dry run with AutoTune preview:  
dry_run = True  
autotune_enabled = True  
autotune_preview_only = True  
  
# Live run with active AutoTune:  
dry_run = False  
autotune_enabled = True  
autotune_preview_only = False  
  
# Static (no AutoTune. Uses Golden Preset Chop settings):  
autotune_enabled = False
```

## Technical Summary

- **Execution path:**  EMA captain → advisor veto → maker orders by default; Quartermaster runs **before** EMA and sells **at market** for take-profit/stagnation (with cooldown + dust guard).
- **Loss minimization:**  Optional hard\_stop\_bps; advisors veto bad crosses; per-product cooldown; daily buy cap; maker pricing with offsets to avoid taker fees on routine trades.
- **Spam loops:**  Quartermaster throttled, clamped to held size, and won't fire again within a short window.
- **State safety:**  Fill reconciliation before AutoTune; immediate fill handling under a lock; CSV logging gated off in dry-run.

## Changelog Summary

- Refactored Swab operations to prevent double pruning and improve state persistence.
- Reduced disk I/O overhead during active trading sessions.
- Enhanced Navigator's BLEND logic with quantization, alpha mapping, and per-knob learning rates.
- Reaffirmed CHOPPY preset as the system's calm-sea anchor regime.
- Improved startup synchronization reliability and reduced log clutter.
- Minor structural cleanups and documentation refinements.