TradeBot v1.0.7 — Detailed Guide

What's New in v1.0.7

- Regime voting decoupled from trading timeframe: AutoTune now uses a dedicated `autotune_vote_interval` (default 15m) for regime analysis, while trading continues on 5m candles.

- Improved KPI telemetry: AutoTune now runs *after* portfolio reconciliation, using real trade data to provide meaningful advisory output.

- Enhanced candle ordering: Candles are now sorted chronologically by timestamp to ensure accuracy in historical lookbacks.

- Cleaner startup order: The bot now performs Reconcile → AutoTune → WebSocket to guarantee up-to-date KPI data before trading begins.

- Improved offset logging: Per-product maker offsets always print clearly after KPI adjustments.

Fleet Command Analogy

TradeBot operates like a naval fleet:

• **Captain** (EMA) – steers overall direction using crossover logic.

• **Navigator** (AutoTune) – reads market weather and adjusts sails (parameters) dynamically.

• **Commodore** (MACD) – validates long-term momentum before executing orders.

• **Skipper** (RSI) – vetoes reckless trades during overbought/oversold conditions.

• **Quartermaster** – secures profits and manages portfolio logistics, ensuring the fleet's efficiency.

• **Deckhand** – the *"broom"* that clears idle trades drifting without purpose, keeping the decks lean and ready for action.

This chain of command keeps the strategy disciplined, adaptive, and self-correcting. Together, they form a balanced command structure that combines momentum, discipline, and adaptability.

Quickstart (PowerShell)

Run from the repo root:

```
python main.py
```

AutoTune (v1.0.7)

AutoTune runs automatically after the initial reconcile and periodically (default every 4 hours). It evaluates recent trading performance and market conditions to adjust RSI, MACD, EMA, and cooldown parameters intelligently. The system now supports a separate voting timeframe via `autotune_vote_interval` (default 15m) and ensures sufficient data coverage through `autotune_vote_min_candles` (default 72).

Reconcile Cadence

Reconcile synchronizes past fills, positions, and PNL data. Startup reconcile now executes before AutoTune for accurate telemetry. Mid-session reconcile sweeps run every 90 minutes by default, and SELL operations trigger quick validation reconciles when enabled.

Key Settings

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

Quartermaster Module (Take-Profit & Stagnation Officer)

The Quartermaster safeguards profits and eliminates idle trades.

• Take-Profit (6%): Automatically executes a market SELL when unrealized profit reaches 6% (≈ 600 bps) or higher. This ensures strong moves are captured without relying solely on EMA crossovers.

• Stagnation (24 h, ±2%): Detects positions held for more than 24 hours with less than ±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

Setting	Default (example)	Purpose
enable_quartermaster	True	Enables take-profit and stagnation logic.
take_profit_bps	600	Trigger threshold (6%) for Quartermaster take-profit exits.
max_hold_hours	24	Hours before stagnation check activates.
stagnation_close_bps	200	±2% band for stagnant positions.
flat_macd_abs_max	0.40	Defines 'flat' MACD for stagnation detection.

Threads in This Program (v1.0.7)

- MainThread — orchestrates startup sequence (Reconcile → AutoTune → WS) and runtime management.

- WebSocket thread — handles live candle and ticker updates from Coinbase.

- Mid-session reconcile thread — performs scheduled portfolio synchronization sweeps.

- Elapsed AutoTune thread — triggers one-shot retuning after a set number of hours (default 4h).

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
```

Release Safety Notes

- Always begin with dry_run mode to validate configuration.

- Keep .state/ and APIkeys.env excluded from version control.

- Preserve KPI and portfolio files to maintain accurate PNL tracking across sessions.

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.