

Bot Log Catalog

All tagged log categories emitted by the premarket screener/trading bot · sip_bot_*.log · as of 2026-09-09

Startup & Session Lifecycle

TAG	SOURCE	WHAT IT RECORDS
[START]	monitor.py	Process start, execution mode (paper/live)
[STARTUP]	intraday_health.py position_manager.py	One-time cleanup at boot (pruning stale health-state entries from a prior day)
[SHUTDOWN]	monitor.py	Signal received, graceful shutdown, cleanup complete
[EOD]	monitor.py	End-of-day force-liquidation and the daily summary line (trades / wins / losses / total P/L)

Candidate Discovery & Filtering

TAG	SOURCE	WHAT IT RECORDS
[PREMARKET]	premarket_scanner.py monitor.py	Universe filtering counts, per-symbol scores, "Selected N candidates"
[FILTER]	symbol_filters.py	Per-symbol rejection during ETF/leveraged/name-complexity filtering (DEBUG level)
[INTRADAY-RESCAN]	monitor.py	The 30-minute full rescans — pool rotation, per-symbol score + health snapshot
[REPLACEMENT]	top_stocks.py	Finding a replacement candidate when a position slot opens
[OPEN]	monitor.py	Market-open transition / stream-start warnings

Health & Entry Evaluation

TAG	SOURCE	WHAT IT RECORDS
[HEALTH]	intraday_health.py	State transitions (e.g. WATCH → STALE), health score, price/vwap slope
[ENTRY]	monitor.py position_manager.py	Shortlist benching ("benched after 5 consecutive not-confirmed attempts"), slot availability
[SLOT]	monitor.py	Open position slot available
[CONFIRMATION]	entry_engine.py	Legacy engine's confirm/reject reasoning — only active when entry.decision_engine = "legacy"
[PIPELINE]	monitor.py	New prediction-pipeline engine's confirm/reject reasoning — state, winner_score, all five component scores + reasons — only active when entry.decision_engine = "prediction_pipeline"

Position Management

TAG	SOURCE	WHAT IT RECORDS
[STOP]	position_manager.py	Initial stop set on entry
[TRAIL]	position_manager.py	Trailing-stop updates, grace extensions on a stop touch
[EXIT]	position_manager.py	Close order submission/fill, or a warning if one's already resting at the broker
[EXIT_REASON]	position_manager.py	Why the position closed (TRAILING_STOP, END_OF_DAY, etc.)
[P/L]	position_manager.py	Realized profit/loss for that trade
[RECONCILE]	position_manager.py	Reconciling local state against the broker's actual positions (throttled to every 30s)

Infrastructure

TAG	SOURCE	WHAT IT RECORDS
[STREAM]	stream.py	SIP connect/reconnect, and historical-backfill success/failure per symbol

Untagged (catch-all)

Roughly 15% of log calls carry no bracket tag — mostly raw exception messages from `alpaca_client.py` (*get_snapshots failed*, *get_minute_bars(...) failed*, order/position API failures) and config-fallback warnings (*Unknown feed '...' in config*). These are lower-frequency and typically only appear on an actual API hiccup or a config typo.

Gap worth knowing: everything above logs at INFO (or DEBUG/WARNING/ERROR as noted) except `[FILTER]`, which is DEBUG-only — and the config's `logging.level` is currently "INFO", so `[FILTER]` lines (and the `[ENTRY]` Shortlist this cycle line, which lives in the same `[ENTRY]` family but isn't broken out as its own tag) never actually reach `sip_bot*.log` today.