QuantMonitor_Multi_Market_Regime_Dashboard_v124

How it works in practice

This document explains the internal logic of the dashboard: data inputs, regime calculation, profile matching, scoring, refresh flow, and practical limitations.

1. What it is

It is a **market-regime monitor**, not a trade executor.

Its job is:

- take the **current market state** for each symbol,
- compare it with a **historically favorable regime profile** loaded from a CSV,
- show a **match score 0-100%** and a label:
- **FAVORABLE**
- **NEUTRAL**
- **UNFAVORABLE**

So the indicator answers:

"Does the market right now look similar to the conditions in which this strategy performed well in the past?"

2. Main inputs

The most important inputs are:

- InpStrategyComments

Example:

XAUUSD:XU_30_913034165_S_HH_CF_SQX, BRENT:Brent hh1, .US30Cash:New Strategy, EURJPY:EJ_15_201514104_S_SM_CF_SQ4

This is a list of pairs:

SYMBOL : STRATEGY_COMMENT

- InpProfileCsv

Path to the CSV with historical regime profiles:

QuantMonitor\QM_MarketRegimeProfiles.csv

- InpRegimeTimeframe

Default is H1

- InpTimerSeconds

Default is 20, so it refreshes every 20 seconds

- InpLookbackBars

Default is 1500, used to build enough history for regime calculations

There are also visual settings for panel position, row height, columns, legend, refresh button, etc.

3. High-level workflow

The dashboard does this cycle:

Step A - parse strategy list

It parses InpStrategyComments into pairs like:

- XAUUSD + XU_30_...
- BRENT + Brent hh1
- .US30Cash + New Strategy

Step B - ensure the symbol exists in MT5

For each symbol it checks whether the symbol is available in Market Watch.

If not, it tries SymbolSelect(symbol, true).

If the symbol is still not available, that card will show an error.

Step C - build current regime snapshot

For each symbol it calculates the **current regime snapshot** from price data and indicators.

Important detail:

It uses **shift 1**, so it evaluates the **last completed bar**, not the currently forming bar.

That is good, because it avoids unstable live-bar values.

Step D - load profile from CSV

Then it opens the CSV and searches for the exact row matching:

- symbol
- strategy_comment

Both must match exactly.

Step E - compute match score

It compares the current snapshot with the historical "ideal" profile from the CSV and computes a score from **0 to 100**.

Step F - render the dashboard

It draws a card for each symbol/strategy with:

- current regime states
- ideal regime states
- numeric metric ranges
- fit flags
- overall status

4. What "current regime snapshot" contains

The engine builds these numeric features:

- close
- ema50
- ema200
- atr14
- atr50
- atr_pct = atr14 / close
- adx14
- rsi14
- er24 = efficiency ratio over 24 bars
- ema50_slope_n = (EMA50 now - EMA50 previous) / ATR14
- ma_gap_atr = (EMA50 - EMA200) / ATR14

It also calculates two volatility percentile thresholds from recent history:

- vol_p33
- vol_p67

Those are used to classify the current volatility bucket.

5. How the regime states are classified

The dashboard converts the numeric features into 6 categorical states.

Direction (`dir_state`)

- **UP** if:
- EMA50 > EMA200
- EMA50 slope normalized >= 0.12
- MA gap in ATR units >= 0.50

- **DOWN** if:
- EMA50 < EMA200
- EMA50 slope normalized <= -0.12
- MA gap in ATR units <= -0.50

- otherwise **SIDE**

This means direction is not just MA cross. It also requires slope and separation.

Structure (`structure`)

- **TREND** if:
- ADX14 >= 25
- ER24 >= 0.30

- otherwise **CHOP**

So both trend strength and movement efficiency must agree.

Volatility state (`vol_state`)

Based on ATR14 / close relative to its recent distribution:

- **LOW** if below 33rd percentile
- **HIGH** if above 67th percentile
- otherwise **MID**

So volatility is relative to the symbol's own recent regime, not an absolute universal threshold.

Phase (`phase`)

Uses ATR14 / ATR50:

- **COMPRESSION** if ratio < 0.90
- **EXPANSION** if ratio > 1.10
- otherwise **NORMAL**

This tells whether volatility is shrinking or expanding relative to its medium-term baseline.

Momentum (`momentum`)

From RSI14:

- **BULL** if RSI >= 60
- **BEAR** if RSI <= 40
- otherwise **NEUTRAL**

Trend strength (`trend_strength`)

From ADX14:

- **WEAK** if < 18
- **MODERATE** if < 30
- **STRONG** otherwise

6. How the CSV profile is used

The CSV stores the historically favorable fingerprint of each strategy.

For each symbol + strategy_comment, the row contains:

Historical performance stats

- total trades
- overall win rate
- overall average net
- overall profit factor
- favorable subset trade count
- favorable subset average net
- favorable subset PF

Best categorical states

- best direction
- best structure
- best volatility
- best phase
- best momentum
- best trend strength

Ideal numeric ranges

For each numeric metric it stores quartiles:

- q25
- median
- q75

For:

- ADX14
- ATR%
- ER24
- RSI14
- EMA50 slope normalized
- MA gap in ATR units

The dashboard mainly uses **q25 to q75** as the "ideal operating range".

7. How the score is calculated

The score is split into:

A. Categorical part = 60 points total

There are 6 state checks:

- direction
- structure
- vol state
- phase
- momentum
- trend strength

Each exact match gives **10 points**.

So max categorical score:

6 x 10 = 60

B. Numeric part = 40 points total

There are 6 numeric checks:

- ADX14 in ideal range?
- ATR% in ideal range?
- ER24 in ideal range?
- RSI14 in ideal range?
- EMA50 slope in ideal range?
- MA gap in ideal range?

Each range hit gives:

40 / 6 = 6.666... points

So max numeric score:

40 points

Final score

score = categorical points + numeric points

Clamped to 0-100.

8. How the label is assigned

After the score is computed:

- **FAVORABLE** if score >= 70
- **NEUTRAL** if score >= 40
- **UNFAVORABLE** otherwise

Color mapping:

- green = favorable
- gold = neutral
- red = unfavorable

So this is effectively a **compatibility score** between current market and the strategy's historical sweet spot.

9. What you see on each card

Each symbol/strategy card shows:

Top line

- symbol + strategy comment
- score + label

Example:

78.3% FAVORABLE

Sub line

- timeframe
- evaluated bar time

Stats line from CSV

- overall PF
- favorable subset PF
- favorable subset trades / total trades

Example:

PF 1.61 | Fav PF 2.05 | Trades 112/436

State rows

- current state
- ideal state
- fit = OK or MISS

Rows:

- Dir
- Struct

- Vol
- Phase
- Mom
- TStr

Numeric rows

- current value
- ideal range
- fit = LOW / IN / HIGH

Rows:

- ADX14
- ATR%
- ER24
- RSI14
- Slope
- Gap

Bottom note

A compact interpretation such as:

- "Regime very close to ideal."
- "Regime mixed vs ideal."
- "Main gap: volatility."

So the card not only gives a score, but also tells you **what is wrong**.

10. Refresh behavior

The dashboard does **not** use `OnTick()` for its logic.

`OnTick()` is empty.

It refreshes via:

- `OnInit()` -> immediate first load
- `OnTimer()` -> automatic refresh every `InpTimerSeconds`

- `OnChartEvent()` -> manual refresh when you click the refresh button

This is a good design for multi-symbol monitoring because it does not depend on ticks of the chart symbol.

11. Important practical detail: it uses the last closed bar

This is one of the most important implementation details.

The engine reads indicator values using **shift 1**, for example:

- EMA50 at shift 1
- EMA50 previous at shift 2
- ATR14 at shift 1
- ADX14 at shift 1
- RSI14 at shift 1

And it uses `rates[1]` for bar time and close.

So the dashboard evaluates the **last completed candle**, not the current unfinished one.

That means:

- more stable regime classification
- less flickering
- no look-ahead bias inside the live dashboard

12. What the dashboard does not do

It does **not**:

- read live trade history from the sqlite database
- recalculate the profile from trades on the fly

- turn strategies on/off automatically
- place, close, or manage trades

Instead, it assumes that the CSV has already been built from your historical analysis.

So the process is:

1. analyze historical trades + market data externally
2. produce QM_MarketRegimeProfiles.csv
3. dashboard loads that CSV
4. dashboard compares current market to that historical profile

13. Why this is useful

This gives you a regime filter like:

- "XAUUSD strategy historically works best in MID vol, COMPRESSION, BULL momentum"
- "Current market is HIGH vol, EXPANSION, NEUTRAL momentum"
- therefore current fit is weak

So instead of blindly running all strategies equally, you can decide:

- increase confidence
- reduce risk
- skip bad conditions
- compare symbols side by side

It is basically a **strategy-to-market compatibility dashboard**.

14. Weak points / limits of the current version

A few important limitations from the current logic:

Exact state matching is rigid

If current state is close to ideal but just below a threshold, it still becomes MISS.

Example:

- ADX 24.9 vs threshold 25.0
- one small move can flip CHOP to TREND

Numeric ranges are simple q25-q75 bands

This is robust, but it ignores shape of distribution and asymmetry.

No weighting by strategy importance

All 6 categorical states are weighted equally.

Maybe for some strategies:

- volatility matters more than momentum
- structure matters more than phase

But current code treats them equally.

No direct execution link

It only informs. Another EA or overlay would be needed to act on the result automatically.

15. In one sentence

QuantMonitor_Multi_Market_Regime_Dashboard_v124 takes the **last closed H1 market regime**, compares it against a **historically profitable regime profile from CSV**, and shows how favorable current conditions are for each strategy.