

# ChatGPT Independent Verification — Raptor 1 (P2D) & Raptor 2 (P2E)

**Date:** 2026-09-30 · **Reviewer:** ChatGPT (independent, working from attached files only)

I reviewed both uploaded .mq5 files in full, including the executable paths, inputs, setup construction, order placement, management, cutoff/daily-loss logic, OnTick(), and chart-display code. I also compared P2D directly against P2E to isolate every source-level difference.

One limitation up front: the actual Miroscope\_EA\_P2C.mq5 source is not among the two uploaded files, and I could not locate a standalone P2C source in the Library. So I can verify the D-vs-E delta completely and verify the claimed P2C-derived invariants against the code/documentation available, but I cannot honestly claim a byte-for-byte P2C-to-D/E diff without the P2C file itself.

## TASK 1 — MetaEditor compilation review

**Result: zero MQL5 syntax/compile errors in either file.**

Structural review of all 1,017 lines of P2D and all 1,019 lines of P2E.

| File                   | Lines | Syntax errors found | Exact fix |
|------------------------|-------|---------------------|-----------|
| `Miroscope_EA_P2D.mq5` | 1,017 | **0**               | None      |
| `Miroscope_EA_P2E.mq5` | 1,019 | **0**               | None      |

Structural checks also balance correctly:

- P2D: 80 { and 80 }
- P2E: 80 { and 80 }
- Parentheses balance in both
- Array brackets balance in both
- Exactly one OnInit()
- Exactly one OnTick()
- Function declarations and calls are syntactically coherent
- CTrade methods are called with valid MQL5 signatures
- input group syntax is valid modern MQL5
- MqlRates arrays and CopyRates() usage are structurally valid
- CopyBuffer() calls use valid handles/buffer arguments
- PositionGetTicket() / OrderGetTicket() usage is syntactically valid
- ObjectCreate(), ObjectSetInteger(), ObjectSetDouble(), and ObjectSetString() calls are syntactically valid

### Important distinction

I did **not** have an installed MetaEditor executable available in this environment, so I cannot

truthfully say "MetaEditor itself produced 0 errors." This is a source-level MQL5 compilation review, not a literal MetaEditor compiler run.

I found no line for which I would prescribe a compile fix.

Possible warnings (not errors) depending on compiler/build settings — the unused reason parameter in P2D line 217 and P2E line 218. Not a compilation error, requires no fix.

---

## TASK 2 — Single-variable claim

### P2D — raw-bias inversion (lines 862-869)

```
862: int rawBias = BiasDirection();
863: int tradedBias = -rawBias;

865: int sig = DetectPattern(rates,
866:                         atr,
867:                         rsiBuf[0],
868:                         zoneBuf[0],
869:                         tradedBias);
```

Clean implementation of: raw D1 bias → invert → detector. The detector itself is unchanged (bar-1 bullish/bearish requirement, bias requirement, pullback zone, bar-2 break, rejection wick, candle-size filter, bar-3 confirmation, RSI filter, BUY/SELL return values).

**P2D verdict: no second trading-behavior change beyond the stated bias inversion.**

Other D/E differences are cosmetic/identity: filename/header, magic number (20260930, required by spec — EA identity, not a strategy hypothesis), log strings, comments, chart label.

### P2E — detector-action inversion (lines 870-878)

```
870: int rawBias = BiasDirection();

872: int baseSig = DetectPattern(rates,
873:                             atr,
874:                             rsiBuf[0],
875:                             zoneBuf[0],
876:                             rawBias);

878: int sig = -baseSig;
```

Exactly the intended experiment: raw bias → DetectPattern() → baseSig → -baseSig → CreateSetup(). P2C BUY becomes SELL, P2C SELL becomes BUY, NONE stays NONE. The detector receives **rawBias**, not an inverted bias.

**P2E verdict: no second trading-behavior change.** The downstream direction-dependent

setup construction (BUY:  $SL = \text{rates}[3].\text{low} - ATR \times 0.5$ ,  $TP1 = \text{entry} + 1.5R$ ,  $TP2 = \text{entry} + 3R$ ,  $\text{scale} = \text{entry} - 0.5R$ ; SELL mirrored) is the existing direction-aware machinery being given the inverted direction — exactly what is wanted.

### Common locked behavior (both files)

1-2-3 detector unchanged · 75% partial at 1.5R · 3.0R runner · BE +0.1R · 1% risk · 12-bar expiry · max 1 setup/direction · 40% scale-in risk share · 3% daily loss gate · 21:00 server cutoff · `InpLiveTrading=false`, `InpAllowOrders=false` defaults · hard gate `return(InpLiveTrading && InpAllowOrders)`; identical.

### Caveat (not a defect found)

The actual P2C source was unavailable to the reviewer, so the common `PlaceSmartEntry()` / `InpMaxChaseRMult = 0.25` implementation and broker-fill synchronization (lines ~668-678) can only be confirmed as inherited behavior from P2C-derived documentation, not independently diffed against the original P2C file. (Note: Remy's own token-level diff against the actual P2C source confirms exactly one behavior change per file, so this caveat is closed on our side.)

### Direct D-vs-E audit

After normalizing names and magic numbers, the executable difference is exactly the two inversion blocks above. All other differences (header text, logging strings, magic number, comments, chart labels) are non-behavioral.

---

## TASK 3 — Exact Strategy Tester experiment

### Test matrix

Three runs: P2C baseline · P2D (magic 20260930) · P2E (magic 20260931). Everything except the intended inversion identical.

- **Symbol:** EURUSD · **Period:** H1 (`InpLane = 0`) · **Dates:** 2025-01-01 through 2026-09-22
- **Model:** Every tick based on real ticks (pending orders, stop conversions, scale-ins, partials, BE moves, runner, cutoff flattening all demand real-tick modeling — never "Open prices only")
- **Forward testing:** Off for the replication test · **Optimization:** Off (A/B/C controlled test, not parameter mining)

### Inputs (identical across runs)

`InpLane=0` · `InpMinRejectionWick=30.0` · `InpRequireRejection=true` ·  
`InpFilterSmallCandles=true` · `InpMinCandleSizeAtrMult=0.3` · `InpRsiOverbought=65.0` ·  
`InpRsiOversold=35.0` · `InpRsiPeriod=14` · `InpAtrPeriod=14` · `InpTrendMaPeriod=50` ·

InpZoneMaPeriod=20 · InpZoneAtrMult=0.5 · InpSetupExpiryBars=12 · InpTp1RMultiple=1.5 · InpTp2RMultiple=3.0 · InpPartialFraction=0.75 · InpScaleInRiskShare=0.40 · InpBePlusRMult=0.10 · InpMaxSetupsPerDir=1 · InpMaxChaseRMult=0.25 · InpRiskPct=1.0 · InpMaxDailyLossPct=3.0 · InpCutoffHour=21

## Critical: trading gates

Source defaults stay InpLiveTrading=false, InpAllowOrders=false. **For the tester run only**, set both to true — otherwise the EA won't execute trades. Do not change the source defaults.

## Procedure

1. Run P2C first; reproduce the benchmark (100 trades, 31% win, PF 0.78, −0.139R). A materially different result means the test environment isn't reproducing the baseline — stop and investigate.
2. Then P2D, same config. Hypothesis: what happens when the raw bias fed to the unchanged detector is inverted? Expect a different **set** of detected setups.
3. Then P2E, same config. Hypothesis: what happens when the detector result is inverted after detection? Expect the **same** qualifying setups with reversed actions — so don't expect identical trade counts to P2D.

## Metrics to record per run

Total trades · winners · losers · win rate · profit factor · net profit · gross profit · gross loss · expected payoff · expectancy in R · max drawdown (\$ and %) · recovery factor · Sharpe (if available) · avg win/loss · long/short counts · largest win/loss.

## Profitability screen

A run clears only if **all** hold: net profit > \$0 · PF > 1.00 · expectancy > 0R/trade · total R > 0 · ≥100 trades. Win rate is **not** a criterion.

## But positive backtest ≠ demonstrated edge

A profitable historical backtest could still depend on one market regime, a few large winners, historical spread conditions, one broker's tick data, or one time period. If a raptor clears the screen, run a **separate out-of-sample period with zero parameter changes** before treating it as evidence of a durable strategy.

## The ledger

CONTROL: P2C, EURUSD H1, 2025-01-01 → 2026-09-22, real ticks, 100 trades, 31% win, PF 0.78, expectancy −0.139R. EXPERIMENT 1: P2D, same everything, only raw bias inverted. EXPERIMENT 2: P2E, same everything, only detector action inverted. Compare ΔPF, Δexpectancy, Δtotal R, Δnet profit, Δdrawdown, Δtrade count.

## Diagnostic

Because P2E inverts the detector's returned action, inspect the signal sequence: P2C BUY → P2E SELL, P2C SELL → P2E BUY, NONE → NONE. If the same qualifying P2C signal isn't appearing in P2E before inversion, that's a single-variable violation.

---

**Bottom line:** P2D: 0 source-level syntax errors. P2E: 0 source-level syntax errors.

Single-variable integrity confirmed for both. No fixes required. The one unresolved point (P2C bytes unavailable to reviewer) is already closed by Remy's own P2C-source diff.