Based off Trade Rush Video:

Testing 87% Win Rate Strategy 100 TIMES (Actually Worked)

Credit for the idea goes to Trading Rush: https://www.youtube.com/@TRADINGRUSH

Not meant for live trading!!! Educational only.

WARNING: This code attempt is a draft version and NOT 100% compliant with the videos rules. Feel free to change as you see fit.

By EgoNoBueno

The EA exits trades before broker end of day. It uses virtual take profit via market order closure.

Run on M5 or M15

Core Concept: Structured Entries

- **Instead of a single entry point and position size, this strategy splits your entry into three distinct points and divides your total intended position size into three smaller, equal parts.**
- **The goal is to achieve a better average entry price and manage risk by scaling into a position.**

Entry Rules:

1. **Identify a Support Level: In an uptrend, mark a support area, typically at a previous swing high. The underlying assumption is that the price is likely to reverse at or above this support.**
2. **Use an Entry Signal: The example uses a MACD indicator. A buy signal is when the MACD line crosses above the signal line, and this crossover occurs below the MACD's zero line. However, the strategy implies other indicators or buy signals could be used.**
3. **First Entry:**
    - **When the first buy signal occurs above the identified support level, take the first entry.**
    - **Risk only one-third of your total intended risk for the trade (e.g., if you normally risk 3%, risk 1% on this first entry).**
4. **Second Entry:**

    - **If the price moves down after the first entry and another buy signal occurs (e.g., another MACD crossover above support), take the second entry.**
    - **Risk another one-third of your total intended risk (e.g., an additional 1%).**
5. **Third Entry (Best Entry Point):**
    - **If the price continues to move down towards or at the identified support level and a third buy signal occurs, take the third and final entry.**
    - **Risk the final one-third of your total intended risk (e.g., the last 1%). This entry is considered the "best entry point" as it's closest to the support.**

Stop-Loss Rule:

- **Consistent Placement: For *all three entries*, the stop-loss is placed at the same "best spot."**
- **Location: This "best spot" for the stop-loss is below the identified support level. This means even for the first entry (which might be significantly above the support), the stop-loss is placed below the support.**

Position Sizing and Risk Management:

- **Split Risk: Divide your total planned risk for the trade by three. Each entry will carry this smaller portion of the risk. For example, if you typically risk 3% on a trade, each of the three entries will risk 1%.**
- **Averaging Down: As you take subsequent entries at lower prices, your broker will likely combine these positions. This results in an average entry price for your total position that is better (lower in an uptrend) than your initial entry.**
- **Total Risk: After all three entries are taken, your total risk on the combined position will be your originally intended total risk (e.g., 3% if each of the three entries risked 1%).**
- **Flexibility: The example uses 1% risk per entry for simplicity, but you can adjust the risk per entry as long as the principle of splitting the total risk across three entries is maintained.**

Key Outcome:

- **The strategy aims to achieve a higher win rate by allowing for multiple entry attempts at progressively better prices, with a pre-defined stop-loss based on the optimal entry scenario.**
- **Even if the first or second entries are stopped out (though the text implies the stop-loss is for the combined position after it averages), the risk is managed in smaller increments. The focus is on the final averaged position.**

Code Review: "Structured MACD Entry" Expert Advisor v1.04 (VSL/VTP)

Overall Strategy:

The "Structured MACD Entry" is designed to implement a specific trading strategy that aims for a potentially high win rate by allowing up to three structured entries into a trade setup. The core signal is based on MACD crossovers relative to its zero line, filtered by the direction of a longer-term Exponential Moving Average (EMA). The EA incorporates a Virtual Stop Loss (VSL) and Virtual Take Profit (VTP) system for precise exit management, while using wider, buffered Stop Loss and Take Profit levels with the broker. Position sizing includes compounding options.

Key Features and How They Work:

1. EA Identification & Setup:

- **Properties: The EA starts with standard `#property` directives for copyright, link, and version ("1.04"). `#property strict` is used for more robust code checking.**
- **Inputs: A comprehensive set of `extern` input parameters allows users to customize:**
  - **Indicator settings (EMA period/timeframe, MACD parameters).**
  - **Stop loss buffer for VSL calculation.**
  - **Minimum price improvement for adding entries.**
  - **Reward:Risk ratios for different entry stages.**
  - **Position sizing modes and risk percentages.**
  - **Slippage.**
  - **VSL/VTP system activation and buffer settings.**
  - **Visual line properties for VSL/VTP.**
  - **Strategy Tester speed controls.**
  - **Spread filter.**
  - **Trading hours filter.**
- **Magic Number:**
  - **`InpBaseMagicNumber`: User-defined base for the magic number.**
  - **`GenerateMagicNumber(int p_baseNum)`: Creates a unique magic**

number for each EA instance by combining the base number, the chart symbol, and the chart period. This is excellent for running the EA on multiple charts without interference.
    - `EA_MagicNumber` (global): Stores the generated magic number used for all trade operations.
- Initialization (`OnInit()`):
    - Generates the `EA_MagicNumber`.
    - Constructs unique base names for VSL/VTP chart lines using the `EA_MagicNumber`.
    - Calculates `pipValueInPoints` based on broker digits.
    - Assigns `slippagePoints` from input.
    - Calculates and stores `g_brokerGmtOffset`.
    - Validates and sets effective parameters like `g_effectiveMaxSpread` and `g_effectiveTradingEndHour`.
    - Prints a detailed summary of the initialized settings, including VSL/VTP status and risk modes.
    - Calls `ResetSeriesState()`, `MonitorSpread()`, and `UpdateHUD()` for initial setup.
- Deinitialization (`OnDeinit()`):
    - Cleans up VSL/VTP lines from the chart using `DeleteLine()`.
    - Prints final trade statistics.
    - Clears the chart comment.
- Main Tick Processing (`OnTick()`):
    - Calls `MonitorSpread()` at the start of each tick.
    - Implements a "once per bar" execution logic using `static datetime lastBarTime`.
    - Checks spread against `g_effectiveMaxSpread`.
    - Checks `IsTradingHoursAllowed()`.
    - If no trade series is active (`!isSeriesActive`), it calls `CheckForInitialEntry()`.
    - If a series is active, it calls `ManageActiveSeries()`.
    - Calls `SlowDownStrategyTester()` and `UpdateHUD()` at the end.

2. Signal Logic & Structured Entry:

- **Trend Filter: Uses an EMA (`InpTrendEMATF`, `InpTrendEMAPeriod`) on a user-defined higher timeframe to determine the overall trend.**
- **Entry Signal (MACD):**
  - **Uses MACD (`InpMACDFast`, `InpMACDSlow`, `InpMACDSignal`) on the chart's timeframe.**
  - **Long Signal: Price closes above the Trend EMA, and on the previous bar, the MACD main line crossed above the MACD signal line, with the crossover occurring below the MACD zero line.**
  - **Short Signal: Price closes below the Trend EMA, and on the previous bar, the MACD main line crossed below the MACD signal line, with the crossover occurring above the MACD zero line.**
- `CheckForInitialEntry()`**:**
  - **Evaluates the conditions for an initial long or short signal.**
  - **If a signal occurs and lot size calculation is valid:**
    - **Determines the `VSL_for_this_potential_series` based on the Trend EMA +/- `InpStopLossBufferPips`. This becomes the `virtualStopLossLevel_Series`.**
    - **Calculates the initial `virtualTakeProfitLevel_Basket` based on the actual open price, VSL, and `InpRR_Entry1_2`.**
    - **Two-Step Order Placement: Sends the initial order with SL=0 and TP=0.**
    - **After confirming the order is open (via `OrderSelect`), it calls `SetBrokerSLTP_ForOrder()` to modify the order with the (wider) broker SL and TP.**
    - **If `UseVirtualSLTP` is true, it draws the VSL and VTP lines.**
    - **Updates EA state to reflect an active series.**
- `ManageActiveSeries()` **(for 2nd/3rd Entries):**
  - **First checks for broker SL/TP hits or manual closures using `CheckSeriesSLTP()`.**

    - **Then, if `UseVirtualSLTP` is true, checks for VSL/VTP hits using `ManageVirtualExits()`.**
    - **If the series is still active and fewer than 3 entries have been made:**
        - **Checks for a new MACD signal in the same direction as the series.**
        - **Requires the `currentPriceForEntry` to be better than the `previousAvgEntryPrice` by at least `InpMinPriceImprovePips`.**
        - **Includes a new safety check: ensures the `currentPriceForEntry` is not already too close to the established `virtualStopLossLevel_Series` (considering `MarketInfo(Symbol(), MODE_STOPLEVEL)`).**
        - **If all conditions pass, it calculates lot size (based on the fixed `virtualStopLossLevel_Series` and the risk percent for that entry stage) and places a new "naked" order.**
        - **After successful placement and selection, it calls `SetBrokerSLTP_ForOrder()` for the new order.**
        - **Calls `UpdateBasketTakeProfit()` to recalculate the basket VTP and update all broker TPs.**

3. Position Sizing:

- `ENUM_SIZING_MODE`**:**
    - **`FixedLot`: Uses `InpFixedLotSize`.**
    - **`PercentRiskEqual`: Each of the three potential entries risks `InpRisk_Entry1`% of the current account balance.**
    - **`PercentRiskCompounding`: This mode's behavior is controlled by `UseUnifiedTurtleSeriesRisk`.**
- `UseUnifiedTurtleSeriesRisk` **(boolean):**
    - **If `true`: The `InpUnifiedTurtleSeriesRiskPercent` (e.g., 2%) is divided by 3, and each of the three entries will risk this fractional percentage (e.g., 0.667%) of the current account balance. This implements a**

"Turtle-like" compounding where the total series risk is defined, and each part contributes equally based on current equity.
    - **If `false`: The EA uses the staged individual risk percentages (`InpRisk_Entry1`, `InpRisk_Entry2`, `InpRisk_Entry3`) for each respective entry, calculated against the current account balance.**
- `GetRiskPercent(int entryNum)`**: Returns the appropriate risk percentage based on the selected mode and current entry number.**
- `CalculateLotSize(double slLvl, int oType, double rPcnt)`**: Calculates the lot size based on the risk percentage (`rPcnt`), the distance to the stop loss level (`slLvl` - which is the VSL when VSL/VTP is active), and the current account balance. It includes checks against min/max lot and lot step.**

4. Virtual Stop Loss / Take Profit (VSL/VTP) System (if `UseVirtualSLTP` is true):

- **Concept: The EA internally tracks a precise VSL and a dynamic basket VTP. Exits are primarily triggered by price reaching these virtual levels, at which point the EA issues market close orders. Wider SL and TP levels are sent to the broker as a safety net and to make initial order modification more acceptable.**
- `virtualStopLossLevel_Series`**: Determined once at the start of a series (based on `commonStopLoss` from EMA +/- buffer) and remains fixed for that series.**
- `virtualTakeProfitLevel_Basket`**: Dynamically calculated by `UpdateBasketTakeProfit()` based on the average entry price of all open trades in the series and the VSL, using the R:R ratio appropriate for the current number of entries.**
- **Broker SL/TP Buffers: `InpVslBrokerSlBufferPips` and `InpVtpBrokerTpBufferPips` define how much wider the broker-side SL/TP should be compared to the VSL/VTP.**
- `SetBrokerSLTP_ForOrder()`**: This function is now responsible for setting the *broker's* SL and TP via `OrderModify()` *after* an order has been successfully opened naked. It uses the VSL/VTP levels plus the defined buffers. It also includes validation to adjust or skip setting**

SL/TP if the buffered levels are still too close to the open price.

- `ManageVirtualExits()`**: Called every tick for an active series. It checks if the current market price (Bid for buy VSL, Ask for sell VSL; Ask for buy VTP, Bid for sell VTP) has breached the `virtualStopLossLevel_Series` or `virtualTakeProfitLevel_Basket`.**
- `CloseSeriesAtMarket(string reason)`**: If `ManageVirtualExits` detects a VSL/VTP hit, this function is called to close all open orders in the current series with market orders. It also logs P/L and updates statistics.**
- **Visual Lines: `DrawUpdateLine()` and `DeleteLine()` manage the display of magenta VSL and blue VTP lines on the chart. Line names are made unique using the `EA_MagicNumber` and the first ticket of the series.**

5. Basket Management:

- `GetAverageEntryPrice()`**: Calculates the volume-weighted average entry price of all currently open orders belonging to the active series and `EA_MagicNumber`.**
- `UpdateBasketTakeProfit()`**:**
  - **Called after a new entry is added to the series.**
  - **Recalculates `virtualTakeProfitLevel_Basket` based on the new `GetAverageEntryPrice()`, the fixed `virtualStopLossLevel_Series`, and the R:R ratio for the current number of open entries (e.g., 0.25 for 1 or 2 entries, 0.5 for 3 entries).**
  - **Updates the VTP line on the chart using `DrawUpdateLine()`.**
  - **Iterates through all open orders in the series and modifies their *broker Take Profit levels* to be `virtualTakeProfitLevel_Basket +/- InpVtpBrokerTpBufferPips`, after validating these new broker TPs against `MODE_STOPLEVEL`.**

6. Auxiliary Features:

- **Trading Hours Filter (`IsTradingHoursAllowed()`): Allows restricting EA activity to specific days and hours (broker server time). Includes a hardcoded daily close period check.**

- **Spread Monitoring (`MonitorSpread()`): Tracks current, average, high, and low spread. `OnTick` checks `g_currentSpreadPoints` against `MaxAllowedSpreadPoints` before attempting new entries.**
- **Strategy Tester Speed Control (`SlowDownStrategyTester()`): Uses busy loops to slow down visual backtesting based on user inputs.**
- **Heads-Up Display (HUD) (`UpdateHUD()`): Provides a comprehensive on-chart display of:**
  - **EA name, symbol, timeframe, magic number.**
  - **Spread information.**
  - **Series status (active, entries, direction, VSL/VTP levels).**
  - **Trend EMA value.**
  - **General EA status (monitoring, hours inactive, etc.).**
  - **Trading hours filter status.**
  - **Live P/L of open series.**
  - **Overall trade statistics (total trades, wins, losses, win rate, P/L).**
- **GMT Offset (`CalculateGMTOffset()`): Calculates and logs the broker's GMT offset.**
- **Logging: Extensive use of `Print(__LINE__, ...)` provides detailed logging with line numbers, crucial for debugging and understanding EA behavior.**
- **Statistics Tracking: Basic tracking of total trades, wins, losses, and P/L in currency and pips. `LogClosedTradeStats` updates these when trades are closed by broker SL/TP or manual intervention (detected by `CheckSeriesSLTP`). `CloseSeriesAtMarket` updates these for VSL/VTP exits.**

How It Works - Workflow Summary:

1. **Initialization (`OnInit`): Sets up parameters, generates magic number, calculates GMT offset, prints settings.**
2. **On Each Tick (OnTick):**
   a. Monitors spread.
   b. Checks if trading is allowed (hours, spread).
   c. If no series is active:

i. CheckForInitialEntry looks for MACD/EMA signals.
ii. If a signal found, calculates VSL.
iii. Sends a "naked" order (no SL/TP).
iv. If successful, selects the order, calculates initial VTP.
v. Calls SetBrokerSLTP_ForOrder to modify the order with wider broker SL/TP and draws VSL/VTP lines (if VSL system is active).
vi. Sets isSeriesActive = true.
d. If a series is active (isSeriesActive == true):
i. ManageActiveSeries is called.
ii. CheckSeriesSLTP looks for broker SL/TP hits or manual closes. If found and series is empty, resets.
iii. If UseVirtualSLTP is true, ManageVirtualExits checks if current price hit VSL or VTP. If so, calls CloseSeriesAtMarket which closes all trades in the series and calls ResetSeriesState.
iv. If no VSL/VTP exit and entriesInSeries < 3, it checks for conditions to add another entry (MACD signal, price improvement, not too close to VSL).
v. If conditions met for an additional entry, it sends another "naked" order.
vi. If successful, calls SetBrokerSLTP_ForOrder for the new order.
vii. Calls UpdateBasketTakeProfit to recalculate basket VTP, update the VTP line, and modify broker TPs of all open orders in the series.
e. Slows down tester if in visual mode.
f. Updates the HUD.

3. **Series Conclusion: A series ends when:**
    - **All its parts are closed by `ManageVirtualExits` (VSL/VTP hit).**
    - **All its parts are closed by broker SL/TP (or manually), detected by `CheckSeriesSLTP`.**
    - **In both cases, `ResetSeriesState` is called to clean up lines and prepare for a new series.**

Overall Code Quality and Practices:

- **Well-Structured for MQL4: The code is organized into logical functions.**
- **VSL/VTP Implementation: The addition of the VSL/VTP system with two-step order placement is a significant enhancement for managing exits precisely and mitigating initial `OrderSend error 130`.**
- **Comprehensive Inputs: Offers good flexibility for users.**
- **Detailed Logging: The use of `__LINE__` is excellent for debugging.**
- **Compounding Options: Provides sophisticated risk management choices.**
- **Error Handling: Basic error checking for `OrderSend`, `OrderModify`, and `OrderSelect` is present. The handling of failed order selection after a successful send (attempting to close the naked order) is a good recovery mechanism.**
- **Potential for Refinement:**
    - **The logic for `InpRR_FarAverage` is still a placeholder and would need a defined condition to be triggered.**
    - **The lot size calculation, while functional for Forex, can be very complex for other instrument types and different broker margin calculation modes.**
    - **Error handling for `OrderModify` in `SetBrokerSLTP_ForOrder` could be expanded (e.g., if modification fails repeatedly, what should the EA do? Currently, it relies on VSL/VTP).**

This EA is a robust framework for the "Three Tries" strategy, with the VSL/VTP system being a key feature for improved trade management. The main challenge, as seen in test logs, often lies in tuning the strategy's core parameters (R:R, signal sensitivity, VSL placement) to achieve consistent monetary profitability across different market conditions and timeframes.