

Multi-Symbol Martingale Strategy EA

0. General Requirements

- (i) "Class" should **NOT** be used in the code.
- (ii) "Pointer" should **NOT** be used in the code.
- (iii) "Trade.mqh" in "Standard Library" **must be used** while other standard library such as "PositionInfo.mqh", "SymbolInfo.mqh" and "AccountInfo.mqh" should be avoided.
- (iv) The source code should be clear, readable, well-organized and easy to modify. "Code Optimization" is necessary. **There should be sufficient description and comment marked in the code (in particular handling existing positions).** The names of functions and variables should be self-explanatory. Words in short form should be avoided and all words should be separated by "_" (i.e. similar to those input name). Code Style should generally comply with MQL5's "Code Styler".
- (v) The EA should be easily amended (i.e. adding new function and deleting function easily). It is preferable that all "repeated" or "relevant" code should be put in "Function". Such approach can allow me to add / remove function easily in future.
- (vi) Please ensure that the EA (especially the filling type) is applicable in "Hedging Account" in "IC Markets" Broker.
- (vii) To explain the source code upon request.

1. Whole_Account_Setting (Applied to the Whole Account)

1.1	Floating_Take_Profit_Dollar	:	Input: 500 [If input == 0, this function is turned off.] To close all position when AccountInfoDouble (ACCOUNT_PROFIT) is <u>higher</u> than USD500 .
1.2	Floating_Stop_Loss_Dollar	:	Input: 3,000 [If input == 0, this function is turned off.] To close all position when AccountInfoDouble (ACCOUNT_PROFIT) is <u>less</u> than "- USD3,000" .

2. All_Symbol_To_Trade

2.1	Multi_Symbol_Trading	:	<p>Input: True/False</p> <p>False :</p> <p>EA will only Trade “Chart Symbol” based on “Group_00_General_Setting”. The trading coding would be run in “OnTick()”.</p> <p>True :</p> <p>EA will only Trade “Symbol Group 00 - Symbol Group 03” based on “Group 00 General Setting - Group 03 General Setting”. The trading coding would be run in “OnTimer()”.</p>
2.2	Symbol_Group_00	:	<p>Inputs for multi-symbols</p> <p>[For example, if the input is “EURUSD;AUDNZD;USDJPY”, the EA will trade those 3 symbol based on Setting_Group_00]</p> <p>[For example, if the input is “Void”, the EA will NOT trade any symbol based on Setting_Group_00]</p>
2.3	Symbol_Group_01	:	<p>Inputs for multi-symbols</p> <p>[the logic same as the above]</p>
2.4	Symbol_Group_02	:	<p>Inputs for multi-symbols</p> <p>[the logic same as the above]</p>
2.5	Symbol_Group_03	:	<p>Inputs for multi-symbols</p> <p>[the logic same as the above]</p>
2.6	Set_Timer_Seconds_Backtester	:	<p>Input: 10 Seconds</p>
2.7	Set_Timer_Seconds_Live_Trade	:	<p>Input: 1 Seconds</p> <ul style="list-style-type: none"> - Set “EventSetTimer()” for “OnTimer()”. - Set difference timer for “backtest” and “live trade” respectively.

3. Group_00_Settings

[EA will trade multi-symbols within "Symbol_Group_00" based on "Group_00_Settings".]

[EA will trade multi-symbols within "Symbol_Group_01" based on "Group_01_Settings".]

[EA will trade multi-symbols within "Symbol_Group_02" based on "Group_02_Settings".]

[EA will trade multi-symbols within "Symbol_Group_03" based on "Group_03_Settings".]

The trading logic of Group_01 - Group_03 are completely the same as Group_00. The Group_00 Setting are as follows:-

Group_00_General_Setting			
3.1	Group_00_Magic_Number	:	Input: 123
3.2	Group_00_Open_Order_Slippage	:	Input: 10
3.3	Group_00_Close_Order_Slippage	:	Input: 10
3.4	Group_00_Buy_or_Sell	:	Enum: 0. Buy_and_Sell 1. Buy_Only 2. Sell_Only
3.5	Bollinger Bands Group_00_BB_Time_Frame Group_00_BB_Period Group_00_BB_Deviation	:	Input: H4 Input: 20 Input: 2.0 Operation: If "Current Price" < Lower Band: Buy If "Current Price" > Upper Band: Sell (applicable to both First Order and Martingale Orders)

4. Group_00_Order_Management_Setting

Group_00_Order_Management_Setting																																				
4.1	Group_00_Points_Distance	:	Input: 400 points Taking “Buy” as example, if the open price of the last buy order is 1.00000, the next Buy Order must be opened at the price <u>less than</u> [1.00000 - Points_Distance].																																	
4.2	Group_00_Time_Distance	:	Input: 240 minutes [If input == 0, this function is turn off.] Example:- If the open time of the last order is 18:30, the next order must be opened after 240 minutes, i.e. after 22:30.																																	
4.3	Group_00_Max_Order_Number	:	Input: 15 [If input == 0, this function is turned off, mean there is no restriction.] [Buy and Sell should be calculated separately.]																																	
4.4	Group_00_First_Order_Lot_Size	:	Input: 0.01																																	
4.5	Group_00_Add_Order_Mode		Enum : Exponential_Mode OR Linear_Mode																																	
<table><tr><th>Level</th><th>Exponential_Mode</th><th>Linear_Mode</th></tr><tr><td>1</td><td>0.01</td><td>0.01</td></tr><tr><td>2</td><td>0.01 x 1.5^1 = 0.02</td><td>0.01 + 0.01 x 1 = 0.02</td></tr><tr><td>3</td><td>0.01 x 1.5^2 = 0.02</td><td>0.01 + 0.01 x 2 = 0.03</td></tr><tr><td>4</td><td>0.01 x 1.5^3 = 0.03</td><td>0.01 + 0.01 x 3 = 0.04</td></tr><tr><td>5</td><td>0.01 x 1.5^4 = 0.05</td><td>0.01 + 0.01 x 4 = 0.05</td></tr><tr><td>6</td><td>0.01 x 1.5^5 = 0.08</td><td>0.01 + 0.01 x 5 = 0.06</td></tr><tr><td>7</td><td>0.01 x 1.5^6 = 0.11</td><td>0.01 + 0.01 x 6 = 0.07</td></tr><tr><td>8</td><td>0.01 x 1.5^7 = 0.17</td><td>0.01 + 0.01 x 7 = 0.08</td></tr><tr><td>9</td><td>0.01 x 1.5^8 = 0.26</td><td>0.01 + 0.01 x 8 = 0.09</td></tr><tr><td>10</td><td>0.01 x 1.5^9 = 0.38</td><td>0.01 + 0.01 x 9 = 0.10</td></tr></table> <p>* Based on the numbers of existing orders for the same direction.</p>				Level	Exponential_Mode	Linear_Mode	1	0.01	0.01	2	0.01 x 1.5^1 = 0.02	0.01 + 0.01 x 1 = 0.02	3	0.01 x 1.5^2 = 0.02	0.01 + 0.01 x 2 = 0.03	4	0.01 x 1.5^3 = 0.03	0.01 + 0.01 x 3 = 0.04	5	0.01 x 1.5^4 = 0.05	0.01 + 0.01 x 4 = 0.05	6	0.01 x 1.5^5 = 0.08	0.01 + 0.01 x 5 = 0.06	7	0.01 x 1.5^6 = 0.11	0.01 + 0.01 x 6 = 0.07	8	0.01 x 1.5^7 = 0.17	0.01 + 0.01 x 7 = 0.08	9	0.01 x 1.5^8 = 0.26	0.01 + 0.01 x 8 = 0.09	10	0.01 x 1.5^9 = 0.38	0.01 + 0.01 x 9 = 0.10
Level	Exponential_Mode	Linear_Mode																																		
1	0.01	0.01																																		
2	0.01 x 1.5^1 = 0.02	0.01 + 0.01 x 1 = 0.02																																		
3	0.01 x 1.5^2 = 0.02	0.01 + 0.01 x 2 = 0.03																																		
4	0.01 x 1.5^3 = 0.03	0.01 + 0.01 x 3 = 0.04																																		
5	0.01 x 1.5^4 = 0.05	0.01 + 0.01 x 4 = 0.05																																		
6	0.01 x 1.5^5 = 0.08	0.01 + 0.01 x 5 = 0.06																																		
7	0.01 x 1.5^6 = 0.11	0.01 + 0.01 x 6 = 0.07																																		
8	0.01 x 1.5^7 = 0.17	0.01 + 0.01 x 7 = 0.08																																		
9	0.01 x 1.5^8 = 0.26	0.01 + 0.01 x 8 = 0.09																																		
10	0.01 x 1.5^9 = 0.38	0.01 + 0.01 x 9 = 0.10																																		
4.6	Group_00_Exponential_Mode_Multiplier	:	1.5 [for “Exponential_Mode” only]																																	
4.7	Group_00_Linear_Mode_Add_Lot_Size	:	0.01 [for “Linear_Mode” only]																																	
4.8	Group_00_Max_Lot_Size	:	0.15 [If input == 0, this function is turned off.]																																	

5. Group_00_TP_SL_Setting

Group_00_TP_SL_Setting			
[Remarks: Virtual Take Profit and Stop Loss: No TP & SL in points will be set in the MT5 Account.]			
5.1	Group_00_Average_TP	:	Input: True/False
5.1A	Group_00_Average_TP_in _Point	:	Input: 200 points
	<p>Average_TP_Point = 200 points</p> <p>Operation: Taking EURUSD “Buy” as example,</p> <ol style="list-style-type: none"> 1. If current price of EURUSD >= the breakeven point of all EURUSD (same magic number), plus Average_TP_Point. 2. Then close all Buy Order of EURUSD (same magic number). <p>[For the avoidance of doubt, each symbol should be calculated separately.]</p>		
5.2	Group_00_Trailing_Stop	:	Input: True/False
5.2A	Group_00_Trailing_Start_in _Point	:	Input: 200 points
5.2B	Group_00_Trailing_Stop_Percent	:	Input: 20%
	<p>Trailing_Stop_Point = 200 points</p> <p>Operation: Taking EURUSD “Buy” as example,</p> <ol style="list-style-type: none"> 1. If “current price of EURUSD” - “the breakeven point of all EURUSD (same magic number)” >= Trailing_Stop_Point, the “Trailing Stop” will be triggered. 2. If the “Trailing Stop” is triggered, double Trailing_Stop_Highest_Point = “current price” - “breakeven point” 3. Keep update the “Trailing_Stop_Highest_Point” if (“current price” - “breakeven point” > Trailing_Stop_Highest_Point) { Trailing_Stop_Highest_Point = “current price” - “breakeven point” } 4. If “current price” < “breakeven point x Trailing_Stop_Highest_Point x (1 - 20%)” { close all Buy Order of EURUSD (same magic number) } <p>[For the avoidance of doubt, each symbol should be calculated separately.]</p>		

5.3	Group_00_Partial_Closing	:	Input: True/False
5.3A	Group_00_Partial_Closing_Start_Level	:	Input: 4
5.3B	Group_00_Partial_Closing_Last_Order	:	Input: 2
5.3C	Group_00_Partial_Closing_First_Order	:	Input: 1 [If input == 0, only “Partial_Closing_Last_Order” will be closed.]
5.3D	Group_00_Partial_TP_in_Points	:	Input: 100 points

For Example, there are **10** Existing Buy Order below:-

Levels	Lot Size	Price	
1	0.01	1.00000	} Partial_Closing_First_Order: 1
2	0.02	0.99500	
3	0.02	0.99000	
4	0.03	0.98500	
5	0.05	0.98000	
6	0.08	0.97500	
7	0.11	0.97000	
8	0.17	0.96500	
9	0.26	0.96000	} Partial_Closing_Last_Order: 2
10	0.38	0.95500	

If:-

- Symbol's Existing Buy Order (same magic number) \geq **4** (Partial_Closing_Start_Level) ; and
- Current Price \geq “Breakeven Points of the **1st** , **9th** and **10th** Level Buy Order” + **100 points**

Then:-

- Close the **1st** , **9th** and **10th** Level Buy Orders

The remaining existing orders are shown as follows:-			
Levels	Lot Size	Price	} Partial_Closing_First_Order: 1
2 1	0.02	0.99500	
3 2	0.02	0.99000	
4 3	0.03	0.98500	
5 4	0.05	0.98000	
6 5	0.08	0.97500	} Partial_Closing_Last_Order: 2
7 6	0.11	0.97000	
8 7	0.17	0.96500	

//----- When The Price continue to increase -----

If:-
(i) Symbol's Existing Buy Order (same magic number) >= 4; and
(ii) Current Price >= "Breakeven Points of the 1st , 6th and 7th Level Buy Order" + 100 points

Then:-
- Close the 1st , 6th and 7th Level Buy Orders

5.4	Group_00_Average_SL	:	Input: True/False
5.4A	Group_00_Average_SL_in_Point	:	Input: 1,000 points
<p>Operation: Taking EURUSD "Buy" as example,</p> <ol style="list-style-type: none"> 1. If current price of EURUSD <= the breakeven point of all EURUSD (same magic number), less 1,000 points 2. Then close all Buy Order of EURUSD (same magic number). <p>[For the avoidance of doubt, each symbol should be calculated separately.]</p>			
5.5	Group_00_Take_Profit_Dollar	:	<p>Input: 100</p> <p>[If input == 0, this function is turn off.]</p> <p>Operation: Taking EURUSD as example,</p> <ol style="list-style-type: none"> 1. If the floating profit of EURUSD Order (same magic number) >= USD100. 2. Then close all EURUSD Buy and Sell Orders (same magic number). <p>[For the avoidance of doubt, each symbol should be calculated separately.]</p>

5.6	Group_00_Stop_Loss_Dollar	:	<p>Input: 500 [If input == 0, this function is turn off.]</p> <p>Operation: Taking EURUSD as example,</p> <ol style="list-style-type: none"> 1. If the floating loss of EURUSD (same magic number) >= USD500. 2. Then close all EURUSD Buy and Sell Orders (same magic number). <p>[For the avoidance of doubt, each symbol should be calculated separately.]</p>
-----	---------------------------	---	---

Group_01 Setting (same as Group_00 Setting in Item 3.1 - 5.6 above)

Group_02 Setting (same as Group_00 Setting in Item 3.1 - 5.6 above)

Group_03 Setting (same as Group_00 Setting in Item 3.1 - 5.6 above)