

# MultiOp

A multi-pair, multi-indicator, grid capable, backtestable dashboard EA  
for MetaTrader 4 and 5

Updated May 19, 2017

Version 0.1

by Marco Z

<https://github.com/marcolovescode>

## Table of Contents

Introduction .....	1
Trade Management Settings.....	1
Filter Value Notation.....	1
Filter name .....	2
Calculation .....	2
Operand .....	2
Optimization Notation .....	2
General Settings.....	3
Instrument Settings.....	3
Trade Settings .....	3
General Trade Settings.....	3
Trade Parameters.....	4
Trade Signal Settings.....	5
Trade Delay Settings .....	5
Grid Settings.....	6
Grid Reset Settings.....	7
Stop Level Settings .....	7
Stop Level Basic Settings.....	7
Break Even Settings.....	8
Trailing Stop Loss Settings.....	9
Jumping Stop Loss Settings .....	9
Basket Exit Settings .....	9
Master Basket Stop Levels .....	10
Master Break Even Stop.....	10
Master Trailing Stop.....	10
Master Jumping Stop .....	10
Symbol Basket Stop Levels.....	10
Symbol Stop Levels .....	10
Symbol Stop Level Offsets.....	11
Master Break Even Stop.....	11
Symbol Trailing Stop .....	11
Symbol Jumping Stop.....	12

Schedule Settings .....	12
Order closing settings .....	13
Delay settings .....	14
Filter Settings .....	14
Subfilter Setting Notation .....	15
Optimization Settings.....	16
Standard Subfilter Settings .....	16
Optimization Values.....	16
Runtime Settings.....	17
Error Settings .....	17
Display Settings .....	17
Cycle Settings .....	18
Frequently Asked Questions .....	18
There is a master entry signal but no trade was entered, why? .....	18

## Introduction

MultiOp is a signal-driven, filter-based dashboard EA. It manages trades and processes indicator signals. It is designed to be flexible with numerous indicator configurations and trading scenarios.

Several goals set apart MultiOp as an exceptional system. MultiOp supports the following:

1. An arbitrary number of indicator filters, e.g., if you want to check ten different parameters on five different timeframes, MultiOp allows for this.
2. Indicator-driven values for SL/TP and similar settings. E.g., you can set SL/TP to any factor of ATR, StdDev, or Spread.
3. Advanced retracement handling, e.g., you can ignore signal retrace entries or fulfill retraces by delay.
4. Advanced SL/TP handling, including moving stops as well as offsetted levels to hide your true stops.
5. Grid setups for pending stop and limit orders.
6. Account baskets and per-symbol baskets, including moving stop levels for symbol baskets.
7. Advanced scheduling, per broker settings or custom set by daily, weekday, or date.
8. Portfolio backtesting and optimization support.
9. MetaTrader 4 and MetaTrader 5 full support.

Total	72.0	Longs 5.3	Shorts 66.7	5001 SSFB Grid				Entry/Exit/Value		MultiOp v0.1 05/2017			
Symbols	Total	Long	Short	O	CSS-H1	CSS-D1	FB-H1	FB-D1	C	FB-H1	ATR-H1	ATR-D1	Spread-Cu
AUDCAD					Short S	Long L					13.9	61.3	5.5
AUDJPY					Short S	Short S		Below O			13.0	86.4	3.2
AUDNZD					Short S	Long L					16.5	63.6	5.7
AUDUSD					Long L	Long L					9.1	46.7	1.4
CADJPY					Short S	Short S		Below O			11.1	80.9	2.5
EURAUD	-7.3	0.0	-7.3		Short S	Short S		Below O			17.6	106.4	2.5
EURCAD				S	Short S	Short S	Below O	Below O	S	Below S	14.4	100.7	3.8
EURJPY	35.2	0.0	35.2		Short S	Short S		Below O			14.7	90.6	1.8
EURNZD					Short S	Short S		Below O			24.6	102.3	9.8
EURUSD	5.5	5.5	0.0	S	Short S	Short S	Below O	Below O			8.1	60.5	1.2
EURGBP	-0.2	-0.2	0.0	S	Short S	Short S	Below O	Below O			7.7	58.9	1.3
GBPAUD					Short S	Short S					22.6	143.1	5.5
GBPCAD					Short S	Long L					16.6	126.1	6.7
GBPJPY					Short S	Short S		Below O			17.3	115.8	5.4
GBPNZD					Short S	Short S					29.6	138.2	9.2
GBPU\$D					Long L	Long L					8.7	87.0	2.0
NZDJPY					Short S	Short S		Below O	L	Above L	13.1	63.3	2.9
NZDUSD					Long L	Long L					10.9	45.7	3.1
USDCAD					Short S	Short S		Below O			9.3	72.6	1.9
USDJPY	38.8	0.0	38.8		Short S	Short S		Below O			13.2	90.9	1.8

## Trade Management Settings

Runtime settings are core to the MultiOp system and encompass such parameters as SL/TP, grid, and trade timing. Some settings are configured using the below notations.

### Filter Value Notation

Settings whose names end in "Calc", such as SL/TP levels and distances, can be set to an exact value or to a filter value using filter value notation:

ATR-H1		Factor		0.5
[Filter name]-[Subfilter name]		Pipe		Calculation   Pipe   Operand

### Filter name

Filter and subfilter to retrieve the value from. Case-insensitive.

Filter name is hardcoded, e.g., ATR, StdDev, and Spread. Subfilter name is defined in the settings, e.g., ATR\_Entry\_Name. Subfilter names should be unique; if a filter value notation tries to reference a non-unique subfilter name, the filter and subfilter initiated first takes precedence.

### Calculation

Optional; takes the filter value and calculates it according to the command. Available values are:

- Factor (Multiply) – Multiply value by operand
- Offset (Add) – Offset value by operand. If operand is a negative number, value is subtracted. If operand is a positive number, value is added.
- Divide – Inverse of Factor.
- Subtract – Inverse of Offset. If operand is a negative number, then value is added to because double-negative arithmetic applies. If operand is a positive number, then value is subtracted.

### Operand

Optional; number to use in calculation. Can be a positive or negative decimal number.

## Optimization Notation

The "Calc" values can be modified in backtest optimization by setting a redirect to an optimization setting (e.g., Opt\_0, Opt\_1, Opt\_2, ...) This is needed because the optimizer can't access Calc notations, so we instead redirect them to Opt\_x where we can specify starting value, steps, ending value, and default (non-optimized) value.

Value redirects take the form:

**10.0 @ 0**

Value | At | Opt Index |

The value (before the @) is always ignored and exists for notekeeping. The value does not need to be specified. The opt index (after the @) refers to the setting Opt\_0, Opt\_1, Opt\_2, ..., and tells MultiOp to pull the value from that setting.

The redirect is always in effect whether or not optimization is being run.

For example, 10.0@0 says to pull the value from Opt\_0.

ATR-H1 | Factor | 2.0@1 says to pull the operand from Opt\_1 in optimization.

The source filter can be optimized as well. ATR@2-H1@3 | Factor | 2.0 says to pull the filter index from Opt\_2 and pull the subfilter index from Opt\_3.

The filter index is hardcoded into MultiOp and will most typically be:

- 0 – ATR

- 1 – StdDev
- 2 – Spread

The subfilter index corresponds to the subfilter count configured in the settings. For example, 0 = subfilter A, 1 = subfilter B, 2 = subfilter C, 3 = D, 4 = E, 5 = F, ...

## General Settings

---

### MagicNumber

Magic number for trades. MultiOp will only manage trades that exhibit this magic number.

### ConfigComment

A text comment that displays on the dashboard, to identify the setfile.

## Instrument Settings

---

### SingleSymbolMode

Use only the chart symbol where MultiOp is attached, and ignore all other symbol settings. Required for real tick mode.

Non-Forex symbols (e.g., indexes, metals) are currently ignored, even if SingleSymbolMode is enabled.

### IncludeSymbols

List of symbols to trade, comma separated. Leave empty to trade all Forex symbols.

Non-Forex symbols (e.g., indexes, metals) are currently ignored, even if SingleSymbolMode is enabled.

### ExcludeSymbols

List of symbols to ignore.

### ExcludeCurrencies

List of currencies to ignore; symbols with these currencies will not be traded.

## Trade Settings

---

### TradeEntryEnabled

Manage trade entries by signal.

### TradeExitEnabled

Manage trade exits by signal. Baskets, SL/TP, and schedule exits will always fire regardless of this setting.

### TradeValueEnabled

Manage SL/TP changes for open trades. If this is disabled, moving stops will not function.

### General Trade Settings

### TradeModeType

Type of trades to enter. Can be:

- Trade with normal market orders – Place market orders immediately. Guarantees trade placement but susceptible to slippage.
- Trade with pending limit orders – Place a limit order at the current price. Guarantees zero slippage but susceptible to not firing. The price action usually triggers this order immediately, but will sometimes miss.
- Trade with grid orders – Place a grid of stop and limit orders, dictated by Grid Settings below.

### SetStopsOnPendings

Set SL/TP initially on a pending order.

If this is false, the SL/TP on the pending order will be zero. If TradeValueEnabled is true, the SL/TP will be set once the pending order is fired.

### IsTwoStep

Set SL/TP on market orders as a two-step modifying order, rather than part of the initial order placement. Required for ECN brokers.

### BrokerPoints

Number of pip decimal places supported by the broker. Most typically 0 for a 4-point broker (quotes whole pips only), and 1 for a 5-point broker (quotes fractions of a pip).

**This setting is critical for internal value processing to work. Make sure this is correct.**

### OrderComment

Comment to attach to orders. Note that the broker often overrides order comments.

## Trade Parameters

### MaxTradesPerAccount

Max trades to allow on the account, including pending orders. Set to 0 to disable checking.

This includes non-EA trades and those not sharing the set MagicNumber. Trades will not be entered if this count would be exceeded.

Verify with your broker the maximum trades allowed. The MetaTrader default is 200 on the broker's side.

This setting is particularly important for grid trades. If all stop and limit orders were enabled, and if you had enough symbols to trade, the count could easily exceed the broker's maximum. The EA may encounter issues if trade placement fails and this setting is disabled.

### MaxTradesPerSymbol

Max trades to allow per symbol, including pending orders. Set to 0 to disable checking.

This setting counts only EA trades with the set MagicNumber.

### MinTradeMarginLevel

Percent of the account margin level to allow trade entries. Set to 0 to disable checking.

If the current margin level is below this percentage, trade entries will be disallowed.

### MaxSpreadCalc

Max spread in pips to allow trades. Set to 0 to disable checking.

### MaxSlippageCalc

Max slippage in pips to allow when entering a market order. The order is not placed if slippage exceeds this value. Set to 0 to disable checking.

This setting has no effect for ECN brokers, and has no effect for pending orders. Slippage is a concern for market makers who can set the price to fulfill orders.

### LotSizeCalc

Lot size to set new trades.

## Trade Signal Settings

### CloseOrderOnOppositeSignal

Close orders when an opposite entry signal is fired, even if no exit signal is currently active.

For example, if a long entry signal is fired, then all short trades for the symbol will exit.

### SignalRetraceOpenAfterExit

Allow entries on a retrace if no trades are active for the symbol.

### SignalRetraceOpenAfterDelay

Allow entries on a retrace after a delay from the last fired master signal, even if trades are currently active. The delay is set in SignalRetraceDelay below.

### SignalRetraceDelay

Delay for entries to be allowed on a retrace. The time units are set in TimeSettingUnit below.

E.g., if SignalRetraceTime = 60 seconds and a first Buy signal was fired at 12:00:30, then a second Buy signal fires at 12:00:50, then a new position is not opened because only 20 seconds have passed since the first signal fired.

But if that second Buy signal fired at 12:01:40, the new position is opened because 70 seconds have passed since the first signal fired.

The retrace delay is updated on every signal. E.g., if the second Buy signal fired at 12:00:50, the third Buy signal needs to fire later than 12:01:50 in order to be valid, as the delay is now based on the second Buy signal.

## Trade Delay Settings

### TimeSettingUnit

Below settings are specified in either Seconds, Milliseconds, or Ticks (applies only in real tick mode).

### EntryStableTime / ExitStableTime



Delay for entry filter signal to be stable.

After all entry or exit signals are stable, the master signal is fired immediately.

### TradeBetweenDelay

Delay between entering trades.

### ValueBetweenDelay

Delay between changing values, e.g., moving stops. This may be helpful if your broker limits requests, but should not be necessary in typical cases.

## Grid Settings

---

TradeModeType in General Trade Settings must be set to “Trade with grid orders” for these settings to take effect.

GridSetStopOrders, GridSetHedgeStopOrders, GridSetLimitOrders, and GridSetHedgeLimitOrders can be combined in multiple ways to create different grid types. For example, enabling GridSetStopOrders and GridSetHedgeStopOrders creates the basic grid, while enabling all four settings as well as GridOpenMarketInitial will create a dual grid as described in <http://forexop.com/trading-with-dual-grid-system/>

### GridSetStopOrders

Set stop orders in the signal direction.

E.g., if the trigger signal is long, then set buy stops above the current price.

### GridSetHedgeStopOrders

Set stop orders in the hedge direction.

E.g., if the trigger signal is long, then set sell stops below the current price.

### GridSetLimitOrders

Set limit orders in the signal direction.

E.g., if the trigger signal is long, then set sell limits above the current price.

### GridSetHedgeLimitOrders

Set limit orders in the hedge direction.

E.g., if the trigger signal is long, then set buy limits below the current price.

### GridOpenMarketInitial

Open a market order in the signal direction immediately upon setting a new grid

### GridCount

# of grid levels to place, per direction.

E.g., if `GridCount` = 5 and levels are placed at both signal and hedge directions, then a total of 10 grid levels will be placed.

### GridDistanceCalc

Distance between grid levels. Can be set to exact value or filter value notation.

## Grid Reset Settings

### GridClosePendingOnSignal

Close pending orders on signal. `CloseOrderOnOppositeSignal` takes effect if enabled.

### GridCloseMarketOnSignal

Close market orders on signal. `CloseOrderOnOppositeSignal` takes effect if enabled.

### GridOpenIfMarketExists

Open new grids if no stop orders exist, and even if market orders are still active.

## Stop Level Settings

The basic settings for SL and TP are the same, just configurable separately if you want separate SL/TP behaviors.

## Stop Level Basic Settings

### StopLossInitialEnabled / TakeProfitInitialEnabled

Set initial stop level when opening the order.

### StopLossInternal / TakeProfitInternal

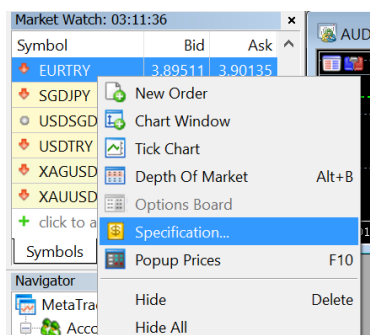
Enable internal tracking and firing of SL/TP by the EA.

This feature must be enabled if you offset SL/TP or enable `MinimumAdd` and want the EA to fire them at the original levels.

### StopLossMinimumAdd / TakeProfitMinimumAdd

Offset SL/TP by broker's minimum stop level per symbol.

The minimum stop level can be found in the symbol's specifications: right-click on a symbol in the Market Watch and select "Specification". It is listed in broker points.

A screenshot of the 'EURTRY contract specification' window. The window title is 'EURTRY contract specification'. The main content area is titled 'EURTRY, Euro vs Turkish Lira'. It contains a table with contract specifications. The 'Stops level' is highlighted in blue and set to 55. Other specifications include Digits (5), Contract size (100000), Spread (floating), Margin currency (EUR), Profit currency (TRY), Calculation (Forex), and Hedged margin (calculate using larger leg).

EURTRY, Euro vs Turkish Lira	
Digits	5
Contract size	100000
Spread	floating
Stops level	55
Margin currency	EUR
Profit currency	TRY
Calculation	Forex
Hedged margin	calculate using larger leg

Enabling this setting can prevent order error 130 (`ERR_INVALID_STOPS`) if SL/TP is below the broker's minimum. All orders' SL/TP will be offset regardless of the original level exceeding the minimum.

To fire at the original SL/TP level, `StopLossInternal/TakeProfitInternal` must be enabled. If these are not enabled, the SL/TP will fire at the offsetted level on the broker's side.

### `StopLossBelowMinimumAction / TakeProfitBelowMinimumAction`

Sets the action to take if the stop level falls below the broker's minimum per symbol.

- Drop New Order / Do not Modify Existing Order
  - Aborts placing a new order. If the order already exists, it is not modified.
- Do not set stop level for new order
  - Places a new order without setting a stop level. If the order already exists, it is not modified
- Set stop level equal to minimum
  - Set stop level to the broker's minimum. This setting is not recommended if `StopLossCalc/TakeProfitCalc` is set or `StopLossInternal/TakeProfitInternal` is enabled, as the original stop level cannot be retrieved with this setting.

### `StopLossCalc / TakeProfitCalc`

Sets an offset for the stop level in pips. Can be set to exact value or filter value notation.

`StopLossCalc` must be a negative value; if using filter value notation, it must be factored by a negative number (e.g., `ATR-H1|Factor|-1.0`). `TakeProfitCalc` is positive.

To fire SL/TP at the original level, `StopLossInternal/TakeProfitInternal` must be enabled. If these are not enabled, the SL/TP will fire at the offsetted level on the broker's side.

## Break Even Settings

---

### `BreakEvenEnabled`

Enable breakeven stop loss after the current price profits a certain distance from the entry price.

Breakeven behavior will proceed even if `StopLossEnabled = false`.

### `BreakEvenProfit`

Stop level in pips to set beyond the entry price.

E.g., if `BreakEvenProfit = 1.5`, then after `BreakEvenJumpDistanceCalc` is hit, set stop loss to 1.5 pips beyond the entry price.

### `BreakEvenJumpDistanceCalc`

Pips after the entry price to engage breakeven stop level. Can be set to exact value or filter value notation.

E.g., if `BreakEvenJumpDistanceCalc = 10`, engage breakeven stop level after the order profits at 10 pips.

## Trailing Stop Loss Settings

---

### TrailingStopEnabled

Enable trailing stop loss that moves behind the current price.

If both `JumpingStopEnabled` and `TrailingStopEnabled = true`, then jumping stop takes precedence over trailing stop. One way you can combine these features is by setting `JumpAfterBreakEvenOnly` while the trailing stop engages before breakeven.

### TrailAfterBreakEvenOnly

Engage trailing stop only after the current price profits at breakeven level, set by `BreakEvenJumpDistanceCalc`.

This setting will take effect even if `BreakEvenEnabled = false`.

### TrailingStopCalc

Pips to trail behind the current price. Can be set to exact value or filter value notation.

## Jumping Stop Loss Settings

---

### JumpingStopEnabled

Enable jump stop loss that jumps behind the current price at set intervals.

If both `JumpingStopEnabled` and `TrailingStopEnabled = true`, then jumping stop takes precedence over trailing stop. One way you can combine these features is by setting `JumpAfterBreakEvenOnly` while the trailing stop engages before breakeven.

### JumpAfterBreakEvenOnly

Engage jumping stop only after the current price profits at breakeven level, set by `BreakEvenJumpDistanceCalc`.

This setting will take effect even if `BreakEvenEnabled = false`.

### JumpingStopCalc

Pip interval to jump behind the current price. Can be set to exact value or filter value notation.

The jump stop will be engaged once the current price hits this level, and the stop will be moved one interval behind the current level.

## Basket Exit Settings

---

Only currently opened orders are counted towards the basket, and only those orders placed by the EA (same magic number) are counted.

Both master baskets and symbol baskets are supported. Symbol baskets allow for indicator-based stop levels, whereas master baskets are specified in exact pips.

### BasketClosePendings

When closing a basket, either close pendings or leave them active.

## Master Basket Stop Levels

---

### [BasketMasterInitialStopLossMode / BasketMasterInitialTakeProfitMode](#)

Enable basket initial stop levels.

### [BasketStopLossValue / BasketTakeProfitValue](#)

Stop level value, exact pips only. [BasketStopLossValue](#) must be negative; [BasketTakeProfitValue](#) is positive.

### [BasketMaxLosingPerDay / BasketMaxWinningPerDay](#)

Max master basket count before disabling trades for the rest of the day. This count resets upon a new day.

### [Master Break Even Stop](#)

#### [BasketMasterBreakEvenStopEnabled](#)

Enable break even stop. This sets the basket stop to [BasketMasterBreakEvenProfit](#) above breakeven once price level crosses [BasketMasterBreakEvenJumpDistance](#).

If [BasketMasterJumpingStopEnabled](#) is true, this must also be set true if you want the first jump to take effect.

### [Master Trailing Stop](#)

#### [BasketMasterTrailingStopEnabled](#)

Enable immediate trailing stop, defined in pips on [BasketMasterTrailingStop](#).

#### [BasketMasterTrailByBreakEven](#)

Set to enable trailing stop always, before, or after break even.

### [Master Jumping Stop](#)

#### [BasketMasterJumpingStopEnabled](#)

Enable jumping stop, defined in pips on [BasketMasterJumpingStop](#).

## Symbol Basket Stop Levels

---

Symbol baskets allow for indicator-based stop levels and offsets.

### [Symbol Stop Levels](#)

#### [BasketSymbolInitialStopLossMode / BasketSymbolInitialTakeProfitMode](#)

Enable basket initial stop levels.

#### [BasketSymbolStopLossCalc / BasketSymbolTakeProfitCalc](#)

Stop level value, in pips. [BasketSymbolStopLossCalc](#) must be negative; [BasketSymbolTakeProfitCalc](#) is positive.

#### [BasketSymbolMaxLosingPerDay / BasketSymbolMaxWinningPerDay](#)

Max symbol basket count, per symbol, before disabling trades for the rest of the day. This count resets upon a new day.

### Symbol Stop Level Offsets

This feature allows the basket to remain active when the profit level is close to stopping out and a new trade is entered. When this happens, the new trade might lower the profit level due to spread and cause a stop out immediately. Setting a dynamic offset allows for the stop level to change temporarily so this does not happen.

The offset is determined by looking at the last one or more trades according to `RelatedSeconds` and `LimitSeconds`. For each trade that falls under these time limits, the offset is increased by `OffsetCalc`.

#### BasketSymbolStopLossOffsetByOrder / BasketSymbolTakeProfitOffsetByOrder

Enable offsetted stop level.

#### BasketSymbolStopLossOffsetCalc / BasketSymbolTakeProfitOffsetCalc

Offset to change the stop level.

#### BasketSymbolStopLossOffsetRelatedSeconds / BasketSymbolTakeProfitOffsetRelatedSeconds

Number of seconds to look back for multiple new trades.

#### BasketSymbolStopLossOffsetLimitSeconds / BasketSymbolTakeProfitOffsetLimitSeconds

Number of seconds to consider a trade. If a trade's starting time is greater than this, the trade is not considered for offsetting the stop level.

#### BasketSymbolStopLossOffsetLimitLower / BasketSymbolStopLossOffsetLimitUpper BasketSymbolTakeProfitOffsetLimitLower / BasketSymbolTakeProfitOffsetLimitUpper

Enable upper and lower profit level limits to engage an offset. If the profit level exceeds either of these limits, the offset is not engaged.

The profit level limit is set by `LimitLowerCalc` and `LimitUpperCalc`, respectively.

### Master Break Even Stop

#### BasketSymbolBreakEvenStopEnabled

Enable break even stop. This sets the basket stop to `BasketSymbolBreakEvenProfit` above breakeven once price level crosses `BasketSymbolBreakEvenJumpDistanceCalc`.

If `BasketSymbolJumpingStopEnabled` is true, this must also be set true if you want the first jump to take effect.

### Symbol Trailing Stop

#### BasketSymbolTrailingStopEnabled

Enable immediate trailing stop, defined in pips on `BasketSymbolTrailingStopCalc`.

#### BasketSymbolTrailByBreakEven

Set to enable trailing stop always, before, or after break even.

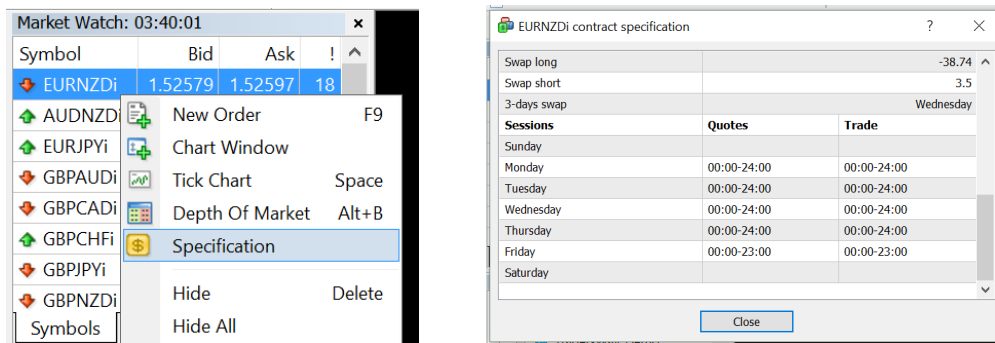
### [Symbol Jumping Stop](#)

#### [BasketSymbolJumpingStopEnabled](#)

Enable jumping stop, defined in pips on [BasketSymbolJumpingStopCalc](#).

## Schedule Settings

Trades can follow either a custom schedule or the broker's specified schedule per symbol. The broker's schedule can be viewed by right-clicking on a symbol in Market Watch and selecting "Specification".



**Note that the schedule may not update in case of holidays or special closings.** You'll also need to check your broker's terms as to when they charge swap. In these cases, you'll need to intervene manually or set a custom schedule.

### [SchedCustom](#)

A custom list of opening and closing times, separated by pipe character `|`.

An opening time starts with the plus character `+` and a closing time starts with the minus character `-`. Times can be specified in the following formats, in 24-hour time:

- `-hh:mm` – Fires every day
- `-d hh:mm` – Fires on a specific day of the week. 0 = Sunday, 1 = Monday, ..., 6 = Saturday
- `-DD.MM hh:mm` – Fires on a specific date of the current year
- `-DD.MM.YYYY hh:mm` – Fires on a specific date and year

It's assumed that orders will open daily at midnight (`+00:00`) unless you specify that orders remain closed at midnight (`-00:00`).

### [SchedCustomType](#)

Specify if [SchedCustom](#) times are in GMT, broker time, or computer local time.

### [SchedCloseCustom](#)

Close orders according to a custom schedule set in [SchedCustom](#).

The [SchedCloseMinutes](#) and [SchedOpenMinutes](#) delays and [SchedGapIgnoreMinutes](#) do not apply to custom closing times: the exact custom times are followed.

The broker schedule settings will continue to apply if enabled, with custom closing times taking precedence. If the custom schedule does not fire a closing time, orders can then still be closed by the Daily, 3-day Swap, and Weekend settings.

### [SchedCloseDaily](#)

Close orders at the end of the session every day.

If [SchedCloseBySwapDaily](#) is enabled, only orders that exceed the swap threshold will be closed.

### [SchedClose3DaySwap](#)

Close orders at the end of the session on the weekday designated as the 3-day swap.

E.g., if the broker's 3-day swap is set to Wednesday, orders will close at the end of the Wednesday session (such as 23:59 on Wednesday).

If [SchedCloseBySwap3DaySwap](#) is enabled, only orders that exceed the swap threshold will be closed.

### [SchedCloseWeekend](#)

Close orders at the end of the session right before a weekend.

### [SchedCloseSession](#)

Close orders at the end of every session. This can be useful if there is more than one session in a single day.

### [SchedGapIgnoreMinutes](#)

If [SchedCloseSession](#) = true, ignore new sessions that start X minutes after the current session. Orders will not close if the gap is less than X minutes.

E.g., if [SchedGapIgnoreMinutes](#) = 15, the current session ends at 23:59, and the next session starts at 12:10, then orders will not be closed because the gap between the sessions is less than 15 minutes. The [SchedOpenMinutesSession](#) delay will also not engage because the session gap is too small to be considered a new session.

This setting does not apply to [SchedCustom](#).

## [Order closing settings](#)

### [SchedCloseOrderProfit](#)

Close only profitable orders, losing orders, or all orders.

### [SchedCloseOrderOp](#)

Choose whether to close long orders, short orders, or both types of orders.

### [SchedClosePendings](#)

Close pending orders or leave them open.

### [SchedCloseBySwapDaily](#) / [SchedCloseBySwap3DaySwap](#)

Only close orders for the respective setting that exceed [SchedSwapThresholdCalc](#).



### SchedSwapThresholdCalc

Swap threshold for closing trades by SchedCloseDaily and SchedClose3DaySwap. Can be set to exact value or filter value notation.

If this value is positive, payable swap must be numerically greater than the threshold for orders to close.

If this value is negative, payable swap must be numerically less than the threshold for orders to close.

If evaluating by 3-day swap, threshold is multiplied by 3.

### Delay settings

#### SchedCloseMinutes

Minutes to close orders before the end of the session.

#### SchedOpenMinutesDaily

Minutes to delay opening orders after a new day starts.

#### SchedOpenMinutesSession

Minutes to delay opening orders after a session starts.

#### SchedOpenMinutesWeekend

Minutes to delay opening orders after a weekend.

## Filter Settings

---

A filter is an indicator that can have any arbitrary number of subfilters with different settings, such as Stoch M15, H1, and H4. Depending on the filter, there can be three types of subfilters:

- Value – Filter provides price or pip values and does not produce signals. Used for SL/TP and other value settings.
- Entry – Filter provides entry signals.
- Exit – Filter provides exit signals.

Per subfilter type, there will be a series of settings corresponding to that type, such as:

- Stoch\_Entry\_Modes, Stoch\_Entry\_Names, Stoch\_Entry\_TimeFrame, Stoch\_Entry\_KPeriod, ...
- Stoch\_Exit\_Modes, Stoch\_Exit\_Names, Stoch\_Exit\_TimeFrame, Stoch\_Exit\_KPeriod, ...

Multiple subfilters can be specified per type, but all settings of that type must have the same number of subfilters defined.

For example, this Stoch filter has an Entry type and an Exit type. There are 3 Entry subfilters and 1 Exit subfilter. All Entry settings have 3 subfilters defined, whereas all Exit settings have 1 subfilter defined.

ab	Lbl_Stoch_1	_____ Stoch Settings [Stoch] _____
ab	Stoch_Entry_Modes	a=1 b=1 c=1
ab	Stoch_Entry_Names	a=M15 b=M30 c=M60
ab	Stoch_Exit_Modes	a=1
ab	Stoch_Exit_Names	a=M15
ab	LbL_Stoch_Entry	---- Stoch Entry Settings ----
ab	Stoch_Entry_TimeFrame	a=15 b=30 c=60
ab	Stoch_Entry_KPeriod	a=5 b=5 c=5
ab	Stoch_Entry_DPeriod	a=3 b=3 c=3
ab	Stoch_Entry_Slowing	a=3 b=3 c=3
ab	Stoch_Entry_Method	a=3 b=3 c=3
ab	Stoch_Entry_PriceField	a=0 b=0 c=0
ab	Stoch_Entry_Shift	a=0 b=0 c=0
ab	Stoch_Entry_BuySellZone	a=22.0 b=22.0 c=22.0
ab	LbL_Stoch_Exit	---- Stoch Exit Settings ----
ab	Stoch_Exit_TimeFrame	a=15
ab	Stoch_Exit_KPeriod	a=5
ab	Stoch_Exit_DPeriod	a=3
ab	Stoch_Exit_Slowing	a=3
ab	Stoch_Exit_Method	a=3
ab	Stoch_Exit_PriceField	a=0
ab	Stoch_Exit_Shift	a=0
ab	Stoch_Exit_BuySellZone	a=30.0

## Subfilter Setting Notation

An arbitrary number of subfilters can be defined like this:

a : 15      |      b : 30  
 Index | Value    | Index | Value

Subfilters are separated by a pipe character, but the setting text must not end with a trailing pipe.

Every subfilter is keyed with a letter and the key must be specified in order (A, B, C, D...). Below is an example of three common settings:

Setting	Values	Description
ATR_Value_Mode	a:1 b:1 c:1	0 - Disabled, 1 - Normal, 2 - Opposite, 3 - Not opposite
ATR_Value_Name	a:H1 b:H4 c:D1	Subfilter name so you can access them on value notation
ATR_Value_Hidden	*:false	Hide them from the dashboard, but continue to calculate.

The Mode setting determines the number of subfilters for the group. All subfilters you need must be specified here. In the example, three subfilters are specified for ATR (A, B, and C).

All other settings do not need to specify all the subfilters, as long as a default is defined for that setting. A default takes an asterisk \* as its key, and fills the value for all unspecified filters for that setting. In the example, the Hidden setting fills all three subfilters with false.

In addition, you can specify per-subfilter values if they are different from the default. This example means to fill subfilter C with true, but all others (e.g., A and B) with false:

`*:false|c:true`

Make sure that all settings in a group define the same number of subfilters (except where a default is listed). If a setting specifies too many subfilters or too few (where a default is not defined), or specifies an index outside of those defined in Mode, then the program force-quits and displays a message.

**Save your setfiles before running the EA; the settings will be lost if MultiOp force-quits.**

### Optimization Settings

See “Optimization Notation” above. Each subfilter value can be redirected to an optimization setting (Opt\_0, Opt\_1, Opt\_2...) so that the values can be modified in the optimizer.

## Standard Subfilter Settings

---

All subfilters will have at least the following standard settings:

### [Filter]\_[Type]\_Mode

Mode to engage the subfilter:

- 0 – Disable: Do not calculate, do not give signals, and hide from the dashboard.
- 1 – Normal: Calculate subfilter and give signals. Master signal is issued if each subfilter produces the same Buy or Sell signal.
- 2 – Opposite: Master signal is issued opposite of the subfilter’s Buy or Sell signal.
- 3 – Not Opposite: Master signal can be issued as long as this subfilter does not disagree with other subfilters. E.g., If the master signal is Buy, it will fire if this subfilter produces an empty signal or a Buy signal, and it will be canceled if this subfilter produces a Sell signal.

### [Filter]\_[Type]\_Name

Subfilter name to allow access by filter value notation and to display on the dashboard. Names should be unique. For exit subfilters, it’s recommended to suffix names with an x, such as “Stoch-M15x”.

### [Filter]\_[Type]\_Hidden

Value subfilters only; display or hide the subfilter from the dashboard. Subfilter will still calculate according to [Filter]\_[Type]\_Mode and can be accessed by filter value notation.

## Optimization Values

---

These settings exist in order to allow the optimizer to access value notations. See “Optimization Notation” above. If a value redirect is specified, the value will always be pulled from one of these settings, whether or not the optimizer is running.

With these settings, you can specify Value (default when optimizer is not running), Start, Step, and Stop.

Variable	Value	Start	Step	Stop	Steps
***** Optimization Values *****					
Notation [value]@[index]	Set start/step/stop below for the setting index				
Opt_0_Comment	Lot size calc				
<input checked="" type="checkbox"/> Opt_0	0.01	0.01	0.01	0.05	5
Opt_1_Comment	GTF enable				
Opt_1	0	0	1	1	
Opt_2_Comment	ATR vs StdDev symbol basket stops				
<input checked="" type="checkbox"/> Opt_2	1	0	1	1	2
Opt_3_Comment	Symbol mod stop factors				
Opt_3	1.95	1.7	.25	2.2	

Settings
Inputs
Agents
Journal

## Runtime Settings

### Error Settings

[ErrorTerminalLevel](#) / [ErrorFileLevel](#) / [ErrorAlertLevel](#)

Severity levels to print for the Terminal expert log, file log, and alerts. Set to None to disable the output.

[ErrorLogFileName](#)

Filename of the file log, if enabled. The file will be located in the profile Files folder.

Leave blank to auto-generate the filename by a millisecond timestamp. If the file already exists, the log will be added onto the existing content.

### Display Settings

[DisplayShow](#)

Show visual dashboard on the chart.

[DisplayShowTable](#)

Show indicator signals on the chart. If this is false, only the basic info header is displayed.

[DisplayShowBasketSymbolLongShort](#)

Show symbol trade statistics. This is enabled automatically if symbol baskets are enabled.

[DisplaySignalInternal](#)

Show trade signals as they are represented internally in code. Used for debugging.

[DisplayColor](#)

Show buy and sell signals as green and red, respectively. If this is false, all signals are shown without color.

[DisplayFontColorDefault](#)

Default font color.

## DisplayScale

Increase or decrease the dashboard size. 0 = Smallest, 1 = Bigger, 2 = Largest.

Size 0 may not display correctly on systems set to 90 DPI, which is default on many systems before Windows 8. Best results will be shown on 144 DPI.

## DisplayBackgroundColor

Set a background color that automatically hides the chart and shows only the dashboard. Set to “None” to disable.

In visual backtesting mode, the chart will always appear, even if this setting is configured.

## DisplayShowChart

If DisplayBackgroundColor is set, this will show the chart instead of hiding it.

## Cycle Settings

System cycle is a performance setting and is separate from “Trade Delay Settings”, set above. The default cycle length of 1 second should suffice for most systems – for example, actual cycle processing can take between 30-300 milliseconds on a Core i5 system, depending on complexity.

If you want faster cycling, you can set a cycle by milliseconds. If your system struggles to complete cycles in time, you can try slowing down the cycle.

However, if you have a slower cycle (such as 5 seconds) and a delay is set to a faster interval (such as 1 second), that delay will only fire once, when a new cycle fires.

## CycleMode

Sets whether to specify cycle length as seconds, milliseconds, or real ticks.

If you set this to real ticks, SingleSymbolMode must also be set.

This setting is independent from TimeSettingUnit, which dictates signal delays. TimeSettingUnit is processed according to actual time passed, not how many cycles have passed (unless it’s set to real ticks.)

## CycleLength

Length of time for each cycle, seconds or milliseconds. If real tick mode is enabled, this is ignored.

## Frequently Asked Questions

---

### There is a master entry signal but no trade was entered, why?

MultiOp checks for entry safety before making the trade. A variety of reasons can prevent a trade from entering, even if nothing is printed in the Expert log:

1. Is TradeEntryEnabled true?
2. Is MaxTradesPerAccount or MaxTradesPerSymbol exceeded?

3. Is MinTradeMarginLevel exceeded?
4. Is MaxSpread exceeded?
5. Has TradeBetweenDelay elapsed?
6. Is there an exit signal that conflicts with the entry signal? (I.e., if there is a long entry signal, is there also a long exit signal that would have exited the trade immediately?)
7. Is the entry signal on a retrace? Check your SignalRetrace settings.
8. Is your configured SL/TP below the broker's minimum stop level requirement? To dodge this failure, check your SL/TPBelowMinimumAction setting and possibly enable SL/TPMinimumAdd and SL/TPInternal.
9. Has the basket count exceeded the daily win/loss limits, if enabled?
10. Is the trade schedule currently in close?