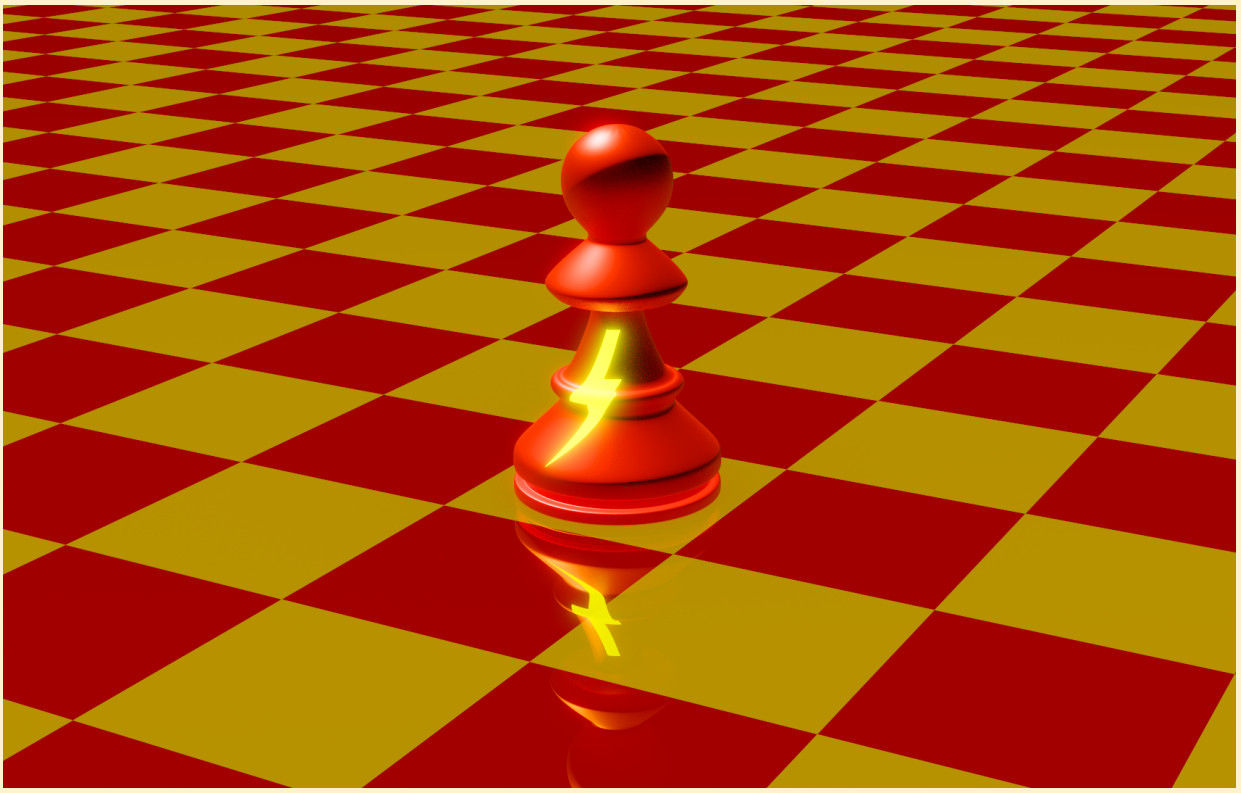
AlgoMaster NNFX



User's Guide

Made by Manuel Alejandro Cercós Pérez.

Index

1. Introduction

AlgoMaster NNFX is a backtesting tool to test, create, improve and get statistics from an NNFX algorithm, which allows you to run the numbers and get multiple answers from your system more easily and quickly:

-Do these settings improve my system or not?

-What effect does avoiding exposure and news have in my final profit?

-What happens if I trade with half risk when EVZ is below 7, instead of 8?

-Etc.

This program has been developed and continually improved since September 2020 to provide the best backtesting and algorithm building experience.

You don't need to read this whole document to be able to use the program. However, it may contain an answer to some problem you may encounter later on, or the explanation of some parameter that you don't know what it does.

Disclaimer

Backtesting and building algorithms automatically can be faster, but it can also be as difficult as doing it by hand. To use this program effectively you will need to get experience and distinguish when a system is good and when it's overfitted (based on its statistics, behaviours and/or forward tests). You can get some practice using the demo version, which allows you to perform backtests with 3 currency pairs (EURUSD, GBPCHF and USDCHF).

Most indicators are supported by AlgoMaster NNFX, however there may be some specific cases that cannot be tested. Either if you rent the product or use it in demo, you can suggest new testing modes by writing a message via MQL5 (if it's not too complicated, and if it can possibly be done).

This program works properly in the **Daily Timeframe**. Some parts of it have limitations when applied to lower (or higher) timeframes: for instance, EVZ only

works on daily (currently) and news are only considered in the next candle so they are not very useful in other timeframes (they would filter too little in lower, and filter everything in higher timeframes).

The performance of an algorithm built using this system shouldn't be taken as a guarantee of future success of that system in real trading. You should visually review the behaviour of the algorithm, check if there are potential errors (in the algorithm, the indicators or the AlgoMaster NNFX program itself) and forward test.

This program may have considerations in place that don't align exactly with your system or your own interpretations of what an NNFX algorithm should be. Still, some nuances and money-management parts of it can be modified. **If you have some interpretation that is not considered in the program**, feel free to **suggest** it to the author and it may be added in next versions (depending on its complexity, and technical viability).

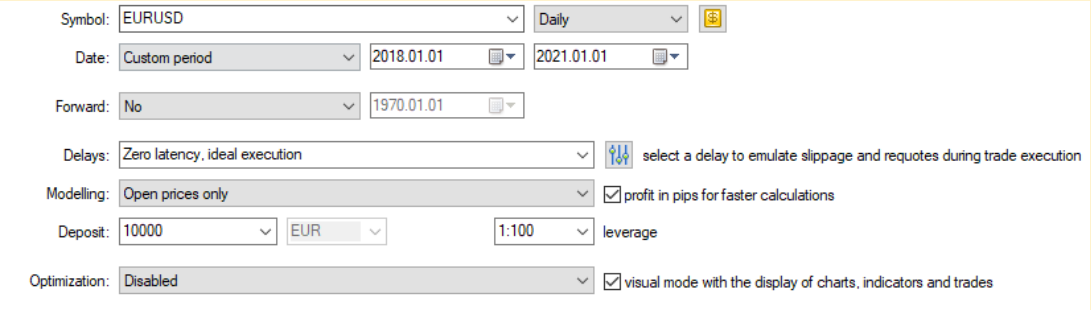
This product is not affiliated with/sponsored or directly recommended by VP (No Nonsense Forex), NNFX Discord and any other person/group related to VP. Some people may consider that this product isn't compatible with the NNFX method, and others may think it's a better way of creating and/or backtesting algorithms. Before renting this product you should consider if it fits your necessities and test it using the demo version to make an informed decision.

How to perform the backtest

The best settings to perform the backtest are with "Open prices only", especially when testing multiple pairs. You can execute backtests in lower time frames, too. The **symbol** that you choose will be the one where the trades and events are drawn, but it doesn't have any preference (however, if that symbol has data gaps it can alter the backtest results).

It's recommended to keep a consistent period to execute the backtests effectively, so you can compare between different results.

The deposit will be used as a reference for the results. The leverage doesn't have influence in the backtest.



If the program detects multiple ticks for the same candle, it will give you a warning to remind you of using the Open Prices Only mode.

```
################################################################################################################
################################################################################################################
WARNING:
We have detected that this optimization was performed using multiple ticks per candle.
This backtester is more efficient when using the Modelling method "OPEN PRICES ONLY"
################################################################################################################
################################################################################################################
```

2. Internal Program Flow

The system executes the same algorithm at each candle open (using values from the 2 last candles). There are 7 possible states: No trade, Main Signal, Main Catch Up, Pullback, One Candle Rule, Continuation and Missed Trade. Depending on the conditions of the indicators, they change from one to another, and a baseline cross overrides all of them. The steps to check the algorithm, which are performed individually for each currency pair, are as follows:

-1. Check active trades

If there is a trade (or trades) active in that currency pair, check if **stop-loss** or **take-profit** got hit. If take-profit got hit, set the second half of the trade to **breakeven** (or close all if not scaling-out). If price has travelled more than the minimum to start moving the **trailing stop**, then it is also updated (at a distance the same as the initial stop).

-2. Check for Exits

Check for exit signals for the active trades from:

- Exit indicator (if using Exit Indicator).
- Opposite baseline cross (if using Baseline).
- Main indicator (if it is set to be used as exit too).
- News (if used, when a trade isn't in breakeven yet, it would be closed).

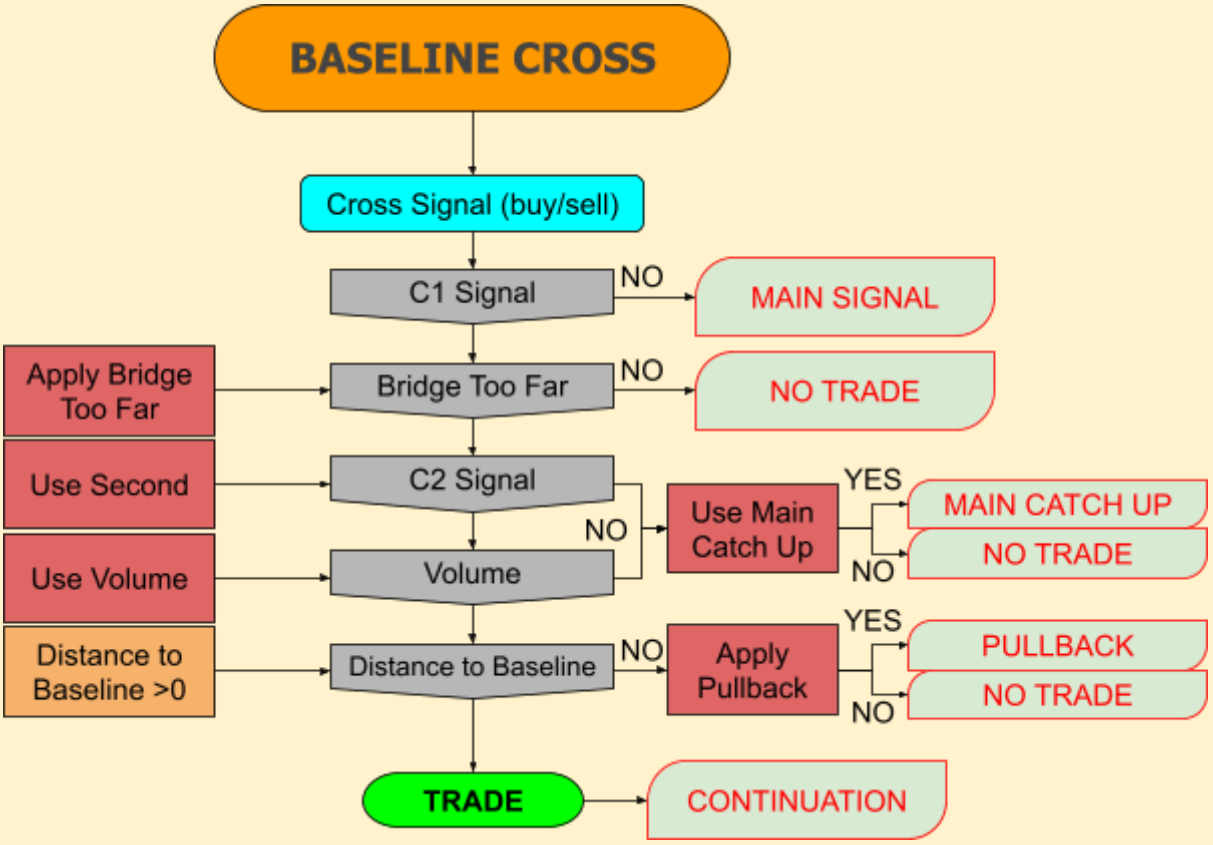
Each one of these exit modes is independent from the rest, so you can have all of them be applied at the same time.

If the exit indicator has given an exit signal, a continuation trade cannot be taken in that same candle (with any of the other exit types, the conditions would already invalidate any continuation trade).

-3. Check for Baseline Cross

This condition is only checked and applied when a baseline indicator is in use. When it is used, it overrides all of the rest of the states.

A baseline cross creates a cross signal (buy or sell). After that, the other conditions that need to be valid and are in use are checked: C1 signal, Bridge Too Far, C2, Volume and distance. If all of them agree, a trade is opened. You can see how the states would be modified in each case in the picture below.



For each condition to be applied (gray blocks), the input on the left needs to be valid (red/orange blocks). If one condition is not fulfilled, the current state changes to the one on its right. If all conditions are valid, a trade is performed and the state also changes (to **Continuation**).

-4. Check Current State

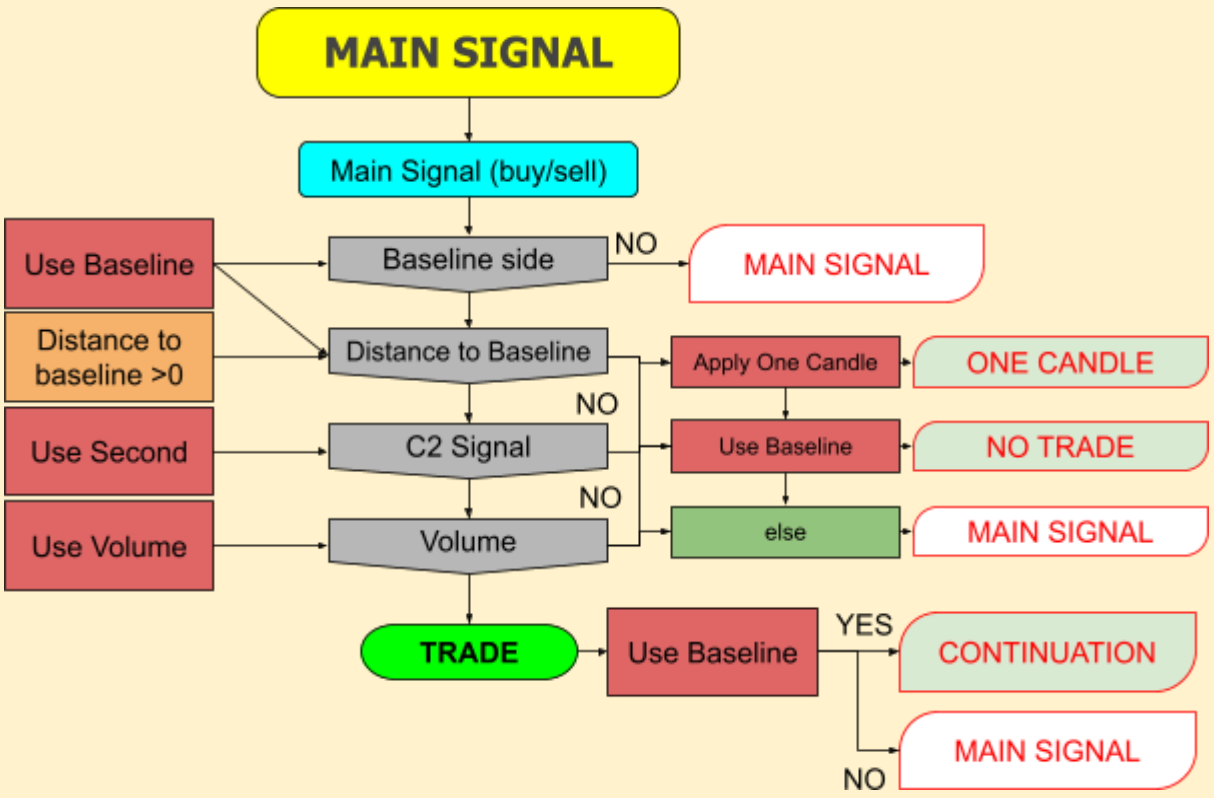
If there is no baseline cross in the current candle, the state of the algorithm is checked. After that check, the current state can continue the same for the next candle or change to another state. There are 7 different states:

No Trade

This state usually occurs after a signal is invalidated by distance to baseline or if pullbacks / one candle rules fail. No signal is considered in this state until C1 flips to the opposite direction. When C1 changes direction, the state changes to **Main Signal** to wait for a valid C1 entry. This state can only occur when a baseline is being used.

Main Signal

When the algorithm is in Main Signal state, it means that it is waiting for a valid C1 signal to happen. There are several filters that apply in a main signal, as can be seen below:



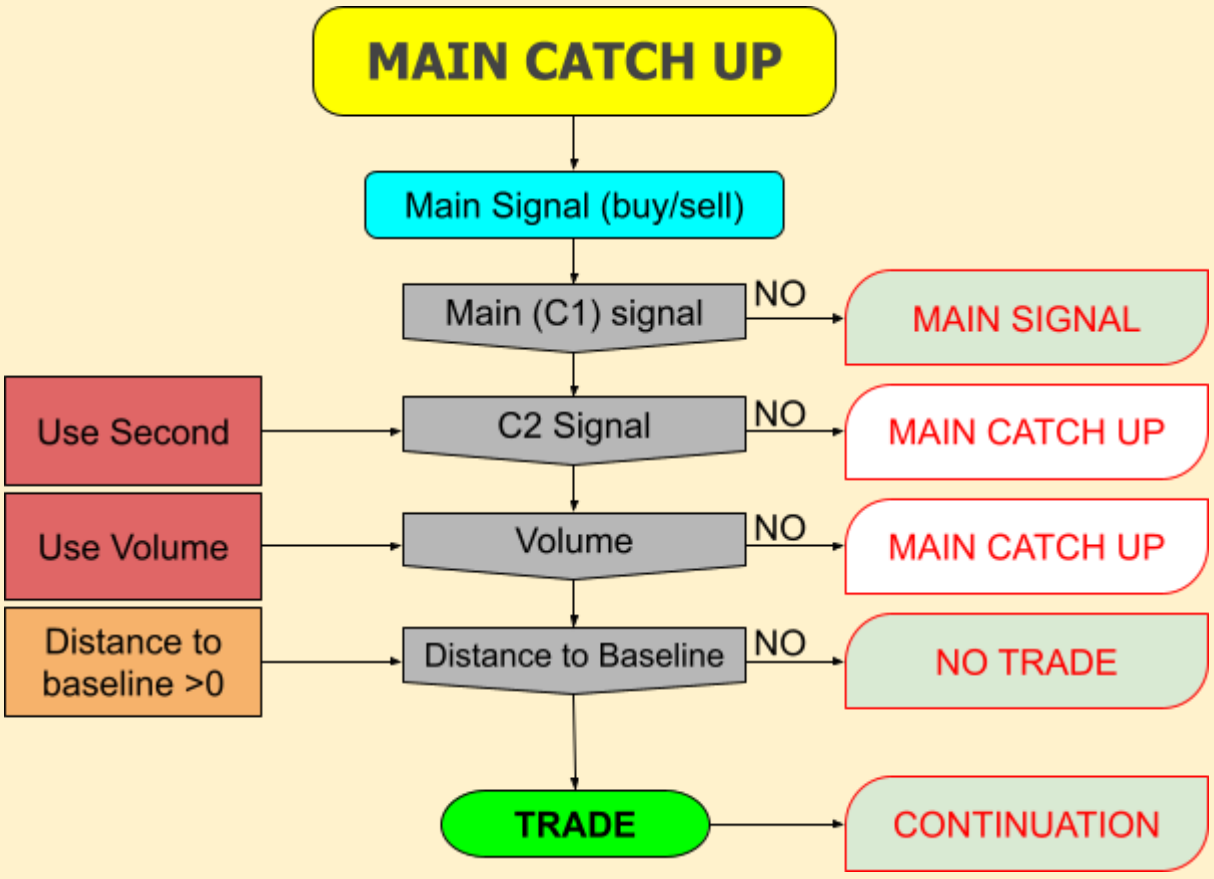
If the baseline side fails, it means that the signal is on the opposite side of the baseline (for example, a sell signal above the baseline), then it would still wait for a valid C1 Signal.

The other filters have the same effect when any of them fails: if One Candle Rule is applied, the state changes to **One Candle**; if baseline is being used but not the One Candle Rule, it changes state to **No Trade**; and if none of the above is applied, it stays at **Main Signal**.

If all conditions are met, the state changes to **Continuation** if a baseline is used, if not, it continues in **Main Signal** state.

Main Catch Up

This state is used after a **Baseline Cross** with C1 confirmation, but where volume or C2 don't align. In that case, the algorithm would wait for them to catch up, and when they do the signal would be taken if it hasn't got too far yet. This state can only be reached if baseline and main catch up are both used.

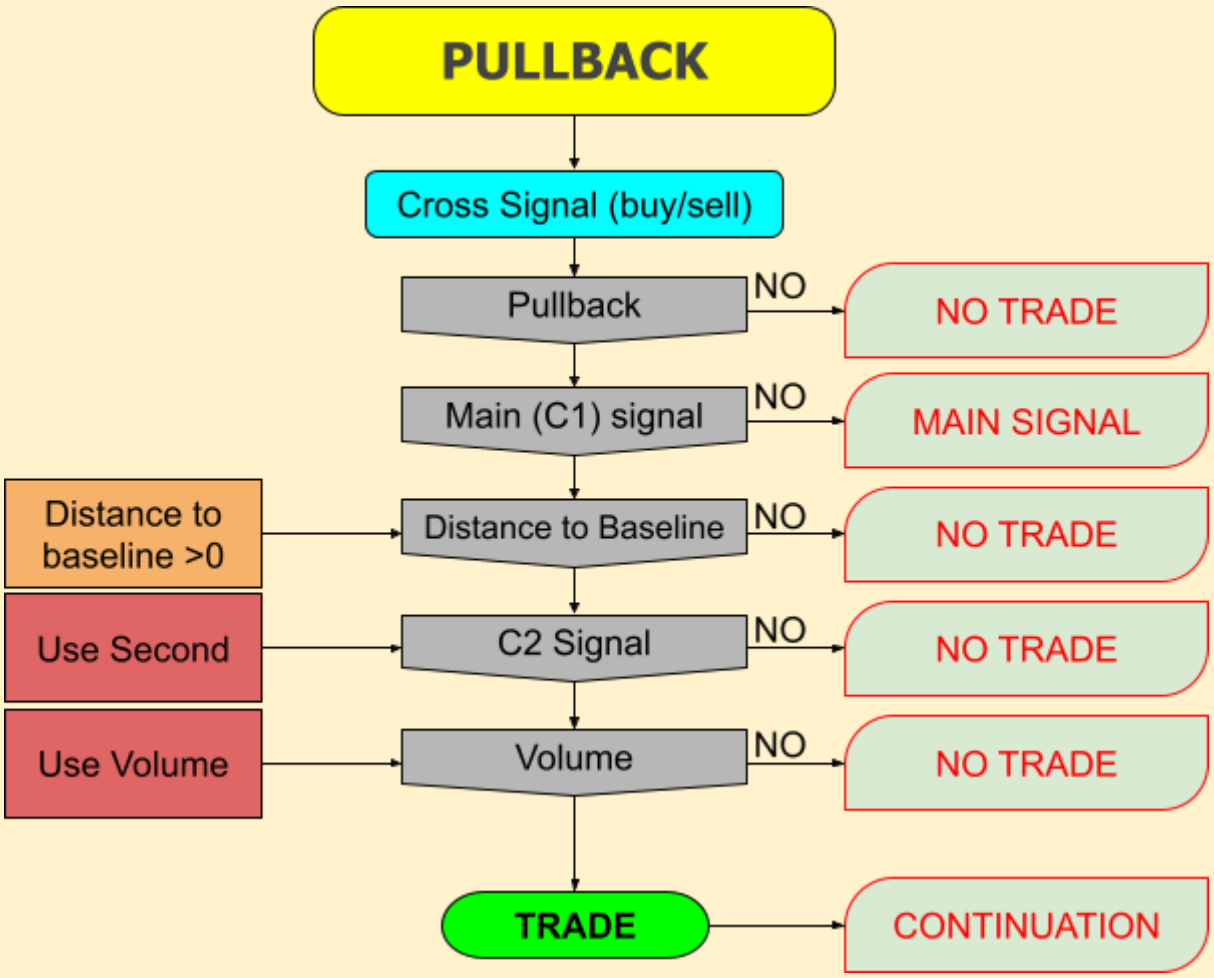


If the C1 signal flips before C2 and/or volume catch up, the state changes to **Main Signal**. If C2 or volume don't catch up, it keeps waiting in the same state. Notice that Distance to Baseline is only considered after C2 and Volume are both valid (the same as the C1 current signal), at that point the signal is given, but if it's too far from the baseline it is invalidated and the state changes to **No Trade**.

If the trade is completed, the state changes to **Continuation**.

Pullback

Pullback state is only reached after a baseline cross signal (with C1, C2 and volume agreeing) that got too far (distance to baseline multiplier * ATR). This state is only executed once after a baseline cross (in the candle next to that cross) if the input Apply Pullbacks is activated.



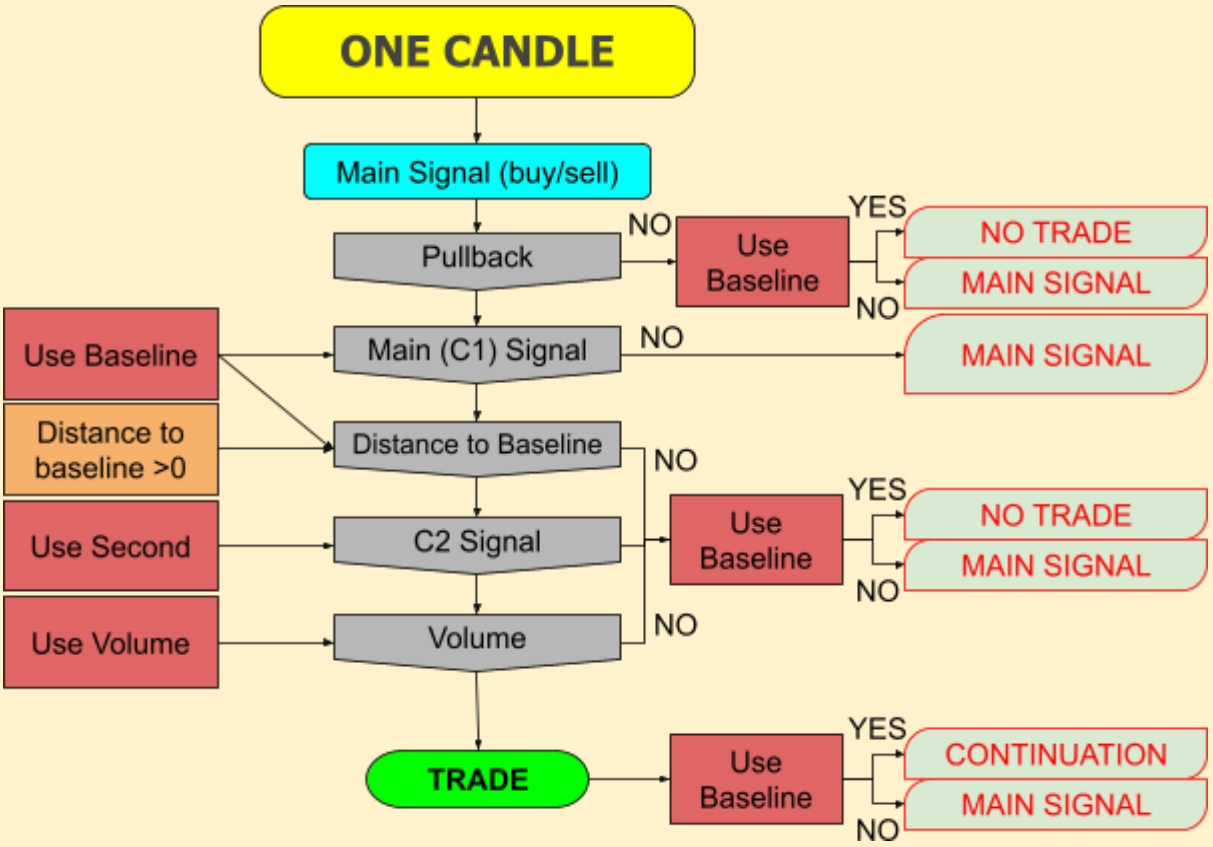
First, if there is no pullback on the candle after (close price is more advanced in the trade direction than it was in last close), then the state changes to **No Trade**.

Second, if C1 flips on the next candle, the state is set to **Main Signal**. Finally, if one of the other indicators (C2 or volume) fails or the distance to the baseline is still greater than the minimum required, the state changes to **No Trade**.

If there is a pullback and the rest of conditions are met, a trade is performed and the next state would be **Continuation**.

One Candle

One Candle Rule is applied after a C1 entry (**Main Signal**) where either volume or C2 fail to agree or the distance to baseline is too big. In that case, the algorithm waits one candle to see if all conditions align **and** there is a pullback in price. This state can only be reached if Apply One Candle Rule is set to true.



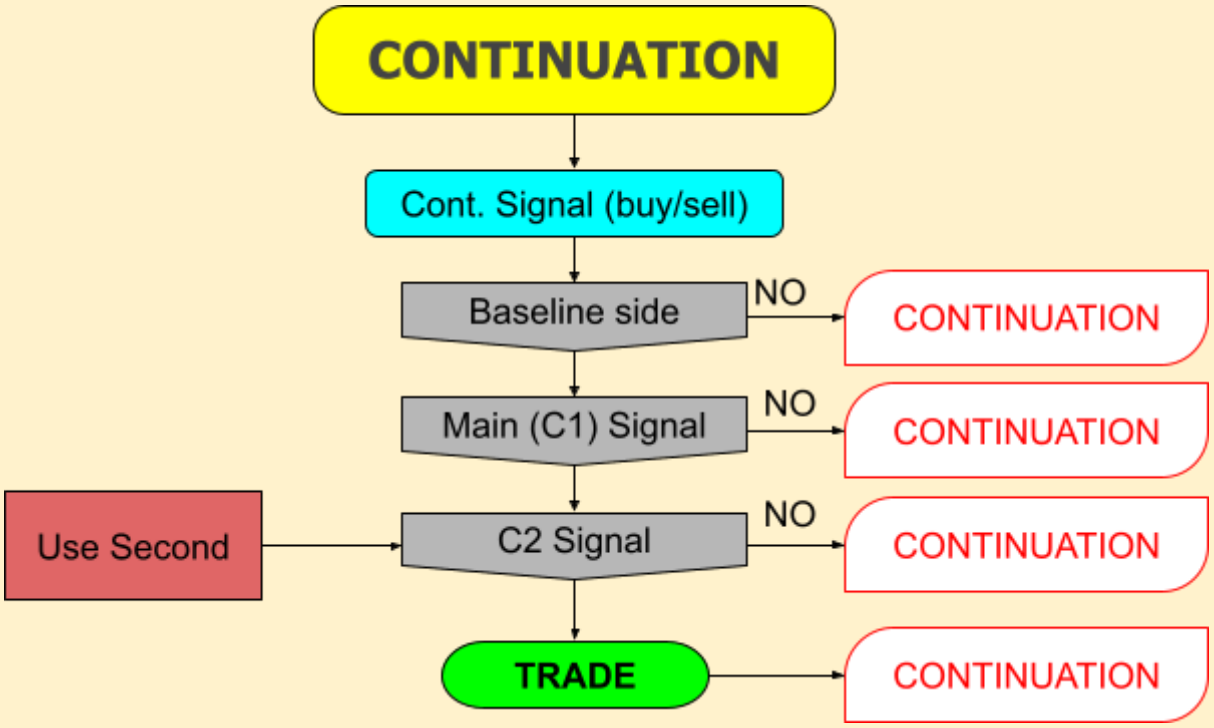
If C1 Signal fails, it means that its signal has reversed in that candle, so no trade is taken and the algorithm waits for a new signal (state **Main Signal**). Pullbacks are checked before all other conditions, and after that (and C1), distance, C2 and volume are checked: if one of them fails the state changes to **No Trade** if a baseline is being used, if there is no baseline it returns to **Main Signal**.

If all conditions are met, a trade is performed and the state changes to **Continuation** (if using baseline) or **Main Signal** (if no baseline is in place).

Continuation

Continuation trades occur after a valid Main (C1) Signal which is closed due to an exit triggered by exit or C1 indicators (or the news). Once this state is reached, it doesn't change until price crosses back the baseline. If continuation trades are not used, the state will change to **No Trade**.

Continuation Signals can be given by the Exit Indicator, Main (C1) Indicator or a different custom Continuation Indicator. C1 and C2 are always checked in every case.



In continuation trades, distance to baseline and volume are not considered. If all conditions are valid, a trade would be performed and the state would stay in continuation to wait for the next continuation signal (after the current trade is closed).

Missed Trade

Missed trade is a special type of state which can be reached if other filtering methods are used (EVZ, News, exposure). If there is a valid trade signal of any type, but either EVZ is below the minimum level, there are news on the next candle in one of the currencies or one of the currencies of that pair has already 2% risk in their direction (long/short), then the trade can't be performed and it's considered a missed trade.

This state can only be reached if a baseline is being used too. If not, then the state stays in **Main Signal** waiting for a new valid signal.

In this state, the algorithm waits until the exit indicator flips or until C1, if it is also being used as an exit, changes its direction. When one of those events happens, the theoretical trade that would have been taken is considered closed, and the state changes to **Continuation**. This state, as with any other, can be overridden by a **Baseline Cross**.

-5. News, EVZ and exposure checks

After the trade signals of all currency pairs have been processed, the trades pass through the rest of the filters (if they are used):

-If EVZ is less than the minimum required, all trades are removed and their state changes to **Missed Trade**. If it's less than the upper limit, but more than the minimum, their risk is reduced to half.

-New entries from currencies that have news events on the next candle are also discarded (their state changes to **Missed Trade**).

-The remaining trades are processed in the exposure filter: entries on pairs where there is already more than the maximum risk allowed in one of their currencies are discarded (**Missed Trades**). The remaining ones split the risk if they share any currency+direction. Maximum risk allowed depends on EVZ if it's being used (if EVZ is lower than the upper limit, maximum risk is half as the default one).

-6. Trade execution

The trades that go through all filters are executed, and the exposure that results from them will be considered in next candles until they are either closed or in breakeven (no risk left).

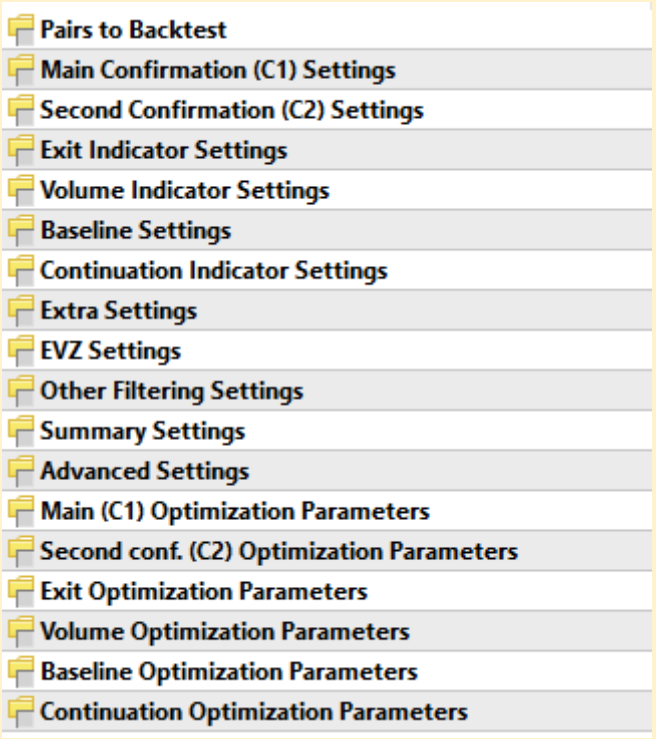
The trades on AlgoMaster NNFX are **virtual** (except in Real Trades mode, in MT5 only), that means that they aren't really performed using real orders (in MT4 this allows backtesting and optimizing on multiple pairs, since it's not possible to trade other pairs apart from the one that is used). It also avoids having imprecissions due to lot-size requirements: any trade is considered to lose exactly the Risk percent determined relative to the balance if it reaches the stop loss, and the rest of profit values (positive or negative) are interpolated from that ratio.

Take into consideration that:

- Virtual trades currently don't take into account the swap commissions and spread, they are an automated reproduction of "manual backtests".
- The lot sizes that would result from the trades are not taken into account, so you should check if the resulting real lot sizes would be possible to trade with your account (limits, leverage...). In forex pairs and daily timeframe this should not be a problem.
- Too small stops may overfit results due to the insane risk-reward ratios that can result from them.
- Gaps and flash crashes aren't currently considered (profits/losses are limited to stop and take profit levels). This may change in future versions.
- Each trade's profit is added when it is **completely closed** (including the half with trailing stop), so take this into account when reviewing equity curves or using compound interest. This may also change in future versions.

3. Parameters summary

AlgoMaster NNFX's parameters are grouped in collapsible input groups:



-Pairs to backtest: Determine which pairs (or a preset of pairs) the program will run in the backtest.

-Indicator settings: **Main Confirmation (C1) Settings, Second Confirmation (C2) Settings, Exit Indicator Settings, Volume Indicator Settings** and **Continuation Indicator Settings**. In each section you can introduce the indicator's name, whether or not it should use them in the backtest, how the program will read them (buffers, entry triggers, levels, etc.) and some specific settings for some of them.

-Extra Settings: Scale-out (or not), wait for indicators to catch up after baseline cross, and apply nuances like pullbacks, one candle rule or bridge too far.

-EVZ Settings: Allows to consider EVZ when backtesting: at which levels it should take trades at half risk, or not take any trade, and whether to scale-out or not when trading at half risk.

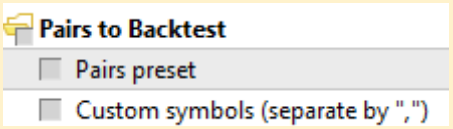
-Other Filtering Settings: Set **News avoidance** or **Exposure control** (avoid trading more than 2% in any specific currency and direction).

-Summary Settings: Determine which files to write (Summary/Optimization), the optimization mode and whether or not to save the complete trade journal in one file (opens and closes). Also, draw event icons in the backtest and whether or not to use the optimization parameters from the groups below.

-Advanced Settings: Modifications of the base of the NNFX algorithm, some of which may be one of your nuances or a potential improvement for your system (stop loss, take profit, ATR period, risk %, etc.).

-Optimization Parameters**:** For **Main (C1), Second conf. (C2), Exit, Volume, Baseline** and **Continuation.** Allows you to modify some of the parameters of your indicators, which permits optimizations to be made within the strategy tester. Up to 10 parameters (not necessarily the 10 first) can be optimised for each indicator.

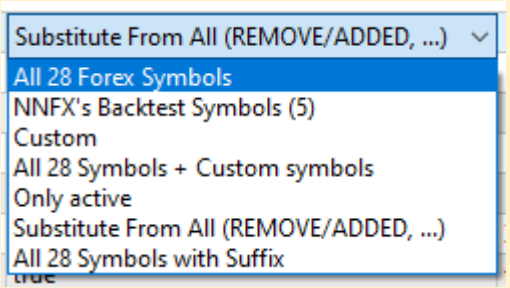
4. Pairs to Backtest



The Pairs to Backtest group is used to determine the symbols on which the backtest will be performed.

This feature doesn't work in the **DEMO VERSION.** In that case, the backtest would be always performed in the symbols **EURUSD, GBPCHF** and **USDCHF**.

AlgoMaster NNFX has been built to perform backtestings on Forex, however you can use it for other symbols like indexes, cryptocurrencies or metals. If you use it on other instruments, you should consider before the position sizes that would result from your system when using 1.5 ATR stop losses (or other ATR values), and whether or not it's possible to trade it.



Currently there are 7 presets that you can use to avoid introducing all the pairs you want the backtesting to be performed in:

-All 28 Forex Symbols: Cross-pairs between the 8 major currencies: EUR, GBP, AUD, NZD, USD, CAD, CHF and JPY.

-NNFX's Backtest Symbols (5): Benchmark symbols that VP recommended to perform initial backtests: AUDCAD, AUDNZD, CHFJPY, EURGBP and EURUSD.

-Custom: Only the symbols that you introduce in the Custom symbols parameter. For example, if you put "SP500, XAUUSD" the backtest would be performed in those symbols only.

-All 28 Symbols + Custom symbols: The backtest would be performed in all 28 major cross-pairs, and the symbols that you add in the Custom symbols parameter.

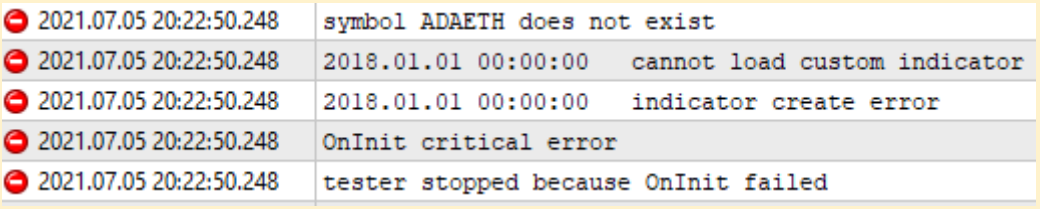
-Only active: Performs the backtest only in the symbol that you use to run the program (See section how to perform the backtest).

-Substitute From All: Backtest all 28 pairs, but change some of the symbols for others. This can be useful if one of the symbols has gaps in your broker's history: you can create a custom symbol in MT5 with other broker data (see this link for more information), and use that one instead. For example, to substitute GBPNZD with a custom symbol called GBPNZD.b, set the Custom symbols parameter to "GBPNZD/GBPNZD.b".

-All 28 Symbols with Suffix: Performs the backtest on the 28 symbols, but with a suffix added to all of them. This can be useful if your broker has suffixes on all currency pairs, or if you use custom symbols for all symbols. For example, if you set the Custom symbols parameter to ".pro", you would use the symbols AUDCAD.pro, AUDCAD.pro... USDJPY.pro.

-Symbol File: Reads a file with the name provided in the Custom symbols parameter. The file must contain one symbol name per line (without commas) to be read correctly. The file has to be in the **Common Files** folder and the extension also needs to be included in the input (for example, "SymbolsFile.txt"). This mode is useful to test a large number of symbols which don't coincide with any preset and which exceed the limit size of string inputs, or to switch quickly between custom presets.

When performing any backtest, make sure that all the symbols that it would use are visible in the market watch (and that they exist), or you may get an error like this:



However, this error (and History synchronization errors) rarely appears after some backtestings are performed since it would be stored in the tester cache.

5. Indicator settings

There are 6 indicators that can be customized: Main (C1), Second (C2), Exit, Volume, Baseline and Continuation. In this section we'll go through each indicator's settings.

Indicator Settings

Each indicator has a settings input where you can introduce a list of custom settings (separated with commas). If left empty, it will use its default parameters. Default parameters are always applied in the **DEMO VERSION.**

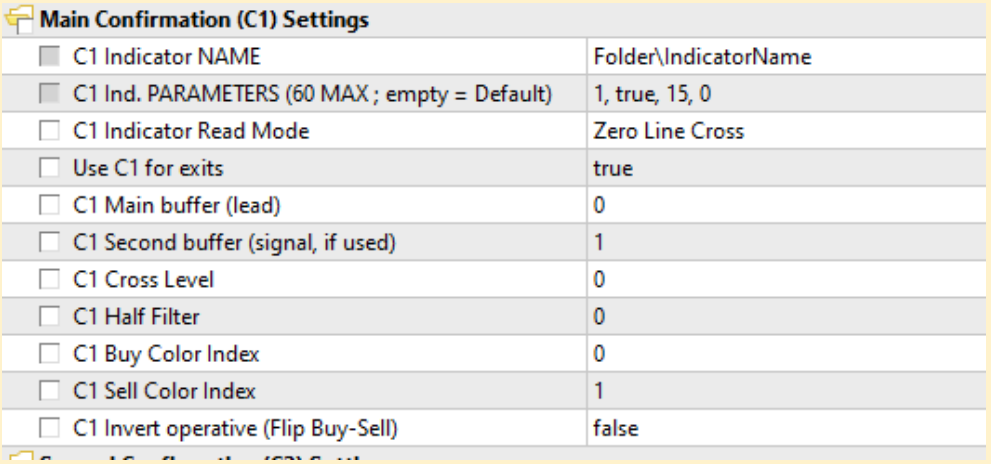
In MT5 it supports **double** values (must have a point in the decimals: 0.0 ; 1.5, this will be important for optimizations), **integer** values (numbers without point: 10 ; 0), **booleans** (true or false), **enumerators** (as integers; inputs that appear as lists, the input is usually its position in order starting with 0, but there are exceptions like **timeframes** and **applied price**) and **strings** (between quotation marks: "USD").

In MT4 only double parameters are supported, this shouldn't affect integer, boolean and enum inputs. However, with string inputs only the default value would be used.

Native Indicators

Both MT4 and MT5 versions allow the use of native indicators. To access them, enter their name on any indicator's name input (between "<" ">"). See Appendix B for more information about the indicators available and their parameters.

Main Confirmation (C1) Indicator



Main Confirmation (C1) Settings	
C1 Indicator NAME	Folder\IndicatorName
C1 Ind. PARAMETERS (60 MAX ; empty = Default)	1, true, 15, 0
C1 Indicator Read Mode	Zero Line Cross
Use C1 for exits	true
C1 Main buffer (lead)	0
C1 Second buffer (signal, if used)	1
C1 Cross Level	0
C1 Half Filter	0
C1 Buy Color Index	0
C1 Sell Color Index	1
C1 Invert operative (Flip Buy-Sell)	false

-Indicator Name: Name and location of the indicator (The folder Indicators is the **base** folder).

-Indicator Parameters: Settings of the indicator in order, separated with commas. If left empty, default settings would be used.

-Indicator Read Mode: See Indicator Read modes.

-Use C1 for exits: whether or not consider opposite signals as exit signals for active trades. Doesn't influence Exit Indicators (both can be applied, one, or none).

-Main Buffer: buffer of the indicator that acts as a "lead" (or Buy buffer in Buffer activation/Chart dots modes).

-Signal Buffer: secondary buffer (signal in two-lines crosses or Sell buffer in Buffer activation/Chart dots modes).

-Cross Level: level which creates signals in Zero line crosses

-Half Filter: half of the distance between the two lines in settings that use "filter zone". For example, with a cross level 0 and half filter of 50, it would result in the levels +50 and -50.

-Color Indexes: color buffer values that correspond to buy and sell signals (in Color Buffer and Chart Dot Color modes). The signal buffer is used as the color buffer. These parameters are only available on **MT5**.

-Invert Operative: flip buy and sell signals.

Second Confirmation (C2) Indicator

Second Confirmation (C2) Settings	
USE C2 INDICATOR	true
C2 Indicator NAME	Folder\Second
C2 Ind. PARAMETERS (60 MAX ; empty = Default)	0.8, 14, 0.3, 6, 0.4, 3
C2 Indicator Read Mode	Two Lines Cross
C2 Main buffer (lead)	0
C2 Second buffer (signal, if used)	1
C2 Cross level	0
C2 Half filter	0
C2 Buy Color Index	0
C2 Sell Color Index	1
C2 Invert operative (Flip Buy-Sell)	false

-Use C2 Indicator: activate/deactivate C2 indicator, if set to false it won't be considered in the backtest.

-Indicator Name

-Indicator Parameters

-Indicator Read Mode

-Main Buffer

-Second Buffer

-Cross Level

-Half Filter

-Color Indexes

-Invert Operative

Exit Indicator

Exit Indicator Settings	
USE EXIT INDICATOR	true
EXIT Indicator NAME	NNFX\Rex
EXIT Ind. PARAMETERS (60 MAX ; empty = Default)	14, 0, 14, 0
EXIT Indicator Read Mode	Two Lines Cross
EXIT Check only signals (not current state)	true
EXIT Main buffer (lead)	0
EXIT Second buffer (signal, if used)	1
EXIT Cross level	0
EXIT Half filter	0
EXIT Buy Color Index	0
EXIT Sell Color Index	1
EXIT Invert operative (Flip Buy-Sell)	false

-Use Exit Indicator

-Indicator Name

-Indicator Parameters

-Indicator Read Mode

-Check Only Signals: consider only signals as exits, or current states also. For example: if it's false, a buy is opened and the next day the exit has the main line below the signal, it's closed. If it's true, it would only be closed if the exit signal was given on **that** candle (if the main line was already below the signal, it would need to go above and cross down again for the trade to exit).

-Main Buffer

-Second Buffer

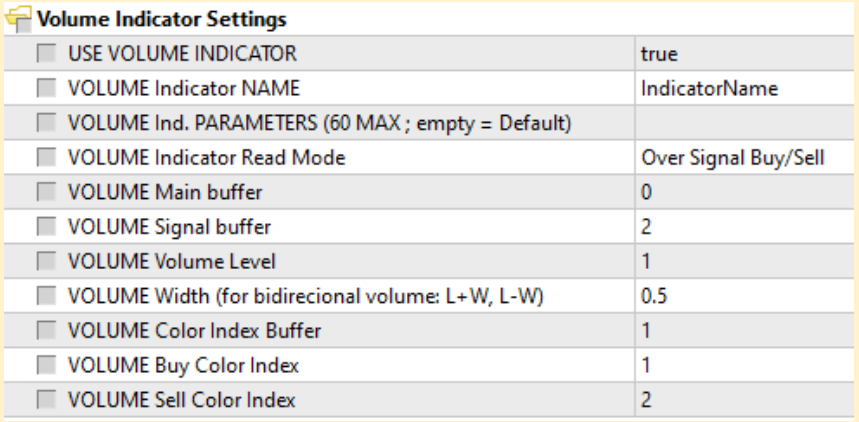
-Cross Level

-Half Filter

-Color Indexes

-Invert Operative

Volume Indicator



Volume Indicator Settings	
USE VOLUME INDICATOR	true
VOLUME Indicator NAME	IndicatorName
VOLUME Ind. PARAMETERS (60 MAX ; empty = Default)	
VOLUME Indicator Read Mode	Over Signal Buy/Sell
VOLUME Main buffer	0
VOLUME Signal buffer	2
VOLUME Volume Level	1
VOLUME Width (for bidirecional volume: L+W, L-W)	0.5
VOLUME Color Index Buffer	1
VOLUME Buy Color Index	1
VOLUME Sell Color Index	2

-Use Volume Indicator

-Indicator Name

-Indicator Parameters

-Indicator Read Mode: Volume Indicators have different settings than the other indicators. See Volume Indicator Read modes.

-Main Buffer

-Signal Buffer

-Volume Level

-Width: As half filters

-Color Indexes: In MT4, each color represents each buffer (buy/sell).

Baseline Indicator

Baseline Settings	
USE BASELINE	true
BASELINE NAME	<MA>
BASELINE PARAMETERS (60 MAX ; empty = Default)	36
BASELINE Buffer	0

-Use Baseline

-Baseline Name

-Baseline Parameters

-Baseline Buffer

Continuation Indicator

Continuation Indicator Settings	
USE CONTINUATION INDICATOR	Use Continuation Indicator
CONTINUATION Indicator NAME	ContInd
CONTINUATION Ind. PARAMETERS (60 MAX ; empty = Default)	9, 7, 7, 8
CONTINUATION Indicator Read Mode	Two Lines Cross
CONTINUATION Main buffer (lead)	2
CONTINUATION Second buffer (signal, if used)	3
CONTINUATION Cross level	0
CONTINUATION Half filter	0
CONTINUATION Buy Color Index	0
CONTINUATION Sell Color Index	1
CONTINUATION Invert operative (Flip Buy-Sell)	false

-Use Continuation Indicator: instead of true/false, it has 4 modes: false, Use Main, Use Exit and Use Continuation (the indicator defined in this group).

-Indicator Name

-Indicator Parameters

-Indicator Read Mode

-Main Buffer

-Second Buffer

-Cross Level

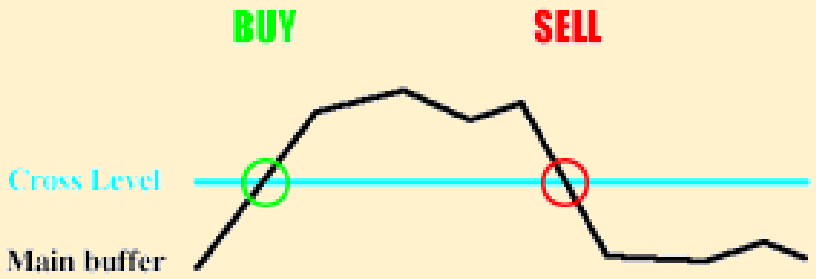
-Half Filter

-Color Indexes

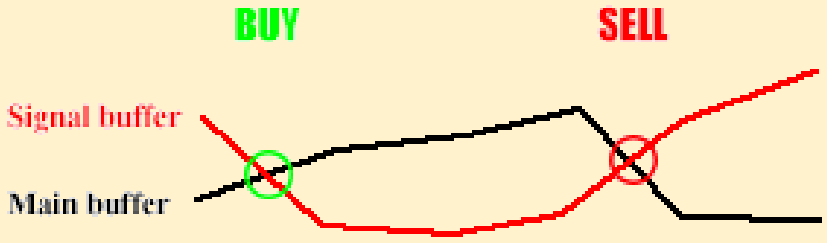
-Invert Operative

Indicator Read modes

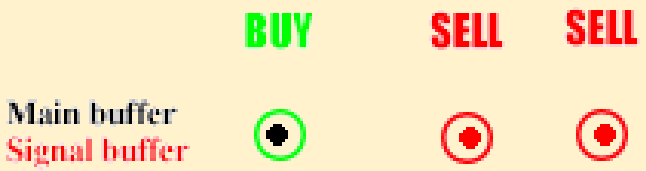
-Zero Line Cross: The signals are given when the **main buffer** crosses up (BUY) or down (SELL) a determined **cross level**.



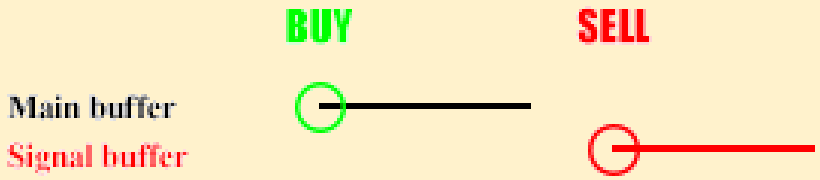
-Two Lines Cross: The signals are given when the **main buffer** and the **signal buffer** cross (if the main buffer crosses up the signal buffer, it's a BUY signal).



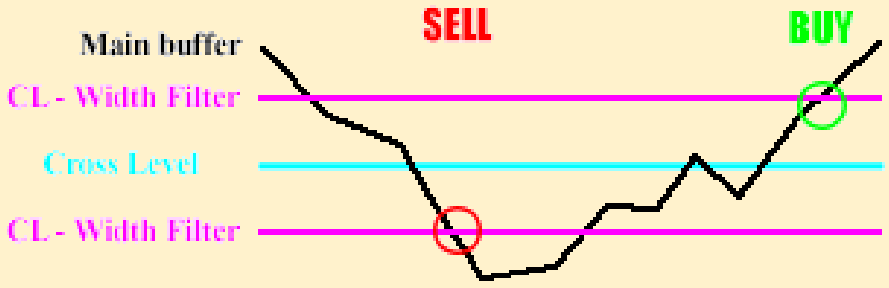
-Chart Dot Signal: When the **main buffer** has a non zero/null value, it's considered a BUY signal. When the **signal buffer** has a non zero/null value, it's a SELL signal. If both coincide, the **main buffer** has preference.



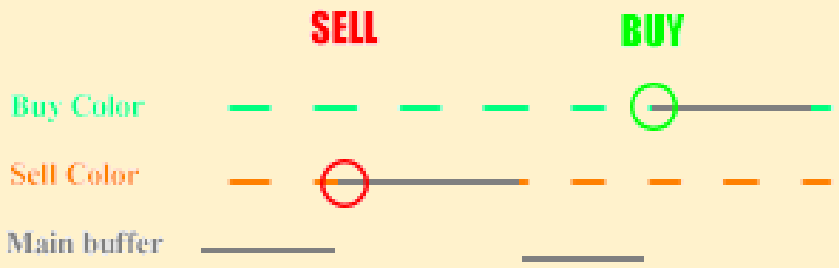
-Buffer Activation: Similar to chart dot signal, but only gives a signal when **any buffer** goes from zero/null to any value. **Main buffer** also has preference.



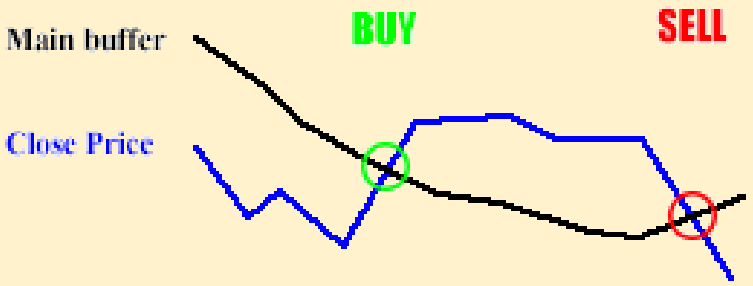
-Zero Line Filter: When the **main buffer** crosses down the level "**cross level - width filter**", a SELL signal is given; when it crosses up the level "**cross level + width filter**", it's a BUY signal.



-Color buffer: The **main buffer** needs to be a COLOR BUFFER for this mode to work properly (more explanation about color buffers in the next section). When the **main buffer** changes from any other value to **sell color**, a SELL signal is given. When its value changes from another value to **buy color**, a BUY signal is given. This mode is not available in MT4 (use Buffer Activation instead).

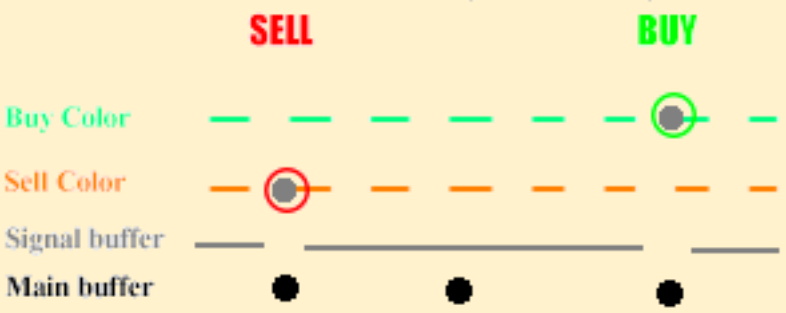


-Cross with price: similar to Two Lines Cross. When the **price** crosses the **main buffer** up, it's a BUY signal. If it crosses down, it's a SELL signal.



-Cross inside Filter: if the **main buffer** crosses down the upper filter (**cross level + width filter**) a SELL signal is given. If it crosses up the down filter (**cross level - width filter**), it's a BUY signal.

-Chart Dot (Color): When the **main buffer** is non zero/null, the **signal buffer** (which is a color buffer) determines if it's a BUY or SELL signal (or none). This mode is not available in MT4 (use Chart Dot instead).

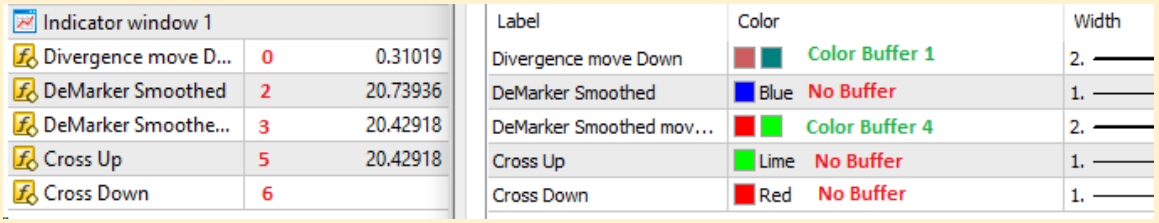


-Over Signal (Colored): When the **main buffer** crosses above the **signal buffer**, it gives a signal, and its color determines the direction (**color buy** or **color sell** in MT5). In MT4 an **extra main buffer** for the sell color is used.

-Over Level (Colored): When the **main buffer** crosses above the **minimum level** value, it gives a signal, and its color determines the direction (**color buy** or **color sell** in MT5). In MT4 an **extra main buffer** for the sell color is used.

Color buffers

Color buffers are used to determine another buffer's color (the buffer below it). They are not visible in the data window. Usually, a good way to find color buffers is looking at the Colors tab of the indicator:



In this case, buffer 1 is the color buffer of 0, and 4 of 3.

You should consider that this rule applies to most indicators, but there can be exceptions (if they hide buffers in the data window or use calculation buffers in between). In those cases, the best way to determine them is trial and error (or looking at the code if possible).

```
SetIndexBuffer(0,Buffer1,INDICATOR_DATA);
SetIndexBuffer(1,ColorBuffer,INDICATOR_COLOR_INDEX);
SetIndexBuffer(2,Buffer2,INDICATOR_DATA);
SetIndexBuffer(3,Buffer3,INDICATOR_DATA);
SetIndexBuffer(4,BufferCalc,INDICATOR_CALCULATIONS);
```

Also, in the "Colors" tab can be determined the indexes of the **Buy Color and Sell Color**, which are needed when using **color buffer** or **chart color dot** modes:

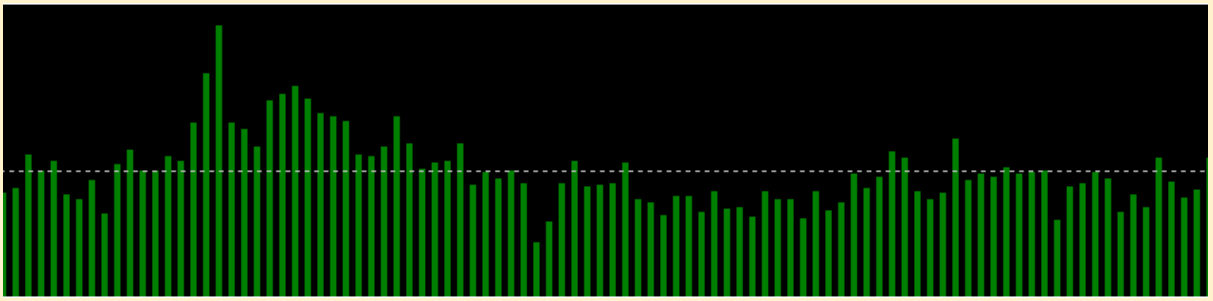
In this example, if you considered green as a buy signal and red as a sell signal, the values would be **Buy Color Index = 0, Sell Color Index = 2**.

In MT4 there are no color buffers, so each instance of color buffer would mean “buffer” instead of “color” (there is one buffer for buy color and other for sell color).

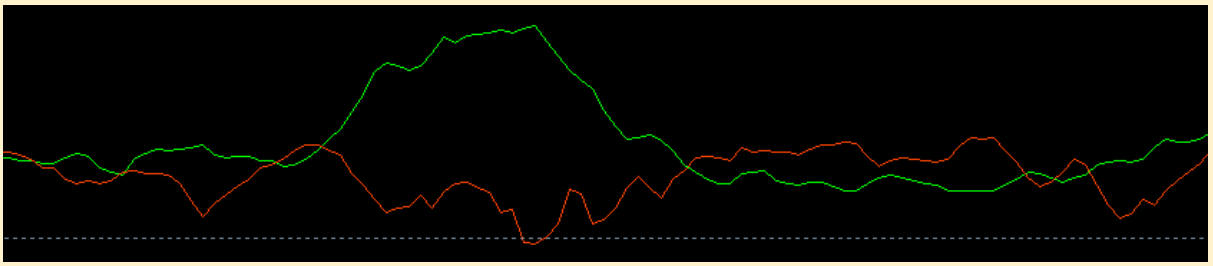
Volume Indicator Read Modes

Volume indicators have specific read modes, different from confirmation/exit signals:

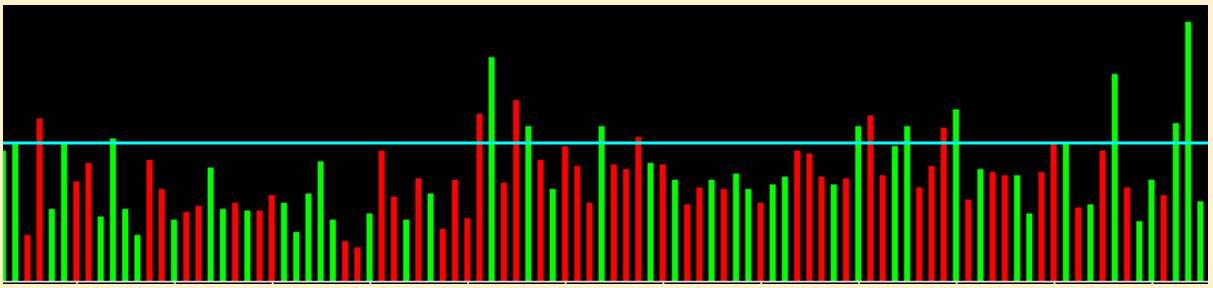
-Over Level: **main buffer** needs to be above a certain **volume level** for trading to be allowed. Most indicators fall in this category.



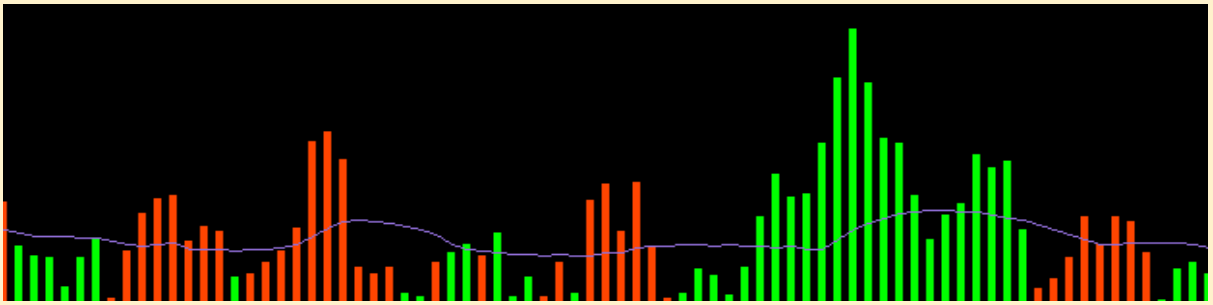
-Over Signal: **main buffer** needs to be above **signal buffer**. In this example, green is the main buffer and orange the signal.



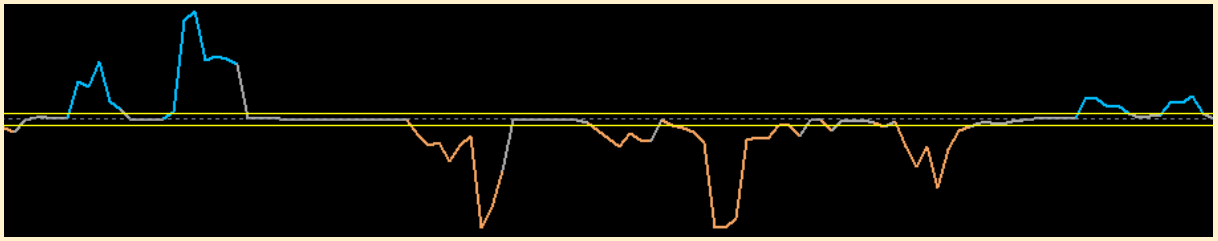
-Over Level Buy/Sell: acts as “over level” mode, but considers the direction of the volume too based on a color buffer (main buffer+1): if its value is either one of **buy color index** or **sell color index** it only allows trades in that direction. If the color value is different, it's considered a neutral signal. Could be considered as a third confirmation indicator combined with volume.



-Over Signal Buy/Sell: Acts as “over signal level”, with the same color buffer filter as “over level buy/sell”.



-Bidirectional Volume: Similar to “over level” but in both directions. The levels are determined by the **width** parameter: if the **main buffer** is higher than **level + width,** only buy signals are allowed. If it’s below **level - width**, then only sell signals are allowed (between level+width and level-width, no signals allowed).



6. Extra settings

Scale Out

When Scale Out is set to true, trades will be split in 2 halves: one gets closed in the take profit level (default=1.0), the other uses a trailing stop.

After price gets to TP, the stop loss is moved to breakeven. The trailing stop starts moving at a distance of 2.0*ATR (default) and is kept at stop loss distance (1.5*ATR default), being moved at the closure of each candle. In the image above you can see an example of a trade that scales out.

If scale out is set to false, all of the trade is closed at take profit level.

EVZ can override the Scale Out parameter if it's set to true. If Scale Out is false, it will never be used.

Wait to Catch Up

When set to true, if a baseline cross signal occurs with C1 indicator validating it but C2 or Volume don't agree, the algorithm would wait for both to agree (Main Catch Up state, see more in Internal Program Flow). If it's false, then it waits until another valid C1 or baseline cross signal is given.

Apply Pullbacks

When set to true, the algorithm will wait for a pullback after a valid baseline-cross trade that has gone way past the baseline (Distance to Baseline parameter, 1.0*ATR default). See more about it on Internal Program Flow.

Apply One Candle Rule

When set to true, the algorithm will wait one candle after a main C1 signal is given if any condition is not met (volume, C2 or distance). A pullback in price is needed for the signal to be valid. See more about it on Internal Program Flow.

Apply Bridge Too Far

When set to true, baseline cross signals in which the C1 signal happened more than 7 candles before are filtered out. Bridge Too Far can be used with any type of C1 signal, not just Two Lines Crosses. The amount of candles (default = 7) can be changed with the parameter "Bridge Too Far Max Candles". See more about how it is applied on Internal Program Flow.

7. Euro FX Vix

You can backtest modifying some money management rules depending on the EVZ and see if it improves (or worsens) your system. However, it is not recommended to be used when optimizing strategies since it's easier to fall in overfitting and much slower. The EVZ value considered is always **EVZ close at trade open candle.** This feature can't be used in the **DEMO VERSION.**

The EVZ indicator is included for free in AlgoMaster NNFX, but you can get the EVZ indicator separately here. It's also available currently for MT4.

There are 3 options that can be modified when using EVZ:

-Scale Out when EVZ < Half Risk: if Scale Out is set to true, you can choose to not scale out when EVZ is below the value at which your system trades with half risk.

-EVZ limit: if EVZ close is below the limit, trades are taken using 50% of the default lot size (and possibly not scaling out, as seen above). If it's above, default risk and scaling out are used. Default value is 7, as with the last VP's update (last was 8).

-Minimum EVZ: if EVZ is below this value, no trades are performed. Default value is 5.

8. Exposure avoidance

Exposure avoidance can be found in the section **Other Filtering Settings**. It only has two options, either apply it or not.

Maximum risk% per currency is 2% as default (the same as the risk input).

When applied, the maximum % per currency and direction (long/short) rule is applied, so if new trades appear where there is already max% exposure (half when having low EVZ) to one currency and direction, it isn't performed (See more about it on Internal Program Flow).

When using exposure avoidance, you can get a more realistic view of how your system would perform in real trading situations. It is not recommended to be used when optimizing strategies since it's easier to fall in overfitting.

The exposure algorithm is an efficient sub-optimal means to perform most trades possible (so that the sample size of the system is more diverse). A MinHeap is used to get the minimum currencies/directions. Its steps are:

-1: Start with the current exposure per currency and direction, and an array of new trade signals.

-2: Delete trades where exposure to one of their currencies (and direction) is bigger than the current maximum exposure (default 2% of account).

-3: Select iteratively currency/direction where the exposure per trade is the lowest (for example, if one currency/direction has already 1% exposure in active trades, maximum of 2%, and the algorithm has 3 new signals, the exposure per trade would be 0.33%).

-4: Add trades to the final trades array and update the new resulting exposure. Repeat steps 3 and 4 until all trades have been processed.

-5: Perform the final trades with its resulting exposure.

9. News Backtesting

The News avoidance system applies VP's principles to avoiding economic events in your trading systems. They can be found in the section **Other Filtering Settings**. News backtesting can't be applied in the **DEMO VERSION,** however, the news indicator can be shown in the charts in Demo (if you have the necessary data)**.** It acts differently in each scenario:

-When a trade hasn't touched Take Profit, if a news event comes on the next candle the trade is closed (either if it's winning or losing).

-If a signal is given before a news event, it's ignored (note in the image below that a trade can be taken at the end of a candle that had a news event).

-When a trade is winning (on breakeven), it isn't closed before a news event (see second trade in the example image).

The news events to be considered are filtered **by name**. The default parameters correspond to the recommendations by VP in his news video (as they are named in forexfactory.com).

EUR News	Monetary Policy Statement, Lagarde, Draghi
GBP News	MPC Official Bank Rate Votes, GDP
AUD News	RBA Rate Statement, Unemployment Rate
NZD News	Unemployment Rate, GDP, GDT, RBNZ Rate Statement
USD News	"Non-Farm Employment Change", FOMC Statement, Fed Chair Powell, CPI
CAD News	BOC Rate Statement, Unemployment Rate, Retail Sales, "CPI m/m"
CHF News	SNB Monetary Policy Assessment
JPY News	Monetary Policy Statement

The news inputs for each currency are separated by commas. There are 2 modes of filtering news, **contains** and **exact**:

-If a news event is written **between** quotation marks (like “CPI m/m” in CAD news), only news that have that **exact** name will be considered.

-If a news event is written **without** quotation marks, any event that **contains** that string will be considered (in the example, the USD news “Fed Chair Powell Speaks” and “Fed Chair Powell Testifies” will be considered by the algorithm).

The News Indicator (which can be found as a separate product here for MT5) can be either shown or hidden in the backtests when using the news filter, but in any case it wouldn’t affect the backtest results since the backtester has its own news management (to avoid unnecessary recalculations).

You can also choose the distance at which the news icons are shown and how separated they are.

10. Summary settings and files

Summary, optimization and output configuration can be found in the **Summary Settings** input group. There are 3 inputs:

-Create Summary/Optimization Files: if true, the program will generate a summary file if it's executed as a normal test, or optimization files if it's part of a genetic or complete optimization.

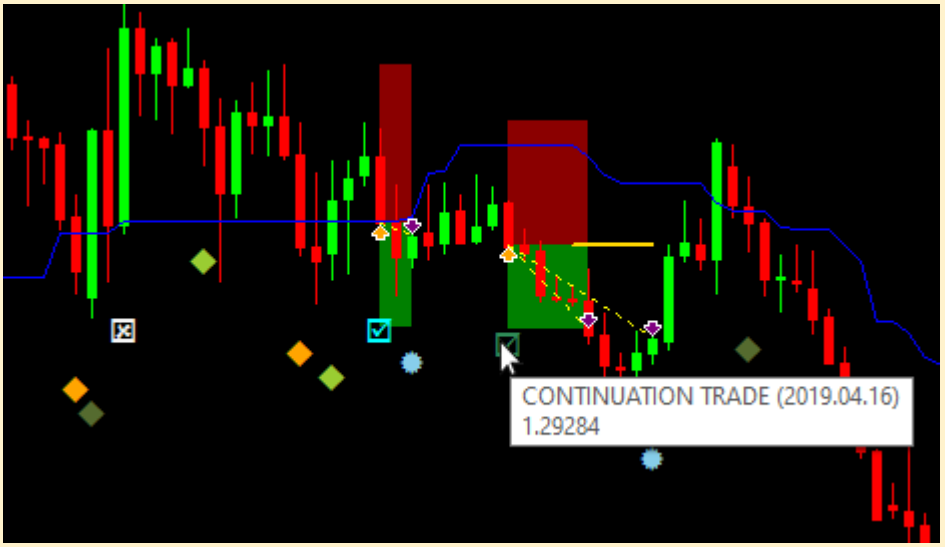
-Optimization Mode: Determines the "score" of the results of different settings. Some modes can be used to generate extra files (but they're not recommended for optimizations). See more in section Optimization Modes.

-Custom Optimization Formula: Formula which is evaluated in the mode Custom Optimization as a system final score. See Appendix C for more information on what can be used in the formula.

-Write Trade Journal: When set to true, a file is created which contains all trade opens and closes (date, symbol and profit). It can be used to search for specific trades, analyze data more closely or find the information of some specific trade.

-Show Extended Summary: When it is true, an extra summary will be printed in the log. For now, this summary contains a description of the closing of trades (which indicator or SL level closed it, and at which profit or loss), but this feature may be extended in next versions.

-Display Event Icons: When set to true, icons are created on the backtesting charts to display every event on the system: C1/baseline entries, entries invalidated by volume or C2, exits, etc.



All of the generated files (except Optimization) are located in the **Common Files** folder, which location is something like this one:

C:\Users\<USER>\AppData\Roaming\MetaQuotes\Terminal\Common\Files

The location of the files is printed after the backtests.

```
==================================
==================================
Summary file "NNFXTESTER_SUMMARY.txt" saved in directory C:\Users\[redacted]\AppData\Roaming\MetaQuotes\Terminal\Common
##########################################################################################################
```

The Optimization files are located in each of the tester core folders. The common root of all is your terminal's tester folder:

C:\Users\<USER>\AppData\Roaming\MetaQuotes\Tester\<TERMINAL_NUM>

This folder contains one sub-folder for each core of your CPU that can process a backtest (**Agent-127.0.0.1-XXXX**), MQL5\Files contains the resulting files of the backtest. An example on how to process the files will be provided in section Optimization Files.

It is recommended that once you find the folders, create direct links to be able to access them with more agility.

Summary File

Summary files contain certain information for each symbol and a general result. The elements are separated with tabulations in order to make it easier to copy and paste to **google sheets**:

```
EURJPY  EURCHF  EURCAD  EURUSD  EURNZD  EURAUD  EURGBP  GBPJPY  GBPCHF  GBPCAD  GBPUSD  GBPNZD.b
-4057.55        -50951.89       -2381.30        -2641.12        -22029.96       -12580.73       -5167.32

WIN RATE        TOTAL TRADES    TOTAL PROFIT    DRAWDOWN        ROI     PROFIT FACTOR   EXP.PAYOFF
48.33   720     111023.25       -25.87  7.39    1.09    154.20  -42000.00       -0.98644
```

	A	B	C	D	E	F	G	H	I
1	EURJPY	EURCHF	EURCAD	EURUSD	EURNZD	EURAUD	EURGBP	GBPJPY	GBPCHF
2	-4057.55	-50951.89	-2381.3	-2641.12	-22029.96	-12580.73	-5167.32	42783.04	26873.62
3									
4	WIN RATE	TOTAL TRADES	TOTAL PROFIT	DRAWDOWN	ROI	PROFIT FACTOR	EXP.PAYOFF	DIST.VALUE	DIST.SHAPE
5	48.33	720	111023.25	-25.87	7.39	1.09	154.2	-42000	-0.98644

The symbol-exclusive information will be different depending on the Optimization Mode chosen:

-Win Rate: wins and loses (separately +/-)

-Total Profit: total profit of symbol

-Total Pips: gross **points** won and lost (separately +/-)

-Drawdown: individual drawdown

-Return on investment: ROI of each symbol

-Profit Factor: profit factor of each symbol (limited at 999)

-Expected Payoff: expected payoff of symbol

-Trade Dist. Value: total profit of symbol[1]

-Trade Dist. Shape: total profit of symbol[1]

-Equity Curve: total profit of symbol (separately +/-)[2]

-Compound Interest + Equity Curve: total profit of symbol (compounded, separately +/-)[2]

The general information contains a quick summary of the stats of the backtest: win rate, total trades, total profit, max drawdown, ROI, PF, expected payoff, distribution value and distribution shape.

[1] These modes also generate Distribution Files

[2] These modes also generate Equity Files

Optimization Files

Optimization files are used to get a quick summary of each optimization pass. Every optimization result is written below the previous results in a set of files:

```
643008.70    50.39   905    643008.70    -17.28   42.83   1.33    710.51   471000.00    4.49493
623477.14    48.40   783    623477.14    -15.20   41.53   1.40    796.27   456000.00    5.34356
849008.08    54.32   694    849008.08    -25.99   56.55   1.53    1223.35  675000.00    7.82130
636383.14    52.18   803    636383.14    -15.42   42.39   1.39    792.51   443000.00    5.03910
504719.67    48.76   765    504719.67    -25.13   33.62   1.27    659.76   359000.00    2.79911
581115.50    47.26   804    581115.50    -20.23   38.71   1.34    722.78   512000.00    5.28801
669220.09    51.22   902    669220.09    -20.21   44.57   1.32    741.93   539000.00    4.81319
539063.10    54.73   455    539063.10    -13.66   35.90   1.64    1184.75  478000.00    9.08716
452960.76    49.39   492    452960.76    -15.17   30.17   1.49    920.65   366000.00    6.67728
```

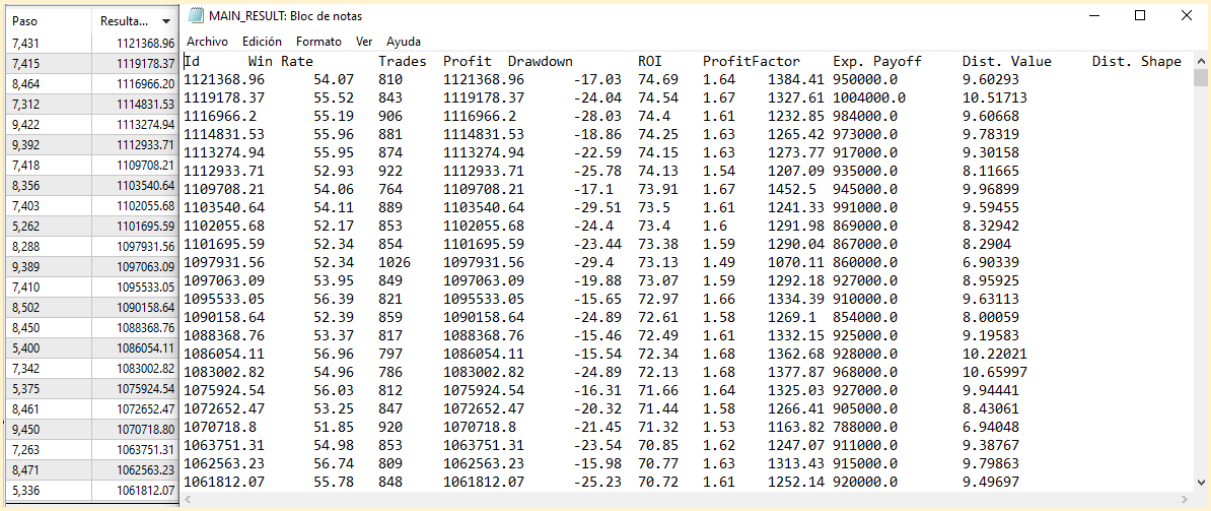
Each row contains: identifier (the tester result), win rate, amount of trades, profit, drawdown, ROI, PF, Expected Payoff, distribution value and distribution shape. **Make sure the files don't exist before performing the optimization** to avoid mixing optimization results (see **cleanFiles.py** at the end of this section).

One different file is generated for each core, so they have to be merged after an optimization to get all of the data in the same place. The following **python 3** program can be an example on how to do that:

```
## process.py
import os

fileName = "NNFXTESTER_OPTIMIZATION_DATA"

balanceList = []
allLines = []

   # FILTER LISTS
for i in range(9): #CORE FOLDERS 0-8

   if os.path.exists("Agent-127.0.0.1-300" + str(i) + \
      "/MQL5/Files/" + fileName + ".txt"):

      f = open("Agent-127.0.0.1-300" + str(i) +  \
         "/MQL5/Files/" + fileName + ".txt", "r", encoding = "utf-16")

      lines = f.readlines()

      # Combine arrays
      allLines = allLines + lines

      f.close()
```

```
   #Convert Strings to Tuples
for line in allLines:
   values = line.split("\t")

   identifier = float(values[0])
   winRate = float(values[1])
   totalTrades = int(values[2])
   totalProfit = float(values[3])
   drawdown = float(values[4])
   roi = float(values[5])
   profitFactor = float(values[6])
   expPayoff = float(values[7])
   distValue = float(values[8])
   distShape = float(values[9])


   if totalTrades == 0: # Ignore rows with no trades
      continue

   tup = (identifier, winRate, totalTrades, totalProfit, drawdown, \
      roi, profitFactor, expPayoff, distValue, distShape)

   balanceList.append(tup) # Add tuple to final list

   #Write merged file
if len(balanceList)>0:

   wr = open("MAIN_RESULT.txt", "w")

   balanceList.sort(key=lambda tup: tup[0], reverse=True)

   wr.write("Id\tWin Rate\tTrades\tProfit\tDrawdown\tROI\tProfit"+ \
      "Factor\tExp. Payoff\tDist. Value\tDist. Shape\n")

   for item in balanceList:
      for subitem in item:
         wr.write(str(subitem) + "\t")
      wr.write("\n")

   wr.close()
```

You would need to change the number of cores depending on how many of them your computer has (or how many folders it generates). After executing this program, located in the **root folder** of your **terminal tester**, you would get a combined file like this:



Paso	Resulta...
7,431	1121368.96
7,415	1119178.37
8,464	1116966.20
7,312	1114831.53
9,422	1113274.94
9,392	1112933.71
7,418	1109708.21
8,356	1103540.64
7,403	1102055.68
5,262	1101695.59
8,288	1097931.56
9,389	1097063.09
7,410	1095533.05
8,502	1090158.64
8,450	1088368.76
5,400	1086054.11
7,342	1083002.82
5,375	1075924.54
8,461	1072652.47
9,450	1070718.80
7,263	1063751.31
8,471	1062563.23
5,336	1061812.07

MAIN_RESULT: Bloc de notas

Archivo Edición Formato Ver Ayuda

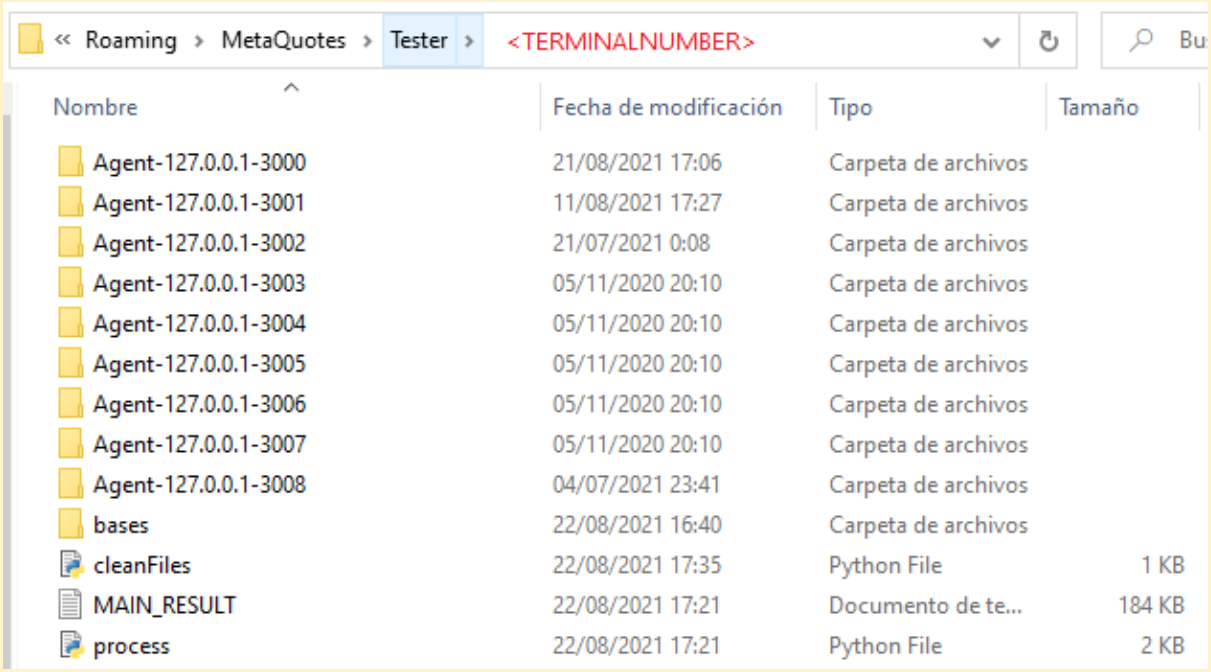
Id	Win Rate	Trades	Profit	Drawdown	ROI	ProfitFactor	Exp. Payoff	Dist. Value	Dist. Shape
1121368.96	54.07	810	1121368.96	-17.03	74.69	1.64	1384.41	950000.0	9.60293
1119178.37	55.52	843	1119178.37	-24.04	74.54	1.67	1327.61	1004000.0	10.51713
1116966.2	55.19	906	1116966.2	-28.03	74.4	1.61	1232.85	984000.0	9.60668
1114831.53	55.96	881	1114831.53	-18.86	74.25	1.63	1265.42	973000.0	9.78319
1113274.94	55.95	874	1113274.94	-22.59	74.15	1.63	1273.77	917000.0	9.30158
1112933.71	52.93	922	1112933.71	-25.78	74.13	1.54	1207.09	935000.0	8.11665
1109708.21	54.06	764	1109708.21	-17.1	73.91	1.67	1452.5	945000.0	9.96899
1103540.64	54.11	889	1103540.64	-29.51	73.5	1.61	1241.33	991000.0	9.59455
1102055.68	52.17	853	1102055.68	-24.4	73.4	1.6	1291.98	869000.0	8.32942
1101695.59	52.34	854	1101695.59	-23.44	73.38	1.59	1290.04	867000.0	8.2904
1097931.56	52.34	1026	1097931.56	-29.4	73.13	1.49	1070.11	860000.0	6.90339
1097063.09	53.95	849	1097063.09	-19.88	73.07	1.59	1292.18	927000.0	8.95925
1095533.05	56.39	821	1095533.05	-15.65	72.97	1.66	1334.39	910000.0	9.63113
1090158.64	52.39	859	1090158.64	-24.89	72.61	1.58	1269.1	854000.0	8.00059
1088368.76	53.37	817	1088368.76	-15.46	72.49	1.61	1332.15	925000.0	9.19583
1086054.11	56.96	797	1086054.11	-15.54	72.34	1.68	1362.68	928000.0	10.22021
1083002.82	54.96	786	1083002.82	-24.89	72.13	1.68	1377.87	968000.0	10.65997
1075924.54	56.03	812	1075924.54	-16.31	71.66	1.64	1325.03	927000.0	9.94441
1072652.47	53.25	847	1072652.47	-20.32	71.44	1.58	1266.41	905000.0	8.43061
1070718.8	51.85	920	1070718.8	-21.45	71.32	1.53	1163.82	788000.0	6.94048
1063751.31	54.98	853	1063751.31	-23.54	70.85	1.62	1247.07	911000.0	9.38767
1062563.23	56.74	809	1062563.23	-15.98	70.77	1.63	1313.43	915000.0	9.79863
1061812.07	55.78	848	1061812.07	-25.23	70.72	1.61	1252.14	920000.0	9.49697

It should contain all optimization passes identified (and sorted) by its result. Then, you can compare them and find good settings based on all statistics.

After processing the file, it's recommended to delete all optimization files (for each core), so they aren't mixed with future optimizations. Doing it by hand can be tedious, the example below is a simple program to delete those files automatically (You could also put this code after processing the files with the other program):

```
## cleanFiles.py
import os

fileNames = ["NNFXTESTER_OPTIMIZATION_DATA"]

#Input to avoid clearing all by mistake
clearB = input("Clear? (Y): ")

if clearB == "Y":

   for i in range(9): #CORE FOLDERS 0-8
      for name in fileNames:
         if os.path.exists("Agent-127.0.0.1-300" + str(i) + \
            "/MQL5/Files/" + name + ".txt"):

            os.remove("Agent-127.0.0.1-300" + str(i) + \
               "/MQL5/Files/" + name + ".txt")
```

With the additional programs and the processed file, the root folder of the terminal tester should look like this:



Optimization Modes

Each optimization mode either uses a different calculation to get a score for the system in the optimizer, writes different results in the Summary File or creates additional files (Equity or Distribution File).

Win Rate

Win rate uses the percentage of wins relative to all trades to determine the score. The result can be influenced by the parameter “Minimum percent to count trades as win/loss” (see Advanced Settings), as trades with very small profit/loss wouldn’t count.

This mode is recommended to get summaries for each symbol, but **not** for **optimization** since it usually tends to overfit (very restrictive settings that would do one or two trades that happen to win: 100.0 score).

The summary file contains total wins and total losses separately, since they give more information than raw win rate (it’s not the same 60% WR with 10 trades than with 100 trades), and they can be still processed in drive sheets.

Total Profit

Total profit mode calculates the supposed win/loss of all trades (without compounding) based on your initial account size. For example, if you risk 2% with an initial account of 10000, each trade you would lose -200 at the stop loss, or win +133.33 at take profit (with the default 1.5:1 risk-reward ratio), trades closed at other points are interpolated.

This mode is probably the easiest-safest one for optimizations, however when searching for good settings in your results check the rest of metrics (PF, amount of trades, drawdown, etc) to avoid overfitted results (don't just pick blindly the one with highest profits).

The summary file contains the net profit for each currency pair.

Total Pips

Very similar to Total Profit mode, but uses raw **POINTS** for the calculations (if your broker has 5-digit symbols, divide the result by 10 to get the number of pips). However, it may be less precise since the trades would have different weights in the final score (a trade on a fast currency pair would have more influence in the result than one on a slow pair). It can be used to get a currency pair summary of wins and loses in pips, if you prefer to measure them like that.

The summary file contains gross points won and gross points lost (separately).

Drawdown

The score is the maximum drawdown percent of the account (always in negative, since the tester optimizes for maximum).

This mode is **not recommended for optimizations** since it doesn't offer much information to differentiate a good system from a bad one (even though a very high DD would definitely be a bad system). It tends to overfit too: a system with no trades would have max score (0% drawdown). However, you can use it to get a summary of all currency pairs drawdowns individually.

Return on investment

This mode's score is the return percent in **one year** (total profit / (initial account * years)). The results are similar to those that you would obtain with Total Profit mode, however it can have less digit precision in the results. It's useful if you want to normalize optimized settings from time periods with different lengths (however, systems tested in short time periods can overfit).

It also gives a summary of each currency pair ROI.

Profit Factor

Profit Factor is the ratio between gross profit and loss (gross profit/gross loss). This mode calculates the profit factor of the system overall as a score. The score is capped at 999.9, to avoid zero division errors (when there are no losing trades). It's **not** recommended for **optimizations** since it tends to overfit and generate very restrictive systems with limited (winning) trades.

It also gives a summary of each currency pair Profit Factor.

Expected Payoff

Expected Payoff is the average return per trade (total profit/total trades). This mode is a good measure for the "quality" of the trades, but it can overfit with a low number of trades so it's **not** really recommended for **optimizations**.

It gives a summary of each currency pair Expected Payoff.

Trade Dist. Value

Trade Distribution Value is the first type of distribution score. See more about how distributions are created in the section Distribution File. The final score is calculated by multiplying each segment amount of trades by the segment value ($), and adding them up. It's similar to total profit but more discrete. Since it's less precise than total profit it's slightly more difficult to overfit, but many equal results are created as a consequence (you would need to manually check every result with the same score as the one you want to search).

The summary file contains the total profit for each pair (not rounded, and not separated between gross loss and profit). A Distribution File is also generated.

Trade Dist. Shape

Trade Distribution Shape is the second type of distribution score. See more about how distributions are created in the section Distribution File. The score gives a quick overview at how centered at 0 is the distribution: a higher value means that the system tends to get more wins, lower value means that it tends to lose more.

The calculation, in summary, is almost the result of dividing Total Distribution Score by the total amount of trades. However, trades that have a bigger profit than +Max Risk have a logarithmic decay, so they give less score in proportion (to reduce the chance of overfitting). This score system is good for optimization when you want to find systems with more "quality" trades (works better when starting from an almost made system and tweaking some settings).

The files generated are the same as with Trade Distribution Value.

Equity Curve

This mode is equivalent to Total Profit mode, but it also generates an Equity Curve file.

The summary file contains gross profit and gross loss for each currency pair, separately.

Compound Interest + Equity Curve

This mode generates an Equity Curve file, but the trade profits are compounded. It's **not recommended** to use this mode in **optimization** since it gives more weight to more recent trades (the profit amount is much bigger than the one of trades at the start). The summary file also has the same information as with Equity Curve or Total Profit modes (with compound interest).

The summary file contains gross profit and gross loss for each currency pair, separately.

Real Trades

This mode performs real trades (built in MT5). It's useful to test the performance of an algorithm in a more realistic environment (with swap commissions and bid-ask spreads).

There are a few considerations that may also affect the results of the system:

- Trades are executed **after candle closure**, so the first swap isn't considered, but the spread may have greater impact. This also causes trades to appear displaced one candle.
- Margin/leverage can have an impact in the system, if margin runs out, some trades may **not** be **opened**. If you test without exposure rules make sure to set a high margin.
- Lot sizes affect how much each trade gives as profit. The lot sizes are always less than the percentage set (not rounded).
- The results of the backtest are always compounded.

When using this mode no summary files are generated, and the tester result is 0 (Custom Max Criteria). However, you can have access to the built-in features of MT5 like the equity curve or the rest of optimization modes (max balance, sharpe ratio, etc.).

This mode is only available in the MT5 version, since it's not possible to perform trades in multiple pairs in an MT4 backtest.

Custom Optimization

Custom Optimization mode allows the evaluation of the system using custom formulas. Testing variables can be accessed to create those formulas (check Appendix C for more information about the variables and functions available).

The formula has to be defined in the input **Custom Optimization Formula**. If there is a division by 0 or the formula is wrongly written, the tester may return a value of 0 (in some instances, it could give an unexpected wrong calculation different than 0).

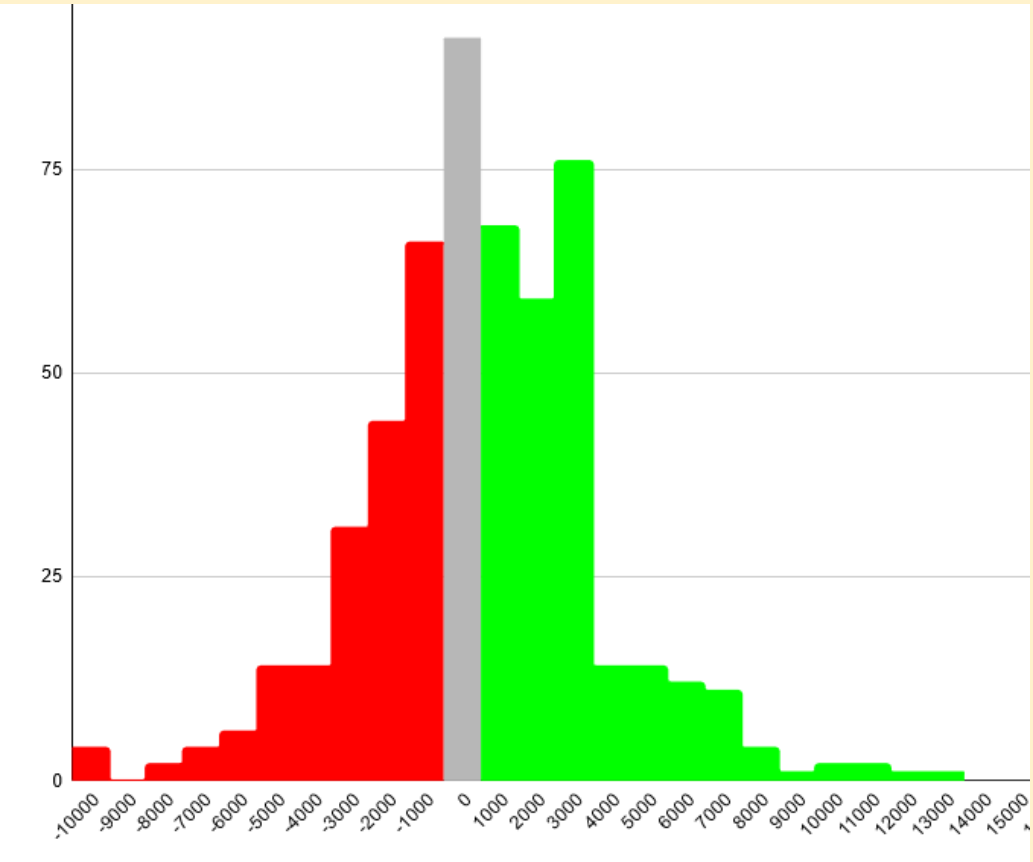
Distribution File

Distribution Files give a quick summary of the overall system's reliability, since they provide a more visual way of displaying the probabilities of a trade winning, losing and how much.

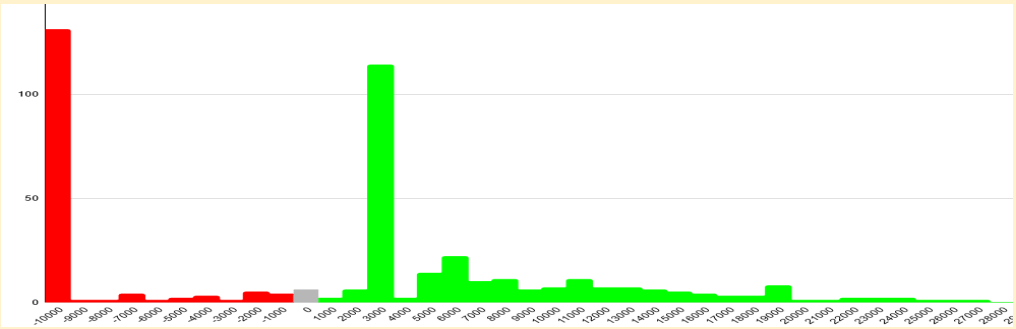
The trades are separated into discrete groups. For example, if your system has 2% risk with initial capital of $50000, the maximum risk would be $1000, and that would be divided in 10 segments of $100. Profitable trades also go in steps of the same amount, but unlike losing trades they have no limited upside (if scaling out). Each trade is classified into the closest group (rounded). The resulting file, which is created in **Common Files** folder, looks like this:

-1000	-900	-800	-700	-600	-500	-400	-300	-200	-100	0	100	200	300	400	500	600
20	6	5	12	11	34	37	60	80	76	61	38	37	170	10	13	5

As with other types of files, this data can be copied and pasted into drive sheets (it's also written horizontally to allow storing multiple distributions of several systems, one for each row). An example of a distribution (which template is provided in the Use Recommendations section) would look like this:



As you can see, the curve is tilted to the profit side (the system of this example had a profit factor in the backtests of 1.73 for instance). The shape of the curve allows one to quickly review the overall system and get a better expectation of how profitable (or not) the trades are. The more the trades, the more statistically reliable[3] it would be.



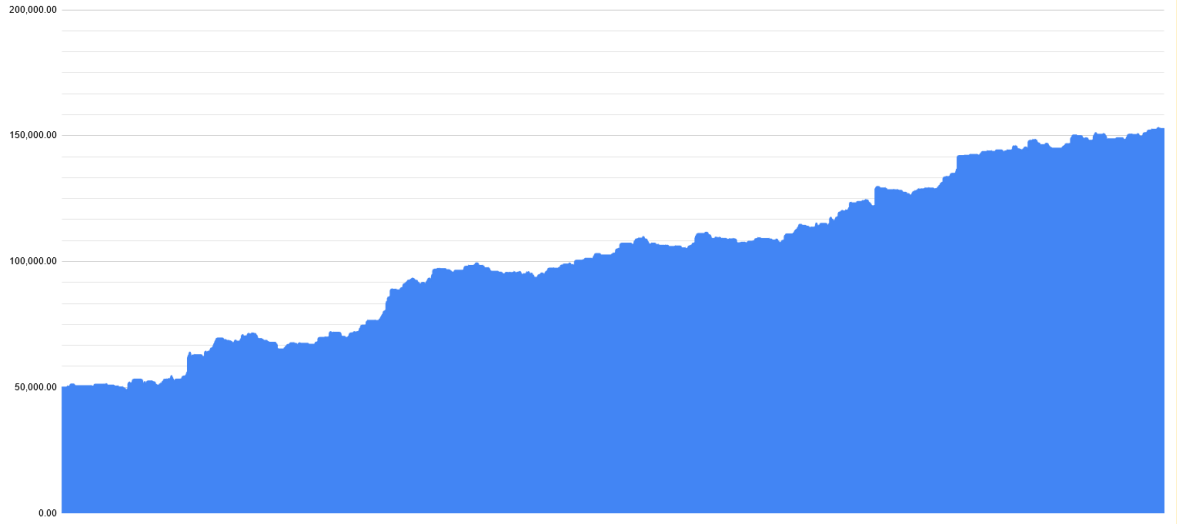
Distributions are not as useful if your trading system doesn't use all indicators (especially baseline/exits) or if it doesn't scale-out, since most trades would fall into the -10th and 7th groups (stop loss and take profit). It is intended to be used in the last steps, when almost all indicators are in place. In the first cases, win rate gives almost the same information as distributions.

Equity File

The Equity file shows the evolution of the theoretical account balance throughout all of the testing period. The balance is measured once per day, not once per trade, so it is normalized by time. It can be useful to recognize periods of drawdown and profitable periods. The file only contains the date and the balance for each candle, but it can be displayed as a graph using drive sheets:

```
2019.03.25 00:00        95479.90
2019.03.26 00:00        95813.23
2019.03.27 00:00        95125.15
2019.03.28 00:00        95629.87
2019.03.29 00:00        95629.87
2019.04.01 00:00        95968.28
2019.04.02 00:00        94751.27
2019.04.03 00:00        94751.27
2019.04.04 00:00        94991.22
```

[3] The distribution takes into account all of the backtesting period, market conditions of some months/years can often give very different results than the distribution in general.



The equity in between days where trades are opened is not considered (the profit of a trade is added when it is completely closed) to avoid slowing down the system too much, so it should be taken as a quick informative summary.

There are 2 optimization modes that generate Equity files: Equity Curve and Compound Interest + Equity Curve. Each one would produce different curves (similar in shape, but the non-compounded would have linear growth and the compounded one exponential growth).

Trade Journal File

The trade journal file can be activated separately with any optimization mode (see the beginning of this section). The file contains the dates of each opening and closing of trades, the currency pair of the trade, position type (buy/sell) and the profit of the trade (for closes).

The data of this file can be useful for reviewing more deeply the system (manually or with external tools) or finding specific trades and its currency pair (for example, the trades with bigger profit to see how the system catched trends).

```
2018.03.05 00:00        EURJPY  SELL CLOSE      -507.18
2018.03.07 00:00        EURGBP  BUY CLOSE       742.97
2018.03.07 00:00        NZDJPY  SELL CLOSE      728.58
2018.03.07 00:00        USDCHF  BUY OPEN        0.00
2018.03.07 00:00        USDCAD  BUY CLOSE       1509.73
2018.03.08 00:00        EURUSD  SELL OPEN       0.00
2018.03.08 00:00        GBPJPY  SELL CLOSE      -318.15
2018.03.12 00:00        EURUSD  SELL CLOSE      -182.55
2018.03.12 00:00        USDCHF  BUY CLOSE       524.51
2018.03.12 00:00        CHFJPY  SELL CLOSE      1192.19
2018.03.14 00:00        GBPCHF  BUY OPEN        0.00
```

11. Advanced settings

The inputs in the Advanced Settings group allow the modification of some of the roots of the NNFX algorithm. Some of them can be a "nuance" of yours, or you can try to create completely new systems that don't align exactly with NNFX.

-Stop Loss ATR: Multiplier applied to ATR to calculate the stop loss level for each trade. Default value is **1.5**. This setting must always be greater than 0.

-Take Profit ATR: Multiplier applied to ATR to calculate the take profit level for each trade. Default value is **1.0**. This setting must always be greater than 0.

-ATR to start moving Trailing Stop: Distance (multiplied by ATR) from the open price where the trailing stop starts being updated. When it gets to that level, the stop loss is updated each day using Stop Loss ATR distance. Default is **2.0**. This setting should be equal or higher than Take Profit ATR to perform effectively.

-Distance to baseline ATR: distance needed (relative to ATR) for a trade to be valid. If price shoots above/below the baseline and gets further than this distance in one candle, a pullback situation is generated (see Internal Program Flow). If a C1 entry is given it also must be closer than this distance. The default for this setting is **1.0**. If set to 0.0, the distance to the baseline would be ignored in any case.

-Bridge too far candles: amount of candles for the Bridge Too Far nuance. After a valid baseline entry, if the C1 signal happened at a higher amount of candles before, the signal is invalidated. Default is **7**.

-ATR period: period setting used for the ATR. It would affect all different relative distances used in the system (SL, TP, baseline, etc.). Default value is **14**.

-Minimal % to count Win/Loss: minimum percent of the trade, relative to the total amount risked in that trade, needed to be counted in the Win Rate stats. It can be considered as some "customizable chalks". The profit amount of the trade is not discounted to the final profit amount in any case. The default value is **0** (all trades count). It's more recommended to be used with a higher value when there are not many indicators in the system, or for C1 testing.

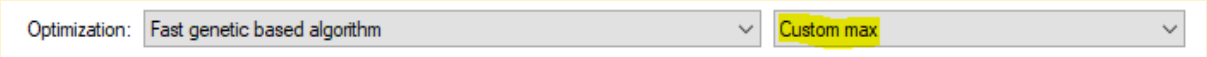
-Risk (% of balance in each trade): percentage of balance that is at risk in each trade (can be modified by EVZ or exposure if they are used). If compounding is used, the balance of reference is the last day's one, if not it would use only the starting balance. Default is **2.0**.

-Use Advanced Trailing Stop: substitutes the default trailing stop with an indicator-based one. Standard trailing stop inputs still apply (Start moving ATR), and the trailing stop cannot go backwards in any case (even if the indicator retraces). If set to **true,** the following parameters will configure the trailing stop: Trailing Stop Indicator Name, Trailing Stop Indicator Parameters, Buffer for Buy Orders and Buffer for Sell Orders.

12. Optimization parameters

Optimization parameters allow modifying predefined inputs of any indicator to perform genetic optimizations. Optimization parameters can't be used in **DEMO VERSION** since it only uses default parameters for the indicators.

For optimizations to work, you need to use the mode Custom Max, and ideally a modelling of "Open Prices only" for better performance. Any other mode will not return any result (0.0).



Optimization inputs have been modified from version 1.05. Below we describe how to use them in their current form.

Optimization Parameters	
Optimization Parameter 1	31
Optimization Parameter 2	0.7
Optimization Parameter 3	12
Optimization Parameter 4	20
Optimization Parameter 5	0
Optimization Parameter 6	0
Optimization Parameter 7	0
Optimization Parameter 8	0

There are 50 available **Optimization Parameters**. All of them can be accessed from any "Indicator Parameters" input field: to use them, set the desired parameter with **"# + number of parameter"**, and set its desired value (or range of values for optimization, start-step-stop) in the corresponding parameter:

C1 Ind. PARAMETERS (60 MAX ; empty = Default)	#1, 12, 25.0

In this example, the first parameter of the C1 indicator would be determined by the value in "Optimization Parameter 1". Values can be reused, so you could put "#1" several times in the same indicator or in other indicator inputs.

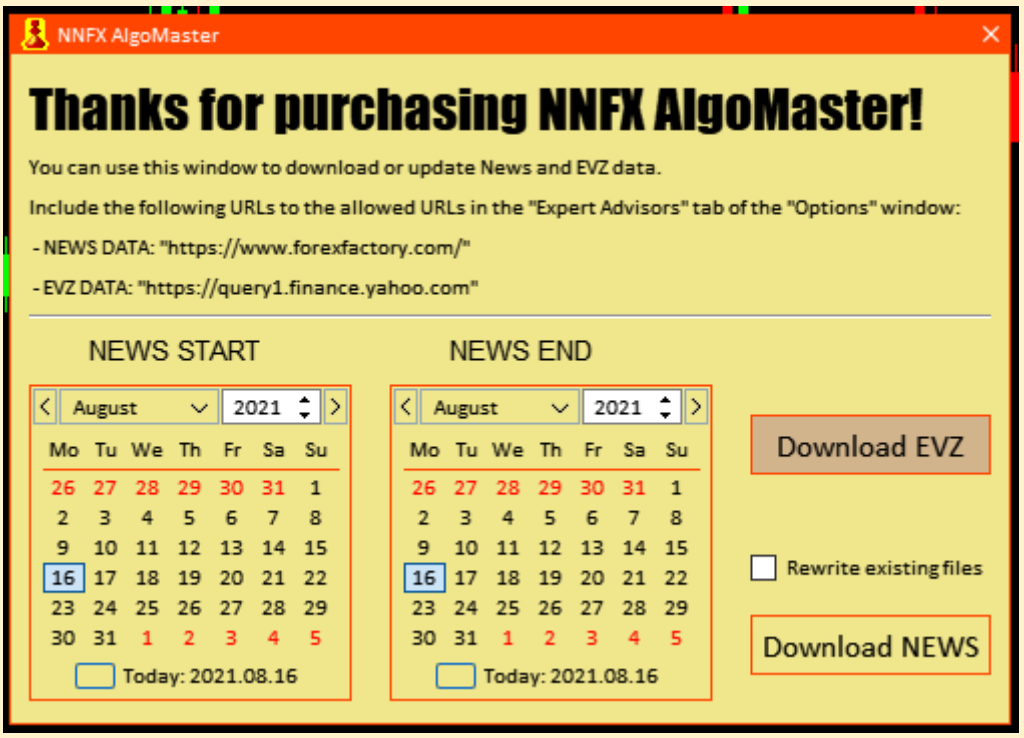
In some instances in MT5, not using the correct data type can cause errors in some indicators. You can force the input conversion by adding a letter at the end of the input: **"i"** to cast as an **integer**, **"b"** to cast as a **boolean** and **"d"** to cast as a **double** (this one is not needed, since by default values are casted as doubles).

C2 Ind. PARAMETERS (60 MAX ; empty = Default)	#1b, #2i, #3d, #2d

13. Graphical interface

AlgoMaster NNFX provides an interface to allow downloading EVZ and News data which can be used in the backtests. This data is equivalent to the data used in the original included products: Forex News Indicator for MT5 and Euro FX Vix (either MT4 or MT5).

To access the graphical interface, drop NNFX AlgoMaster into one of your charts (don't mind the expert parameters):



Since it needs to be applied to live charts, this feature is not available in the demo version.

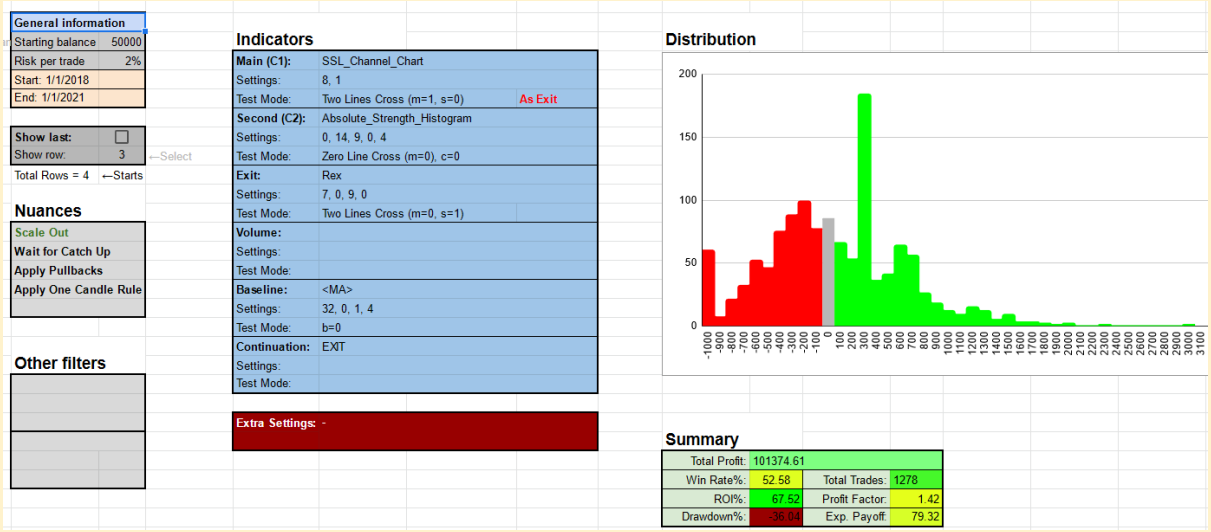
To download EVZ data, simply click the **Download EVZ** button and it will get all information from 2008. If you perform backtests with recent data, you may need to update it more frequently.

To download news data, you will need to select a **start** and an **ending** date and then click **Download News**. Future dates can be chosen too as news are scheduled, but they may change after. If you download News for the current week it is recommended to **rewrite files** after. If the news system was updated, you will have to rewrite them also to get the most recent features.

14. Use recommendations

AlgoMaster Drive Sheets template

You can get a template drive sheet to get the most of this program in this link. To use it, create a local copy in your drive and modify it as you want (delete what you won't need or use, change fonts/colors, add new data to the graphs, etc.). There, you'll be able to keep track of all your backtests in one place, compare them and see if your system keeps improving (or keep multiple systems in one place).

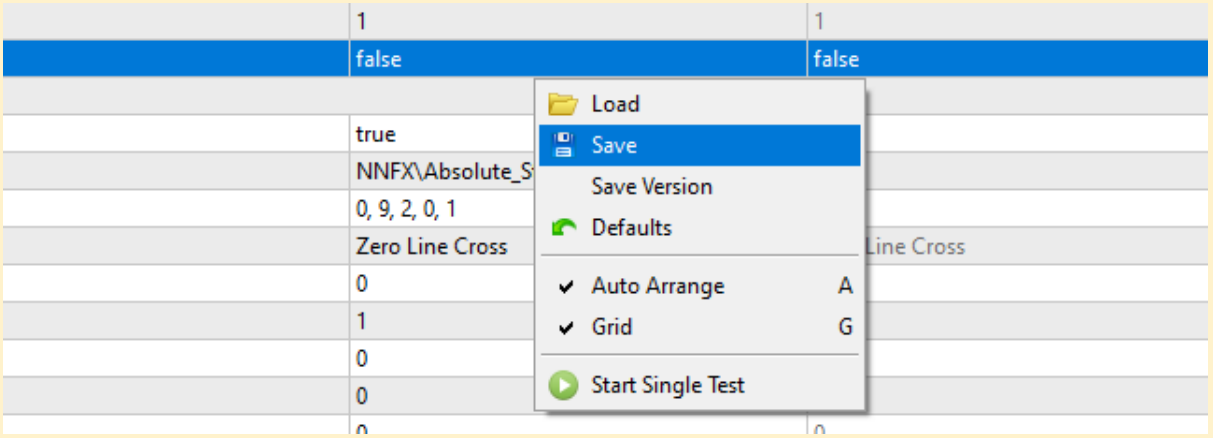


Using templates to change parameters

Since there are many parameters in the program, it can be easy forgetting to set some input before doing a backtest. One way to solve some of these issues is using templates/presets, where you can save some settings in a file and load them in one click.

For example, let's say that every time you test an algorithm you want to test it first without exposure, EVZ and news, but later you want to test it with them to see how it performs in a more "realistic" way. You could create 2 templates, one to activate all of those inputs and other to deactivate them.

First, right click on the inputs to save a new template (better if you have the desired inputs as you want them to be), set a name and save it.



Then, open the newly (with any notepad) created file and **remove all lines** that you don't need (or don't want changed). For this example, to activate all 3 filters you would need only something like this:

```
; EVZ Settings
useEvz=true||false||0||true||N
scaleOutHalfRisk=false||false||0||true||N
; Other Filtering Settings
useExposure=true||false||0||true||N
useNews=true||false||0||true||N
```

In this example, you would also set it to **not** scale out when EVZ is lower than half risk.

The values to mostly take into consideration are the ones highlighted in red: the parameter name and the value used in a backtest. The other values on the right are used for optimizations (start, step and stop), and the last one is whether or not the box is checked for optimizations ("N" means it is not checked, "Y" means it is checked). These last values won't appear in non optimizable inputs (like strings). The inputs that start with ";" are input group names, and they don't have any effect on the file (they can be removed without problem, but they are useful not to get lost in the file).

You can use templates for almost anything: to save your best performance system with all the settings exactly as they were originally, to save a preset of different news events, to clear all optimization parameters or to save some preset of symbols like metals or crypto.

To end with last example, the file needed to deactivate all of these inputs would look like this:

```
useEvz=false||false||0||true||N
useExposure=false||false||0||true||N
useNews=false||false||0||true||N
```

Note that in this file groups and the input “scale out at half risk” have been removed too: groups are not really needed and the other input is not used in any case when “use EVZ” is false.

Different timeframes and instruments

AlgoMaster NNFX has been made and tested for forex pairs and daily timeframes. It can be used in more timeframes, but the results may not be as reliable: if you use it on very low timeframes, take into account that this program does not consider **ticks, gaps** in price, **commissions** and **spreads** (it is the equivalent of an automated-manual backtest).

EVZ doesn’t work on other timeframes apart from Daily (in future updates, it may be able to work on weekly-monthly, but in lower timeframes it’s almost impossible to get accurate data).

News may not be as useful in other timeframes since it filters out trades or exits them at the beginning of candles that have them: in lower timeframes it wouldn’t stay out for very long, and in higher timeframes it may get events to avoid almost each candle on each pair. The lower timeframe issue may be adapted in future updates, but for higher timeframes it’s not recommended to use (or with less news).

Why do I get the same values in optimization?

If all the optimization passes get exactly the same result, make sure you are accessing the correct optimization parameters. Also make sure that you are using the mode **Custom Max** (unless the optimization mode is **Real Trades**, then you would have to use any other mode).

If you get different results but there are many that have the same values, probably one or more of the parameters that you are optimizing **don't affect** the indicator (maybe because of an error in configuration, or because some settings have effect in only some combinations of parameters).

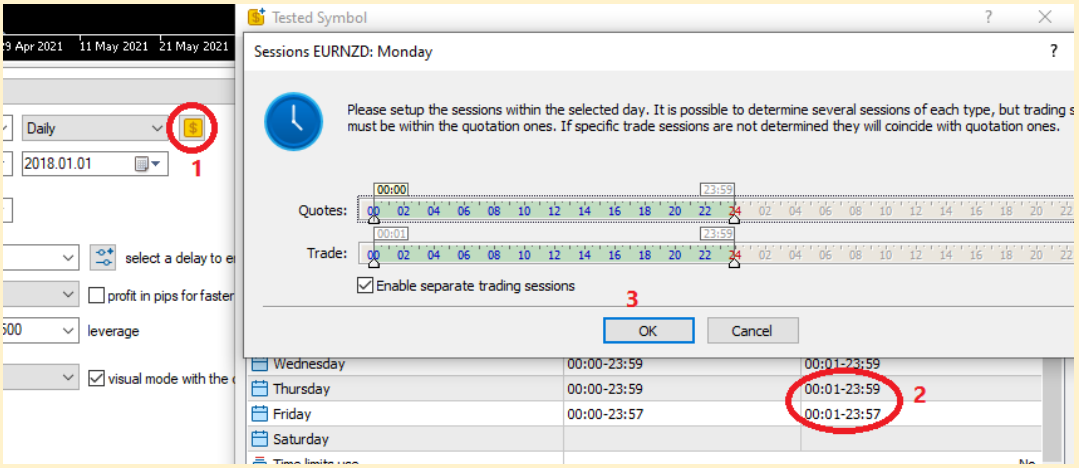
If you have all of that checked and fine, but you still get all 0's in your optimization, check if your **indicators** work properly in the strategy tester and make sure that they are not using very **restrictive settings** (this can be common with volume indicators).

Why do I get "market closed" errors in Real Trade Mode?

When using some brokers' data, some trades would probably be executed in close time, since D1 candle open is always at 00:00. You can check this in each symbol specification, to work properly, the time range for trading must include the time 00:00. Below you can see an example where that would be a problem.

Sessions	Quotes	Trade
Sunday		
Monday	00:00-23:59	00:01-23:59
Tuesday	00:00-23:59	00:01-23:59
Wednesday	00:00-23:59	00:01-23:59
Thursday	00:00-23:59	00:01-23:59
Friday	00:00-23:57	00:01-23:57
Saturday		

If you backtest only one pair, you can solve this by modifying the symbol's specification directly, but when using more pairs you can't access that.



To solve this problem there are 2 ways: the least precise would be using a demo Metaquotes account and setting **custom commissions** with the same values as the broker you want to test. This would be faster but the swaps can be very different between accounts.

The second way would be creating copies of the symbols you want to test (without the time limitation) and recreating the original commissions:

Creating the symbol copies

For that, you can use the ex5 script in the Examples files “SymbolClone”: drop it a chart with the symbol you want to copy, set a suffix (use the same one preferably) and hit accept (you can optionally remove the weekend candles that some brokers have). To save time, it will copy only the candles of the active timeframe (D1 is enough to run the backtests in that same timeframe).

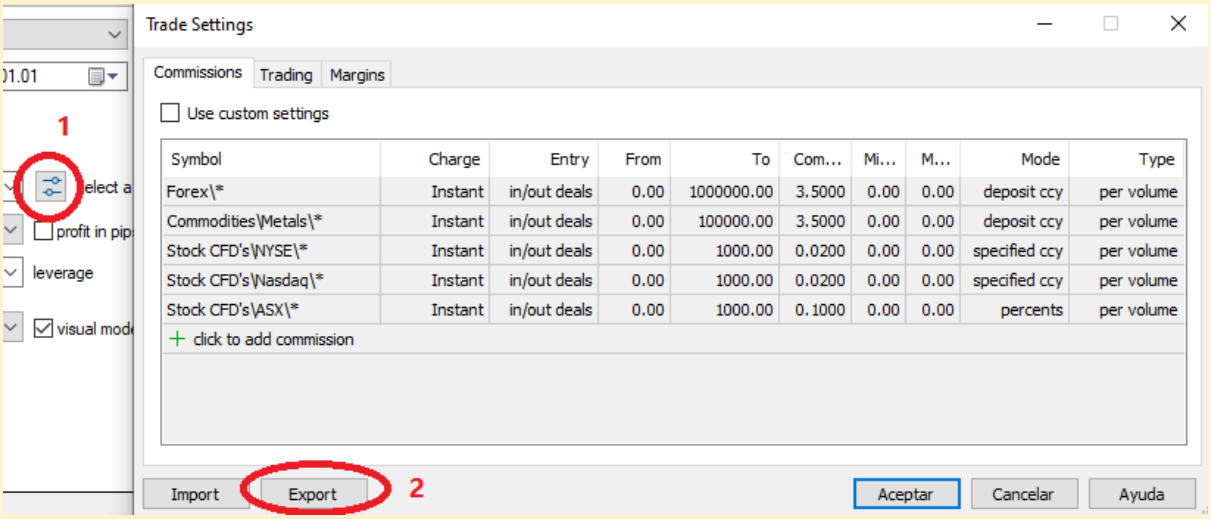
AUDCAD.b, Australian Dollar vs Canadian Dollar		
Sector	Currency	
Digits	5	
Contract size	100000	
Spread	floating	
Stops level	0	
Margin currency	AUD	
Profit currency	CAD	
Calculation	Forex	
Initial margin	100000	
Chart mode	By bid price	
Margin rate	**Initial**	**Maintenance**
Market buy	1.0000000	0.0000000
Market sell	1.0000000	0.0000000
Buy limit	0.0000000	0.0000000
Sell limit	0.0000000	0.0000000
Buy stop	0.0000000	0.0000000
Sell stop	0.0000000	0.0000000
Buy stop limit	0.0000000	0.0000000
Sell stop limit	0.0000000	0.0000000
Trade	Full access	
Execution	Market	
GTC mode	Good till cancelled	
Filling	Immediate or Cancel	
Expiration	All	
Orders	All	
Minimal volume	0.01	
Maximal volume	100	
Volume step	0.01	
Swap type	In points	
Swap long	-1.89	
Swap short	-0.95	
3-days swaps	Wednesday	
Sessions	**Quotes**	**Trade**
Sunday		
Monday	00:00-24:00	00:00-24:00
Tuesday	00:00-24:00	00:00-24:00
Wednesday	00:00-24:00	00:00-24:00
Thursday	00:00-24:00	00:00-24:00
Friday	00:00-24:00	00:00-24:00
Saturday		

AUDCAD, Australian Dollar vs Canadian Dollar		
Sector	Currency	
Digits	5	
Contract size	100000	
Spread	floating	
Stops level	0	
Margin currency	AUD	
Profit currency	CAD	
Calculation	Forex	
Initial margin	100000	
Chart mode	By bid price	
Margin rate	**Initial**	**Maintenance**
Market buy	1.0000000	1.0000000
Market sell	1.0000000	1.0000000
Buy limit	0.0000000	0.0000000
Sell limit	0.0000000	0.0000000
Buy stop	0.0000000	0.0000000
Sell stop	0.0000000	0.0000000
Buy stop limit	0.0000000	0.0000000
Sell stop limit	0.0000000	0.0000000
Trade	Full access	
Execution	Market	
GTC mode	Good till cancelled	
Filling	Immediate or Cancel	
Expiration	All	
Orders	All	
Minimal volume	0.01	
Maximal volume	100	
Volume step	0.01	
Swap type	In points	
Swap long	-1.89	
Swap short	-0.95	
3-days swaps	Wednesday	
Commissions	**Instant, in/out deals**	
0 - 1000000	3.5 USD per lot	
Sessions	**Quotes**	**Trade**
Sunday		
Monday	00:00-23:59	00:01-23:59
Tuesday	00:00-23:59	00:01-23:59
Wednesday	00:00-23:59	00:01-23:59
Thursday	00:00-23:59	00:01-23:59
Friday	00:00-23:57	00:01-23:57

The properties of the copied symbol should be exactly the same, except for the maintenance margin (which can't be accessed directly from the script) and the commissions (some brokers may have different extra properties). The first one you can modify manually, the commissions are explained below.

Applying custom commissions

Commissions won't work by default in the custom symbols (except swaps). To copy them, you'll first need to export them as a file:

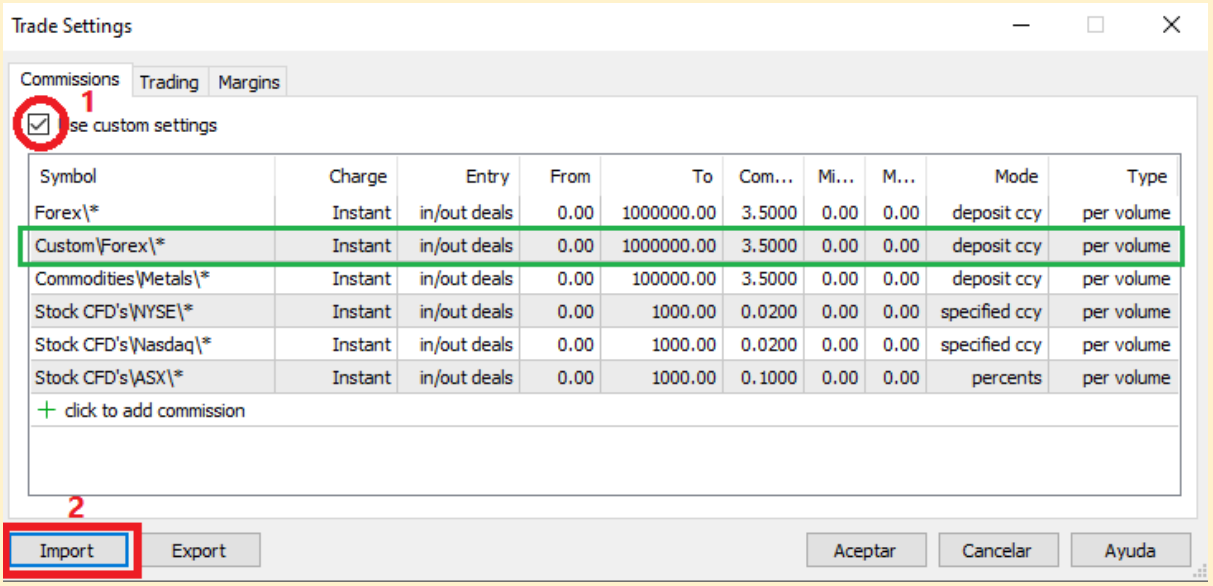


Next, you need to access the exported file (from a notepad is enough) and make these 2 changes to the file:

- Set CommonUseSettings from 0 to **1.** Number "1" is for Custom Settings.
- Copy the Forex section and add **"Custom\"** at the beginning (that's the location of the copied symbols). You'll have to add more sections if the backtest includes metals, for example.

```
CommonUseSettings=1
CommonOrdersLimit=200
CommonPositionsLimit=200
MarginMode=2
MarginFreeMode=1
MarginSOMode=0
MarginFreeProfitMode=0
MarginFlags=0
MarginCall=100.00
MarginStopOut=50.00
CommissionSymbol=Forex\*
CommissionCharge=2
CommissionRange=0
CommissionEntry=0
CommissionValue=3.5000
CommissionRangeTo=1000000.00
CommissionMode=0
CommissionType=1
CommissionSymbol=Custom\Forex\*
CommissionCharge=2
CommissionRange=0
CommissionEntry=0
CommissionValue=3.5000
CommissionRangeTo=1000000.00
CommissionMode=0
CommissionType=1
```

After saving the file, you can import it as custom settings for commissions:



Now, when you run a backtest in the new custom symbols it will have the same broker commissions as the original one.

As a drawback, this method may have less precise bid-asks and a low “Modelling Quality”, which is expected since it only uses D1 data. You can always increase the predefined broker commission to represent the spread impact.

A. Release notes

Version 1.00: Initial version

Version 1.01:

- Redesign of color buffers: in MT4 an extra sell buffer has been added; in MT5, volume color buffer has been removed (main buffer +1 will be used).
- Compatibility corrections of some volume modes in MT4.
- Added 2 indicator modes: Over Signal Colored and Over Level Colored.

Version 1.02: Added Advanced Trailing Stop settings: substitutes common NNFX method with a custom indicator's based trailing stop.

Version 1.03: Minor fixes (MT5): bug correction in <TEMA> indicator.

Version 1.04:

- Handling of invalid symbol error when any symbol isn't found: now they are omitted from the backtest to avoid critical errors.
- Pair preset mode "Symbol File" added.

Version 1.05:

- Real Trade Mode (only MT5) and Custom Optimization Mode added.
- Extended Summary added (closing of trades).
- Major changes in Optimization Parameters: now they can be written directly on the inputs and accessed several times or from several indicators. There are 50 optimization inputs which can be used in any indicator indistinctly.

B. Built-In indicator list

Gray = both versions; blue = only MT5; orange = only MT4

Between parenthesis is the data type which the parameter is parsed to, if the type is an enumerator (Applied Price, MA Method) it is parsed as an integer. "Default" is the complete list of parameters in the default settings (if it had to be written in the corresponding "Indicator Parameters" input).

- **<AMA>**: Adaptive Moving Average
  -Parameters: AMA Period (int), Fast MA Period (int), Slow MA Period (int), AMA Shift (int), Applied Price (int)
  -Default: 9, 2, 30, 0, 1[4]
- **<ADX>**: Average Directional Movement Index
  -Parameters: ADX Period (int), Applied Price (int)
  -Default MT5: 14
  -Default MT4: 14, 0[5]
- **<ADXW>**: Average Directional Movement Index by Welles Wilder
  -Parameters: ADX Period (int)
  -Default: 14
- **<BB>**: Bollinger Bands
  -Parameters MT5: Bands Period (int), Bands Shift (int), Deviation (double), Applied Price (int)
  -Default MT5: 20, 0, 2.0, 1
  -Parameters MT4: Bands Period (int), Deviation (double), Bands Shift (int), Applied Price (int)
  -Default MT4: 20, 2.0, 0, 0
- **<DEMA>**: Double Exponential Moving Average
  -Parameters: MA Period (int), MA Shift (int), Applied Price (int)
  -Default: 14, 0, 1

[4] Close Price is equivalent to 1 in MT5: Applied Price enum doesn't start with "0".

[5] In MT4, Close Price is 0 as usual. In this case, the applied price parameter is only available in MT4.

- **<ENVELOPES>**: Envelopes
  -Parameters MT5: MA Period (int), MA Shift (int), MA Method (int), Applied Price (int), Deviation (double)
  -Default MT5: 14, 0, 0, 1, 0.1
  -Parameters MT4: MA Period (int), MA Method (int), MA Shift (int), Applied Price (int), Deviation (double)
  -Default MT4: 14, 0, 0, 1, 0.1
- **<FRAMA>**: Fractal Adaptive Moving Average
  -Parameters: MA Period (int), MA Shift (int), Applied Price (int)
  -Default: 14, 0, 1
- **<ICHIMOKU>**: Ichimoku Kinko Hyo
  -Parameters: Tenkan Sen (int), Kijun Sen (int), Senkou Span B (int)
  -Default: 9, 26, 52
- **<MA>**: Moving Average
  -Parameters: Period (int), Shift (int), Method (int), Applied Price (int)
  -Default: 10, 0, 0, 1
- **<SAR>**: Parabolic Stop and Reverse
  -Parameters: Step (double), Maximum (double)
  -Default: 0.02, 0.2
- **<STDEV>**: Standard Deviation
  -Parameters: Period (int), Shift (int), Method (int), Applied Price (int)
  -Default MT5: 20, 0, 0, 1
  -Default MT4: 20, 0, 0, 0
- **<TEMA>**: Triple Exponential Moving Average
  -Parameters: Period (int), Shift (int), Applied Price (int)
  -Default: 14, 0, 1
- **<VIDYA>**: Variable Index Dynamic Average
  -Parameters: CMO Period (int), EMA Period (int), Shift (int), Applied Price (int)
  -Default: 9, 12, 0, 1

- **<ATR>**: Average True Range
  -Parameters: MA Period (int)
  -Default: 14
- **<BEARS>**: Bears Power
  -Parameters: MA Period (int)
  -Default: 13
- **<BULLS>**: Bulls Power
  -Parameters: MA Period (int)
  -Default: 13
- **<CHAIKIN>**: Chaikin Oscillator
  -Parameters: Fast MA Period (int), Slow MA Period (int), MA Method (int), Applied Volume (int)
  -Default: 3, 10, 1, 0
- **<CCI>**: Commodity Channel Index
  -Parameters: MA Period (int), Applied Price (int)
  -Default MT5: 14, 6
  -Default MT4: 14, 5
- **<DEMARKER>**: DeMarker
  -Parameters: MA Period (int)
  -Default: 14
- **<FORCE>**: Force Index
  -Parameters MT5: MA Period (int), MA Method (int), Applied Volume (int)
  -Default MT5: 13, 0, 0
  -Parameters MT4: MA Period (int), MA Method (int), Applied Price (int)
  -Default MT4: 13, 0, 0
- **<MACD>**: Moving Averages Convergence/Divergence
  -Parameters: Fast EMA Period (int), Slow EMA Period (int), Signal Period (int), Applied Price (int)
  -Default MT5: 12, 26, 9, 1
  -Default MT4: 12, 26, 9, 0

- **<MOMENTUM>**: Momentum

  -Parameters: MOM Period (int), Applied Price (int)

  -Default MT5: 14, 1

  -Default MT4: 14, 0
- **<OSMA>**: Moving Average of Oscillator

  -Parameters: Fast EMA Period (int), Slow EMA Period (int), Signal Period (int), Applied Price (int)

  -Default MT5: 12, 26, 9, 1

  -Default MT4: 12, 26, 9, 0
- **<RSI>**: Relative Strength Index

  -Parameters: MA Period (int), Applied Price (int)

  -Default MT5: 14, 1

  -Default MT4: 14, 0
- **<RVI>**: Relative Vigor Index

  -Parameters: MA Period (int)

  -Default: 10
- **<STOCHASTIC>**: Stochastic Oscillator

  -Parameters: K Period (int), D Period (int), Slowing (int), MA Method (int), Price Field (int)

  -Default: 5, 3, 3, 0, 0
- **<TRIX>**: Triple Exponential Moving Averages Oscillator

  -Parameters: MA Period (int), Applied Price (int)

  -Default: 14, 1
- **<WPR>**: Williams' Percent Range

  -Parameters: Calc Period (int)

  -Default: 14
- **<AD>**: Accumulation/Distribution

  -Parameters MT5: Applied Volume (int)

  -Default MT5: 0

  -No parameters in MT4

- **<MFI>**: Money Flow Index

  -Parameters: MA Period (int), Applied Volume (int)

  -Default MT5: 14, 0

  -Default MT4: 14
- **<OBV>**: On Balance Volume

  -Parameters MT5: Applied Volume (int)

  -Default MT5: 0

  -Parameters MT4: Applied Price (int)

  -Default MT4: 0
- **<VOLUMES>**: Volumes

  -Parameters MT5: Applied Volume (int)

  -Default MT5: 0

  -No parameters in MT4
- **<AC>**: Accelerator Oscillator

  -No parameters
- **<ALLIGATOR>**: Alligator

  -Parameters: Jaw Period (int), Jaw Shift (int), Teeth Period (int), Teeth Shift (int), Lips Period (int), Lips Shift (int), MA Method (int), Applied Price (int)

  -Default MT5: 13, 8, 8, 5, 5, 3, 2, 5

  -Default MT4: 13, 8, 8, 5, 5, 3, 2, 4
- **<AO>**: Awesome Oscillator

  -No parameters
- **<FRACTALS>**: Fractals

  -No parameters
- **<GATOR>**: Gator

  -Parameters: Jaw Period (int), Jaw Shift (int), Teeth Period (int), Teeth Shift (int), Lips Period (int), Lips Shift (int), MA Method (int), Applied Price (int)

  -Default MT5: 13, 8, 8, 5, 5, 3, 2, 5

  -Default MT4: 13, 8, 8, 5, 5, 3, 2, 4

- **<BWMFI>**: Market Facilitation Index

  -Parameters MT5: Applied Volume (int)

  -Default MT5: 0

  -No parameters in MT4

C. Custom Optimizer Reference

Variables

#WR : Win Rate (0-100).

#DD : Maximum Absolute Drawdown. The value returned is **positive.**

#PF : Profit Factor.

#EP : Expected Payoff.

#SV : Stats Value (Check Trade Dist. Value).

#SS : Stats Shape (Check Trade Dist. Shape).

#ROI : Annualized Return on Investment (%).

#NW : Number of winning trades.

#NL : Number of losing trades.

#NT : Number of trades (#NW + #NL).

#GW : Gross wins (profit of winning trades).

#GL : Gross loss (profit of losing trades, this amount is always **negative**).

#FP : Final profit (#GW + #GL).

Operators

a + b : Addition

a - b : Subtraction

a * b : Multiplication

a / b : Division

a ^ b : Power (a raised to b)

Parenthesis change the precedence of operations: **(a + b) * c**

Functions

Functions don't have a variable number of arguments, if there are more or less arguments there will be an error in the evaluation of the formula (0 or a wrong result).

min(a, b) : returns the minimum value between a and b.

max(a, b) : returns the maximum value between a and b.

sqrt(a) : returns the square root of a.

abs(a) : returns the absolute of a.

logn(a) : returns the natural logarithm of a.

log10(a) : returns the logarithm of a by base 10.

round(a) : returns the closest integer to a.