50% -V3 Multi-currency EA for real MT5 accounts.

**Description**

The EA applies the MAX block indicator. All indicator calculations should be transferred to the EA. The EA should be reliable and continue to work correctly after restarting the terminal/PC, as well as after resetting the EA on the chart or after changing the version. After changing the settings, trading should resume normally (provided that the changes do not affect the main logic). There should be no critical errors causing the EA to stop, including zero divide errors and array overflows. The EA should use computing resources as efficiently as possible and work as fast as possible. It is to be used on real accounts/demo/tester. It should work correctly on all instruments available in the terminal, including all currency pairs, CFDs, raw materials, stocks, cryptocurrencies, including instruments having non-standard names, like GBPUSD-Pro, GBPUSD.m and similar ones. The EA should see them as the same instrument. It also should work correctly if a trader works manually or applies other EAs on the same account. If a deal opened by the EA is closed manually, this should not lead to an error. The details on what to do are described below. Support for Instant Execution, Market Execution and Exchange Execution. There should be two operation modes: for netting and hedging account. Switching should be done in the settings.

**Options**

*General settings*

- netting: yes/no

- magic

- comment

- Trading start date and time

- disable unidirectional transactions by currency: yes/no

- number of simultaneously traded instruments: (1-28)

- close no less than a specified number of instruments before opening new ones:

- total profit for closing all positions

- total loss for closing all positions

- maximum number of positions

- close all deals if the maximum number of positions is reached: yes/no

- complete trading: yes/no

- minimum equity

- in deposit percentage

*Instrument 1: (for example, EURUSD)*

-Slip

***Indicator settings***

- calculate the block size from ATR

- ATR period

- ATR multiplication ratio

- calculate the block size from spread: yes/no

- normal spread:

- spread averaging period: (specify the number of minutes)

- spread multiplication ratio:

- TF1 (BS) block size: (set in price format, for example 0.00001)

- Number of blocks to build: (integer from 1 to 1,000,000)

- Ratio of adding the next timeframe: 0 - not used (set in the price format, for example 0.00001)

- Next timeframe multiplication ratio (0 - not used, fractional, 0.000001 - 100):

- rounding precision (for example, 0.00001)

- Calculate the excess percentage: yes/no

- Build blocks in the chart window: yes/no

- Build in a separate window: yes/no

- Growing block line color

- Falling block line color

***Settings***

- complete trading: yes/no

- initial timeframe for analysis: (1-10,000)

- allow Buy positions: yes/no

- allow Sell positions: yes/no

- lot

- dynamic lot: yes/no

- deposit percentage (assume that 0.01 lot=$1000 and adjust on cross rates)

- adjust the lot based on the point price: yes/no

- calculate the lot size based on the block size ratio to the daily volatility

- ATR period

- normal ratio of the block size to volatility

- potential profit to open a position (set in % with the accuracy of 0.01):

- manage all positions in the series: yes/no

- minimum number of blocks to define an excess

- maximum number of blocks to define an excess (0 - only the minimum value):

- selection step

- use percentage: yes/no

- adjust by the reference period: yes/no

- reference period (0 - not used)

- opening percentage

- closing percentage

- calculate the opening percentage via the probability: yes/no

- opening percentage

- calculate the closing percentage via the probability: yes/no

- closing percentage

- calculate the closing percentage via the root: yes/no

- control the profit via the timer: yes/no

- profit control frequency (in seconds from 0.01 to 3600)

- maximum loss in $

- maximum profit in $:

- ratio from the lot: integer numbers from 1 (0 - not used)

- compensate the spread: yes/no

- adjust closure points: yes/no

- adjustment ratio

- use take profit: yes/no

- use stop loss: yes/no

- number of timeframes to check (view several ones, select the maximum one)

- wait for a new block to appear on TF+1: yes/no

- minimum distance between positions in minutes (0 - not used, the value range is 1-240)

-block point multiplication ratio for the distance between positions of a single TF

- block point multiplication ratio for the distance between positions of different TFs

- maximum number of the series positions:

- close the series in case of the maximum positions in the series: yes/no

- maximum number of positions for the instrument:

- close all positions on the instrument in case of the maximum positions on the instrument: yes/no

- number of the timeframe, after which the profit falls:

- profit reduction factor:

- maximum timeframe:

- close all after reaching the maximum timeframe: yes/no

- open positions on each block: yes/no

- positions on each block if an additional series is created: yes/no

- create additional series: yes/no

- allow counter deals of additional series: yes/no

- maximum number of scales

- TF for the start of the additional series:

- number of blocks of the basic TF to define % of the new series:

- % for confirming the new series:

- use probability for confirmation: yes/no (add a word in the text for confirmation)

- probability for the new series (set in %)

- use the series profit for early closing: yes/no

- profit ratio for early closing: (accuracy 0.01, 0-10)

- loss compensation ratio

- close positions from the end: yes/no

- profit ratio to close positions from the end: (accuracy 0.01, 0-10)

*Instrument 2 (for example, GBPUSD):*

………………

……………..

……………..

*Instrument 28 (for example, CADCHF):*

……………...

**Settings comment**

If the instrument is not specified, it is not used in trading. Only one instrument can be specified. If netting is selected, only one robot can work on a single instrument. Other robots will not use the instrument. No manual trading is to be performed on the instrument. In case of an accidental closure, the EA should remember the volume opened at this price.

1. **Operation algorithm**

Instead of basic charts and timeframes, trading is performed on synthetic timeframes created by MAX bloсk indicator (moved to the EA code). In fact, the EA first builds the arrays of synthetic timeframes and trades on them afterwards. First, I will describe the basic algorithm. It will feature the ability to select modified operation modes. For example, alternative percentage options for opening/closing or additional series.

**Operation**

All trading is carried out in a logical series of positions.

The robot calculates the excess percentage of growing or falling blocks (depending on what is prevailing now) on a certain number of blocks. The number of blocks for searching the percentage is set as a range. The excess is searched on each sample from the range with a certain step. If the growing blocks excess percentage is detected, Sell positions are opened. In case of a falling blocks excess, Buy positions are opened. Positions are opened in the series on each subsequent block by a certain algorithm. The algorithm is activated once per block and works on formed bars (excluding profit and loss control). At the same time, the algorithm looks for an opportunity to move on to the higher synthetic timeframe. If such an opportunity is found, the entire work is moved to the higher timeframe. Position closure also occurs by the excess percentage. It should decrease down to the set value. For example, positions opened up to 95% of the excess and closed up to 60%. The number of blocks, which decreased the excess, forms the profit. After opening the first position series and reaching a certain timeframe (if the series is not complete), sub series logically related to the main series are created. Each subseries may have its own subseries. The series are numbered from 1 to n. The first series is the main one, the second one belongs to the first one, the third one belongs to the second one, etc. The purpose of additional series is to obtain additional profit, which may not be used anywhere or can be used in order to quickly close the main series and complete the entire logically related series. In this case, the profit of each subseries is used to compensate the profit of its parent series. After completing all series for the instrument, the logic is reset to zero and the search for the new series starting point is performed.

The algorithm works independently for 28 instruments defined in the settings. All these instruments are united by common settings exercising control.

**1.1 Main algorithm**

At the start, there are two settings: "**allow Buy** positions" and "**allow Sell** positions". Permissions for the instrument to open Sell and Buy positions are set here. If both are allowed, there are no limitations. If Buy=no, only Sell series can be started. In this case, Buy position opening conditions are ignored, and the search is repeated on a new block till the conditions for opening Sell positions are found.

The EA is designed to work with 28 instruments simultaneously. In fact, all these instruments work independently of each other but are combined by several common settings and belong to a single EA. Each instrument has a separate block of settings. These setting blocks are the same for all instruments but parameters are unique for each instrument. Next, I will describe the operation logic of one instrument. The remaining ones follow it as well.

Several positions can be opened on a single instrument. They are combined in a series. Positions can be oppositely directed if the appropriate setting is selected (the settings are described below). A new series starts after opening the first position on the instrument.

MAX block indicator is used to build blocks of all necessary synthetic timeframes. The indicator code should be moved to the EA. In fact, the data arrays for different synthetic timeframes are built. The indicator works on formed real bars. Its values are recalculated every time a new real bar is closed. Further on, I will call the timeframes created by the indicator TF1, TF2 and TFn. The indicator works only on closures of real bars, while the EA works only on closures of synthetic timeframe blocks. The EA applies TF1 and subsequent timeframes. TF1 is always used to calculate the number of blocks. You can use any initial timeframe for analysis. It is set in the "**initial timeframe for analysis**". For example, the initial timeframe for analysis may be 20. There is no point in building timeframes between TF1 and TF20 in the indicator here if they are not used. But TF1 is always needed to count the time.

Definitions:

- TF1, TF2, TFn – synthetic timeframe index

- TFB – basic timeframe

- Nb(base) – basic number of blocks

- %PB – percentage of prevailing blocks

- excess percentage – percentage of prevailing blocks

- Nb – number of blocks to analyze the percentage of prevailing blocks (set by the range)

- Nb1, Nb2, Nbn – number of blocks in each sample from the range

- Pt – number of points to be added to the closure price of the basic timeframe block

- BSВ – basic timeframe block size

- BStf1 – TF1 block size

- Bstf2-… -Bstfn – block size of the remaining timeframes

**Define the necessity to start the series**

The necessity to start the new series should be defined before launching the new series.

Calculation of the percentage of growing or falling blocks is performed on the initial timeframe (set in the "**initial timeframe for analysis"**) on the number of blocks set using the "**minimum number of blocks to define the excess"** and "**maximum number of blocks to define the excess" range.** The percentage of prevailing blocks is calculated in the EA (further on, designated as **%РВ** or excess percentage). If growing blocks are prevailing, the percentage of growing ones is calculated. If falling blocks are prevailing, the percentage of falling ones is calculated. In other words, the type of prevailing blocks is defined first. The indicator itself counts the excess percentage of growing blocks. We should take it but, since the EA uses the percentage of prevailing blocks (%РВ), we need to recalculate the indicator readings to fit the form used in the EA. If the excess percentage value calculated by the indicator exceeds 50%, for example 65%, it is taken in its initial form and it is assumed that growing blocks exceed 50% (in the current example - 65%). If the excess percentage is less than 50% (for example, 40%), then 100%-40%=60%. This is how the percentage value of falling blocks is obtained. If the excess percentage = 50, this means the percentage of growing and falling blocks=50%. There is no excess and no prevailing blocks.

If the maximum number of blocks is set to 0, the range is set by a single sample and the minimum number of blocks is used. First, the minimum number of blocks is selected from the range and %РВ is defined. Next, "**selection step"** is used to select the next number of blocks and %РВ is defined on it. This happens till the algorithm reaches the upper end of the range. After obtaining the array of values, we need to define the necessity to open a position. To do this, the percentages of prevailing blocks are compared with the "open percentage" value sequentially, starting from the first value. If %РВ exceeds or is equal to "**opening percentage" on one of the samples**, we need to take the sample with the highest %РВ. This sample will be a basic sample and the timeframe will become basic. If %РВ turns out to be the same on several samples, the preference is given to the sample consisting of the greater number of blocks. It is necessary to clarify that, if growing blocks prevailed on the minimum number of blocks, then %РВ is calculated for growing blocks for the remaining blocks and timeframes as well. If %PB exceeds or is equal to the one defined in the settings, then the decision is made to start the series for the instrument. After finding the basic timeframe, we need to remember and fix the real bar date and time the basic timeframe is based on. After making a series start decision, the date and time of both basic timeframe and TF1 are fixed. At first, the basic timeframe and TF1 may be equal to each other if the initial timeframe for analysis is 1. Then the basic timeframe will increase, while TF1 remains and is needed for further calculations. The indicator has the "**fix time**" setting. Here the bar time is fixed, and the basic timeframe and TF1 are built to the right of the real bar with the fixed time. In other words, the number of blocks of these timeframes is increased. This is necessary for the basic timeframe and TF1. Then the basic timeframe will change (increase). The basic timeframe time should be fixed for all instances. The time is fixed in a separate place for each new basic timeframe. The time is not fixed for all non-basic timeframes except for TF1.

Example: no positions on the instrument, **the minimum number of blocks for defining the excess =**16**, the maximum number of blocks for defining the excess=**25**, selection step=**2, **opening percentage=**75. %РВ is defined on 16 blocks (of the initial timeframe). If it is less than 75, wait for the next block. The new block appears, %РВ>=75%, falling blocks are prevailing. Next, we need to define %РВ (of falling blocks) on the next samples. To achieve this, 16+2=18, 18+2=20, 20+2=22, 22+2=24, 24+2=26. Suppose that %РВ (of falling blocks) on 24 blocks exceeds or is equal to the value on 16 blocks. Then the basic sample is the one consisting of 24 blocks. The decision is made to start the series on the instrument.

Since the blocks are closed by close prices once during the bar formation, several blocks can be built during the bar formation. The last built block is always used as a starting point. If 5 blocks were built during the bar formation, there is no point in repeating all procedures on each of the built blocks. The calculations are performed only on the last one.

**Moving on to a higher timeframe**

After deciding to open the series, we need to check %РВ on the nearest timeframe. As an example, let's assume that the initial timeframe is always 1 unless otherwise stated. If a decision to start the series is made on TF1, %РВ is checked on TF2. If %РВ was for growing blocks on TF1, then the check is performed for growing blocks on TF2. If %РВ on TF2 exceeds or is equal to "**opening percentage",** the basic timeframe is TF2 (its time is fixed after TF2 stops being basic, its time becomes unfixed and it is again built to the left of the last real bar). The process is repeated for TF3 and so on, till %РВ of the checked timeframe is not less than the opening percentage. Just like in the first timeframe, %РВ from the range of the number of blocks defined in the settings is checked on each new timeframe. In other words, the check is performed on all samples within the specified range with the specified step, rather than on 16 blocks. Just like in the first timeframe, the priority is given to the sample consisting of a larger number of blocks. For example, the excess is detected on 26 blocks on TF1, as well as on 16 blocks on TF2. The basic timeframe may increase, while the basic number of blocks may decrease, but the first position is not opened yet. Several timeframes are considered for moving on to the next timeframe, rather than one. Their number is specified in the "**number of timeframes to check**". If set to 1, TF2 is viewed after TF1 followed by TF3 etc., till we receive the answer that %РВ is less than the one defined in the settings. If the **number of timeframes for the check=**3, the three nearest timeframes are viewed simultaneously and the priority is given to the highest timeframe if its %РВ exceeds or is equal to the one specified in the settings. Suppose that the necessary excess is found on TF1, while the number of timeframes is 3. This means we need to have a look at %PB on TF2, TF3 and TF4. If %PB exceeds or is equal to the value specified in the settings for one of them, view the next three timeframes and select the basic timeframe and the basic number of blocks. Repeat the process till the ability to move on to the next timeframe disappears. After finding the maximum timeframe with a specified number of blocks, on which %PB exceeds or is equal to the one specified in the settings, while "wait for a new block"=no, check the potential profit of the position, open it, start tracking it and calculate the closing point. Before checking the potential profit, it should first be calculated. The profit calculation is described in point 1.2 and the check method is also described there. Closing and tracking conditions will be described in a separate paragraph. If "**wait for a new block**"**=yes**, no position should be opened yet. Count the necessary number of TF1 blocks to the right (wait till they are formed) from the close price of the last basic timeframe block (the one, on which the trading decision has been made, but this decision has been postponed by the "wait for a new block" setting). The basic timeframes values ***Nb(Base)\*BSB = PN(Base)*** and ***Nb(Tf+n)\*BS(Tf+n)=PN(Tf+n)*** are used to achieve that. Now ***PN(Tf+n)- PN(Base) = Pt*** – number of points that should pass from the basic timeframe block close price. Next, Pt should be divided into the TF1 block size. ***Pt/BStf1=Nbtf1***. The obtained value Nbtf1 may be fractional and should be rounded up to the nearest larger integer (for example, 1.1 is rounded up to 2). A fractional number of blocks will not do because the minimum discretization frequency = 1 TF1 block, while TFn may contain an unclosed TF1 block. Therefore, if the last TFn block is closed not in TF1 closure point, these two blocks should be built from the price of the last Tf1 block, which closed inside TFn. See the example in Figure 1. The obtained Nbtf1 value is the maximum value, after which a decision to check the profit for opening a position is made. But the ability to move on to a higher basic timeframe with each new block on TF1 should be checked till the number of blocks reaches Nbtf1. In other words, the check is performed not only after reaching Nbtf1, but also after each new block on TF1, while Nbtf1 is the last number of blocks, on which the check is performed.

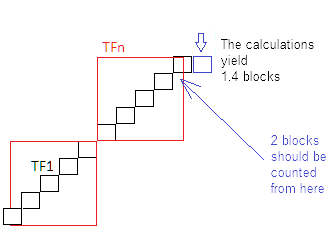


Figure 1

Keep in mind that all timeframes are based on close prices of formed real bars, therefore thee may be a situation, in which the number of passed TF1 blocks exceeds the calculated value after a new candle appears. This is fine provided that the number of passed TF1 blocks is not less than the calculated value. In any case, all operations are to be performed with the close price of the last block of each timeframe.

While the number of Nbtf blocks is not passed, the blocks of several subsequent timeframes (their number is specified in the settings) are built from the basic TFB+1 timeframe on each new closed TF1 block and %PB is checked. If %PB is less than the one specified in the settings on all new timeframes (after the last NBtf1 block appears), the basic timeframe remains the same. The expected profit on the potential position is checked. If it fits the conditions, the position is opened. If %PB exceeds or is equal to the value specified in the settings on one of the timeframes and the number of blocks, the basic timeframe is increased up to the one, at which the condition is fulfilled. A higher timeframe and number of blocks also have a priority here. The timeframe size has the highest priority followed by the number of blocks. The highest timeframe is selected first, while the maximum number of blocks is selected within it. After increasing the basic timeframe, we need to check the closest N timeframes, recalculate Nbtf1 using new data of new timeframes and wait for the necessary number of TF1 blocks from the new basic timeframe. Repeat the process till it is possible to open a position according to the conditions. Before opening a position, calculate its close price and check if the minimum profit condition is met from the open price to the potential close price.

If the potential profit condition is fulfilled, open the position. If not, the series is considered complete without opening a position. The search for a new series opening condition is started afterwards. If the position is not the first one in the series, the next actions depend on the "**enable control for all positions**" parameter. If 'yes', the series should remain open but the position is skipped and everything is done according to the algorithm afterwards. If 'no', the check is not performed for the second and subsequent positions in the series.

**Example: Open percentage =87.5%, number of timeframes to check =3, block range 16-20, selection step=2, BStf1 = 0.00100, timeframe multiplication ratio =1.05, initial timeframe for analysis = 1. %PB = 87.5% detected on TF1 on 16 blocks (14 falling and 2 growing blocks).** The check is performed on 18 and 20 blocks, %PB is less than 87.5% on them. This means the basic timeframe =TF1, 16 blocks. The time for TF1 is fixed (blocks are built to the right). Next, we need to check %PB on the nearest timeframes. The number of timeframes to check =3, which means the check is to be performed on the nearest three ones. The check is performed on TF2, TF3, TF4 (each one is checked on 16, 18 and 20). All timeframes feature the option, in which %PB = 87.5%. So we select the highest one, which is TF4. The maximum number of blocks, on which %PB exceeds or is equal to the threshold value, is defined. The largest value has the higher priority. As a result, %PB >=87.5% (let it be 87.5 for ease of calculation) on 16 TF4 blocks. Now TF4 (16 blocks) becomes the basic timeframe. On TF1, the time remains fixed. The three closest timeframes (TF5, TF6 and TF7 on 16, 18 and 20 blocks) are checked in a similar way. %PB on them is less than the one specified in the settings. This means TF4 (16 blocks) remains the basic one. Now if "wait for a new block"=no, the potential profit on a position is checked. If the conditions are fulfilled, BUY is opened (since %PB was found for falling blocks) and its further tracking begins. If "wait for a new block"=yes, wait till the price moves a certain number of TF1 blocks from the close price of the last TF1 block inside the block on TF4, in which the decision to make TF4 the basic one was made. To do this, we need to find the number of Nbtf1 blocks. Define the block size for TF4. It is equal to (TF1 = 0.00100 = BStf1), 0.00100\*1.05=0.00105=TF2\*1.05=0.00110=TF3\*1.05=0.00115=TF4. Since the block size on TF4=0.00115, the number of blocks is 16, ***PN(Base)= Nb(Base)\*BSB = 16\*0.00115=0.0184***. We need to check %PB on the nearest three timeframes (TF5, TF6 and TF7). To define the number of TF1 blocks, the minimum TF is used (one of those where the check is performed). Therefore, we need to define the block size on TF5. It is equal to 0.00115\*1.05= 0.00120. We need to take the minimum number of blocks from the range (in the current example, it is 16). Next, ***PN(Tf+n)= Nb(Tf+n)\*BS(Tf+n)= 16\*0.00120=0.0192***.

***Pt= PN(Tf+n)-PN(Base) = 0.0192-0.0184 =0.00080. Next, Nbtf1=Pt/BStf1 = 0.00080. The resulting number is less than 1. It is rounded to the nearest highest value, i.e. 1. From the close price of the last TF1 block inside the TF4 block, we need to wait till the price moves one TF1 block. After the price moves the necessary number of blocks (keep in mind that formed bar close prices are used, so more blocks may be passed), we need to build TF5, TF6 and TF7 from the current price.*** Now we need to check %PB on TF5, TF6 and TF7 timeframes, as well as on 16, 18 and 20 blocks on each timeframes. If %PB is less than the one specified in the settings, a position is opened. If %PB exceeds or is equal to the threshold value on one of the timeframes and number of blocks, repeat the adjustment of the basic timeframe, i.e. check %PB on the nearest timeframes whose number is specified in the settings. The loop is formed: detect an excess, search the nearest TFs, raise TF if necessary, wait for the necessary number of TF1 blocks and check the nearest TFs. If an excess is detected, check the nearest TFs again and wait for the necessary number of TF1 blocks. The loop repeats till it is possible to raise TF. After the algorithm receives the answer (after waiting for the necessary number of TF1 blocks) that the excess on the nearest TFs is below the threshold one, the loop is stopped. For example, %PB = 87.5% on TF6, 18, blocks. The next three timeframes (TF8, TF9, TF10) are checked, the basic one is defined. Then wait for the new TF1 block with the last block equal to Nbtf1 from the block closure price inside the basic timeframe. The process continues till the profit check is performed to open a position.

The objective of all these actions is to find the maximum timeframe and the maximum number of blocks, on which %PB exceeds or is equal to the number specified in the settings, and define the maximum basic timeframe before opening a position. Keep in mind that if "initial timeframe for analysis" features %PB for falling blocks, the percentage of falling blocks is defined on all other timeframes as well.

The initial indicator features the "Fix time" ability. It is needed for the basic timeframe for TF1. Since the indicator is built from right to left from the close price of -1 bar, the Close time of the bar is fixed after defining the basic timeframe. The indicator now is constructed from left to right, i.e. new blocks of the basic timeframe are added. Old blocks remain. There is no need to recalculate the already built blocks, new ones should be added instead. Calculation resources should be used as efficiently as possible. This function is needed only for the basic timeframe and TF1. If a timeframe is no longer basic, this fixation should be removed for it to save calculation resources. At this stage, the time can be fixed only for two timeframes simultaneously. After completing the series, the time on TF1 is reset as well and not fixed before an excess is found on the necessary timeframe.

Figure 2 shows an example, in which the excess is defined on 10 TF1 blocks. "**Initial timeframe for analysis**"=1 here. Block size on TF1 = 0.00010. TF generation mode is selected in the indicator settings as adding ratio = 0.00010. This means the block size on TF2 is 0.00020. For simplicity, "**minimum number of blocks to define an excess**"=10 and is set with no range. Opening percentage=100%. The percentage of prevailing growing blocks of 100% is found on 10 TF1 blocks. Check an excess on the nearest three timeframes. The check shows the percentage is less than 100. The decision to open a Sell position is made but "wait for a new block on TF+1: yes", therefore we need to calculate the number of TF1 blocks to wait for. The block size on TF1 of 0.00010 is taken and multiplied by the number of blocks to define the excess (10). 0.00010\*10=0.00100

TF2 = 0.00020 is multiplied by the number of blocks to define the excess. There are 10 of them. 0.00020\*10=0.002000. Now 0.00200-0.00100=0.00100 points. 0.00100/0.00010=10 TF1 blocks should be waited out till the price moves from the block with the fixed time. The nearest timeframes are checked along the way after each fully formed TF1 block with the aim of early transition to the higher basic timeframe.

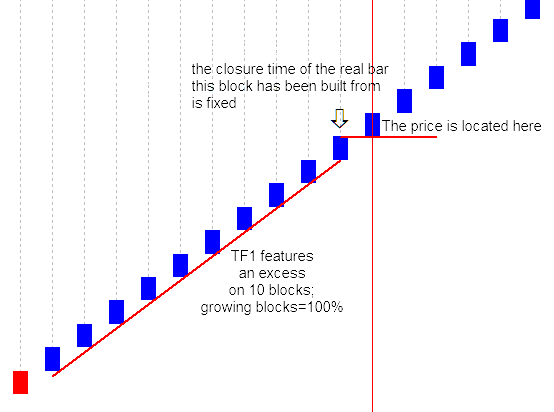


Figure 2

Figure 3 (green blocks) shows the price moving 0.00100 points, as well as forming of 10 TF2 blocks featuring the excess of 100% as well. The nearest number of timeframes defined in the setting is viewed, and the answer informing that higher timeframes feature no excess is received. TF2 becomes the basic timeframe now. The close time of the bar the last green block is built for is fixed here. The decision about opening a position and waiting for the necessary number of blocks to form on TF3 is made. The close time remains fixed on TF1.

0.00020\*10=0.002000, block size on TF3=0.00020+0.00010=0.00030. 0.00030\*10=0.00300.

0.00300-0.00200=0.00100, now 0.00100/0.00010=10 – maximum number of blocks to wait till the price moves from the block with the fixed time on TF2. The nearest timeframes are checked along the way after each fully formed TF1 block with the aim of early transition to the higher basic timeframe.

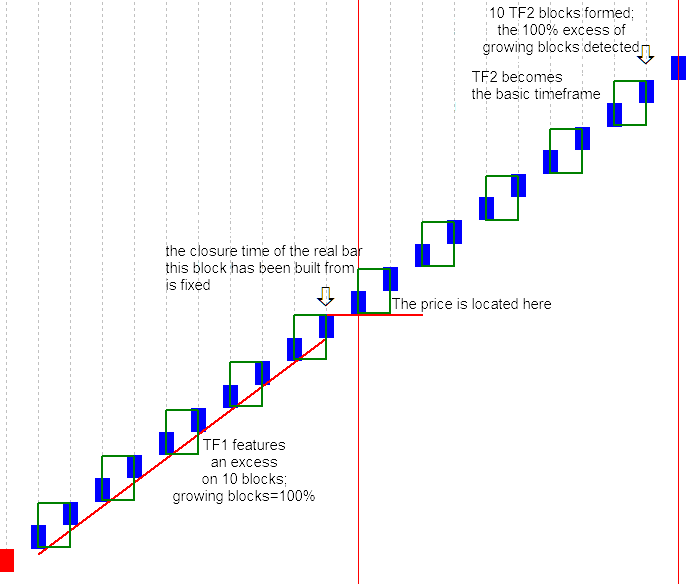


Figure 3

Figure 4 shows that the price has moved yet another 10 TF1 blocks and the excess of 100% has formed on 10 TF3 blocks. TF3 becomes the basic timeframe now. We need to check the nearest timeframes for an excess. If it is detected, move on to a higher timeframe. If not, calculate the number of TF1 blocks to wait in order to make an attempt to move on to TF4. The nearest timeframes are to be checked along the way after each fully formed TF1 block with the aim of early transition to the higher basic time frame. The calculation has been done. Now wait for 10 TF1 blocks to check the ability to move on to TF4. I.e. the calculated number of 10 blocks is the maximum number of blocks to be skipped.

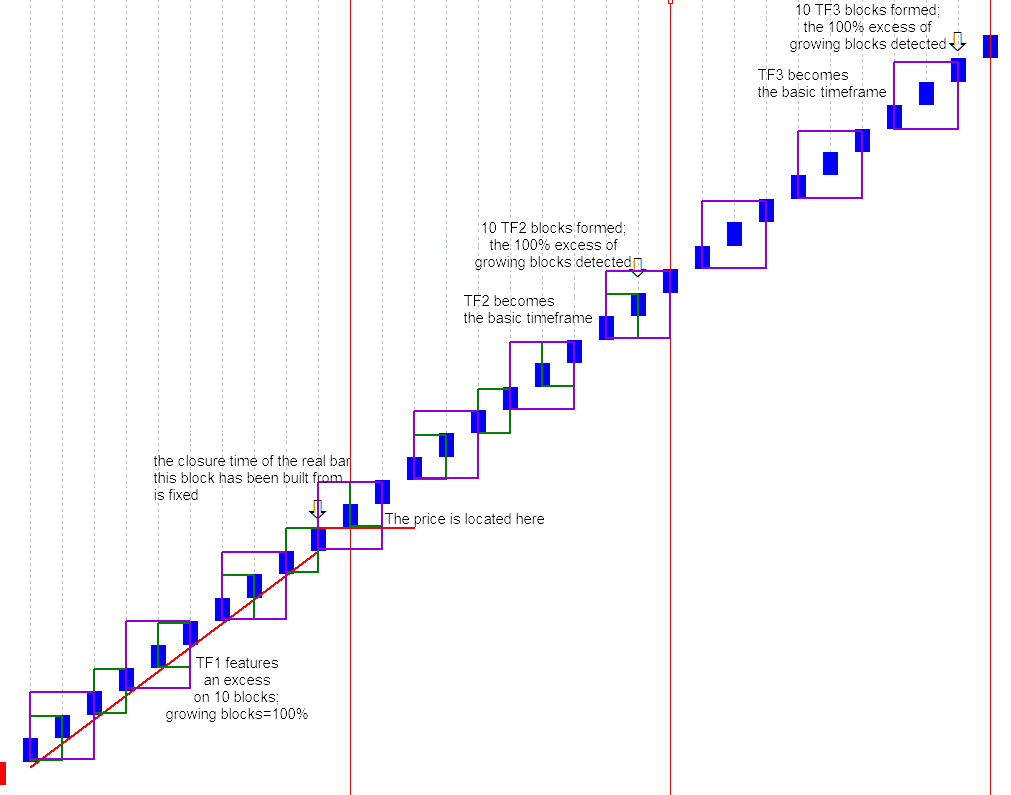


Figure 4

Figure 5 shows that 10 TF1 have passed and the transition to TF4 has not happened. This means it is time to open a Sell position on the block marked with the arrow on the right.

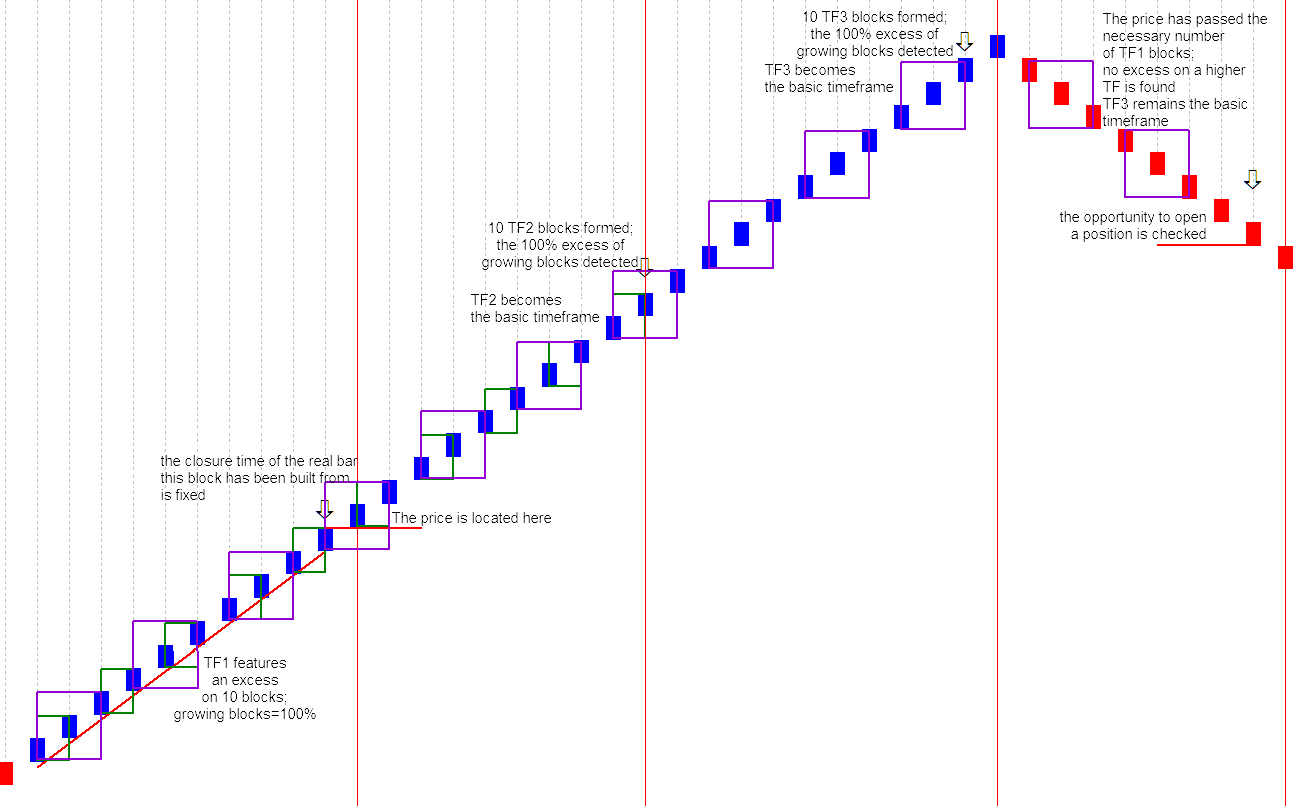


Figure 5

Before opening the first position, we should check whether this action is feasible. To do this, calculate the point where the series should close with profit and compare it with the value defined in the settings.

**Maximum timeframe.** The settings provide the ability to limit the maximum timeframe, after which the timeframe stops increasing. This is defined in the "**Maximum timeframe**". After reaching the timeframe, there are two operation options. If **"close all after reaching the maximum timeframe"**= yes, the series simply completes and all positions belonging to the series are closed. Further on, I will describe how to work with multiple series. If the first series reaches the maximum timeframe and the second one is open, then only the first one is over, while the second one has its own maximum timeframe. The second series becomes the first one, etc.

If **"close all after reaching the maximum timeframe"**= no, the timeframe does not increase anymore. Positions are opened according to the algorithm at each new block of the basic timeframe if necessary.

The timeframe increasing algorithm works on each newly formed block of the basic timeframe and is launched before a new position in the series is opened. To open a position in the series, we need to determine whether the basic timeframe has changed.

* 1. **Close with profit**

- %CP – percentage to close positions

- Nb(Base) – number of blocks of the basic timeframe, on which an excess is found

- NPb – number of prevailing blocks

- Nbа – number of basic timeframe blocks

- Nb+1(Base) – current number of basic timeframe blocks

- NbТ – total number of blocks for closing, on which %CP is closed

- NbC – number of remaining blocks

- FB – block with a fixed time

- FBC- block closing price with a fixed time

- FBL – closing price of the last formed basic timeframe block

- FB1, FB2, FBn – closing price of blocks after a block with fixed time

- BSB – block size of the basic timeframe

- PC – series closing price

- PT – target profit

- PR(ask) – current Ask price

- PR(Bid) – current Bid price

- PB - number of profitable blocks

The closing point is calculated before opening a position.

After opening the first position on the instrument, the new series is considered to have started. The series objective is to close all positions with a total profit. The point of closing with profit is calculated first for a position and then for the entire series. To calculate the point, we need to know the basic timeframe number, block size on that timeframe, the number of blocks, as well as the number of growing and falling blocks. To calculate the point of closing a position, the "closure percentage" is defined in the settings. We need to take the number of prevailing blocks in the sample (NbP) and divide it into the percentage for closing positions (%СР). Since %CP is defined in the settings in %, it should be divided by 100. The equation looks as follows: ***NbТ = NbP/(%СР/100)***. This is how we obtain the closest number of blocks, on which positions can be closed with profit. Now we need to find how many blocks of the opposite (to the prevailing ones) direction should appear. To do this, subtract the number of prevailing blocks from the total number of blocks. ***NbT-Nbа=NbС***. Round the obtained value up to an integer one. Now calculate the price for closing a position/position series (РС). To do this, the number of points to be passed by the price is subtracted from the close price of the last formed block of the basic timeframe. ***РС= FBL-(BSB\*NbC)***. Upon reaching the price, all series positions should be closed and the series is to be completed. Wait for the start of the new series. After forming each new block of the basic timeframe, the close price should be recalculated considering new data and adjust the closure point. Here FBL means the last block to show the general equation. During the calculations, it is first replaced with FBC, and then, upon incoming new blocks, FB1, FB2, …, FBn.

Let's consider an example in Figure 6. The necessary %PB is found on 11 blocks. It is equal to 90.9%. In the settings, %СР closure percentage =62.5, the block size is 1. For more simplicity, assume that the open price of the very first block (to the left)=0, then the close price of block 11 =9.

***NbТ = NbP/(%СР/100)=10/(62.5/100)=16***. The series is to be closed on the total of 16 blocks if the situation develops as in the Figure. The number of potential opposite (to prevailing ones) blocks is calculated. In the current case, bullish blocks are prevailing, which means bearish blocks should appear.

***NbC=NbT-Nba=16-11=5***. There should be 5 bearish blocks. The series closing price is now calculated. PC= ***FBC -(BSB\*NbC)=9-(1\*5)=4***. The resulting closing price is 4. Now we should close the series if the price is equal or less than 4. This is how it works if nothing changes.

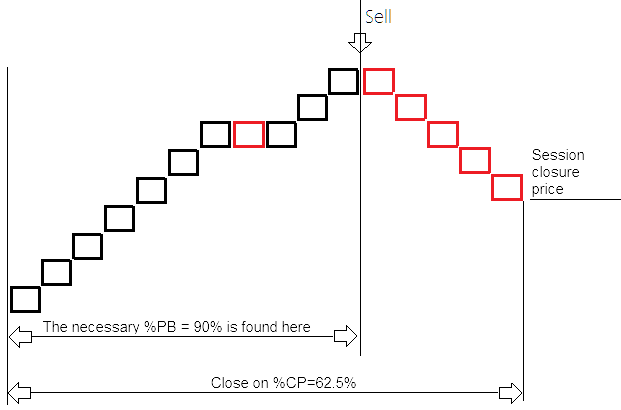


Figure 6

With each new block on the basic timeframe, the series close price should be recalculated after the FB block because the price may not follow this scenario.

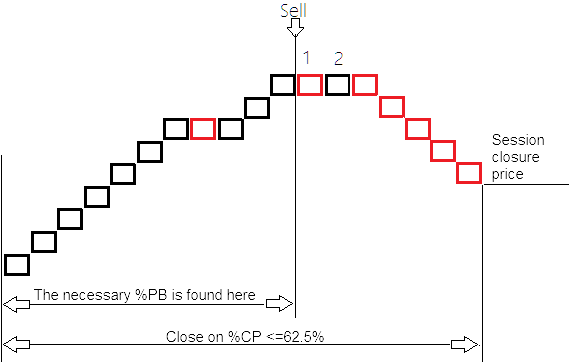


Figure 7

If we follow the example in Figure 7, this will look as follows: Block 1 appears, after the sample, on which the necessary excess percentage has been found, ***NbТ = NbP/(%СР/100)=10/0.625=16***

***NbC=NbT-Nbа=16-12=4***. This means 4 more bearish blocks are to appear. Nba is increased by 1 here since there are 12 blocks in total now. ***PC= FB1-(BSB\*NbC)=8-(1\*4)=4***. Here the close price FB1 = 8, since the block was bearish. It turns out the series close price has not changed remaining at 4.

Block 2 appears afterwards and the close price should be recalculated again. Here:

***NbТ = NbP/(%СР/100)=11/0.625=17.6***. Here NbP is increased by 1 since the total number of growing blocks becomes equal to 11. ***NbC=NbT-Nbа=17.6-13=4.6***, rounded to the nearest integer NbC=5. 5 blocks should appear downwards from the current point but the FB2 close price = 9, therefore ***PC= FB2-(BSB\*NbC)=9-(1\*5)=4***. The series ends when the price reaches 4 or a lower value. Adjustment of close prices continues as new blocks appear till the position is closed.

The timeframe increasing module continues working. Therefore, if the conditions are met and the basic timeframe is increased, all calculations are performed for the new basic timeframe. The module works with the basic timeframe and tracks closing new blocks on the basic timeframe, which is dynamic in turn. Accordingly, when the basic timeframe is increased, FB always changes and block closing prices change as well.

There are two profit control options here set in "**control profit via timer: yes/no**". If "no", the crossing of the position closing level by the price is controlled with each new basic timeframe block. After a new block appears, the condition of closing with profit is checked. If the price is lower or equal to the closure point for Sell positions, the position is closed. For Buy positions, the price should be above or equal to the closure point. If **"profit via timer": yes**, the timer is set to control the profit. A crossing of the specified threshold by the price is controlled by the timer. The timer is set in "profit control frequency". Its range is 0.1 - 3600 seconds. In case of 1 second, the check for reaching the specified threshold by the price is done once per second. If the threshold is reached, positions are closed. For Buy positions, the price should exceed or be equal to the threshold, while for Sell positions, it should be less or equal. There are two options here in regard to which the price is to control for crossing the Bid or Ask threshold. The switch is called "compensate spread" in the settings.

If "compensate spread" =no, Buy position is closed if the Bid price exceeds or is equal to the specified threshold. Sell position is closed if the Ask price is less than or equal to the threshold.

If "**compensate spread**" =yes, Buy position is closed if the Ask price exceeds or is equal to the specified threshold. Sell position is closed if the Bid price is less than or equal to the threshold.

After activating the closure, all series positions should be completed regardless of the price direction during the closure.

The settings provide the ability to enable take profit. If "use take profit=yes", a take profit is set for a close price of each position in the series. After that, it is modified with the change of the series closing point. If the closure price does not change, the order remains unmodified. However, the activation of the parameter does not disable the profit control by the EA supplementing it instead. The profit is checked before placing an order. If a series should be closed, there is no point in placing an order. If an order cannot be placed because the price is too close to the market, the EA controls the profit without orders. Here we need to check the broker limitations to avoid the order placement error. A situation may arise when the EA sends a request to close positions, while they are closed by take profit. This should not cause a critical error. The EA should understand that its positions are closed according to the operation algorithm and close the series. If a decision to close positions (by any of the methods) is put into action, there should be no unclosed positions remaining. If some positions are closed by take profit or stop loss, a decision is made to complete the series and forcefully close all the remaining series positions regardless of whether the price reaches the threshold value or not.

**Adjusting the closing point**

There is an additional parameter "**Adjusting the closing point".** If it is set to "no", the calculated closure point is used with no adjustment. If it is set to "yes", the calculated closure point should be adjusted depending on the profit obtained on open positions. The adjustment does not work for the first position in the series. In other words, if the first position is opened, only the control of opening by the potential profit is performed (described below). But if the series has more than one position, then, after calculating the closure point, we need to calculate the profit the series obtains if positions belonging to it are closed in that point. To do this, we need to find the average position open price. It is found as the sum of open prices divided by the number of positions. In case of Buy positions, subtract the average open price from the close price. The result is the number of points. If the resulting value is positive, there is nothing to adjust. If negative, the closure point should be adjusted. All is similar for Sell positions. The only difference is that the closure point price is subtracted from the average open price.

To adjust closure points, subtract BSB\*(adjustment ratio) from the average open point for Sell positions. BSB size of the basic timeframe block in the price format (for example, 0.00129). "**Adjustment ratio**" – fractional number with the accuracy of 0.1. It can take values starting from 0. The same goes for Buy positions. BSB\*(adjustment ratio) is added to the average open price.

If adjustment is enabled, the adjusted closure point is used in all calculations where necessary.

**Position open control by potential profit**

Before opening the first position in the series, we need to calculate its potential closure point and make sure the potential profit exceeds or is equal to the one defined in the settings. If the potential profit is less than the one defined in the settings, the first position in the series is not opened. The series is deemed complete, while the algorithm moves on to searching for the point of opening a new position. This is done by the "**potential profit for opening a position"** parameter. The parameter is set in %. To manage opening a Buy position, the current Ask price is subtracted from the calculated close price. ***PT=PC-PR(Ask)***. The number of profitable blocks is calculated further on ***PB=PT/BSB***. The obtained number of profitable blocks is divided by the remaining number of NbC blocks and multiplied by 100. The obtained value should exceed or be equal to the value defined in "**potential profit for opening a position".**

***(PB/NbC)\*100>=***"**potential profit for opening a position".** If the condition is met, the first position in the series can be opened. If not, the position cannot be opened and the series completes without opening a position. The search for the new series start point is performed.

The algorithm is similar for opening Sell positions, however the target profit equation is different, ***PT=PR(Bid)-PC***, ***PB=PT/BSB***, ***(PB/NbC)\*100>=***"**potential profit for opening a position".** "**Potential profit for opening a position"** is fractional with the accuracy of up to 0.01, for example 80.52.

The algorithm is executed only for the first position in the series. The remaining positions are opened with no control if "**enable control for all positions in the series**"=no.

If "**enable control for all positions in the series**"=yes, the control algorithm is executed for each position in the series. There is no need to complete the series for the remaining positions if ***(PB/NbC)\*100*** is less than the one specified in the settings. The position is simply skipped. But the execution of the entire basic algorithm goes on considering that the skipped position becomes virtual. It takes part in further calculations, but is not open. The algorithm should remember that there should be a position but it is not opened and all further calculations should be made taking into account that there should have been a position in this place. Accordingly, if the second position should be opened on a higher timeframe than the first one, the closure point of the first position should be recalculated as if the second position is open.

**Decreasing the series profit after reaching a certain timeframe**

For positions opened after the profit decrease activation, the opening control is disabled if it was enabled.

After the basic timeframe becomes greater than or is equal to the value defined in "**timeframe number, after which the profit falls**", the profit starts decreasing on each subsequent timeframe. The profit is calculated on each subsequent timeframe via "**profit decrease ratio**". We need to find the average price of opening positions (MP) belonging to the series. The prices of opening these positions are summed up for that and divided into the number of positions. Now for Sell positions, subtract the series close price (PC) from the average price to obtain the number of points before closure (SC).

For Sell: ***MP-PC=SC***

For Buy: ***PC-MP=SC***

The number of points before closure (SC) should be multiplied by the profit multiplication ratio (Kp). Next, we need to calculate the new series close price (PC2).

For Sell positions: ***MP-SC\*Kp=PC2***

For Buy positions: ***MP+ SC\*Kp=PC2***.

The profit decreases at each basic timeframe increase. All this is related to the case when the timeframe has reached the specified value and the profit should fall. If the timeframe is increased again, the equations become as follows

For Sell positions: ***MP-SC\*Kp\* Kp =PC2***

For Buy positions: ***MP+ SC\*Kp\* Kp =PC2***

Multiply by Kp as many times as the number of timeframes passed from the timeframe, at which the profit started to fall.

For example, the profit starts decreasing on timeframe 10, equations for Sell positions:

***MP-SC\*Kp=PC2***, for Buy positions: ***MP+ SC\*Kp=PC2***. The timeframe 11 already features Sell equations:

***MP-SC\*Kp\*Kp=PC2***, for Buy positions: ***MP+ SC\*Kp\*Kp=PC2***. And so on.

If the function is enabled, PC is used in the calculations in point 3 (using additional series profit). The price is used wherever a series close price is needed.

**Other options of closing with profit**

The additional possibility of closing the series with profit is set in "**maximum profit in $**". 0 – not used. It indicates how many USD (deposit currency) the entire series should earn in order for it to complete. All positions belonging to the series (including closed ones) are considered. The profit control is performed via the timer. It is common for all. Upon reaching the value, the series is complete. Here I am talking about all interconnected logical series. Further on, point 3 states that there can be more than one series on the instrument. Therefore, multiple additional series can be opened and closed. The profit obtained on all of them is summed up with the current profit on open positions. After the specified value is reached, all series are closed. All logical series are reset and trading starts anew. In other words, this is the maximum possible profit the entire series is able to obtain together with its logical subseries.

If "**lot ratio**" is equal to a non-zero value, the maximum profit (after which all logical series are complete) is calculated depending on the working lot of the instrument. The point price for the instrument and for the used lot should be calculated here. For example, for 0.01 lot on GBPUSD, 1 point on 0.00001 quotes= $0.01. In case of Gazprom shares, 1 lot costs 0.01 rubles. The point price is multiplied by "**lot ratio**" to obtain the profit to be obtained so that the entire series is closed. The closure is performed as described above. The only change is the method of specifying the maximum profit. Example: Point price of 0.01 lots on GBPUSD = $0.01, the lot ratio =100,000. The point price is multiplied by 100,000 yielding $100. When the total profit of all logically connected series becomes greater than or equal to $100, all positions are closed and trading starts anew.

* 1. **Opening additional positions in the series**

After opening the first position in the series, the new block of the basic timeframe appears after the fixed time block (FB). If Sell is opened, the close price of the new block (FB1) is above the FB block close price. No moving to a higher timeframe is performed. One more Sell position should be opened. The same conditions should be met in case of Buy positions. However, the FB1 close price should be below the FB close price. Just like in the first position, the position open decision is made here. But it is postponed by the basic timeframe increase module till all the calculations are done to check the ability to move on to a higher timeframe. The timeframe increasing module performs all the necessary calculations. If the decision is made that it is currently impossible to move on to a higher timeframe, check the distance from an open position to the current price. If the distance is greater than or equal to the block size of the basic timeframe, a position should be opened. If the distance is less than the basic timeframe block, then we need to track it till it exceeds or becomes equal to the basic timeframe block and open positions. If a position does not open before the next block of the basic timeframe, it is skipped, and the algorithm continues its work, i.e. determines the need to open a position on a new block. For Sell positions, the new position open price should exceed the open price of the previous one for the distance no less than the basic timeframe block, while for Buy positions the new position open price should be less than the open price of the previous one. This is repeated on each new block of the basic timeframe.

When moving on to a higher timeframe occurs after opening one of the positions, a position of the appropriate direction should also be opened. This is assumed to be a new basic timeframe block. The distance between the previous and new positions should not be less than the basic timeframe block (of the new basic timeframe). Here we should also control the distance till a position is opened, a new block appears or a transition to the next timeframe is performed. Note: the distance control before opening a position is valid only before a new block of the basic timeframe appears. After the new block appears, the need to open a position is determined anew.

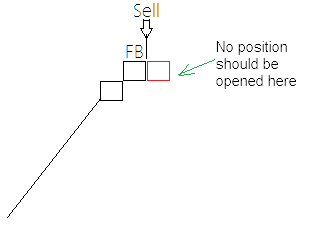
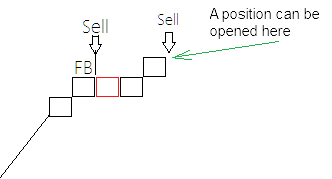
 

Figure 8

Figure 8 shows schematically when positions can be opened. The provided example is for Sell. For Buy, all is done similarly, but in the opposite direction. This continues after each new block. The general idea is as follows: if the price goes against an open position (a loss is increasing), new positions should be opened every time the price moves a certain number of points. Calculating the points is described here. The minimum distance between positions can be adjusted in the settings. It can be set separately for positions from a single timeframe and for positions from different timeframes. This is done via the two parameters defining the block size multiplication ratio to control opening new positions.

Setting: "**block point multiplication ratio for the distance between positions of a single TF**" – fractional value with the accuracy of 0.00001. It is set as the basic timeframe block size multiplication ratio. The value can exceed 1 or be less. The block size is multiplied by this ratio, and the distance from the new position to the previous one should not be less than the obtained value. The open price value should be taken from the open position properties. For example, the block size is 0.00010, the ratio is 1.1, thus 0.00010\*1.1=0.00011. For Sell positions, take Bid prices. For Buy positions, take Ask prices. If multiplication yields the value less than the minimum price step, the value is assumed to be 0.

"**Block point multiplication ratio for the distance between positions of different TFs**" – fractional value with the accuracy of 0.00001. It is set as the multiplication ratio of the basic timeframe block size. It is used to decrease or increase the minimum distance between positions when moving on to a higher basic timeframe. The open price value should be taken from the open position properties. Example: the block size of the basic timeframe is 0.00100. The transition to a new basic timeframe has been performed. The block size has increased up to 0.00200. Thus, 0.00200\*ratio. This will be the minimum distance between positions. The ratio can be less than 1 or more. If multiplication yields the value less than the minimum price step, the value is assumed to be 0. For Sell positions, take Bid prices. For Buy positions, take Ask prices.

The settings feature additional parameters limiting the distance between positions. "**Minimum distance between positions in minutes**" is set in minutes from 1 to 240. 0 – not used. The parameter means that a position cannot be opened closer than the specified number of minutes to the previous one. The logic is as follows: receive a response that a position should be opened, check the distance in points and then in minutes. If all is well, the position is opened. If minutes do not fit the conditions, wait a necessary number of minutes and check the distance in points again. This is valid till a new block of the basic timeframe appears. If the position is not opened, all is reset after a new block of the basic timeframe appears and the position is considered skipped.

**Opening positions on each block**

Another operation option of "**Open positions on each block**"=yes. If the prevailing blocks are growing ones, a new position appears after forming each new growing block regardless of the price location relative to the previous position. If the prevailing blocks are falling ones, a position is opened after each new falling block appears. Since blocks are constructed based on close prices of real bars, there may be several blocks. In this case, the decision about opening a position is taken based on the last block. Example: growing blocks were prevailing, five growing blocks appeared during the forming of a real bar, one Sell position is opened.

If the setting is activated, "**block point multiplication ratio for the distance between positions of a single TF**" does not work, while "**block point multiplication ratio for the distance between positions of different TFs**" works. As soon as a new basic timeframe is found, trading switches to it. In other words, moving on to a higher timeframe is a priority. Positions are opened on a new timeframe according to a similar principle. See the example in Figure 9:

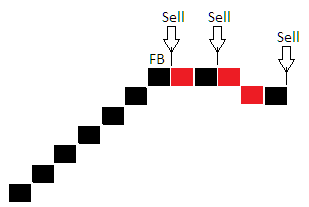


Figure 9

The ability to open positions on each block can be blocked by the function of moving trading to another scale to be described in section 3 (opening additional series). The blocking ability is set in "**do not open on each block if an additional series is created**". If "yes", opening positions on each block is disabled when an additional series appears. This does not apply to positions opened when switching to a new timeframe.

**1.4 Closure with a loss**

The series can also be closed with a loss. If the loss on the entire series on the instrument exceeds or is equal to "**maximum loss in $**", all positions of the series are closed. All positions belonging to the series (including closed ones) are considered here. A single series cannot bring more losses than is set in this parameter. The number is written and controlled in the deposit currency. The overall loss and profit are controlled via the timer. There is an ability to set stop losses. To do this, activate "**use stop loss".** If stop losses are activated, they are all set into a single price for all positions within the series. The price is calculated based on the common lot of open positions, average open price, one point price for the lot (and the instrument) in the deposit currency, as well as the total profit of the already obtained series in closed positions (profit can only be negative).

Point price/(maximum loss + profit of closed series positions)= number of points before a stop loss

For Buy positions, SL= (average open price) - (number of points before a stop loss)

For Sell positions, SL= (average open price) + (number of points before a stop loss).

A stop loss changes and moves every time a position in the series is opened or closed.

The second closing option with a loss:

**minimum equity**. Set in the deposit currency. This is a general setting for the entire robot, it is monitored continuously (at each tick). 0 – control is disabled. If the equity becomes less or equal to the number, all positions opened by the EA are closed, trading is stopped and no more new positions are opened. We need to make sure that all positions belonging to the EA instance are closed and no orphaned positions remain. If the equity has increased when closing positions, positions should still be closed and no longer open. For example, the minimum equity is $1000, during closure it increases up to $1100, positions are closed till all are closed and the EA opens no new positions any more.

**1.5 Maximum number of positions**

The settings provide the ability to limit the maximum number of positions the series is able to open. This is done in "**maximum number of the series positions**", value range 1-10,000. If the number of positions in the series reaches this value, new positions are not opened. The EA waits for a position to be closed according to one of the conditions either in profit or in loss. If an additional series (described below) is started, positions may be opened again on the new series. In other words, this parameter relates only to one series. If the number of positions has decreased in the series due to some reasons and become less than the maximum one, positions can be opened again. The setting has an additional parameter "**close the series in case of maximum positions in the series**". If "yes", all positions of the series are closed upon reaching the maximum number of positions in the series regardless of their profit, and the series is completed. If additional series are opened (for example, the second one), the second series becomes the first one.

There is an additional limitation of the number of positions for the instrument. If the number of open positions on the current instrument exceeds the value specified in "**maximum number of positions on the instrument**" (1-10,000), new positions are not opened on that instrument till the number of positions becomes less than the specified one or till all positions are closed according to one of the conditions. There is also an additional setting "**close all positions on the instrument in case of the maximum positions on the instrument**". If "yes", then upon reaching the maximum number of positions, all positions on the instrument are closed, all series are closed and trading starts anew.

1. **Percentage modification for opening and closing the series**

"**Open percentage**" and "**close percentage**" defined in the settings are used in the main algorithm for opening and closing positions. This is the simplest operation mode, but these percentage values may be set in another form. If "**calculate the opening percentage via the probability**"=yes, the percentage for opening is no longer taken from the open percentage but instead it is calculated according to the equation based on the number of blocks. The open percentage is still used in all calculations. However, it is taken not from the setting but from the calculations. The same goes for the close percentage. If "**calculate the closing percentage via the probability**"=yes, the close percentage is calculated using the equations and depends on the number of blocks. These settings work independently of each other. For example, "**open percentage**" can be defined in the settings, while the close percentage is calculated or vice versa. Perhaps, both percentages will be calculated rather than one.

If "**calculate the opening percentage via the probability**"=yes, then the probability for calculating the open percentage is set in "**opening probability**". The probability is set in %, for example, 70.5 with the accuracy of 0.000000000000000001. The same goes for the case when "calculate the closing percentage via the probability"=yes. The "closing probability" is set here. The open and close percentages are used in the basic algorithm for work and calculation, and they are to be used further on. However, they are not taken from the settings but are calculated for each number of blocks via the open and close probabilities. Thus, if the number of basic timeframe blocks changes, the open/close percentage values automatically change as well. For example, to start the series, the number of blocks is set in the range of 16-24. This means the open percentage will be different for each number of blocks and calculated as described below. The same goes to the close percentage. If during the work, the number of blocks of the basic timeframe is increased (it is increased if the price starts fluctuating in a narrow range), the close percentage is calculated anew for each subsequent number of blocks.

- %Open – open percentage

- %Close – close percentage

- Со – opening probability

- Сс – closing probability

- C – number of possible combinations for this option

- P – event probability

For each number of blocks encountered during the operation, we need to build the table of the open/close probability corresponding to the open/close percentage since the open/close percentages are used in the main algorithm calculations. Take the considered number of blocks and build the table with the probabilities of all possible combinations for it according to the equation:

С=n!/(m!\*(n-m)!

P=C/2^n

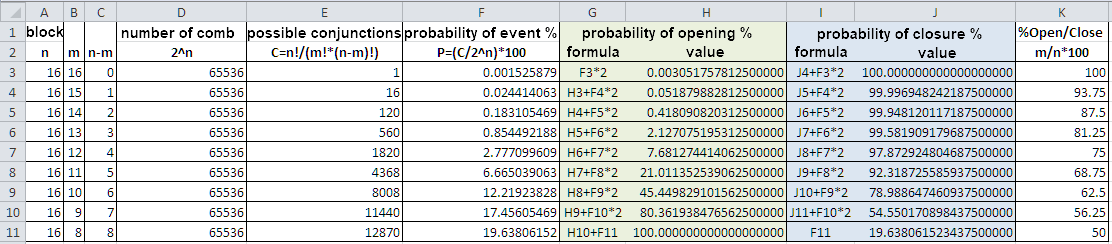


Figure 10. Even n

Figure 10 shows the table and equations for calculating its values. Such a table should be made for each number of blocks encountered during the EA operation and saved in order not to recalculate it every time. The table is calculated a bit differently for an even and an odd number of blocks. If 16 blocks have already been encountered, the table should be saved and a data is to be taken from it. If the number of blocks has not been encountered yet, for example 64, the table should be created and saved for future use to save resources in the tester/optimizer and in real trading. A single table is created for all instruments and should remain in RAM. There is no need to save it to a hard disk. If the EA is relaunched, it should create new tables as necessary. This means the same tables are used for EURUSD, GBPUSD and all 28 instruments if the number of blocks is similar. There is no point in performing calculations every time a value should be obtained. m numbers are calculated by successive subtraction of one from n till m is equal to n/2 in case of an even n. If n is an odd number, subtraction is performed up to n/2 rounded to the nearest larger integer. If n/2=17/2=8.5, it is rounded to 9. The same is done to all odd numbers. The difference between even and odd n can also be seen when calculating open and close probabilities in the table. In case of an even n, the probability of opening is calculated in the table starting from the top table row. Here the event probability is multiplied by 2, the second row is calculated as the previous row + the next event probability multiplied by 2 etc. The multiplication ratio of 2 is present everywhere except for the lower row where it is not needed. In case of an odd n, the opening probability is calculated in a similar way from the upper line but the lowest line already has the multiplication ratio of 2. The equations are set in the table near the value to the left and labeled "equations".

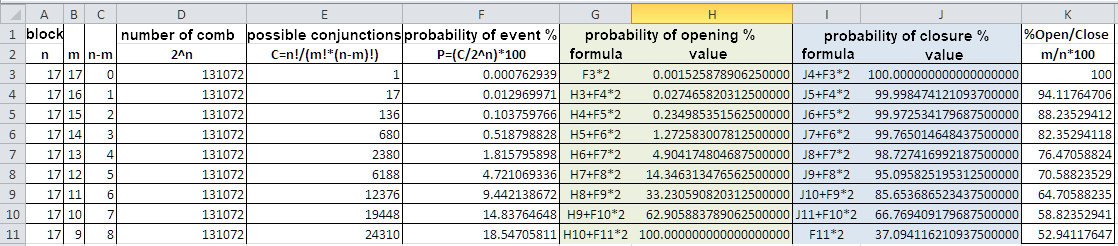


Figure 11. Odd n

The same goes for the closing probability in the table. The closing probability in the table is calculated starting from the lowest row. If n is even, the event probability is taken without multiplication and is multiplied by 2 in all other rows. If n is odd, the event probability is multiplied by 2 in both the bottom and all other rows. The tables feature all the equations. Figure 11 shows the table with n=17.

**Example:** The opening probability is set to Со=0.5%, 16 blocks have passed. The opening probability column contains the number less than or equal to 0.5. As we can see from the table, the nearest value that is less or equal to 0.5% is 0.41. It corresponds to 87.5%. Since the main algorithm applies the open percentage, the open percentage of 87.5% is used. The same goes for the closing probability. If the closing probability settings Сс = 92%, the nearest value less or equal to 92% is found in the table. In the table, it is 78.98%. It corresponds to the close percentage of 62.5%.

If you know a simpler and more resource-saving way to perform the calculation, you are welcome to use it.

When a series is opened, the close percentage is always calculated for the increasing number of blocks as new blocks appear on the basic timeframe. For example, the excess has been found on 16 blocks and a position has been opened. The close percentage is recalculated on each newly appeared block before the position is closed. I.e. first, on 16, then on 17, 18, 19, 20 blocks etc. till the transition to the next TF is performed or the series is complete. A separate close percentage is used on each new number of blocks. This is possible thanks to the mechanism of closing with profit described in 1.2.

**The second option of calculating the close percentage**

The close percentage may be calculated in two ways depending on what is selected in the settings. The close percentage can be calculated as described above but if "**calculate the closing percentage via the root"=** yes, the close percentage is easier to calculate. To perform the calculation, take the current number of blocks of the basic timeframe consisting of the number of blocks where the excess has been found + the number of blocks passed after the block with the fixed time (let it be n for simplicity). When moving on to the next timeframe, n should always be redefined. Besides, n should be redefined after each new block appears or the basic sample size is increased. Equation for defining the close percentage

***(root(n))/2+n/2 = nb***, next ***nb/n\*100=close percentage***.

Example: 100% of prevailing bullish blocks are found on 10 blocks, a Sell position is opened and the FB block time is fixed. The close percentage is calculated as ***nb=(root(10))/2+10/2 = 6.581***. Then, the close percentage = ***6.58/10\*100=65.81%***. Next, the percentage is inserted to the algorithm for defining the close point. If the FB block is followed by yet another block, the close percentage should be adjusted on each new block. Figure 12 shows 10 new blocks that have appeared sequentially.

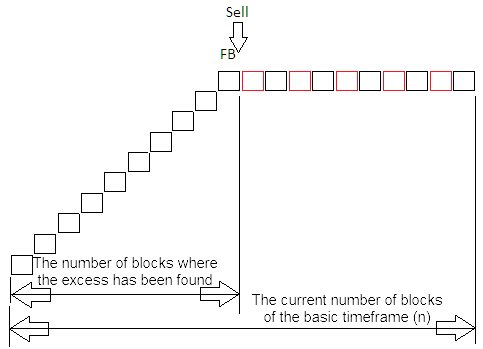


Figure 12

The close percentage here is ***nb=(root(20))/2+20/2=12.24***. ***Close percentage = 12.24/20\*100=61.2***.

If the basic timeframe is increased, the percentage is recalculated. If the number of blocks is added, the percentage is recalculated.

The setting works instead of the close percentage. It is also able to work together with the opening probability or with a rigidly set open percentage. If the setting is activated, the rigidly set close percentage is not used.

1. **Creation and operation of additional series**

There may be more than one series on one instrument at a time. New series are logically linked to each other, although they work independently. New series appear when the timeframe of the first series becomes large enough and fluctuations of lesser timeframes fit into it. Further on, the first opened series will be called basic or first. Each subsequent series will belong to its parent series. In other words, the first series will be the parent one for the second series, the second series will be the parent one for the third series etc. Additional series may have positions of the same direction as the first one. This is defined by the "**enable opposite deals of additional series**". If "yes", positions in the second series can be Sell, while positions in the first series can be Buy, positions in the third series are Sell again, etc. If "no", positions in the additional series can only be in the direction as in the first series. If the first series features Buy positions, then all additional ones are Buy, too.

**3.1 The algorithm of forming the new series**

If the first series is open, the search for the second series starts after the timeframe of the first series exceeds or is equal to the one specified in "TF to start an additional series". To start the second series, the search for the open percentage (defined in the settings or calculated via the opening probability) for analysis should be started on the initial timeframe. Like in the first series, the opening percentage is searched for on the number of blocks set by the range in the settings. If the percentage of prevailing blocks exceeds or is equal to the opening percentage on one of the samples, we need to define whether the second series should be started.

There are two methods of defining prevailing blocks.

1. If "**allow opposite deals of additional series"** =no, %PB is calculated for blocks of the type present in the first series. For example, if the first series featured an excess on the growing blocks and position type was Sell, we should look for the necessary %PB for growing blocks as well. Accordingly, if the first series featured Sell positions, only Sell positions can be opened in the second series.
2. If "**allow opposite deals of additional series"** =yes, %PB is calculated for prevailing blocks regardless of which type of blocks prevailed in the first series. For example, in the first series, prevailing blocks are growing while in the second series, prevailing blocks may turn out to be falling or growing. Accordingly, positions of the second series may turn out to be either Sell or Buy regardless of the first series position type.

We need to take all samples where the necessary %PB has been found. To confirm the need for the second series, we need to calculate the range of blocks (out of the ones where the necessary %PB has been found), on which the percentage of prevailing blocks should be checked for its correspondence to the value set in "% for the new series confirmation" (set in %), for each sample. To confirm the series, the percentage of prevailing blocks in the newly calculated range set by the Bmin(m1) - Bmax(m1) number of blocks should be less or equal to the value set in "**% for confirming the new series**".

NPb(s2)m1 – number of prevailing blocks series 2, sample 1. Here in brackets (s2) – belonging to the series 2, m1 – belonging to the first sample.

NPb(sn)mn – number of prevailing blocks, series n, sample n.

%PB(s2)m1 – percentage of prevailing blocks series 2, sample 1. Labeling logic is the same as above.

%PB(sn)mn – percentage of prevailing blocks series n, sample n.

%PV – percentage of confirming the new series from the settings.

Bmin(m1) – minimum number of blocks of sample 1 to confirm the start of the new series.

Bmax(m1) – maximum number of blocks of sample 1 to confirm the start of the new series.

The Bmin(m1) - Bmax(m1) ranges are calculated for all samples where the excess has been found and each sample is checked in its own range.

The lower boundary of the block range, on which the percentage for confirming the new series is to be searched for, is calculated as

**Bmin(m1)=NPb(s2)m1/(%PV/100) –** calculated for the least sample out of the ones, on which the necessary %PB is detected.

**Bmin(m2)=NPb(s2)m2/(%PV/100) –** calculated for the next sample. It consists of the greater number of blocks compared to the first one, etc., till all samples, on which the necessary %PB has been found, take part in the calculations.

To calculate Bmax(m1), we need to take the number of formed blocks of the basic timeframe of the first series set in "the number of basic TFs to define % of the new series". Calculate the number of blocks of the current timeframe (the one, on which the necessary %PB has been found on the second series) fitting into the specified number of blocks of the basic timeframe. Add the number of blocks of the initial timeframe of the second series not fitting into the last block of the basic timeframe because it has not yet been formed. The resulting value is Bmax(m1). "**The number of blocks of the basic TF for defining % of the new series**" defines the number of the already formed blocks. But large blocks are formed slower than small ones, therefore built small blocks will almost always be present. They should be added to the number of small blocks fitting into the specified number of large ones. If a small block is closed at the same time as the large one and there are no small blocks not fitting into the large ones, 0 is added. Suppose that two blocks of the basic timeframe of the first series are specified in the settings. The number of blocks of the initial timeframe should be calculated inside these two blocks. These are to be the last two large formed blocks to the right.

Now we need to check the percentage of prevailing blocks in the Bmin **–** Bmax range for each sample. The check is carried out sequentially over the entire Bmin **–** Bmax range by adding one block till the range is complete or till the percentage of prevailing blocks becomes less or equal to the one specified in "**% for confirming the new series**" (or another percentage calculated in an alternative way).

Figure 13 shows the entire calculation. Suppose that the first series is opened. It features 20 timeframe blocks. According to the settings, we need to search the point for opening an additional series upon reaching the 20th timeframe. The initial timeframe = 4. The range is set in one number = 7 blocks, %PB =85.7, %PV=50, the number of blocks of the basic TF for defining the new series =2.

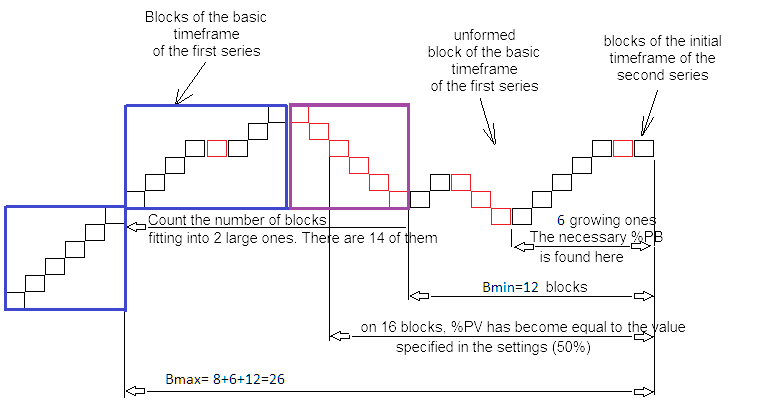


Figure 13

On the last 7 blocks of the initial timeframe of the second series, the excess has become more or equal to the value defined in the settings. The decision should be made about starting the series. ***Bmin(m1)=NPb(s2)m1/(%PV/100) = 6/(50/100)=12***. Thus, the minimum number of blocks, on which %PV is checked, is 12. Next, we should calculate Bmax(m1). To do this, we take the number of blocks of the basic timeframe for defining the new series (2) and the number of blocks of the initial timeframe within them is calculated. Two blocks of the basic timeframe feature 14 blocks. Next, we need to calculate the number of blocks within the incomplete large block. There are 12 of them. ***Bmax=14+12=26***. The resulting range is 12-26 blocks, in which the series start should be confirmed or refuted. Check the percentage of prevailing blocks starting with 12 blocks to the right. To confirm the possibility of starting the second series, the percentage of dominant blocks on one of samples 12-26 should be less than or equal to %PV. 12 blocks feature 8 growing and 4 falling ones, 8/12\*100=66.67%. This is more than %PV, which means further check is required.

13 blocks 8 growing and 5 falling ones, 8/13\*100=61.54%

14 blocks 8 growing and 6 falling ones, 8/14\*100=57.14%

15 blocks 8 growing and 7 falling ones, 8/15\*100=53.33%

16 blocks 8 growing and 8 falling ones, 8/16\*100=50.00% <=%PV- the match is found. No further checks are needed. The new series can be created. The value here cannot be less than 50 since the percentage of prevailing blocks is always calculated. If the equality is not fulfilled, the second series cannot be started. If the condition is not fulfilled within the specified range, wait for forming the new small block and repeat the procedure. Since the time is still not fixed for small blocks, they are built to the left when closing each new real bar. Therefore, the check for opening the second series is performed after closing each new real bar.

Since the percentage of prevailing blocks of a certain type has been initially found (for example, growing blocks are prevailing), %PV should be calculated for this type of blocks (for example, for growing ones).

Since the initial timeframe in the indicator is built from right to left and the number of blocks is specified in the settings, it may turn out that this number of blocks will not be enough for filling the necessary number of blocks of the basic timeframe. If the number of blocks is insufficient, we need to multiply this number of blocks by 2 and try again till the number of blocks is sufficient.

**Further work with the new series**

After the ability of creating the new series has been confirmed, the series follows the algorithm, just like in the first series. In other words, the necessity to move on to higher timeframes and the high-priority number of blocks are defined, positions are opened, profits are calculated if necessary etc. The entire algorithm is identical to the first series.

**Blocking position opening on each block**

If the first series features the ability to open positions on each block and

"**Open positions on each block**"= yes, it is possible to disable opening positions on each block if the second series appears. To do this, set "**positions on each block where an additional series is created**" to "no". In such a combination, the first series should stop opening positions on each block till it has the second series. At the same time, the second series is able to open positions on each block till the third one appears, etc. If additional series are complete, the first series is able to open positions on each block.

**Opening new series**

After opening the second series, it can be closed or the third series can appear. Each series is able to create its subseries. The first series creates the second one, the second series creates the third one, etc. The second series creates the third one if the basic timeframe of the second series is increased up to the value specified in "**TF for starting an additional series**". Each series is able to create an additional one if its timeframe reaches this value.

It may so happen that the first, second and third series have been created. The first and second ones have been closed with profit, while the third one has remained. In this case, it becomes the first one and the process continues till all series and subseries are closed.

If the second series has closed with profit, while the first one has not closed yet, another second series can be opened if the conditions are right. The process may be repeated till the series has the timeframe higher than the one specified in the settings. It is able to create new series.

**3.2 Alternative percentage of confirming a new series**

"**% for confirming the new series**" can be specified in the settings and remain stable or it can be specified via the probability and calculated depending on the number of blocks it is measured on. The calculation algorithm still applies the percentage for confirming the new series but it will be adjusted for each number of blocks. As before, the number of blocks for confirmation is set as the Bmin **-** Bmax range, but the percentage changes for each number of blocks. For example, in case of Bmin, this is one percentage value, while for Bmax, this is another percentage value. For each subsequent number of blocks exceeding Bmax, the percentage value is different. The function is enabled in "**use probability for confirmation**": yes/no. If yes, the percentage is calculated via the probability and set in "**probability for the new series**" in %. The probability accuracy is set as in point 2 (0.0000000000000000001), the parameter is set in % in the format 50.0000000000000000000. The calculation is similar to the one specified in point 2 with the only difference that the probability of opening and closing was calculated in the previous case, while here we calculate the percentage for confirming the new series. A sample calculation is shown in Figure 14.

As in 2, the tables are created for each number of blocks and stored in RAM. The table with the given number of blocks is specified for all series and instruments. Therefore, it is not removed after completing the series but is used as needed. The sample tables are shown in Figures 14 and 15. Figure 14 shows an example for the even n number, while Figure 15 demonstrates an example for the odd n number. The difference between the tables is in m value and in the fact that the lowest event probability for the even number of blocks is not multiplied by 2 but is multiplied in case of the odd number. The equations are set in the table next to the calculated values.

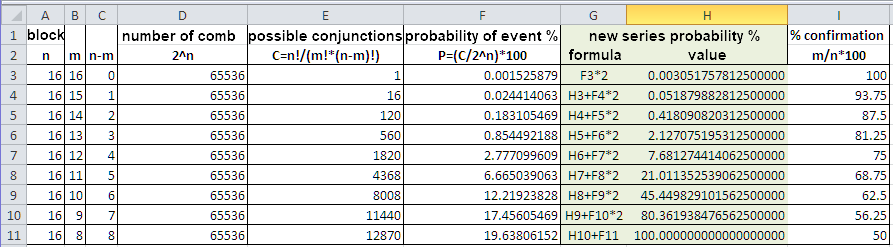


Figure 14. Even n

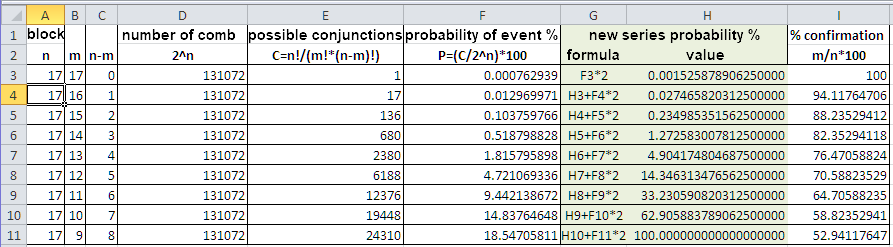


Figure 15. Odd n

If you know a more resource-saving option, you are free to use it. Make sure that the logic described here remains intact.

For example, the number of blocks 16, probability of the new series = 70%. In the "new series probability" column, we need to find the number greater or equal to 70. The nearest value is 80.36, which corresponds to 56.25%. This means the percentage for confirming the new series is 56.25%. This corresponds to 9 blocks in one direction and 7 in another. If such a combination is found on 16 blocks, the necessity to open a new series is confirmed.

**3.3 Using the profit of additional series**

If we get profit after closing an additional series, it can be used to partially compensate a loss/profit on the main series. Here we have two options of compensating a loss/profit which are enabled in the settings. The sequence of the series is important here. The profit in the second series is used to compensate a loss/profit of the first one, the profit of the third series is used to indemnify for a loss/profit of the second series, etc. Only one compensation algorithm is able to work at a time. Therefore, the situation, in which both parameters are set to "yes", is not used and considered an error. In other words, it is not important how these blocks can interact, since they are not meant to work simultaneously. However, both parameters may be set to "no". They may be disabled and not used at all. In this case, the profit on closed series is not taken into account anywhere.

1. "**use the series profit for early closing**"=yes. For more simplicity, we will consider

the case when the first series opens the second one and the profit of the second series is used to compensate the first series. The idea behind the third, fourth and all subsequent series is similar. After the second series gets a certain profit, it is closed. The profit is multiplied by the ratio set in "**profit ratio for early closure**". The ratio is fractional up to 0.01 and its value range is 0-10. The profit of the second series is marked as Profit 2 and now the profit value is stored in memory. The positions of the first series have the supposed closure point calculated. Now we need to calculate the amount of profit in the deposit currency the first series is to receive if all positions belonging to it are closed in this point. To achieve this, we need to sum up all lots of open positions and find their average open price. It will be equal to the sum of open prices divided by the number of open positions. Now we need to find the number of points from the average open price up to the point of the expected closure, calculate the point price for the common lot and multiply the number of points by the point price. Here the point is the minimum price change. Here is how we get the estimated profit. For Sell positions, the average open price minus the close price = number of points. On the contrary, for Buy positions, the proposed closure price minus the average open price = number of points. If the profit is negative, this is a loss. After describing the algorithm with the positive profit, the algorithm with the negative profit is described. Then we should subtract the profit of the second series from the proposed (positive) profit to get the adjusted proposed profit. Now we should close positions of the first series when the profit for them becomes more or equal to the adjusted profit. In other words, the closure point does not move, but the EA starts to additionally control the current profit on the first series positions. The control is performed via the common timer or after forming each new block depending on what is selected in the settings. The positions are closed according to the condition that comes the first, in the closure point or according to the adjusted profit. The adjusted profit should be recalculated each time the proposed closure point or the profit obtained by the second series changes, or a position of the first series is added/removed when the average open price changes.

The second series can be closed more than once. The profit from all second series is summed up after being multiplied by the ratio.

Figure 16 shows an example of open positions and the expected point of their closure.

The first series features 4 positions of lot 0.01 at 0.0005, 0.0009, 0.0012 and 0.0013. Estimated close price = 5. Total lot = 4\*0.01=0.04. Average open price is equal to

(0.0005+0.0009+0.0012+0.0013)/4= 0.000975. Let's assume that the minimum price change value = 0.0001. Let the price of the minimum quote change of 0.0001 (for the lot of 0.04) is $1. We need to find the number of points up to the closure price of 0.000975 -0.0005=0.000475. Now the profit from this number of points is equal to $1/0.0001\*0.000475=$4.75 (estimated profit). Suppose that the profit of $1.2 remains from the second series. Multiply it by the ratio from the settings (let it be 0.8) 1.2\*0.8=0.96. Now 4.75-0.96=3.79. When the profit of open positions exceeds or is equal to 3.79, they should be closed. The first series is considered complete. If positions are not closed with a profit, while the new second series is opened and closed instead, its profit should be multiplied by the ratio and added to the previous profit of the second series. Let the profit of the new second series be $2, then 2\*0.8=1.6. Then 0.96+1.6=2.56 – the profit of the second series. Suppose that the estimated closure point and the expected profit have not changed, then 4.75-2.56=2.19. When the profit on open positions reaches 2.19, the first series can be completed.

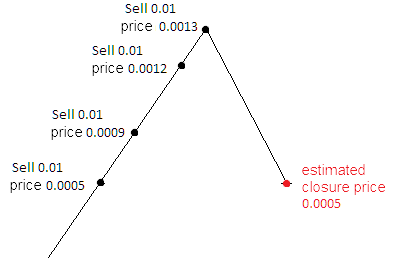


Figure 16

It works similarly for the rest of the series. The profit of the third series is used for the early closure of the second series etc. If the second series features the third one and the second series is closed considering the profit of the third one, the profit of the second one is calculated as the profit from closed positions + the profit of the third series that participated in closing the second series. If we take the first series as an example, its profit is 2.19+2.56=$4.75, as if it has closed in its calculation point.

If the profit of the first series in the closure point is negative according to the calculation, then this is a loss. The profit of the second series is added to the current loss of the first series. The additional parameter "**loss compensation ratio**" (called **CCL**) is added here. Next, we calculate the profit in the deposit currency we are able to obtain from the number of points received via BSB multiplied by the loss compensation ratio. ***BSB\*CCL=PP.*** To get profit, we need to find PP in the deposit currency for the current lot. Here we should know the point price in the deposit currency for the current common lot. The obtained value will be called **PPi**. To allow the series complete early without reaching its estimated closure point, the equality ***(the current loss of the first series) + (the profit of the second series)>=PPi*** should be fulfilled. The equality is checked using the timer. Accordingly, the current loss constantly changes together with the profit of the second series.

Example: in the estimated closure point the first series shows the loss of $10, 5 positions are opened 0.01 lots each = 0.06 lots. The current loss of the first series -$12, the profit on the second series comprises +$18. BSB=0.00238, CCL=0.5. Next, BSS\*CCL=0.00238\*0.5= 0.00119. Let the price of 0.000001 for 0.06 lots is $0.06. Then the price of 0.00119 is equal to PPi=$7.14. Next, we need to check the equality ***(the current loss of the first series) + (the profit of the second series)>=PPi***. To do this, -$10+$18>=$7.14 - the equality is fulfilled. The first series should be closed. If the equality is not fulfilled, the series should not be closed. Instead, we need to check the equality again via the timer.

If the profit of the first series is negative in the closure point, while the profit of the second series is negative, there is no need to do anything since there is nothing to sum up or calculate. In this case, the first series can be closed only in its calculated closure point.

1. "**close positions from the end**"= yes.

If the second series was closed, while the first one remains, calculate the profit (in the deposit currency) obtained when closing the second series and remember this value. Further on, it will be called Profit 2. If the value is positive, this is a profit, if it is negative, this is a loss. Now we need to find the position of the first series with the greatest loss in the deposit currency (the loss is a profit with a minus sign) and add the profit obtained when closing the second series to it. The loss of such a position is marked as ML1 (1 – the position belongs to the first series). The profit on the position should be negative for this calculation. If there are no positions with a negative profit, the algorithm is different. I will describe it below. Next, the profit obtained when closing the second series, multiplied by "**profit ratio for closing positions from the end**", is added to the maximum loss of the position of the first series. If the obtained value exceeds or is equal to 0, the position from the first series featuring the maximum loss should be closed. Subtracting the profit of the closed series from the loss of the closed position results in a residual value. Next, we should find yet another position of the first series with the maximum loss and add the residue from the previous time to it. If the value exceeds or is equal to zero again, this position should also be closed, etc. till the residual value is no longer sufficient for closing loss-making positions or till loss-making positions are over. This residual value is stored in memory. Then the second series may be opened and closed with a profit. In this case, that profit is multiplied by the ratio and the residual value from the previous series is added. After that, the process is repeated.

If the profit is not sufficient for closing the most loss-making position after multiplying the profit of the second series by the ratio, the check is repeated after the new block of the basic timeframe of the first series appears.

If the positions are not closed but the new second series is opened and closed with a profit, this profit is also multiplied by the ratio and added to the profit of the previous second series. The profits of all second series are summed up and residues are summed up as well. All this continues till the first series is closed. After that, there is no need to store the profit of the second series in memory any more. In other words, the part of the profit obtained from all second series is used to compensate the part of the loss of the first series positions. In fact, the profit of all second series is summed up. If the residual value from compensating loss-making positions remains, it is summed up as well. The objective of the second series is to accumulate the necessary profit so that the first series gets a certain profit.

**R = ML1(1)+Profit 2\* КPt** – the equation of closing the first loss-making position of the first series

**R2= ML1+R –** the equation of closing the second loss-making position of the first series.

КPt – "**profit ratio for closing positions from the end**"

ML1 – current loss value for a position with the maximum loss of the first series. If the value is negative, this is a loss. If there are no loss-making positions, a position with the least profit is taken and the value is positive.

Profit 2(1) – profit of the second series closed first

Profit 2(2) – profit of the second series closed second

Profit 2(n) – profit of the second series closed n th

Profit n – profit of the n th series

R – residual profit of the second series.

**If the first series features no loss-making positions,** find a position with the least profit in the first series

(the algorithm is launched only after there are no loss-making positions in the first series). It already has a calculated and defined point of closure with a profit. We need to calculate the profit in the deposit currency the position should obtain after closing in the calculated point. Now this profit should be compared with the one remaining after closing the second series. If the profit remaining from closing the second series exceeds or is equal to the profit the position is able to obtain, the position should be closed. The residue from the profit of the second series is reduced by the potential profit of the closed position. Next, yet another position with a lesser profit is detected and the procedure is repeated. After closing each subsequent position, the profit from the second series is reduced by the value of the potential profit of these positions.

After each closed second series, the profit of the second series grows. Each new profit of the second series is multiplied by the ratio and added to the previous ones and residues from the previous ones. After each closed position, the profit of the second series falls by the value of either a loss or a potential profit depending on which position is closed.

**Example:** Let КPt=0.8. The first series is opened, it features three positions, the first one is -$10, the second one is -$5 and the third one is +$2, the potential profit for the profitable position is $8. The first series features the second one with two positions -$2 and +$1, while the second one features the third series. For more simplicity, let the profit for open positions remain static. The third series is closed with the profit of +$1. This is insufficient to close loss-making positions of the second series. Trading is continued. The second series has yet another third series closed with the profit of $5. The overall profit of the third series is equal to 1\*0.8+5\*0.8=$5. $5 is enough to close a loss-making position of -$2 of the second series. -$2+$5 =$3. Thus, $3 remains from the third series. The second series features no more loss-making positions. There is only a profitable one but its potential profit should be $6, while the current one is $1. 1+3<6, this means the position of the second series should not be closed. But another third series opened and closed with the profit of $8. $8\*0.8=$6.4, $3 of the residual value is then added. 6.4+3=$9.4.

Now there is the profit of the third series of $9.4. This is sufficient to close the position of the second series with the potential profit of $6. 6-1=$5 is insufficient for reaching the closure point. 9.4-5=$4.4. This is more than necessary to get the potential profit and the position of the second series is closed. The remaining residue is $4.4. This means the profit of the second series comprises the sum of the positions closed with a profit and of the profit remaining from the third series. ($4.4+$0+$6)\*0.8=$8.32 (0 – profit of a loss-making position closed with a compensation, $6 – profit of a profitable position closed with a compensation, 4.4 – profit remaining from the third series). The maximum loss of the first series position of -$10 and the profit of the second series are not sufficient for covering the loss. The trading is resumed. The ability to close the most loss-making position of the first series is checked on each new block of the basic timeframe. Yet another second series has opened and closed according to the algorithm. The profit of +$15 has been received on it. This value is multiplied by the ratio of КPt 15\*0.8=12. The profit remaining from the previous series 12+8.32=$20.32 is added to the obtained value. Now we have the profit on the second series = $20.32. This value is sufficient for closing a loss-making position of -$10+$20.32=$10.32 remains. The first series has yet another loss-making position of -$5+$10.32=$5.32. The first series has all loss-making positions closed, while the residual value is not sufficient for closing a profitable position (the potential profit is $8). If yet another second series is opened, the profitable position of the first series is closed in the point without reaching the previously calculated $5.32. In other words, the residual profit of the second series of $5.32 remains and the last position of the first series is closed when the profit on it remains equal to 8-5.32=$2.68.

These compensation algorithms are valid for all series. So the profit of the second series participates in closing positions of the first series, while the profit of the third series takes part in closing positions of the second series, etc. After closing the series, all profit obtained by the series considering the series belonging to it is stored in memory. For example, if the second series featured the third one, while the third series had the fourth one. The second, third and fourth series closed, and the entire profit obtained after the closure of all the third and fourth series is considered the profit of the second one. To put it differently, if the second series closed, it obtained a certain profit. However, this profit may be formed by the third and further series (all positions of the second series may theoretically be loss-making).

1. **Lots**

The position lot may be set in the EA by an instrument in different ways.

1. "**Lot**" – a stable lot, for example 0.01.
2. **"dynamic lot" –** if yes, the lot is calculated from the deposit. To do this,

the percentage set in "**percentage from the deposit" should be defined with the accuracy of 0.1, for example 10.1.** Here it is assumed that each $1000 on the account = 0.01 lots = 100%. For $10,000=0.1 lots = 100%.

The calculation is simple: deposit\*deposit percentage/10,000,000= lot. The obtained value is rounded to the nearest correct value. If the broker allows using the lot of 0.01, the rounding is performed up to 0.01. If the broker allows the lot of 0.1, then the rounding is done accordingly. There are brokers allowing the usage of the minimum lot of 0.1 but with the accuracy of 0.01. Here we should avoid errors when opening positions. If the deposit is $10,000 and the percentage is 50, then 10,000\*50/10,000,000=0.05. The possible percentage range is 0.1- 1000.

1. **"adjust lot based on price"**.

If "yes", the lot size is adjusted based on the point price in the deposit currency.

First, we need to calculate the point price from the lot of 0.1 and then divide the lot by this value. For example, the point price with the lot 0.1 USDCAD= 0.77. Thus, the previously obtained lot (from the setting or obtained after calculating the dynamic lot) should be divided into the point price. If the lot is 0.5/0.77=0.649, it is rounded to the nearest correct one, for example 0.65. The parameter works both with static and dynamic lots.

1. **"calculate the lot size based on the block size ratio to the daily volatility"**

After the signal for opening the first position of the first series arrives, we already know the size of the block of the basic timeframe BSB this position is opened on. So now we need to define the position lot based on the ratio of the block size to the daily volatility. In the future, all positions logically related to this series will be opened with this lot (including subseries positions). Daily volatility is defined by ATR launched on D1 chart of the instrument. The averaging period is defined in "**ATR** period". BSB in price format, for example 0.00125, ATR in the same format, for example 0.00450. The result is ***ATR/BSB=Knorm***. "**Normal ratio of the block size to volatility**" (KnS) defines the ratio, at which lots are not adjusted. If Кnorm exceeds the value, the lots are proportionally reduced, if less, then they are proportionally increased. KnS/Knorm= Kc – adjustment ratio. Now, lot/Kc= lot for opening. Here the lot is used by one of the methods, depending on which receiving option is selected in the previous settings. While a stable lot from the settings is simply taken for the calculation, a dynamic lot is used with all the settings enabled. So, the parameter can be applied to a previously calculated lot.

Regardless of the used method, the lot is defined once for a logically connected series before opening the first position. Further on, all logically connected series are traded using a stable lot. As soon as no logically connected series remain, the lot is redefined before the start of a new series and opening a position.

1. **Block size auto calculation**

The indicator sets TF1 block size and all the remaining timeframes, including the initial and basic ones, are calculated from it. TF1 block size may be used manually in "**TF1 (BS)** block size". It can also be calculated automatically. There are several options for block size auto calculation

1) "calculate block size from ATR" ATR indicator is launched on the same timeframe the EA is set on. It has "ATR period" and "ATR multiplication ratio" (Katr) defined. Further on, the block size is defined as ATR\*Katr. The parameter cancels the ability to set a block size manually. The block size is defined once per the logically related series before its start. After a logically connected series is complete, the block size is redefined before starting a new series. Just like the lot, the logically connected series are the first series, the second one belonging to it, etc. The block size changes only after all series are closed.

2) "**calculate block size from spread**" is able to work with the previous setting. In this case, the block size is not less than the one obtained in the previous calculation. A spread multiplied by the spread multiplication ratio is used to calculate the block size. If this method is selected, the robot gathers statistics on the average spread for this instrument for the selected period. In other words, it should take the sum of all spread data for the period and divide it by the number of measurements. The measurement frequency is set in "**spread measurement frequency**". It is set in seconds from 0.1 to 3600. This is how we obtain the average spread for the period specified in "**spread averaging period**". The averaging period is set in minutes from 10 to 1440. It turns out something like this: the average spread is 0.00010 with the sample frequency of 1 second per 120 minutes. In this example, we get the average of 7200 spread samples. The obtained value should be multiplied by "**spread multiplication ratio**". This is how we get the TF1 block size. After launching the EA, it features no spread statistics right away. Therefore, the "**normal spread**" is initially set for the instrument. That spread is used for multiplying by the ratio. After the statistics is gathered for the specified period, the normal spread is no longer used.

1. **General functions and settings**

**- Trading start date and time**

The parameter specifies the data the EA starts trading from. If in the tester, the EA does not trade before that date and time. In a real or demo account, the function is used to set the EA and define the date and time, after which it starts trading.

**- number of simultaneously traded instruments: 1-28**

Here we specify the number of instruments the series can be opened on simultaneously. If 2 instruments are set and the series is launched on one instrument, only a series for one more instrument can be started. If 2, the EA cannot open positions on more than 2 instruments simultaneously. Only instruments belonging to the EA are considered here. Positions opened by other robots or manually are ignored.

**- disable unidirectional transactions by currency**

After a signal to open a new series of positions appears, check if there are open positions belonging this EA instance. If not, positions are simply opened. If open positions are already present, define the instruments and direction. After that, divide the instrument into 2 currencies (for example, AUDUSD consists of AUD and USD) and define the position direction on separate currencies. When buying a currency pair, the first currency is bought and the second is sold (the opposite is true for selling a pair). Thus, Buy AUDUSD positions are divided into 2 components:

Or

This is how we get separate currencies and position directions for them. This data is saved (till the series is open). Now when a series start signal (on another currency pair) arrives, we need to divide the currency pair, at which a position is to be opened, into separate currencies and disable the start of the new series if there is a match of the position type and currency in the new series with the ones previously opened in the series.

**Suppose that we already have active Sell positions on AUDUSD, thus we have Sell AUD and Buy USD. A signal to open a new Sell series on AUDCHF arrives.**



In other words, Sell on AUD and Buy on CHF are to be opened, but Sell on AUD already exists in the previous series, which means opening such a series should be disabled before the next series opening signal. The check is repeated on the next signal. In case of Buy AUDCHF series signal, such a series can be opened, since Buy AUDCHF = (Buy AUD and Sell CHF). No such combinations have been found yet.

This is how the list of currencies with position directions assigned to them is made. When a new series signal arrives, the check is performed. If there are matches, the series is not opened.

A new series with the following instruments and directions cannot be opened for the open Buy AUDUSD series:

Buy – EURUSD; GBPUSD; NZDUSD; AUDNZD; AUDCAD; AUDCHF; AUDJPY

Sell – USDCAD; USDCHF; USDJPY; EURAUD; GBPAUD.

The series in the following combination can be started:

Sell – EURUSD; GBPUSD; NZDUSD; AUDNZD; AUDCAD; AUDCHF; AUDJPY

Buy – USDCAD; USDCHF; USDJPY; EURAUD; GBPAUD.

Or on any other currency pairs, in any direction (Buy/Sell).

The function should work correctly with symbols having non-standard names, like GBPUSD-pro, mGBPUSD, GBPUSDm – this is a single instrument and, in case of a Sell deal, GBP is sold and USD is bought. This can be implemented in different ways. If something should be added for the implementation, it is fine. Most importantly, the EA should understand this is one instrument.

**- close no less than a specified number of instruments before opening new ones**

Positions can be opened on multiple instruments. The setting limiting this number of instruments is described above. If the number of traded instrument reaches its maximum defined in the "**number of simultaneously traded instruments**", the series cannot be started on new instruments. Here we can set how many instruments should be closed in order to open series for new instruments. Example: the maximum number of traded instruments is 4 and the EA has already opened series on 4 instruments. "**Close no less than a specified number of instruments before opening new ones"**=2. Series on GBPUSD, EURUSD, CADJPY and NZDCHF have been opened. New series cannot start. Series on at least 2 instruments should complete before we are able to launch the new series. GBPUSD is complete. No new series can be opened as 2 instruments should be completed. After CADJPY is complete, we are able to start the new series.

Here we are talking about logically connected series. If a series is opened on GBPUSD and the (second) subseries should be opened, this parameter does not affect the main algorithm inside a single instrument. In fact, this parameter is a tool for limiting the number of instruments traded simultaneously.

**- total profit to close all positions/ total loss to close all positions**

The parameter defines amount of profit/loss in the deposit currency on all positions belonging to the EA. If the current profit on open positions belonging to the EA exceeds a specified threshold, all positions are closed, the series are reset and the work starts anew. The work with the current loss is handled similarly.

**- maximum number of positions**

The parameter defines the maximum number of simultaneously opened positions belonging to the EA. If the number of positions reaches the maximum value, no new positions are opened. Here there is an additional parameter "**close all deals in case of the max number of positions**". If "yes", all open positions belonging to the EA are closed, the series are reset and the work starts anew.

- **complete trading**

If "yes", no series are opened on new instruments. All the algorithms perform their work till the logical conclusion condition. After completing the series on the instrument, the new one is not opened. This allows the algorithm to complete its work normally and avoid starting it anew. The parameter is activated during work, therefore its activation should not affect the algorithm logic. This is the case when changing the settings should not lead to errors.

- **minimum equity**

The parameter defines the minimum equity, at which all positions should be closed and algorithm execution interrupted. No new positions should be opened in case of completion by the minimum equity. If the equity has grown and exceeded the threshold value when closing positions, positions are closed anyway. No new positions are opened before the EA restarts. "**Deposit percentage**" features an additional parameter. 0 – not used. If the parameter is set to a value from 1 to 100, the position closure directory is calculated as the deposit\*percentage/100= closure threshold. The "minimum equity" remains a top priority setting. The equity cannot fall below it even if the deposit percentage calculation yields a lower value.

1. **Displaying info on the screen and in the logs**

1) Basic timeframe blocks, TF1 blocks, as well as the blocks of the closest timeframes analyzed by the EA are to be displayed on the chart. They all have different colors. This should be displayed on the chart of the instrument the EA is launched on and on open charts of other instruments specified in trading settings. "**Build blocks in the chart window**" is enabled/disabled in the indicator settings

2) Data on each instrument is to be displayed in the chart window. It can be arranged as a drop-down list. If a user wants to view the data on the instrument, he/she should simply click on it.

The necessary data:

- number of basic timeframe blocks (where an excess was found)

- opening percentage

- current number of basic timeframe blocks

- block date and time with a fixed time

- current percentage of prevailing blocks

- type of prevailing blocks

- basic TF index

- number of open positions

- closing percentage

- close price

- current profit on open positions (the minus sign is for a loss)

- number of opened additional series

- number of additional closed series

- profit obtained by additional series (defined separately for the second, third and all subsequent series)

3) Data in the logs. We need to write excess detection points, blocks and percentage, as well as mark transition to each new level of the basic timeframe. Also, write position opening reasons and a series it belongs to. When positions are closed, we need to write closure reasons, the appropriate subseries and a profit it obtained. After the main series is closed, inform of that and display a profit obtained by it. When opening and closing positions, show a spread and a time required to open/close a position.

Extended log. The settings provide the ability to enable an extended log. It is needed at the development stage. It is to feature all intermediate calculations and all invisible actions performed by the EA. This is necessary to check equations and intermediate calculations.

Use Latin letters in logs, parameters and data displayed on the screen.

Each instrument should feature the "reset series" button. It is used in case positions are closed manually. After clicking the button, the EA understands that the positions are closed on purpose, so all the data should be reset and trading on the instrument should start anew. The button should be protected in some way to avoid accidental clicks.

1. **Extra**

Source codes are to be provided for everything. All settings should contain Latin letters as well. The check is considered complete after a week of uninterrupted error-free work on a demo account with periodic server reboots. Everything should work without glitches.

The EA should not "forget" to close its positions. If there was a signal to close a position, it should be closed. There should be no "orphaned" positions. If a requote occurs when opening a position or it is not opened due to some other reason, each instrument has the "Slip" parameter. This is a maximum deviation of the price from the one the position should have been actually opened at. If a position has not opened but less than Slip has passed before forming the next block from the perfect price, an attempt to open a position should be repeated. Attempts are repeated till the position is opened or till the price moves beyond Slip. The parameter is set in points. If quotes have 5 digits, 30 points= 0.00030. The same goes for any other number of digits. The EA should not open multiple positions in a single point due to an error. It should control opening positions. For example, there may be the following error: an order has been sent and the answer has been received informing that the order has been rejected, but a position has been opened. Another request is sent, the same answer is received but the position is opened again. This should be avoided. If connection with the server is lost when closing positions, then these positions are to be checked and closed as soon as the connection is restored.

If a position is closed accidentally (manually or otherwise), the EA should restore it if the price approaches the level the position was originally opened at within the Slip distance. In other words, the EA should remember position open levels and what logical parts these positions belong to, so that if any part lacks a position, it is restored if possible.

The code should contain a date, till which the compiled version works so that it can be changed. After this date, the EA should stop working and display the expiration notice.

The specification (including the equations and the interaction of different parameters) may contain logical inaccuracies and errors. If you find any, fell free to discuss them with me and fix them during the current work. The main objective is to develop an algorithm which works as intended even if the requirements specification has a contradiction to the commissioned logic.

The operation logic will be checked in the tester, while the operation itself will be assessed both in the tester and on a demo account. The algorithm will be used on the demo account within a week. If no errors are detected within a week, then all is good. During the tests, the terminal/PC will reboot. This should not cause malfunctions.

The task has been developed and compiled by Maxim Romanov

e-mail: [223231@rambler.ru](mailto:223231@rambler.ru)

MQL5 profile: <https://www.mql5.com/en/users/223231>

LinkedIn profile: [www.linkedin.com/in/максим-романов-05475610b](http://www.linkedin.com/in/максим-романов-05475610b)