



Strategy Testing

The terminal allows not only write expert advisors, but also [test](#) them before using. This useful function allows to check operativeness and efficiency of the trading system on history data. Testing allows to start automated trading with the full knowledge about expert conduct under different market conditions. The special "[Tester](#)" window was built into the terminal for this purpose. Using this window, one can [optimize expert inputs](#), as well.

Strategy Tester — Setup



Setup of Testing Parameters

Before starting to test expert advisors, one has to perform the setup. This means that one has to:

- [select an expert and set up its inputs](#)
- [select a symbol and its period](#)
- [select one of three modeling methods](#)
- [set the testing time range \(optional\)](#)

To test and optimize experts, a special "[Tester](#)" window is used in the terminal. All above parameters are set in the "Settings" tab of this window.

An Expert Advisor and Its Parameters

One has to select an expert to be tested in the "Tester — Expert Advisor" field. One cannot select any expert file in this field. Only those experts that are available in the client terminal can be selected here. To be selected, they must be compiled and located in the /EXPERTS folder.

After an expert has been selected, one has to perform additional setting of testing parameters and inputs. This can be done by pressing of the "Expert properties" button. A new window with three tabs will appear then:

- **Testing** — general testing parameters are set in this tab. These are volume and currency of the initial deposit to be given in the corresponding fields. It is this deposit that will be operated by the expert during testing. Types of positions to be opened at testing are defined here, as well: Only Long — open only long positions; Only Short — open only short ones; Long and Short — open both long and short positions. Whatever expert algorithm is, it will open positions only in the defined directions. One can include an optimization genetic algorithm and select a parameter to be optimized (maximization by the balance value, the profit factor, expected payoff, or minimization by the maximal drawdown value or drawdown percent).
- **Inputs** — the list of all inputs is given here as a table. Inputs are variables that influence the expert operation and can be changed directly from the client terminal. There is no need to change the expert code in order to change these parameters. The amount of input variables can change depending on the expert. At testing, the expert inputs are defined in the "Value" field. Data written in the fields of "Start", "Step", and "Stop" do not influence expert testing and are necessary just for [optimization of its parameters](#). How to work with these parameters is described in the "[Optimization Setup](#)" section.
- **Optimization** — settings in this tab allow to limit testing passes at optimization. Changing of parameters in this tab does not influence single testing passes of the expert.

Symbol and Its Period

To start testing, it is not enough just to select an expert and set it up. One has to select a symbol and a period (timeframe) for testing. These are data that will be used for testing. At testing, one can select an available in terminal symbol or use an external data file. History data files of *.FXT format stored in the /TESTER directory are used in testing. These file are created automatically at testing if an available in the terminal symbol was selected. If external data are used, one has to save the corresponding file in the /TESTER directory manually and disable "Recalculate" in order not to overwrite this testing succession.

The symbol is defined in the field of the same name, and timeframe is in the "Period" field. If no data file for this symbol, period and modeling method does not exist yet, it will be created automatically. If the necessary file has already been created and the "Recalculate" option is enabled, the data file will be generated again. If there are no history data for the symbol or period, the tester will download the last 512 history bars automatically.

Attention: If there are some data outside the latest 512 bars for the symbol, the history data will be downloaded automatically, up to the last available one. This can cause sharp increase of the incoming traffic.

Methods of Modeling

Historical data are saved in the terminal only as bars and represent records appearing as [TOHLCV \(HST format\)](#). These data can be used for modeling of price changes at testing of experts. In some cases, such information is not enough for testing. For example, for the daily timeframe, price changes within a bar can result in triggering of the expert. At the same time, no triggering can occur at testing. In other words, testing of an expert based on only bars can be inaccurate and give a false idea about the expert efficiency.

The trading terminal allows to test experts by various methods of historical data modeling. Using historical data from smaller periods, it is possible to see price fluctuations within bars, i.e., price changes will be emulated more precisely. For example, when an expert is tested on one-hour data, price changes for a bar can be modeled on one-minute data. Thus, modeling brings historical data near the real price fluctuations and makes expert testing more authentic.

One of three methods of historical data modeling can be chosen for testing:

- **Open prices only (fastest method to analyze the bar just completed)**

Some automated trading systems do not depend on properties of modeling within a bar, they trade on completed bars. The bar is completed if the next one has appeared. These are such experts for that this modeling method was developed.

In this mode, the bar opening is modeled first (Open = High = Low = Close, Volume=1) what allows the expert to identify the completion of the preceding bar precisely. It is this incipient bar that is used to start testing of the expert. At the next step, the fully completed current bar will be given, but no testing is performed on it!

- **Control points (the nearest less timeframe is used)**

The control points modeling method is intended for a crude estimate of experts efficiency that trade within the bar. The historical data of the nearest less timeframe must be available to apply this method. In some cases, the available data of the less timeframe do not completely cover the time range of the timeframe under test. If the data of the less timeframe are missing, the bar evolution is generated on the basis of predefined wave templates as it was in the preceding, third version of MetaTrader 3 Client Terminal.

As soon as historical data of the less timeframe appear, these new data will be interpolated. However, the really existing OHLC prices appear as control points. In the most cases, the results of testing experts by method of control points can also be considered as estimated ones, not as final ones. Such results are of intermediate, estimated nature.

- **Every tick (based on all available least timeframes)** This is the most accurate method of modeling prices within a bar. Unlike that of "control points", this method uses for generation not only data of the nearest less timeframe, but also those of all available nearest timeframes. At that, if there are data of more than one period for the same timeframe at the same time, the data of the least timeframe will be used for modeling. Like in the preceding method, control points are generated on the basis of OHLC data of the least available timeframe. To generate price movements between control points, interpolation based on predefined templates is also used, so one-minute data are highly desirable to be available that would cover the entire testing range. It is possible that several similar ticks are modeled one after another. In this case, the doubled quotes will be filtered out, and the volume of the last of them will be fixed.

One has to consider the possible large amount of tick data modeled. This can influence the consumed resources of the operation system and testing speed.

Attention:

- it is not recommended to launch testing on every tick if there are no available less timeframes that completely cover the period under test, otherwise, the results will not be accurate;
- modeling on control points is basically used at optimization of experts, and all ticks modeling is for a close testing.

The modeling quality can be checked in the "Report" window. The "Modeling quality" field and a colored band are intended for this. The band is a scheme of the modeling process. It can be of three colors:

1. **Gray** — this part of available data did not participate in testing. Gray color can appear if the date range was specified for testing (described below);
2. **Red** — modeling was not performed in this space because of missing data of a less timeframe. At that, only data of the timeframe selected for testing were used;
3. **Green** — modeling was performed in this space. And the brighter is the color, the higher the modeling quality was. For example, at testing on H1 period, the deep-green band can mean that data of M30 period were used for testing, and the brightest one does that M1-period data are used.

After modeling parameters and date range (described below) have been changed, the data file must be created afresh. To do so, one has to flag the "Recalculate". If the above setting were not changed, there is no need to recalculate. In this case, it is recommended to disable the above option in order to reduce the testing time.

Time Range

The range of dates allows to test experts not on all available data, but within a certain time space only. This can be useful if there is a need to test a certain part of history data. Date range can be used not only for expert testing, but also for modeling of the testing succession of bars (file of data modeled to be used for testing). It is often no need to model data of the entire history, especially for every-tick modeling where the amount of unused data can be very large. That is why, if data range was allowed to be set at the initial modeling of testing succession, bars that are beyond this range will not be modeled, but just transcribed into the output succession. The data will not be excluded from the succession in order the correct calculation of indicators on the entire received history to be possible. It must be noted that the first 100 bars will not be modeled either. This limitation does not depend on the date range defined.

To enable date range limitation, one has to flag "Use date" and specify the necessary values in the fields of "From" and "To". After all settings have been made, one can press the "Start" button and start testing. After testing has started, the approximate time of completing of this process can be viewed in the lower part of the window.

Testing Visualization

If the "Visualization" is flagged, after the "Start" button has been clicked, the chart will be opened automatically, on which the modeled tick sequence will be played. The playback speed can be regulated. The playback can be suspended by clicking the "||" button. The repeated click on this button resumes the income of the modeled ticks. Pressing of F12 causes immediate appearance of the next tick even in the suspension mode. Visualization can be skipped up to a certain date. After the desired date is set and the "Skip to" button is clicked, visualizing will stop and then resumed after the tester reaches the defined date.

Attention: If "Optimization" is enabled, [expert parameters will be optimized](#), not tested, after the "Start" button has been pressed.

Strategy Tester — Results

Testing Results

After the testing has been completed, one can see the results thereof in the tabs of "Results", "Graph", "Report", and "Journal".

Results

Information about all trade operations performed is given in this tab as a table:

- **#** — the trade operation sequence number;
- **Time** — time at which the operation was performed;
- **Type** — type of the operation (sell, buy, s/l, t/p, modify, close at stop, etc.);
- **Order** — ticket number of trade position or pending order (not to be mixed up with the trade operation sequence number described above);
- **Lots** — amount of lots traded;
- **Price** — symbol price during operation;
- **S/L** — the Stop Loss order value. No entries in this field mean that the order was not placed;
- **T/P** — the Take Profit order value. No entries in this field mean that the order was not placed;
- **Profit** — profit/loss. The profit/loss value is entered only at closing of positions;
- **Balance** — balance value. The balance value is recorded only at closing of positions.

Having clicked with the left mouse button on any column header, one can sort out all entries in the table in increasing or decreasing order. Using the "Copy" context menu command or accelerating keys of Ctrl+C, one can copy the selected lines of results to the clipboard for further use in other applications. If no line has been selected, the entire table will be copied to the clipboard. As well, to copy the entire table to the clipboard, one can execute the "Copy All" command. The report of results can be saved in the hard disk as an HTML file. To do so, one has to execute the "Save as Report" context menu command. The commands of "Set Date "From"" and "Set Date "To"" allow to set the time span for testing. At that, the selected operations dates will be written in the fields of "Use date from:" and "Use date to:" in the [testing settings](#). This can be useful if there is a need to test an expert closely or to optimize its parameters within this range of dates.

Graph

The graph representing the account balance (the blue line is "Balance") and general account status considering open positions (the green line is "Lots") is drawn in the "Graph" tab automatically. If only balance line is displayed in the graph, it means that the lines of "Balance" and "Lots" coincide during the entire testing period. Graph allows changes in trading results during testing even more visualized. If lot sizes were changed during testing, the chart of lot size changes will be shown in the bottom of the graph.

A double click with the left mouse button on any point in the graph will switch to the ["Results"](#) button, the corresponding line being selected. Using the "Copy" context menu command or accelerating keys of Ctrl+C, one can copy the graph to the clipboard to be used in other applications. A graph can also be saved in the hard disk as a GIF file. To do so, one has to execute the "Save as Picture" context menu command or press the accelerating keys of Ctrl+S. The commands of "Set Date "From"" and "Set Date "To"" allow to set the time span for testing. At that, the selected operations dates will be written in the fields of "Use date from:" and "Use date to:" in the [testing settings](#). This can be useful if there is a need to test an expert closely or to optimize its parameters within this range of dates.

Report

Generalized results of the expert testing and some key data are represented in the "Report" tab. Such reports allow to quickly compare various experts to each other. The following data are published in reports:

- **Bars in test** — the amount of history data modeled in bars;
- **Ticks modeled** — the amount of ticks modeled;
- **Modeling quality** — the quality of ticks modeled during testing, in per cents. Modeling is shown elementarily as a band in the next line of the report. The band can be of one of three colors:
 1. **Gray** — this part of available data was not involved into testing. Gray color can appear if there was a date range specified in the [testing settings](#);
 2. **Red** — modeling was not performed in this space because of missing available data of a less timeframe. At that, only data of the timeframe selected in the [testing settings](#) were used;
 3. **Green** — modeling was performed in this space. And the brighter is the color, the higher modeling quality was. For example, at testing on the H1 period, the deep-green band means that data of M30 period were used for testing, and the brightest one does that M1 data were used;

Attention: If the fastest method ("Open prices only") was selected in the [testing settings](#), the entire band will be red. At that, "n/a" (no modeling was performed) will be written in the "Modeling quality" field;

- **Initial deposit** — the amount of initial deposit;
- **Total net profit** — financial result of all trades. This parameter represents the difference between "Gross profit" and "Gross loss";
- **Gross profit** — the sum of all profitable trades in monetary units;
- **Gross loss** — the sum of all unprofitable trades in monetary units;
- **Profit factor** — the ratio between the gross profit and the gross loss in per cents. One means that these sums are equal;
- **Expected payoff** — mathematical expectation of win. This parameter to be calculated statistically represents average profit/loss factor of one trade. It can also be considered as showing the expected profitability/unprofitability of the next trade;
- **Absolute drawdown** — the largest loss is below initial deposit value;
- **Maximal drawdown** — the largest loss of the local maximum in the deposit currency and in per cents of the deposit;
- **Total trades** — total amount of trade positions;

- **Short positions (won %)** — amount of short positions and the profit percentage thereof;
- **Long positions (won %)** — amount of long positions and the profit percentage thereof;
- **Profit trades (% of total)** — the amount of profitable trade positions and their part in the total trades, in per cents;
- **Loss trades (% of total)** — the amount of unprofitable trade positions and their part in the total trades, in per cents;
- **Largest profit trade** — the largest profit among all profitable positions;
- **Largest loss trade** — the largest loss among all unprofitable positions;
- **Average profit trade** — average profit value within a trade (sum of profits divided into the amount of profitable trades);
- **Average loss trade** — average loss value within a trade (sum of losses divided into the amount of unprofitable trades);
- **Maximum consecutive wins (profit in money)** — the longest series of profitable trade positions and sum of their wins;
- **Maximum consecutive losses (loss in money)** — the longest series of unprofitable trade positions and sum of their losses;
- **Maximal consecutive profit (count of wins)** — the maximal profit within one series of profitable trades and the corresponding amount of profitable trades;
- **Maximal consecutive loss (count of losses)** — the maximal loss within one series of unprofitable trades and the corresponding amount of unprofitable trades;
- **Average consecutive wins** — average amount of profitable positions in consecutive profitable series;
- **Average consecutive loss** — average amount of unprofitable positions in consecutive unprofitable series.

The report can be copied to the clipboard or saved in the hard disk as an HTML file. To do so, one has to execute the context menu commands of "Copy" and "Copy as Report", respectively. The report can also be copied to the clipboard by pressing of accelerating keys of Ctrl+C.

"Journal" Tab

Information about testing process with all trade operations is automatically published in the "Journal" tab. This journal is the same as [that of the "Terminal — Experts" window](#), except for that messages published in the tester window inform about testing of the expert, but not about its operation at the market. After testing has been completed, these data will be output in a separate directory named /TESTER/LOGS. The testing journal files are stored in the /EXPERTS/LOGS directory, file names corresponding with the data of journal — YYYYMMDD.LOG. To view the journals stored, one has to execute the "Open" context menu command and select the desired file. To copy the message to the clipboard, one has to execute the corresponding command of the context menu. The "Clear All Journals" command allows to remove all journals from the directory and clears the tab. This can be useful to clear the hard disk of too many files.

Strategy Tester — History Files in FXT Format

History Files in FXT Format

In its operation, tester uses an *.FXT file with generated succession of bars. Each record of the generated succession represents the bar status at either moment within one bar. When modeling bars, tester takes other bars from this file and updates the current bar or adds another one if it has just begun to be formed.

One may refuse to apply the standard bar modeling, but use one's own data file for testing/optimization. To do so, one has to disable "Recalculate" and place the desired FXT file into the /TESTER/HISTORY folder. The file name must appear as "[symbol name][period in minutes]_[type of modeling (0 — every tick, 1 — control points, 2 — open prices)].FXT" (without spaces). For example, it can be "EURUSD1440_1.FXT" where "EURUSD" is the security symbol, "1440" is period of D1 (1440 minutes, 24 hours), and "1" means modeling by control points.

A short description of the format is given below. It begins with the header:

```
//+-----+
//|                                     |
//+-----+
struct TestHistoryHeader
{
    int         version;           // 404
    char        copyright[64];     // copyright
    char        symbol[12];
    int         period;
    int         model;             // for what modeling type was the ticks sequence generated
    int         bars;              // amount of bars in history
    time_t      fromdate;          // ticks generated from this date
    time_t      todate;           // ticks generating stopped at this date
    double      modelquality;      // modeling quality
    //---- general parameters
    char        currency[12];      // currency base
    int         spread;
    int         digits;
    double      point;
    int         lot_min;           // minimum lot size
    int         lot_max;           // maximum lot size
    int         lot_step;
    int         stops_level;       // stops level value
    int         gtc_pendings;      // instruction to close pending orders at the end of day
    //---- profit calculation parameters
    double      contract_size;     // contract size
    double      tick_value;        // value of one tick
    double      tick_size;         // size of one tick
    int         profit_mode;       // profit calculation mode { PROFIT_CALC_FOREX, PROFIT_CALC_CFD, PROFIT_CALC_FUTURES }
    //---- swap calculation
    int         swap_enable;       // enable swap
    int         swap_type;         // type of swap { SWAP_BY_POINTS, SWAP_BY_DOLLARS, SWAP_BY_INTEREST }
    double      swap_long;
    double      swap_short;        // swap overnight value
    int         swap_rollover3days; // three-days swap rollover
    //---- margin calculation
    int         leverage;          // leverage
    int         free_margin_mode;   // free margin calculation mode { MARGIN_DONT_USE, MARGIN_USE_ALL, MARGIN_USE_PROFIT, MARGIN_USE_LOSS }
    int         margin_mode;       // margin calculation mode { MARGIN_CALC_FOREX, MARGIN_CALC_CFD, MARGIN_CALC_FUTURES, MARGIN_CALC_CFDINDEX };
    int         margin_stopout;    // margin stopout level
    int         margin_stopout_mode; // stop out check mode { MARGIN_TYPE_PERCENT, MARGIN_TYPE_CURRENCY }
    double      margin_initial;    // margin requirements
    double      margin_maintenance; // margin maintenance requirements
    double      margin_hedged;     // margin requirements for hedged positions
    double      margin_divider;    // margin divider
    char        margin_currency[12]; // margin currency
    //---- commission calculation
    double      comm_base;         // basic commission
    int         comm_type;         // basic commission type { COMM_TYPE_MONEY, COMM_TYPE_PIPS, COMM_TYPE_PERCENT }
    int         comm_lots;        // commission per lot or per deal { COMMISSION_PER_LOT, COMMISSION_PER_DEAL }
    //---- for internal use
    int         from_bar;          // fromdate bar number
    int         to_bar;           // todate bar number
    int         start_period[6];  // number of bar at which the smaller period modeling started
    int         set_from;         // begin date from tester settings
    int         set_to;           // end date from tester settings
};
```

```
//----
int      freeze_level;      // order's freeze level in points
//----
int      reserved[61];
};
```

Then, the array of modeled bars follows:

```
#pragma pack(push,1)
struct TestHistory
{
    time_t      otm;          // bar time
    double      open;        // OHLCV values
    double      low;
    double      high;
    double      close;
    double      volume;
    time_t      ctm;        // the current time within a bar
    int         flag;        // flag to launch an expert (0 - bar will be modified, but the expert will not be launched)
};
#pragma pack(pop)
```