

Getting Started

- Terminal Start
- Opening of Accounts
- Authorization
- Security System
- Live Update

Client Terminal Settings

- Server
- Charts
- Objects
- Trade
- Expert Advisors
- Email
- Publisher
- Events

User Interface

- Main Menu
- Toolbars
- Market Watch
- Data Window
- Navigator
- Terminal
- Tester
- Fast Navigation

Working with charts

- Chart Opening
- Chart Setup
- Chart Management
- Charts Print
- Templates, Profiles

Analytics

- Graphical Objects
- Technical Indicators

Trading

- Order Types
- Trailing Stop
- Types of Execution
- Trade Positions

Auto Trading

- *Expert Advisors*
- *Strategy Testing*
- *Optimization of Experts*
- *Custom Indicators*
- *Scripts*

Tools

- History Center
- Export of Quotes
- Global Variables
- Contract Specification
- Languages Support

Auto Trading

To work at financial markets more effectively, one can develop one's own successful system of trading. It is very difficult to act within a chosen system of trading in the manual mode due to significant influence of normal human emotions. Mechanical trading systems do not suffer from this disadvantage.

Client Terminal gives a large range of means for development and use of mechanical trading systems (MTS, experts, advisors). The development environment allows to create, debug, and test expert advisors. Experts are able not only alert about recommendation trading signals, but undertake the complete control over trading activities online.

MetaQuotes Language 4, MetaEditor and strategy testing tools are built in the terminal. One can create the following using these means:

- **Expert Advisors** — mechanical trading systems that allow complete automation of analytical and trading activities;
- **Custom Indicators** — independently written technical indicators intended for analyzing of price changes;
- **Scripts** — programs to be executed only once, on request.

MetaQuotes Language 4

MetaQuotes Language 4 (MQL 4) is the language for programming of trade strategies built in the client terminal. It allows to write custom expert advisors that automate trade processes and ideally suit for implementation of traders' own strategies. Moreover, traders' own custom indicators, scripts and DLL's can be created in MQL 4.

Syntax of MQL 4 is quite similar to that of C language. A large amount of functions necessary to analyze quotes, manage positions, call technical indicators, and others, are included in MQL 4. The MetaEditor, an editor for expert advisors, is used for writing the source code of programs. The MetaQuotes Language Dictionary that contains descriptions of all language constructions and functions is built in the editor.

Programs written in MQL 4 are different in tasks and properties:

- **Expert Advisor** is a mechanical trading system (MTS) to be run when a new tick incomes. It can work not only in the mode alerts, but also perform trade operations independently. Terminal allows to test trading strategies on history data in order to detect the expert features under different market conditions;
- **Custom Indicator** is a technical indicator written individually. Custom indicators are used only for analyzing of price changes;
- **Script** is a program that is intended for a single execution of some actions. Unlike experts, scripts are launched not tick by tick, but on a command.

MetaEditor

MetaEditor is a constituent of the client terminal. It is intended for creation, editing, and compilation of program source codes written in MetaQuotes Language 4 (MQL4). It can be used in developing of experts, custom indicators, scripts and DLL's.

Expert Advisors Wizard is built in the editor to assist in fast creation of new MQL4 programs. The new file will be saved in the corresponding folder of the client terminal automatically. Templates can be used in creation of an MQL4 program, as well. After that, one can start writing of the program text. The supplemental information can be found in MQL4 Dictionary. The full description of the language with examples of how to use it is given there. After the program has been written, one has to compile it in MetaEditor. As a result of successful compiling, the executable program code will be created that can be launched or tested in the terminal.

A detailed description of MetaEditor can be found in its Help Files.


Expert Advisors

Expert Advisors (experts) are programs in the terminal that have been developed in MetaQuotes Language 4 (MQL 4) and used for automation of analytical and trading processes. They allow to perform prompt technical analysis of price data and manage trading activities on basis of signals received. The entire routine work of technical analysis and trading can be given to experts. An expert can perform analytical and trading operations for any symbols or periods independent on whether the corresponding chart was opened or not.

Working with experts means:


- **Creation of an Expert**
To create and compile an expert, one has to use the built-in "MetaEditor". It is a constituent of the client terminal and represents a convenient development environment of MQL4 programs.
- **Expert Setup**
Before using of experts, one has to set up them first. Working parameters common for all experts are set in the client terminal settings. Besides, each expert can have its own settings.
- **Launch of an Expert**
To launch an expert, one has to impose it into the chart. As soon as a new tick incomes, the expert will start executing.
- **Expert Shutdown**
An expert advisor is shut down after it has been removed from the chart.

Creation of an Expert

To create experts, one has to use MetaQuotes Language 4 (MQL 4) and MetaEditor. To launch the expert editing program, one has to execute the "Create" command of the "Navigator — Expert Advisors" window context menu, or the "Tools — MetaQuotes Language Editor" menu command, or press F4 or the  button of the "Standard" toolbar. After that, the Expert Creation Wizard will be opened automatically that allows to immediately start working at a new MQL4 program. One has to flag the "Expert Advisor" as a type of object to be created and fill out the following fields:

- **Name** — expert name;
- **Developer** — the developer's name;
- **Link** — link to the developer's site;
- **Inputs** — the list of expert inputs. To add a new parameter, one has to press the "Add" button or the "Delete" button to delete a parameter.

After that the new expert window with the defined inputs will open in editor. The source file (*.MQ4) of the expert will be saved in the /EXPERTS folder of the client terminal automatically. At this moment, one can start to write the expert code.

After the expert development has been completed, it must be compiled. To do so, one has to execute the "File — Compile" command in the expert editor, or press F9 or the  button of the toolbar. As a result of successful compilation an executable program file with *.EX4 extension will be created and saved in the /EXPERTS folder automatically. The list of compiled experts can be viewed in the "Navigator — Expert Advisors" window in the client terminal. If the expert has not been compiled successfully, its icon will be gray. This means that this expert cannot be used.

Editing of Experts

To start editing of the existing expert from the terminal, one has to execute the "Modify" command of the "Navigator — Expert Advisors" window context menu. At that, the experts editor will open where the source code of the selected expert. After the expert source code has been modified, one has to recompile it and get a new executable EX4 file. Otherwise, terminal will use the previous version of the expert, the non-modified one.

Attention: If the expert has not been compiled successfully, its icon will be gray. This means that this expert cannot be used.

Expert Setup

Before using of experts, one has to set them up first. Working parameters common for all experts are defined in the client terminal settings window. This window can be opened by the "Tools — Options" menu command or by pressing of accelerating keys of Ctrl+O. To set up expert parameters, one has to select the "Expert Advisors" tab. The following settings are available in it:

- **Enable Expert Advisors**

This option allows to enable or disable the use of all experts. If it is disabled, the start() function will stop execute for all experts when a new tick incomes. At that, the smileys at the expert names in the upper right corners of charts will be replaced with daggers. To enable experts, one has to flag this option and press "OK". They will start functioning when a new tick incomes, and daggers in the upper right corners will be replaced with smileys again.

Attention:

- the expert will start function when a new tick incomes, and it often completes its work before the next tick incomes. Disabling of experts does not interrupt working of those that are active already;
 - this option covers only expert advisors and does not allow to disable custom indicators and scripts automatically.
- **Disable experts when the account has been changed**

This option represents a safety tool that disables experts when the account has been changed. It can be useful, for example, when one switches from a demo account to a real one. To enable it, one has to flag it and press "OK".
 - **Disable experts when the profile has been changed**

A large amount of information about the current settings of all charts of the workspace is saved in profiles. Particularly, information about the attached experts is saved in profiles. Experts included into a profile will start working when a new tick incomes. Having enabled this option, one can disable launching of experts when the profile has been changed.
 - **Allow live trading**

Experts can work automatically, without trader's participation. They can both analyze price changes and trade. This option is intended for limiting of trading functions of experts. These limitations can be useful for testing of analytical capacities of an expert in the real-time mode (not to be mixed up with testing of an expert on history data).
 - **Ask manual confirmation**

This option will only be enabled if experts are allowed to trade. If this option is enabled and the expert tries to trade, one will be able to confirm the trade or interrupt it manually. In other

words, trading activities of experts can be managed through this manual confirmation option. This can be useful for testing of trading functions of an expert in the real-time mode (not to be mixed up with testing of an expert on history data)..

- **Allow DLL imports**

Experts can use DLLs to enlarge their functionalities. If this option is enabled, the libraries can be used without any limitations. Disabling of the option results in that no expert can use any external DLLs. It is recommended not to allow DLL imports when working with unknown experts.

- **Confirm DLL function calls**

This option will only be enabled if DLL imports are allowed. It allows to control over execution of each called function manually. If it is disabled, the imports from external libraries will not be controlled. This option is recommended to be enabled when working with unknown experts.

- **Allow external experts imports**

If there is a need to export functions from other experts or MQL4 libraries during the expert functioning, one has to enable this option. If it is disabled, no expert already launched will be able to call functions from other MQL4 programs. This option is recommended to be disabled when working with unknown experts.

Launch of an Expert Advisor

After general parameters have been set up, experts can be launched. To do so, it is enough just to attach the expert to the chart. The "Attach to a chart" command of the "Navigator — Expert Advisors" window context menu or double click with the left mouse button on the selected expert in the same window allows to impose it into the active chart. "Drag'n'Drop" technique allows to impose the expert into any chart. At that, the window of the expert special settings will appear. In its "Common" tab, it is possible:

- **positions** — select the direction of position opening:
 1. Long&Short — in both directions;
 2. Only Long — only for buying;
 3. Only Short — only for selling.
- **Enable alerts** — enable/disable the expert to alert;
- **Disable alert once hit** — disable alerts after the first alert has been given;
- **Allow live trading** — enable/disable live trading;
- **Ask manual confirmation** — request for manual confirmation of position opening when trading;
- **Allow DLL imports** — enable/disable imports of functions from DLL files;
- **Confirm DLL function calls** — request for confirmations at each function call from DLL files;
- **Allow import of external experts** — enable/disable calling of functions from external experts.

External variables of the expert can be changed in the "Inputs" tab. these are variables of extern class. To save an input, one has to double-click with the left mouse button on its value and write the new one. At that, one can change the value of each variable or download the set of inputs already saved (the "Load" button). One can save the current set of external variables using the button of the same name.

The "Reset" button returns all default settings. Parameters defined in the terminal settings are given in the "Common" tab. And parameters defined in the program source code are set as inputs. To attach the expert with the new parameters to a chart, one has to press "OK". To cancel the expert imposing, one has to press the button of the same name.

Attention: Only attached experts can be set up individually. However, while current executing, the window of the expert properties cannot be opened. This can be done only during intervals between calls of the start() function. At that, the expert will not be launched until its properties window is closed. If the expert inputs were changed, the expert will be re-initialized with its new inputs at pressing of "OK".

After an expert has been set up, it will be initialized and, as soon as a new tick incomes, execute. Expert

is attached if its name and a smiley can be seen in the upper right corner of the chart. If live trading is disabled in the expert settings, a L will appear instead of the smiley. A dagger (†) means that all experts are disabled.

Attention: Only one expert can be attached to a chart. If another expert is imposed, the previous one will be deleted from the chart.

Expert Shutdown

To shut down an expert, one has to remove it from the chart. Expert must have been deinitialized before it is shut down. To remove an expert from the chart, one has to execute the chart context menu "Expert Advisors — Delete" command or attach another expert to the same chart. Besides, the expert can be removed from the chart at profile or template change.

Attention:

- at client terminal shutdown, all experts are shut down, too;
- at chart closing, the expert attached to this chart will be shut down;
- at imposing of another expert, the previous one will be removed after confirmation;
- deletion of the expert from the "Navigator" window does not shut down the expert of the same name imposed in the chart;
- disabling of experts in the client terminal settings does not provide complete disabling of experts. This option stops launching of the start() function of each expert, but the init() will continue to execute.

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.

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.
- **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

History 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.

Terminal allows to test experts by various methods of history data modeling. Using history 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 history data near the real price fluctuations and makes expert testing more authentic.

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

- **Open prices only (fastest method to analyze the bar just completed)**
Some mechanical 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 which 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 (based on the nearest less timeframe with fractal interpolation of 12 control points)**

The control points modeling method is intended for a crude estimate of experts efficiency that trade within the bar. The history data of the nearest less timeframe must be available to apply this method. In the most 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 further bar development will be generated on close prices of 12 preceding bars. It means that the changes within the bar are the same as those of price within the last 12 periods. It is fractal interpolation.

As soon as history data of the less timeframe appear, fractal interpolation will be applied to these new data. But there will be used not 12, but just 6 preceding bars. It means that really existing Open, High, Low, Close prices are reproduced, and two generated prices more. Values and locations of these two generated prices depend on that on 6 preceding bars.

- **Every tick (based on all available least timeframes with fractal interpolation of every tick)** 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 by fractal interpolation. It is also used for modeling of price changes between control points. 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.

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

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, only 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;
 0. **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.

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;
    char        copyright[64];          // copyright
    char        symbol[12];
    int         period;
    int         model;                  // for what modeling type was the succession generated
    int         bars;                   // amount of bars in history
    time_t      fromdate;
    time_t      todate;
    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
    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 }
    //----
    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         reserved[64];
};
```

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)
```

Optimization of Experts

Optimization represents successive passes of the same expert advisor with different inputs on the same data. At that, such parameters can be sorted out at which the expert efficiency will be maximal. Terminal possesses some built-in means that allow to automate this process. To optimize an expert, one has to flag the option of the same name in the "Tester" window and press the "Start" button.

Optimization Setup

Optimization represents consecutive passes of the same expert with different inputs on the same data. At that, such parameters can be taken that make the expert efficiency maximal. The terminal possesses in-built means that allow to automate this process. Before beginning to optimize expert parameters, one has to set them up. It means that one has to:

- select an expert and its inputs
- select a symbol and its timeframe
- select one of three bar modeling methods
- set up the time span for optimization (optional)

A special window named "Tester" is used for testing and optimization of experts in the terminal. All settings listed above can be made in the "Settings" tab of this window.

Expert Advisor and Its Parameters

One has to select expert the parameters of which should be optimized in the "Tester — Experts" window. Not any expert file can be selected in this field, but only those that are available in the client terminal. For this, they must be compiled and placed in the /EXPERTS folder.

After the expert has been selected, one has to make an additional setup and set the inputs. This can be done by pressing of the "Expert properties" button. At that, a new window containing the following three tabs will appear:

- **Testing** — general optimization parameters are set in this tab. They include the initial deposit volume and currency to be specified in the corresponding fields. It is this deposit that will be operated by the expert during optimization. Types of positions to be opened are selected in this tab, as well: Only Long, Only Short, or Long and Short. Whatever expert algorithm is used, it will open positions only in the directions defined here;
- **Inputs** — all inputs are listed here as a table. Inputs are variables that influence the expert operation and can be changed directly from the client terminal. For these parameters to be changed, there is no need to change the expert code. The amount of inputs can vary depending of experts. At optimization, the expert inputs are set in the fields of "Start", "Step", and "Stop".

Initial values, change interval, and final values of external variables will be set in these fields, respectively. There are checkboxes to the left of variable names that include the parameter into optimization process. If a variable is not checked in this checkbox, it will not be involved into optimization. Its value will not be changed within the optimization process, and the parameter given in the "Value" field will be written here. The amount of expert passes depends on these parameters directly. Data written in the "Value" field do not influence the expert optimization and are necessary only for its testing.

The set of inputs already saved before (including those given in the fields of "Start", "Step", and "Stop") can be downloaded. This can be done by pressing of the "Load" button and having selected the preliminarily saved set of parameters. The current set of external variables can be saved by pressing of the corresponding button.

- **Optimization** — this tab allows to manage limitations during optimization. If any of conditions is met during a separate pass, this pass of the expert will be interrupted. Limiting parameters are:
 0. **Balance minimum** — the smallest balance value in the deposit currency;
 0. **Profit maximum** — the largest profit in the deposit currency;
 0. **Minimal margin level %** — the lowest margin level in per cents;
 0. **Maximal drawdown %** — the largest drawdown in per cents;
 0. **Consecutive loss** — the largest loss within a series. Loss series is a number of consecutive loss trades;
 0. **Consecutive loss trades** — the largest amount of loss trades within a series;
 0. **Consecutive win** — the largest total profit within a series. Profitable series is a number of consecutive profitable trades;
 0. **Consecutive win trades** — the largest amount of profitable trades within a series.

To enable a limit in condition, one has to flag it in the checkbox to the left of it. Double click with the left mouse button in the "Value" field can be used to change the existing parameter.

Symbol and Its Period

It is not enough just to select an expert and set it up to start optimization: a symbol and its period (timeframe) must be selected for tests. These are the data on which all tests will be made. A symbol available in the terminal or an external data file can be used for tests. History data files in *.FXT format to be stored in the /TESTER directory are used in tests. These files are created at tests automatically if the corresponding symbol available in the terminal was selected. If external data are used, one has to place the corresponding data file into the /TESTER directory manually and disable the "Recalculate" option in order not to overwrite this testing succession.

Symbol is defined in the "Symbol" field, and timeframe is in the "Period". If there is no data file for this symbol, period, and modeling method, it will be created automatically. If the necessary file has already been created and the "Recalculate" option has been enabled, the data file will be generated again at optimization. If there are no history data for the symbol and period, the tester will download 512 latest history bars automatically.

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

Modeling methods

History 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.

Terminal allows to test experts by various methods of history data modeling. Using history 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 history data near the real price fluctuations and makes expert testing more authentic.

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

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

Some mechanical 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 which 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 (based on the nearest less timeframe with fractal interpolation of 12 control points)**

The control points modeling method is intended for a crude estimate of experts efficiency that trade within the bar. The history data of the nearest less timeframe must be available to apply this method. In the most 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 further bar development will be generated on close prices of 12 preceding bars. It means that the changes within the bar are the same as those of price within the last 12 periods. It is fractal interpolation.

As soon as history data of the less timeframe appear, fractal interpolation will be applied to these new data. But there will be used not 12, but just 6 preceding bars. It means that really existing Open, High, Low, Close prices are reproduced, and two generated prices more. Values and locations of these two generated prices depend on that on 6 preceding bars.

- **Every tick (based on all available least timeframes with fractal interpolation of every tick)** 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 by fractal interpolation. It is also used for modeling of price changes between control points. 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.

After modeling parameters 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.

Attention:

- If "Optimization" is disabled, the expert will be tested, not optimized at pressing of the "Start" button;
- at optimization, like at testing, one can use one's own history files.

Optimization Results

After optimization has been completed, its results can be viewed in the tabs of "Optimization Results" and "Optimization Graph".

Unlike testing, optimization is supposed to perform many passes of the mechanical trading system (MTS) with different inputs. This is aimed at determining of such expert parameters at which its efficiency will be the highest. To optimize an expert, one has to flag "Optimization" in the testing settings tab and press "Start". After that, two new tabs will appear in the window: "Optimization Results" and "Optimization Graph".

There are not all operations listed in the tab of "Optimization Results", unlike that of testing results, but only final reports about each pass. The entire information is represented as a table with the following fields:

- **Pass** — the pass number;
- **Profit** — the pure profit (gross profit minus gross loss);
- **Total trades** — the total amount of open trade positions;
- **Profit factor** — ratio between gross profit and gross loss in per cents. One means that these values are equal;
- **Expected payoff** — mathematical expectation of win. This value to be calculated statistically represents the average profit/loss factor of one trade. It can also be considered to represent the expected profit/loss factor of the next trade;
- **Drawdown \$** — the largest drawdown related to the initial deposit, in the deposit currency;
- **Drawdown %** — the largest drawdown related to the initial deposit, in per cents;
- **Inputs** — changeable values of inputs at each pass.

Having clicked with the left mouse button on any column header, one can sort out all records in the table in increasing or decreasing order. At execution of the "Set Input Parameters" context menu command, the data of the selected pass will be written as the expert basic inputs (expert properties window, "Inputs" tab). At that, it will be switched to the "Settings" tab, and the optimization will be disabled. Having pressed the "Start" button, one can start to test the expert with the selected inputs. Double click with the left mouse button on the pass line in the Optimization Results tab allows to do the same. Using the "Copy" context menu command or accelerating keys of Ctrl+C, one can copy the selected results lines to the clipboard for further use in other applications. If no line has been selected, the entire table will be copied to the clipboard. To do the same, one can also execute the "Copy All" command. The report about optimization results can be saved on the hard disk as an HTML file. To do so, one has to execute the "Copy As Report" context menu command. Other context menu commands allow to set up representation of results:

- **Skip Useless Results** — show/hide the results of unprofitable passes;
- **Show Input Parameters** — show/hide the "Inputs" column;
- **Auto Arrange** — arrange the column sizes automatically when the window size has been changed.
The same action can be done by pressing of A;
- **Grid** — show/hide grid to separate the columns.
The same actions can be done by pressing of G.

Optimization Graph

The graph of profit of all passes is drawn automatically in the "Optimization Graph" tab. The graph allows to estimate the profitability of the use of different inputs combinations visually. The graph representing the amount of profitable (green color) and unprofitable (red color) trades at each pass is given in the bottom of the window, as well.

Double click with the left mouse button on any point of the graph switches to the "Optimization Results" tab and selects the corresponding pass. 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. The graph can also be stored on the hard disk as a GIF file. To do so, one has to execute the "Save as Picture" context menu command or press accelerating keys of Ctrl+S.

Custom Indicators

Custom indicator is a program independently developed in MetaQuotes Language 4 by the user and functioning as a technical indicator. Technical indicator is a mathematical transformation of security price and/or volume in order to forecast future price changes. The use of indicators allows to answer the question about whether the current trend will remain the same and where it will turn. Indicators are intended for relative simplifying of the complicated process of trading decision making. Algorithms of indicators are also used for development of trading tactics and expert advisors.


Attention: Custom indicators are intended only for analyzing of symbol price changes, but not for trading itself.

Working with custom indicators means:

- **Creation of an Indicator**
To create and compile custom indicators, one has to use the built-in "MetaEditor". It is a constituent of the client terminal and represents a convenient development environment of MQL4 programs.
- **Indicator Setup**
Before using of custom indicators, one has to set them up first. Working parameters common for all indicators are defined in the window of client terminal settings. Besides, every indicator can have its own settings.


- **Imposing of an Indicator**
Parameters of the indicator are calculated and the indicator itself is drawn when imposed into the chart.
- **Deletion of an Indicator**
If there is no need of an indicator anymore, the indicator can be deleted from the chart.

Creation of Custom Indicators

To create custom indicators, one has to use MetaQuotes Language 4 (MQL 4) and MetaEditor. To launch the editor, one has to execute the "Create" command of the "Navigator — Custom Indicators" window context menu, or the "Tools — MetaQuotes Language Editor" menu command, or press F4 or the  button of the "Standard" toolbar. At creation of the MQL4 program, the Experts Wizard that allows to create new MQL4 programs will open automatically. "Custom Indicator" must be selected in it as the object to be created, and the necessary fields must be filled out:

- **Name** — indicator name;
- **Developer** — developer's name;
- **Link** — the developer's web site;
- **Inputs** — the list of indicator inputs. To add a new parameter, one has to press the "Add" button, and for deletion, one has to press the "Delete".

Then it is necessary to decide whether the new indicator will be created in a separate sub-window and what range it will have. Besides, it is necessary to define the amount and parameters of the indicator arrays. Values of their elements are used to draw lines in the chart. In other words, when indicator arrays are defined, the lines of the future indicator are defined, too. After that, the window of the new indicator with the defined settings will open. A file with the source code (*.MQ4) of the indicator will be placed into the /EXPERTS/INDICATORS folder of the client terminal automatically. From this point onwards, one can start to write the text of the custom indicator.

After the indicator has been developed, it must be compiled. To do so, one has to execute the "File — Compile" editor menu command, press F9 or the  button of the toolbar. After the indicator has been successfully compiled, an executable program file with *.EX4 extension will be created to be automatically placed into the /EXPERTS/INDICATORS folder. The list of custom indicators can be viewed in the "Navigator — Custom Indicators" window of the client terminal.

Modifying of Custom Indicators

To start modifying of the existing indicator from terminal, one has to execute the "Modify" command of the "Navigator — Custom Indicator" window context menu. At that, the MetaEditor will open where the selected indicator source code has already been downloaded. After this code has been changed, one will have to recompile it and get a new executable EX4 file. Otherwise, the previous, non-modified version of the indicator will be used in the terminal.

Attention: If the indicator has been compiled with errors, it is impossible to attach it to the chart. At that, the indicator properties window will not appear, and a record about impossibility to open a file with an executable code will appear in the experts journal.

Custom Indicators Setup

Indicators must have been set up before they are used. Working parameters common for all custom indicators (and experts) are set up in the client terminal settings. The corresponding window can be opened by the "Tools — Options" menu command or by pressing of accelerating keys of Ctrl+O. To set up working parameters of indicators, one has to select the "Expert Advisors" tab. Only two options influence working of custom indicators:

- **Allow DLL imports**
Custom indicators can use DLLs to enlarge their functionalities. If this option is enabled, the libraries can be used without any limitations. Disabling of the option results in that no MQL4 program can use any external DLLs.
- **Allow external experts imports**
If there is a need to export functions from other experts or MQL4 libraries during the indicators work, this option must be enabled. If it is disabled, no launched indicator will be able to request for functions from other MQL4 programs.

How to Attach Custom Indicators to a Chart

After general setup has been completed, one can attach custom indicators to the chart. A custom indicator can be imposed into the active chart by a double click with the left mouse button on it in the "Navigator — Custom Indicators" window or by execution of the context menu command "Attach to a chart". The "Drag'n'Drop" technique allows to impose the analytical tool into any chart. At that, the setup window will appear automatically that has several tabs:

- **Common**
Import from DLLs and MQL4 can be managed in the "Common" tab. Besides, if indicator is drawn in a separate window, one can set up its range from here. To do so, one has to flag the corresponding options and set the desired values in the fields.
- **Inputs**
External variables that can be managed directly from the terminal are grouped in the "Inputs" tab. To modify the desired variable, one has to double-click on its value in the table and write a new one.
- **Colors**
The "Colors" tab is intended for managing of the indicator elements to be shown in the screen. Besides colors, one can also modify thickness and style of lines.
- **Levels**
Horizontal lines at any level can be set from the "Levels" tab in the indicator window. To create a new level, one has to press the "Add" button, and to delete it, one has to press the "Delete" button. Colors, thickness and style of levels can also be changed from this tab.

Attention: This tab is not available for indicators drawn directly in the price chart.

- **Visualization**
One can limit the use of an indicator by timeframes in the "Visualization" tab. This can be useful if the same indicator must have different settings for different timeframes. One can, for example, impose two copies of the same indicator with different settings into the same chart and limit their use by timeframes: the first indicator will work only for smaller timeframes, and the second one will do for larger timeframes. The "Show in the Data Window" option allows to hide/show data about the given indicator in the Data Window.

Immediately after that, recalculation of the indicator values and drawing thereof in the chart will start. Custom indicators, like technical ones, can be drawn in a separate indicator window with its own vertical scale (for example, MACD) or imposed directly into the price chart (for example, Moving Average).

Removing of a Custom Indicator from the Chart

To shut down an indicator, one has to remove it from the chart. At that, its drawing and recalculation of its values will stop. To remove an indicator from the chart, one has to execute its context menu commands of "Delete Indicator" or "Delete Indicator Window", or the chart context menu command of "Indicators List — Delete".

Attention: Removing of a custom indicator from the "Navigator" window will not shut down the indicator of the same name imposed in the chart.


Scripts

Script is a program written in MetaQuotes Language 4 (MQL 4) and intended for a single performing of any actions. A script can fulfil both analytical and trading functions. Unlike experts, scripts are executed on request, not by ticks. In other words, where an expert works almost continuously, a script, having completed the function once, stops functioning by itself.

Working with scripts means:


- **Creation of a Script**
The built-in "MetaEditor" is used to create and compile a script. It is a constituent of the client terminal and represents a convenient development environment of MQL4 programs.
- **Script Setup**
One has to set up scripts first before using them. Working parameters common for all scripts are defined in the client terminal setup window. Every script has its own settings, as well.
- **Launching of a Script**
To launch a script, one has to attach it to the chart. The script algorithm will be launched immediately after that.
- **Deletion of a Script**
The script completes its working after it has been deleted from the chart.

Creation of a Script

To create a script, one has to use MetaQuotes Language 4 (MQL 4) and MetaEditor. To launch the experts editing program, one has to execute the "Create" command of the "Navigator — Scripts" window context menu, or the "Tools — MetaQuotes Language" menu command, or press F4 or the  button of the "Standard" toolbar. At an MQL4 program creation, Expert Creation Wizard will open automatically that allows to create new MQL4 programs promptly. "Script" must be selected in it as the object to be created, and all necessary fields must be filled out:

- **Name** — script name;
- **Developer** — developer's name;
- **Link** — the developer's web-site.

After that, the new script window will open in the editor. File containing the script source code (*.MQ4) will be placed into the /EXPERTS/SCRIPTS folder of the client terminal automatically. Then one can start to write the source code of the program.

After the script has been created, it must be compiled. To do so, one has to execute the "File — Compile" menu command in the experts editor, press F9 or the  button in the toolbar. After the script has been successfully compiled, the executable file with *.EX4 extension will be created and placed into the /EXPERTS/SCRIPTS folder automatically. The list of all scripts can be viewed in the "Navigator — Scripts" window of the client terminal.

Editing of Scripts

To start editing of the existing script from the terminal, one has to execute the "Modify" command of the "Navigator — Scripts" window context menu. At that, the experts editor with the selected script source code already downloaded will open. After the source code has been modified, one has to recompile it and get a new executable EX4 file. Otherwise, the previous, non-modified version of the MQL4 program will be used in the terminal.

Scripts Setup

Before starting to use scripts, one has to set them up. Working parameters common for all scripts are stated in the terminal settings window. This window can be opened by the "Tools — Options" menu command or by pressing of accelerating keys of Ctrl+O. To set up script parameters, one has to open the "Expert Advisors" tab. Only five options influence the operation of scripts:

- **Allow live trading**
Like experts, scripts can work in automated mode. They can both analyze price changes and trade. This option is intended for limiting of trading functions of scripts and expert advisors.
- **Ask manual confirmation**
The "Ask manual confirmation" option is only activated if scripts are allowed to trade. If this option is enabled and the script tries to perform a trade operation, one will be able to either confirm it or stop it manually. In other words, manual confirmation allows to control over trading activities of experts and scripts.
- **Allow DLL imports**
To have their functionalities enlarged, scripts can use DLLs. If this option is enabled, the libraries can be used without any limitations. If this option is disabled, no script will be able to use any external DLLs.
- **Confirm DLL function calls**
This option can only be enabled if DLLs are allowed to be imported. It assist to control over each called function execution manually. If it is disabled, import from external libraries will not be controlled.
- **Allow external experts imports**
If, at the script operation, there is a need to import functions from other expert or MQL4 libraries, this option must be enabled. If it is disabled, no script will be able to call functions from other MQL4 programs.

Launch of Scripts

After general parameters have been set up, the script can be launched. To do so, it is enough just to attach it to the chart. At that, of there is a "#property show_inputs" instruction in the program source code, the script setup window will appear automatically. In the "Common" tab of this window, one can:

- **positions** — select direction of position opening:
 - 0. Long&Short — both long and short;
 - 0. Only Long — only to buy;
 - 0. Only Short — only to sell.
- **Enable alerts** — enable/disable script alerts;
- **Disable alert once hit** — disable alerting after the first alert has been made;
- **Allow live trading** — enable/disable live trading;
- **Ask manual confirmation** — request for manual confirmation of position opening when performing trade operations;
- **Allow DLL imports** — enable/disable importing of functions from DLL files;
- **Confirm DLL function calls** — request for confirmation at each call of functions from DLL files;
- **Allow import of external experts** — enable/disable calling of functions from external experts.

External variables of the script can be changed in the "Inputs" tab. They are variables of extern class. To change a parameter, one has to double-click with the left mouse button on its value and write the new one. At that, one can change the value of each variable or download the set of inputs already saved before (the "Load" button). One can save the current set of inputs with the button of the same name.

The "Reset" button returns all default settings. Parameters defined in the terminal settings are set in the "Common" tab. And parameters set in the source code of the program are defined as inputs. The script with the defined parameters can be confirmed by pressing of "OK" and canceled by pressing of button of the same name.

Attention: Unlike those of experts or custom indicators, special properties of the script are set only at its launch.

Script will be launched immediately after that. Double click with the left mouse button on the selected script in the "Navigator — Scripts" window or execution of the "Execute on Chart" command of the script context menu will attach the script to the active chart. The "Drag'n'Drop" technique will allow to attach the script to any chart. If the "Remove Script" command appears in the chart context menu, it means that script is working. This command is active only while this MQL4 program is working.

Attention: Only one script can be attached to a chart.

Script Shutdown

Normally, a script shuts down by itself. But it can be removed manually. At that, it must be deinitialized first and then rolled out from the memory. To remove a script from the chart, one has to execute the chart context menu "Remove Script" command or attach another script to the chart. Besides, a script will be removed from the chart after symbol or timeframe has been changed.

Attention:

- all scripts will be removed after terminal has been shut down;
- removing of script from the "Navigator" window will not shut down the MQL4 program of the same name imposed in the chart.