

and previously unused UserRecord::balance\_status field. MetaQuotes ID is used like a phone number. After specifying MetaQuotes ID in the desktop version of the client terminal, users can send notifications of different trade events to their mobile devices. MetaQuotes ID is also supported on [MQL5.community](#): after specifying an ID in the profile, a user is able to receive important notifications from the site and communicate with other community participants via personal messages. You can find more details in the article [MetaQuotes ID in MetaTrader Mobile Terminal](#).

### Added the new method for licensing the applications

Added CManagerInterface::LicenseCheck method for checking the presence of the license for the application using Manager API in the server license. This method is necessary for distribution of third-party applications for MetaTrader 4 platform. In particular, this type of licensing is to be used in the app store that will soon be launched on [MetaQuotes Support Center](#) website.

Permission to use the application (application module name) is registered in the trade server license according to the client's request or when purchasing the application in the Market. In CManagerInterface::LicenseCheck method, an application developer passes the name of the licensed module. In case the license contains that module, the trade server returns RET\_OK. The developer can define the application's operation logic in case the trade server returns the answer different from RET\_OK. For example, the application may be disabled completely or it may work only in demo mode.

### Added ability to display custom messages in Manager API journal and disable logging

Added CManagerInterface::LogsOut method for displaying your own messages to Manager API journal. This improves convenience of logging an application operation, as all logs are collected in a single place.

Also, CManagerInterface::LogsMode method has been added. It allows selecting one of the two Manager API journal modes:

- LOG\_MODE\_ENABLED - Manager API will send its entries to the journal.
- LOG\_MODE\_DISABLED - Manager API will not send its entries to the journal.

## All Platform Components

### Entry time in the journals is now displayed up to milliseconds

Improved the accuracy of the time displayed in the journals of all MetaTrader 4 platform components: all terminals, servers and APIs. Now, the time is displayed up to milliseconds.

If you use Server API plugins, Manager API apps or any other programs analyzing logs, make sure that they work with the new Journal entry format correctly.

Previous entry format:

```
2013.11.27 14:42:08    192.168.0.100:400    '5': login
```

New Journal entry format:

```
2013.11.27 14:42:08.249 192.168.0.100:400    '5': login
```

### Fixed errors reported in crash logs

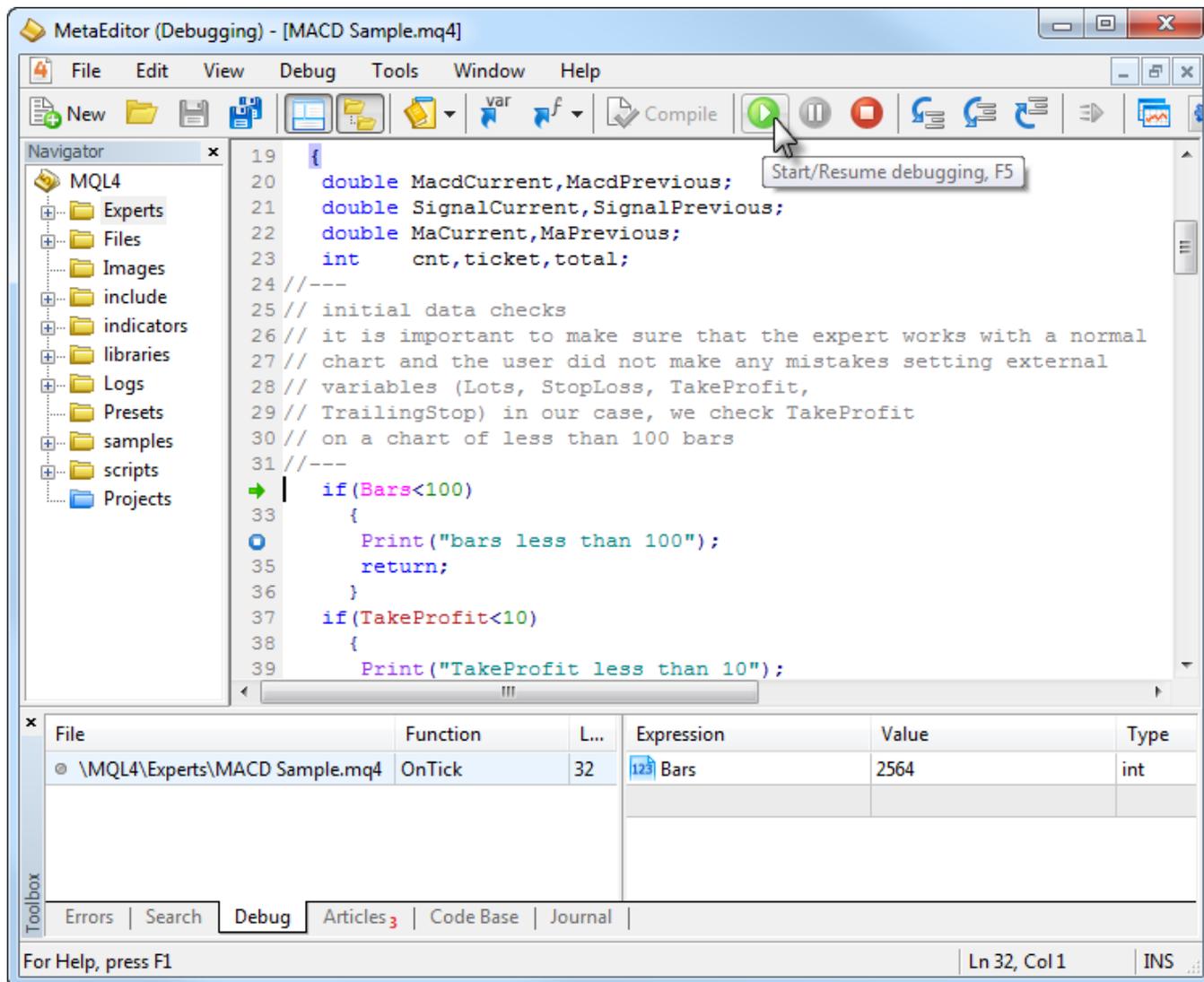
Each platform component contains the special system that is triggered in case of the program's emergency completion. This system generates anonymous reports on the technical details of the program crash and sends them to MetaQuotes Software Corp. for analysis. This mechanism of error detection and timely processing is very important for such complex system as MetaTrader 4 online trading platform. It ensures that each system component will be promptly fixed improving the system's general resilience.

## MetaTrader 4 Client Terminal with Updated MQL4 and Market of Applications

Over the past seven months since the release of the public build 509, we put in a lot of effort to improve MetaTrader 4 client terminal. MQL4 programming language for developing trading strategies has undergone the most significant changes. These changes have provided new features for developing trading robots and made possible the launch of [MetaTrader 4 Market](#) - secure app store for MetaTrader 4 terminal.

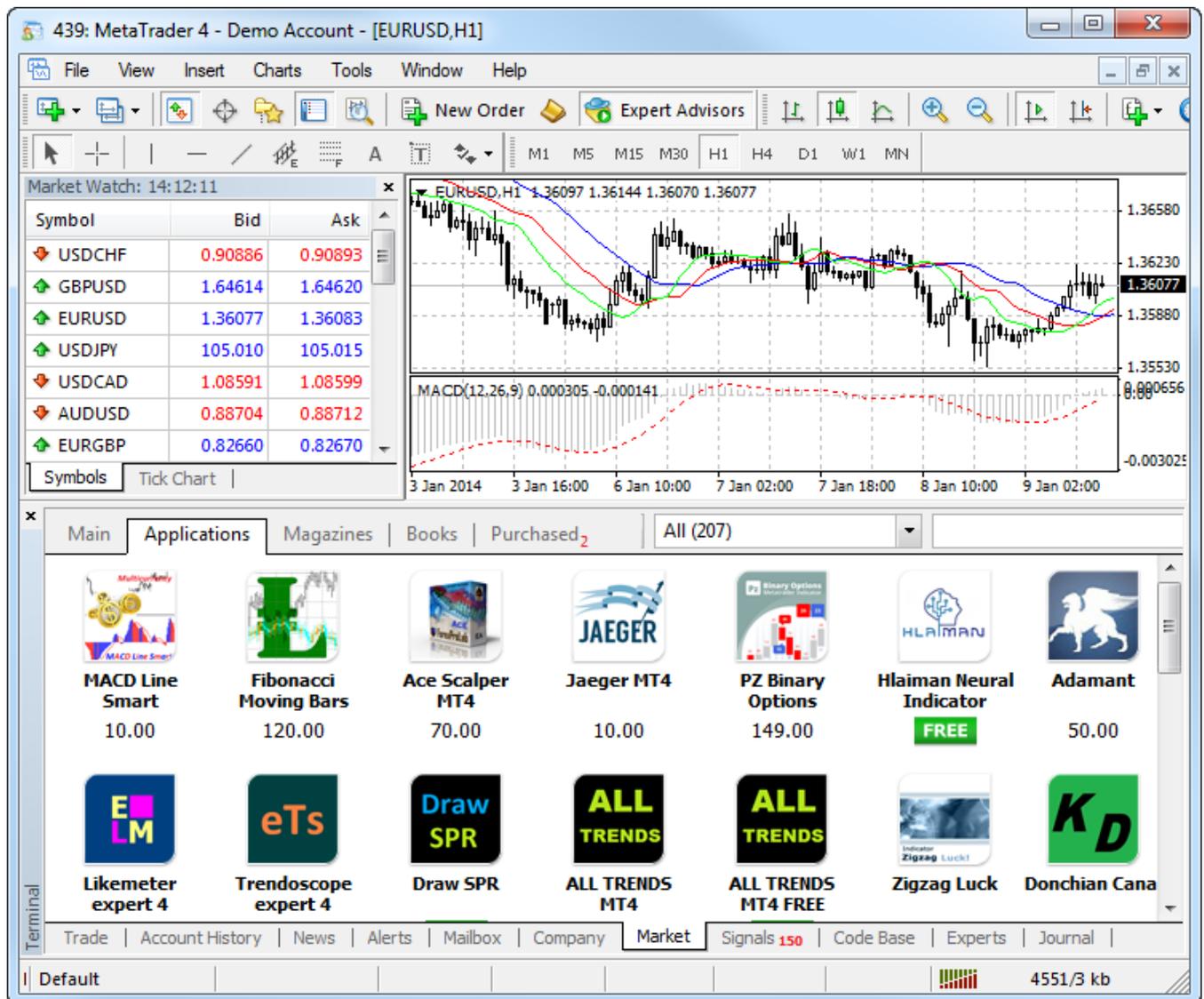
### MetaTrader 4 Client Terminal

1. Terminal: MQL4 language for programming trading strategies has been substantially revised and its functionality has been brought closer to that of MQL5. Classes and structures have been added and the language syntax has been extended up to C++ level allowing developers to implement all the advantages of object-oriented programming. It means that trading robots and technical indicators can be created much easier and faster now, while Expert Advisor developers have gained new opportunities in a single MetaEditor development environment - debugging, profiling, code completion, etc. Find out more details on the new features in a separate section of [MetaEditor](#) article on MQL5.



Increased operation speed of MQL4 apps, provided built-in support of DLL and Visual Studio, implemented access to [MQL5 Storage](#) and much more.

- Terminal: Added Market tab for buying and downloading paid and free Expert Advisors, signals and scripts from [MetaTrader 4 Market](#). Now, any MQL4 developer can become a Seller and [offer products to millions of traders](#) using MetaTrader 4.



You can download a demo version or buy a full-featured paid MQL4 application on MetaTrader 4 Market directly from the terminal. Each purchase is completely secure allowing a user up to 5 activations on different PCs and has no limitations on working time, account type or a broker. Each paid indicator or an Expert Advisor can be launched in MetaTrader 4 strategy tester in visual mode before purchasing in order to evaluate the product.

- Terminal: Changed the terminal data structure and storage location - now, the data of each installed terminal is stored in a separate terminal data folder.

Since the [release of MetaTrader 4 platform](#) in 2005, selected **terminal installation directory** was also the place where all data necessary for work (terminal logs, templates, profiles, quotes, etc.) has been stored. But with the release of Windows Vista in 2008, all Microsoft operating systems implement serious limitations of user privileges in order to protect them against malicious programs and to prevent applications under one user account to change or damage the same program's data necessary for another user account.

Starting with Windows Vista, User Account Control (**UAC**) mechanism was introduced. The main idea of implementing the mechanism is that the application's unchangeable data stored in the installation directory should be separated from user's data modified by this application during its operation. Thus, all data of each user should be stored in a separate folder. You can find more information about UAC in Windows help or the [Wikipedia article](#).

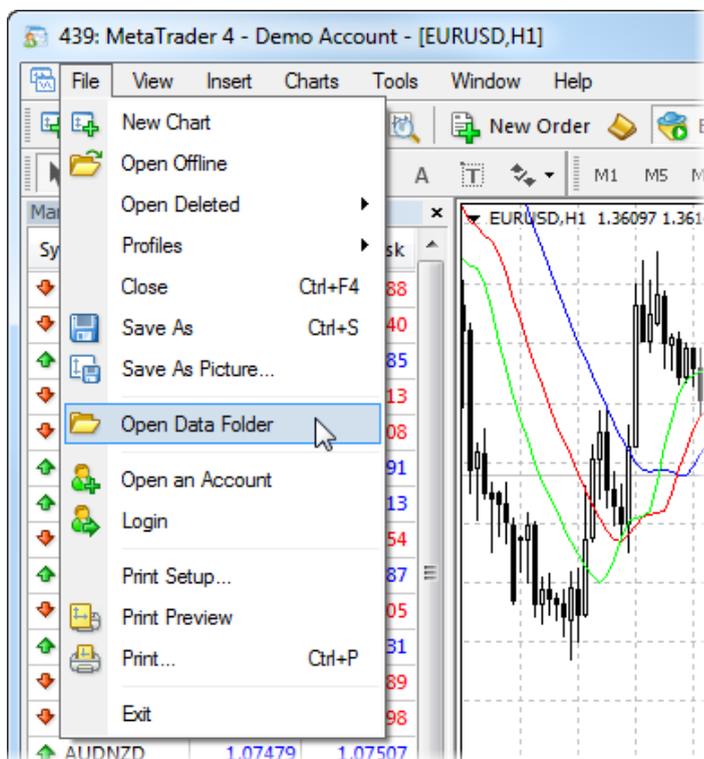
In the new version, all data of a certain user working with MetaTrader 4 terminal from a separate installation directory are stored in the **terminal data folder**. This folder can be found on a system disk (a disk with installed Windows operating system) along the following path

```
C:\Users\User_account_name\AppData\Roaming\MetaQuotes\Terminal\Instance_id
```

where:

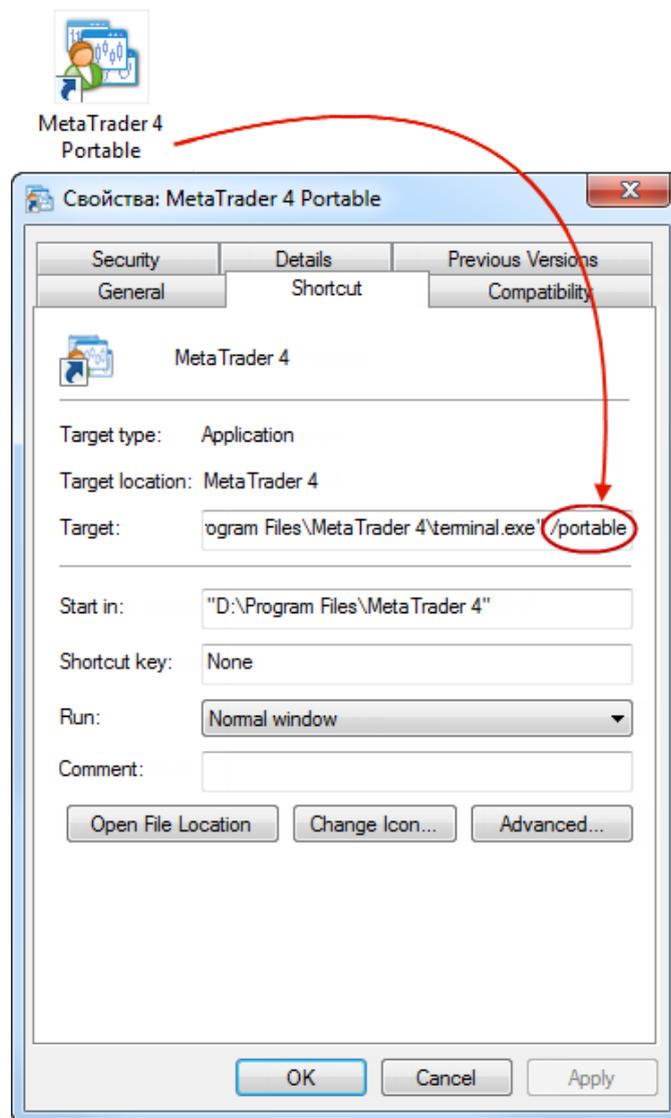
- C: - name of a system disk;
- User\_account\_name - user's account for working in Windows;
- Instance\_id - unique name of the folder where all the user's data for working with a certain copy of MetaTrader 4 terminal is stored.

"Open Data Folder" command has been added to File menu of the terminal for searching and opening the data folder.



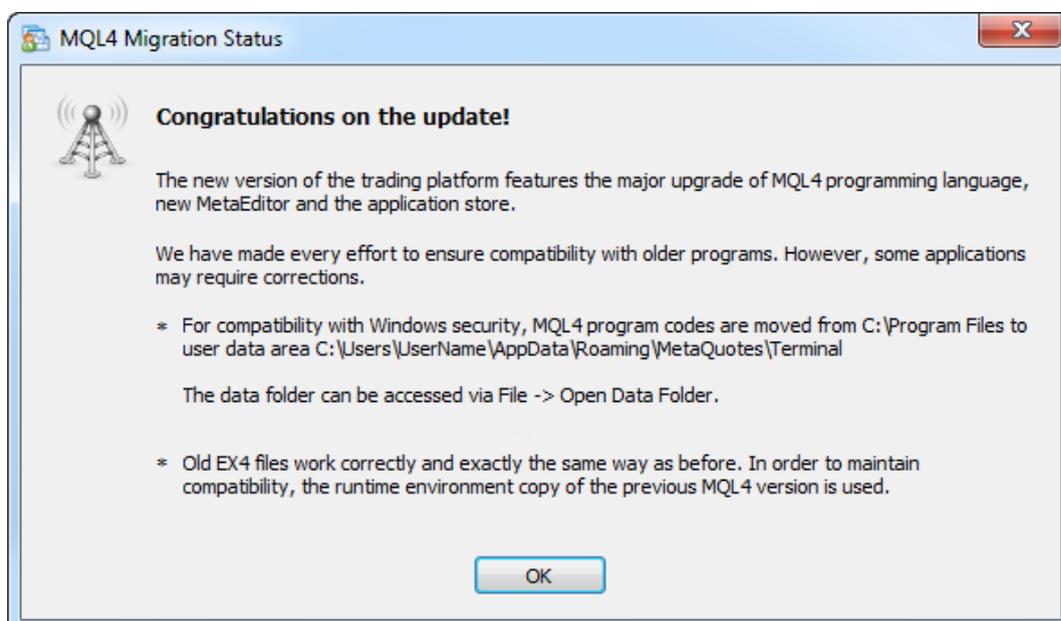
The root of each terminal data folder also contains **origin.txt** file where you can find the path to the installation folder of the terminal this data refers to. This allows users to reversely match each terminal data folder with a certain terminal installation directory, for example, in case when several copies of MetaTrader 4 terminal are installed by a user. This type of working with the terminal when the data folder is separated from the installation one is the **main mode** and highly recommended for security reasons.

The terminal can also be launched in the so-called **Portable mode** (as in previous MetaTrader 4 builds). In that mode, the data is stored and recorded in the installation folder. To do this, use **"/portable" key**. For more convenience, you can create an additional terminal launch shortcut with the appropriate name on your desktop and add the key directly to the shortcut:



4. When updating the old terminal version, existing user data is copied to the new location in two cases:
- installation directory is different from the data one;
  - installation directory contains custom MQL4 programs (Expert Advisors, indicators, scripts, etc.) not included in the standard delivery of the terminal's old version.

After the user data is copied successfully and the update is complete, users will see a dialog window containing the full paths of the source folder and new data folder:



Additionally, the terminal's Journal will contain messages about all the movements of the user data performed during the update.

- Terminal: Changed the algorithm of placing trade requests. In previous builds, an additional network connection has been created when placing trade requests. This connection has remained active for some time after a last trade request was sent in order to provide fast sending of a large number of trade requests.

Now, when placing trade requests, all of them are passed via the main connection with the trade server. Thus, the time of processing the trade requests (especially the first one) has been considerably reduced as there is no need to wait for connection to the trade server being established any more. In order for the new request sending algorithm to be operable, the trade server should also be updated - requests are still set in a separate connection on older trade servers.

Due to transition to the new model of requests processing, the ability to cancel the trade requests not yet accepted by the server (Cancel button in the trade request status dialog) has been disabled.

- Terminal: Changed RateInfo history data storage format. RateInfo structure was presented as follows in the old version:

```
struct RateInfo
{
    unsigned int    ctm; // bar open date
    double          open; // Open price
    double          low; // Low price
    double          high; // High price
    double          close; // Close price
    double          vol; // volume
};
```

In the new format, RateInfo structure features fields for storing spread and trading volume:

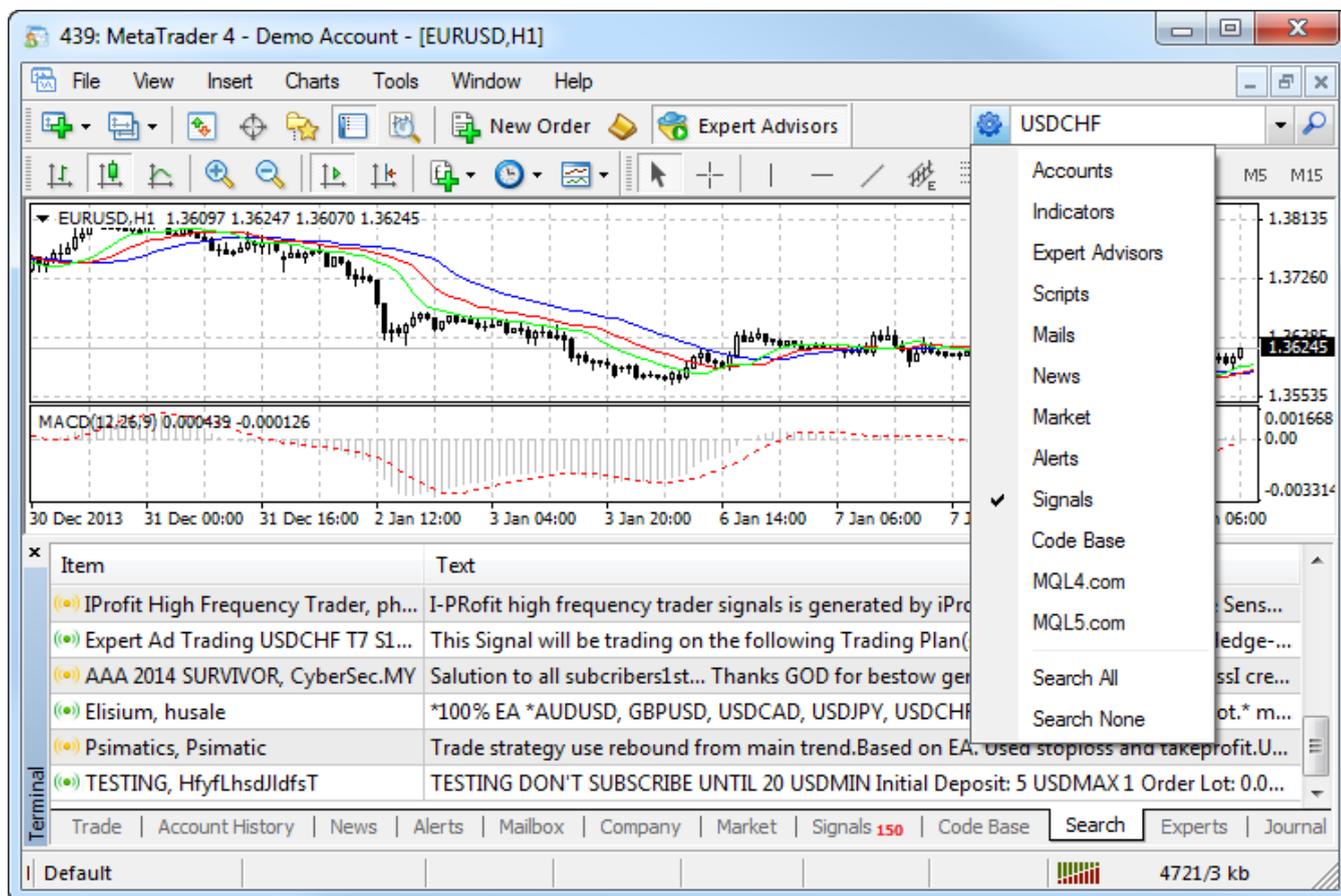
```
//--- Standard quote presentation in the new terminal version
struct RateInfo
{
    INT64          ctm; // open date and time
    double         open; // Open price (absolute value)
    double         high; // Low price
    double         low; // High price
    double         close; // Close price
    UINT64         vol; // tick volume
    INT32          spread; // spread
    UINT64         real; // trade volume
};
```

Thus, if MQL4 programs contain DLLs for passing/accepting price data, the corresponding functions in the source codes should be rewritten and recompiled considering format changes to ensure proper operation.

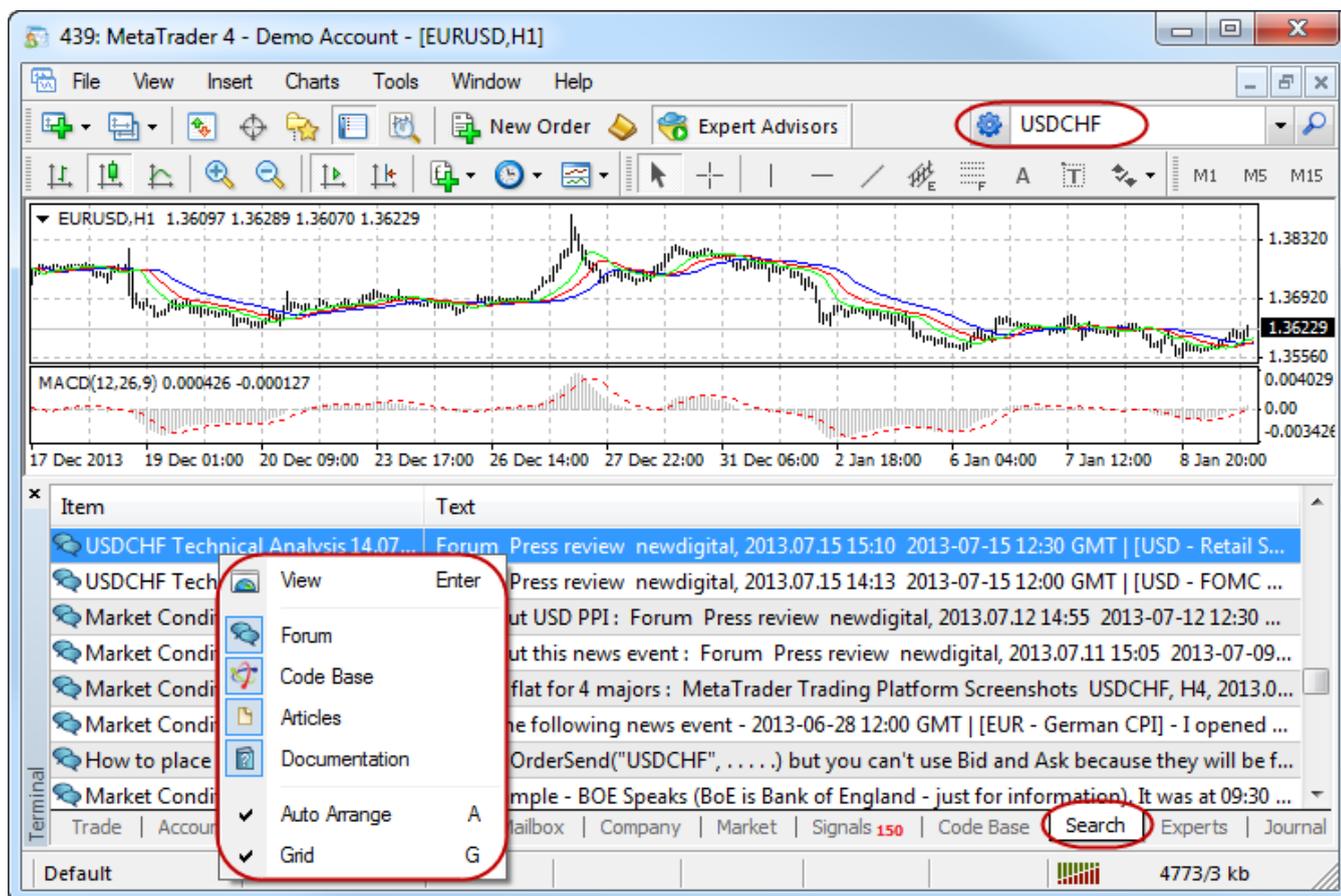
**Old EX4 applications and DLLs based on old RateInfo format will not work in the new terminal. Conversion to the new format is required.**

- Terminal: Added global search in MetaTrader 4 client terminal and at MQL4.com. The search in the terminal allows you to quickly find information in all sections of the client terminal, [Code Base](#), as well as on MQL4.com and MQL5.com websites.

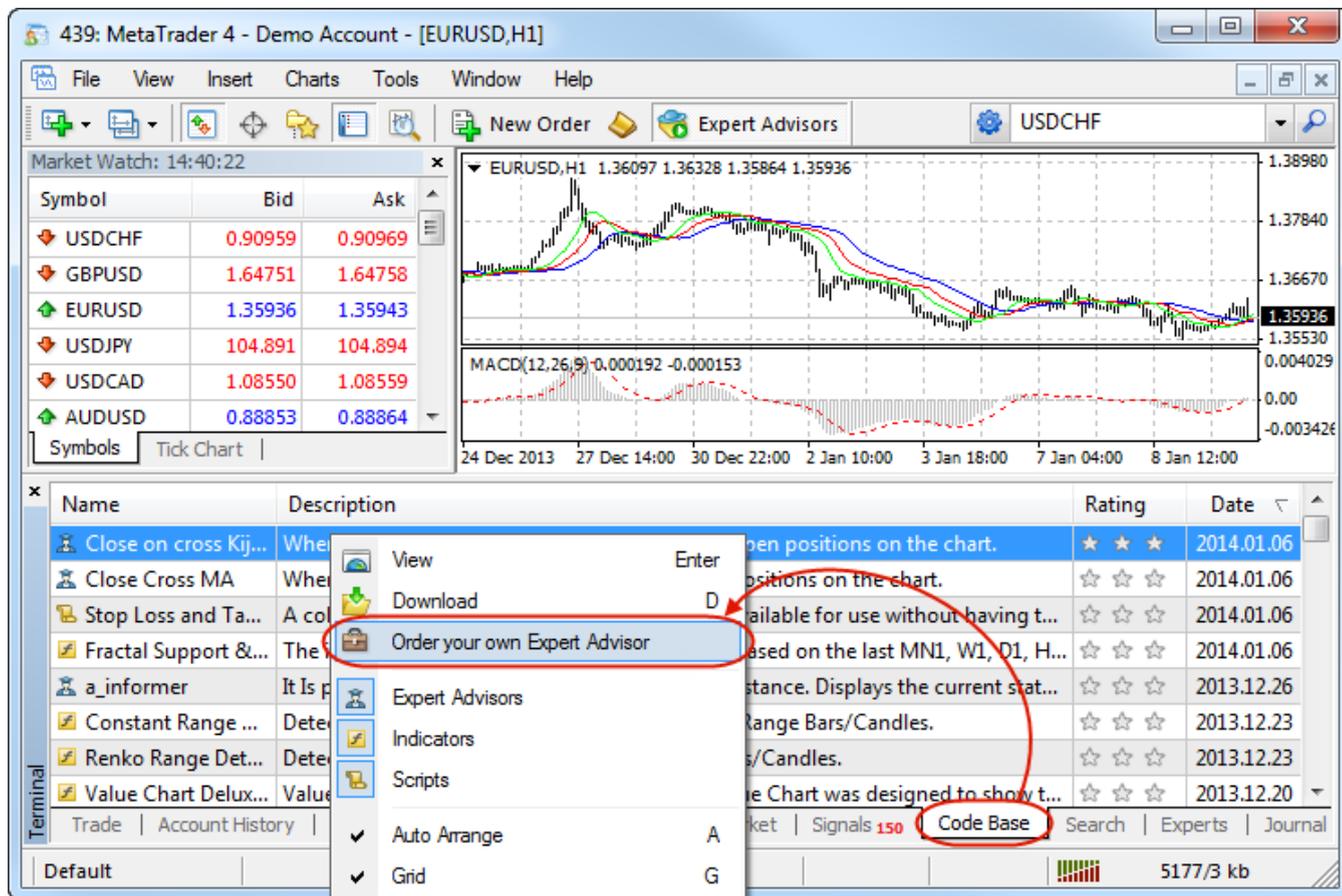
Word-by-word search with regard to Russian and English morphology is used. The search is performed in the sections specified by a user in the settings:



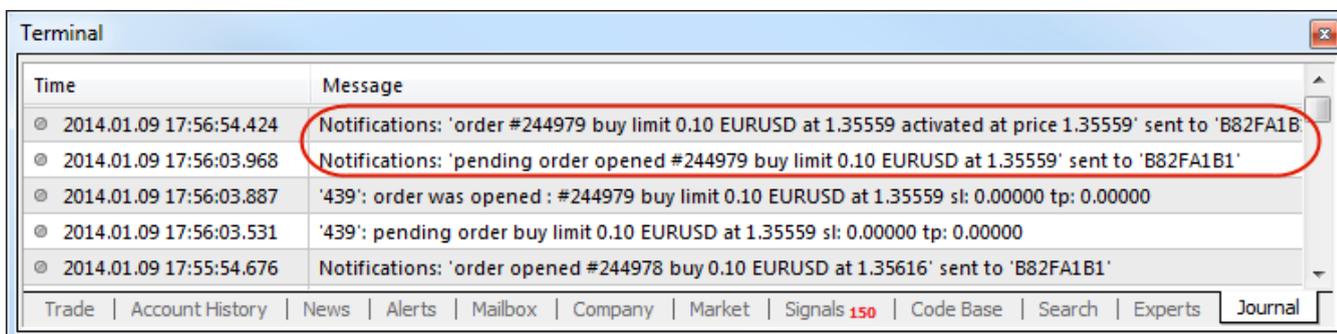
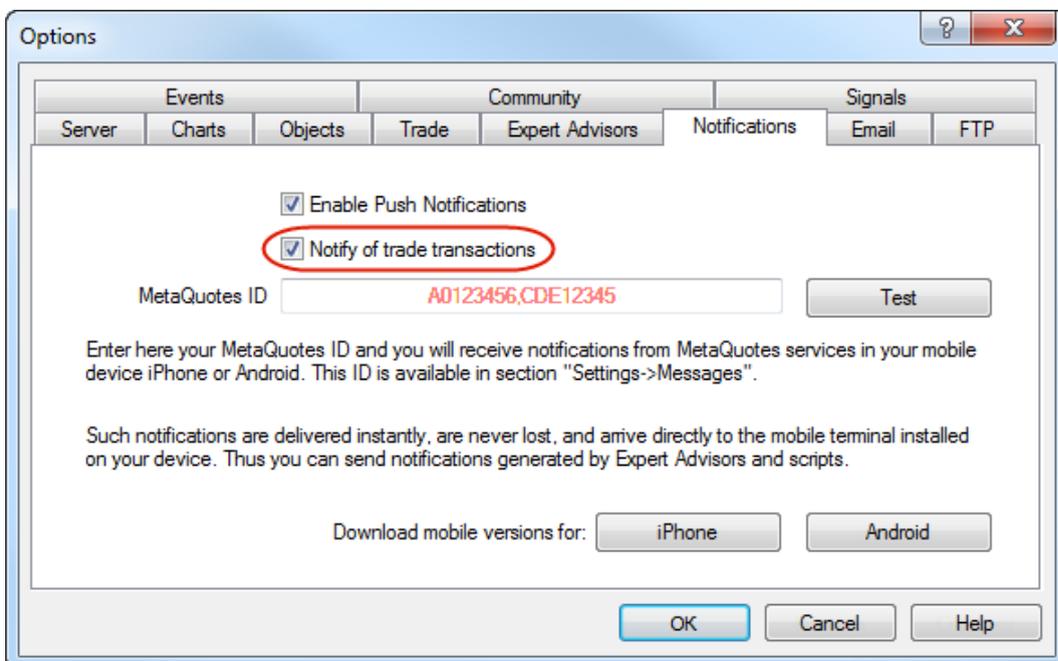
The context menu of the search results tab contains additional commands depending on the type of results:



- Terminal: Added the command of placing an instant order for the development of MQL4 applications in [Jobs](#) section. Now, each trader can not only download any code from the free library but also find a programmer able to develop a necessary indicator or a trading robot according to the user algorithm. The process of an order execution is described in the article [How to Order an Expert Advisor and Obtain the Desired Result](#) and provides protection for both parties - a developer and a customer.



- Terminal: Added ability to send push notifications of transactions occurring on the client account: placing, changing and removing orders, activation of pending orders and SL-TP, Margin Call and Stop Out events. Push notifications are sent to the mobile terminals installed on iOS and Android powered devices, do not depend on the phone number and can be delivered if Internet connection is available. Find out the details in the article [MetaQuotes ID in MetaTrader Mobile Terminal](#).

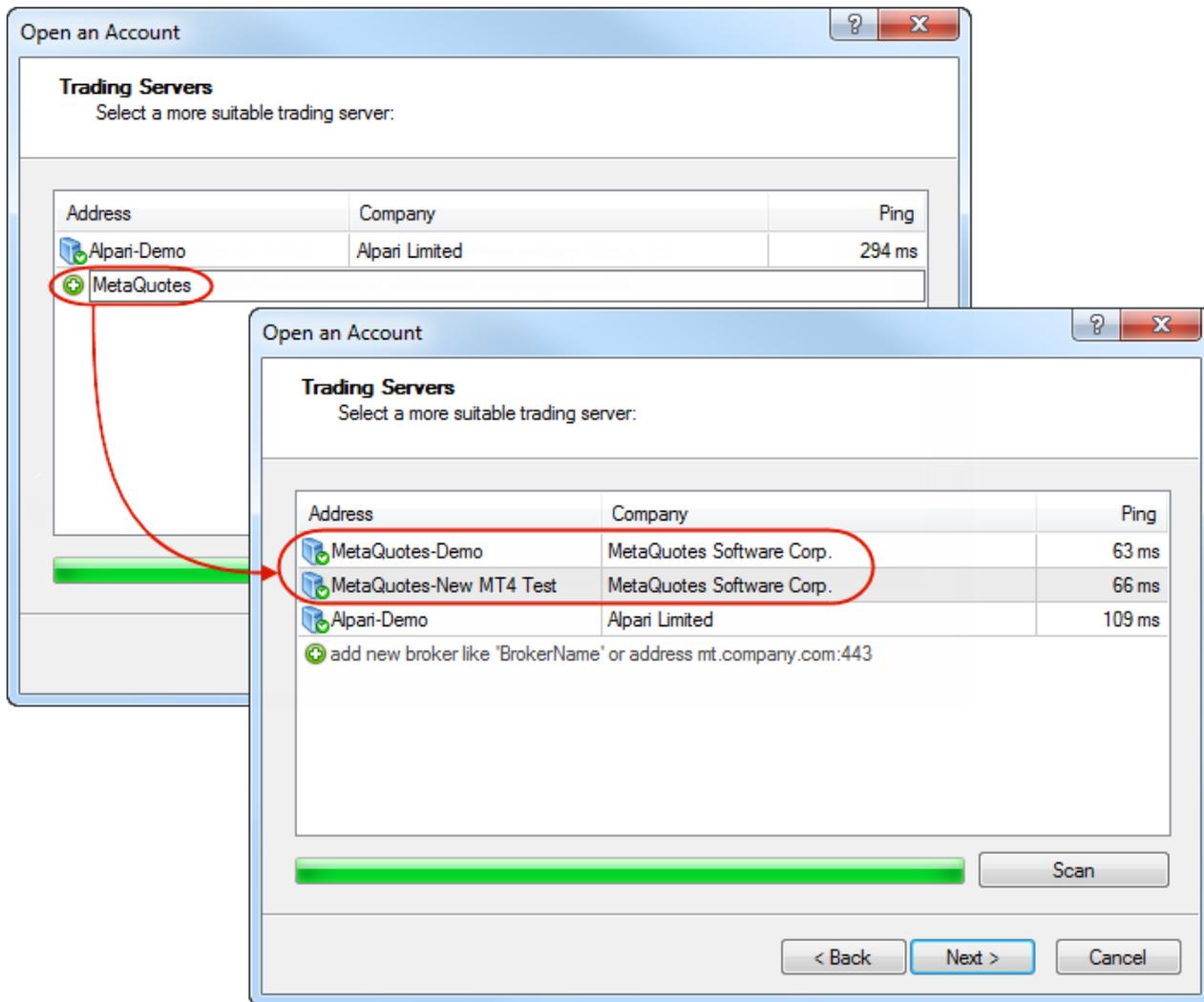


10. Terminal: Added display of alerts on the chart. The alerts can be managed via drag'n'drop. Also, the ability to set an alert expiration time has been implemented.

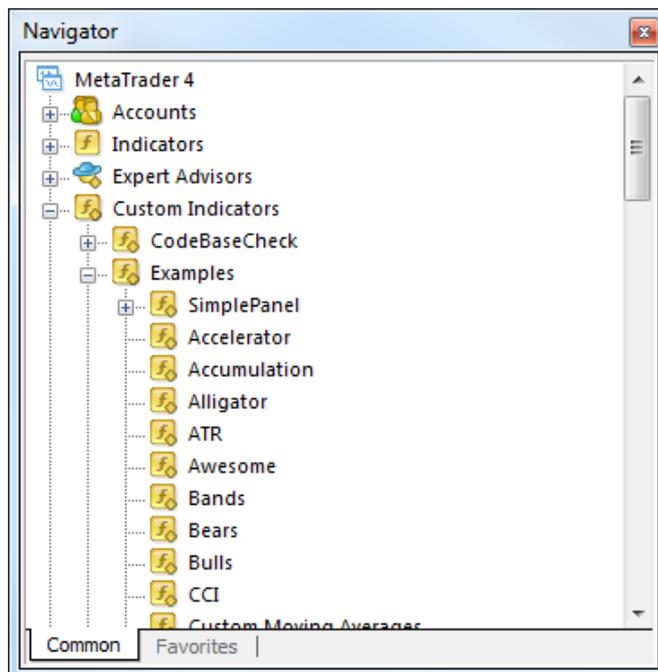


When management of trading levels from the chart is allowed, alert's price value can be changed by its dragging to a new price level. Alerts can be disabled or removed using a context menu on the appropriate chart level.

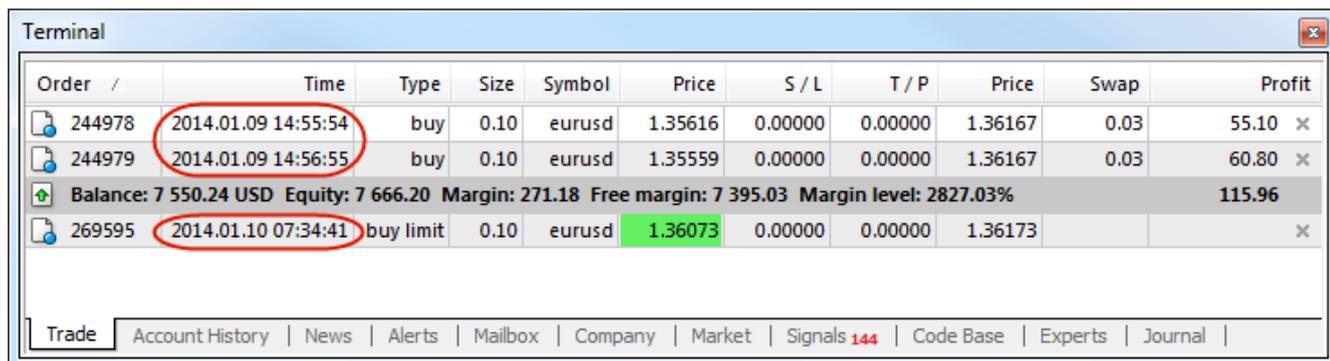
- 11. Terminal: Added search for a server by a company name in the demo account opening window. Just enter a company name (or a fragment of a server name) and the terminal will automatically add all servers of the company (the servers containing the specified line in their names) to the list:



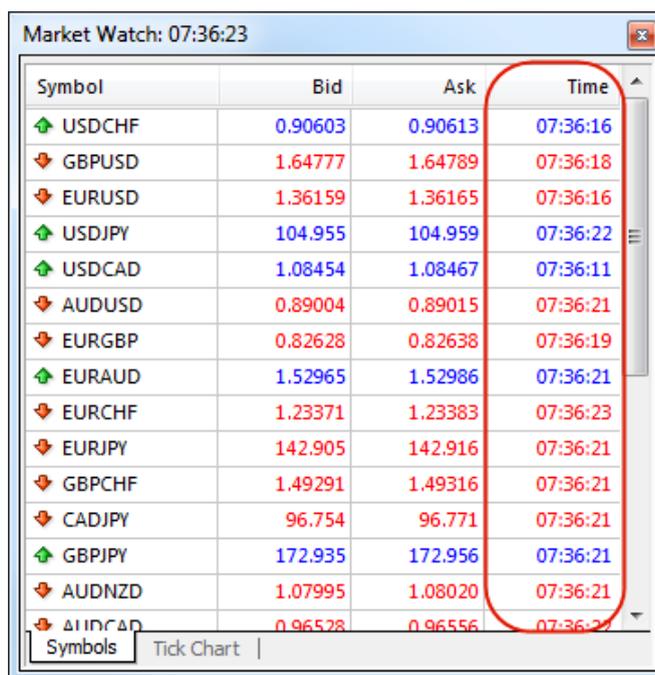
- 12. Terminal: Added display of subfolders and the structure of Expert Advisors, scripts and indicators in Navigator window.



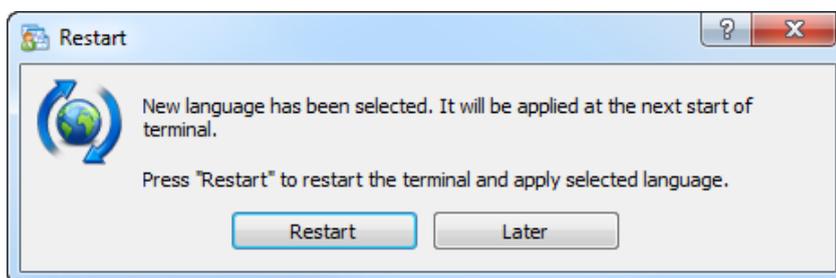
- 13. Terminal: Improved accuracy of opening and closing time of trade orders and positions displayed in the lists of open orders and positions, trading history and trade reports - seconds have been added.



- 14. Terminal: Improved accuracy of the last quote's time displayed in Market Watch and Popup Prices windows - the last quote's time is shown up to seconds (instead of minutes).

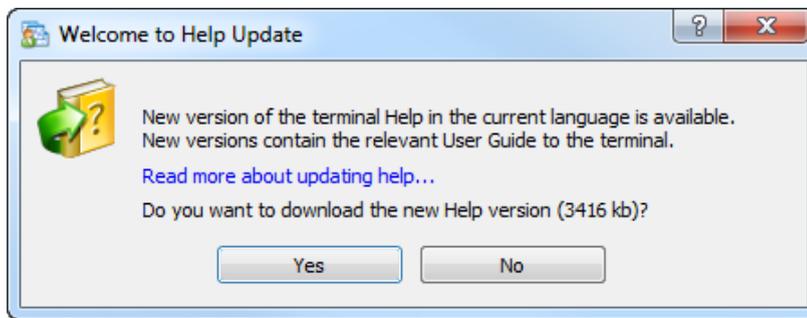


- 15. Terminal: Added automatic terminal restart request after changing the interface language. The client terminal should be restarted in order for the change to take effect. Previously, users had to close and launch the terminal for themselves. Now, after changing the interface language, users are immediately prompted to restart the terminal - it can be closed and restarted at a single click.



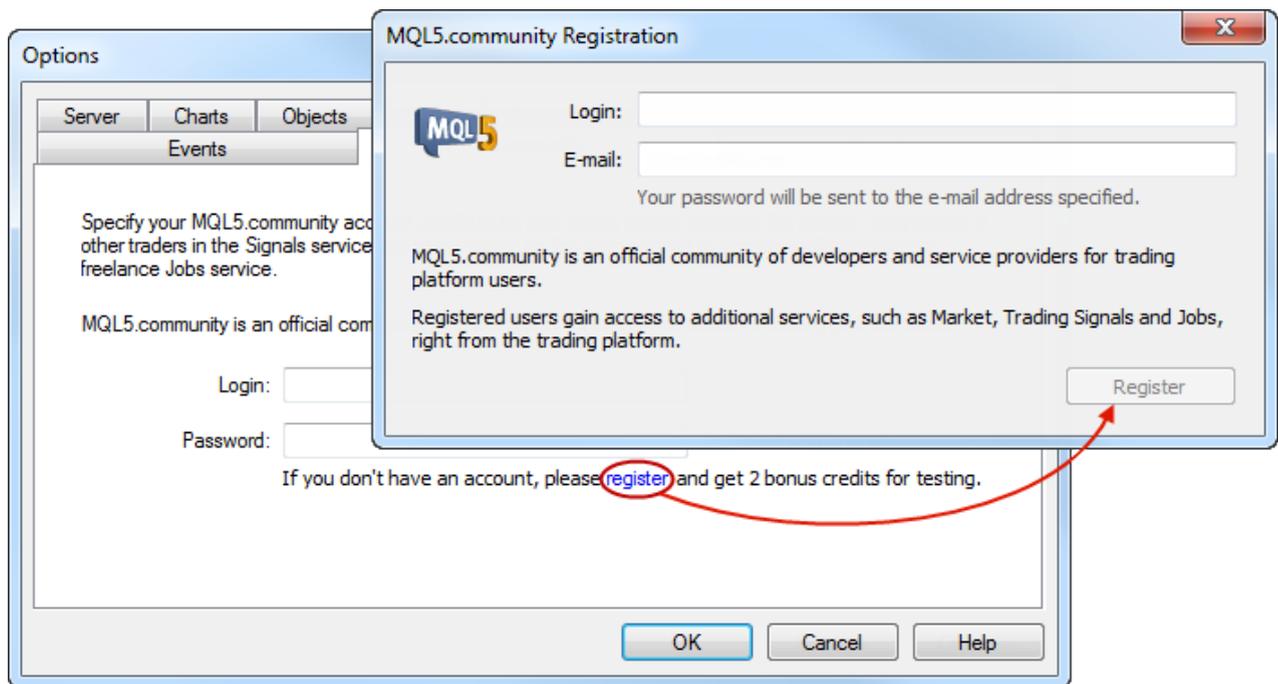
- 16. Terminal: Fixed tooltips of trading levels in case the indicators using a separate subwindow are present on the chart. When hovering the mouse cursor over Stop Loss or Take Profit level on the chart, a tooltip containing profit/loss (in points and currency) to be received in case of the level's activation is displayed. Previously, if an indicator working in a separate subwindow (for example, standard MACD) was present on the chart, that tooltip was not displayed sometimes.

- 17. Terminal: Added automatic update of the client terminal built-in user guide files similar to the one implemented in MetaTrader 5. Now, help files are not included in the terminal's installation and LiveUpdate packages reducing the traffic volume. The user guide is downloaded the first time a user opens it. In all subsequent cases, the system checks if the new versions are present. If there is a new version available, the user is prompted to update:



The new system of working with help files also reduces the traffic consumption due to the fact that the language version of the downloaded user guide corresponds only to the language specified for the client terminal's interface (no excessive language versions are downloaded). This system also provides regular update of the user guide regardless of the client terminal update.

18. Terminal: Added MQL5.community fast registration dialog in case a user has no account. Now, an MQL5.community account can be created without the need to leave the terminal.



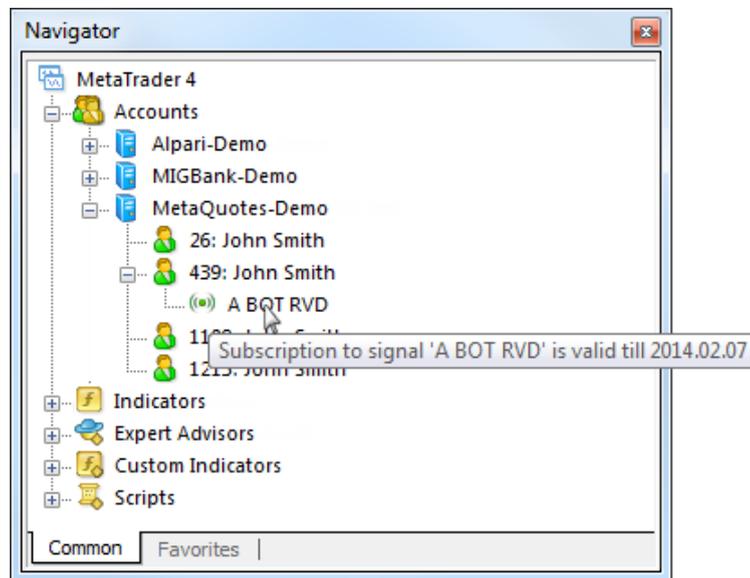
Specify login and email address in the registration window. After clicking Register, an email for MQL5.community account activation is sent to the specified address.

MQL5.community account allows traders to use additional powerful services:

- MetaTrader 4 AppStore - users can buy MetaTrader 4 apps or download them for free directly from the terminal. [MetaTrader 4 AppStore](#) offers hundreds of various applications and their number is constantly increasing.
- Signals service - users can [subscribe to trading signals provided by professional traders](#) and make profit. Trading operations are automatically copied from provider's account to subscriber's one. The service also allows selling your own trading signals. A trading account can be connected to the monitoring system in a few clicks.

19. Terminal: Trading accounts are now displayed much clearer in Navigator window:

- Accounts are grouped by their appropriate trade servers.
- If an account is subscribed to trading signals, the icon with a signal name is displayed nearby. When hovering the mouse cursor over the signal, subscription expiration date is shown.

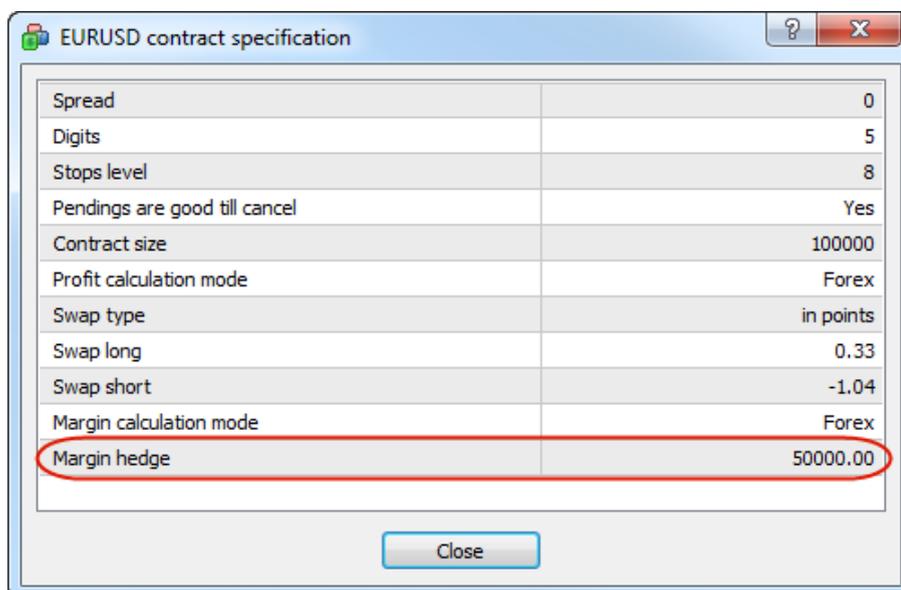


20. Terminal: If the current trading account is subscribed to a signal, the appropriate icon is displayed in Trade tab of Symbols window. When hovering the mouse cursor over the icon, the data on the signal is shown. There have been cases when traders forgot that their account was subscribed to a signal. That could happen, for example, when subscription has been made to check the service's operation. The subscription icon makes working with signals more convenient.

Order	Time	Type	Size	Symbol	Price	S / L	T / P	Price	Swap	Profit
244063	2014.01.07 13:0...	buy	0.10	chfjpy	115.166	0.000	0.000	115.139	0.00	-2.59
244064	2014.01.07 13:0...	buy	0.10	nzdjpy	86.599	0.000	0.000	86.594	0.00	-0.48
244065	2014.01.07 13:0...	buy	0.10	nzdusd	0.82938	0.00000	0.00000	0.82939	0.00	0.10
244066	2014.01.07 13:0...	sell	0.10	gbpchf	1.48848	0.00000	0.00000	1.48898	0.00	-5.52
244067	2014.01.07 13:0...	buy	0.10	audusd	0.89276	0.00000	0.00000	0.89286	0.00	1.00
244068	2014.01.07 13:0...	sell	0.10	usdchf	0.90659	0.00000	0.00000	0.90681	0.00	-2.43
244069	2014.01.07 13:0...	buy	0.10	usdjpy	104.409	0.000	0.000	104.409	0.00	0.00
244070	2014.01.07 13:0...	buy	0.10	eurjpy	142.436	0.000	0.000	142.421	0.00	-1.44
244071	2014.01.07 13:0...	buy	0.10	eurusd	1.36424	0.00000	0.00000	1.36408	0.00	-1.60
244072	2014.01.07 13:0...	buy	0.10	eurgbp	0.83087	0.00000	0.00000	0.83073	0.00	-2.30
244073	2014.01.07 13:1...	sell	0.10	eurjpy	142.426	142.826	141.426	142.433	0.00	-0.67
<b>Balance: 7 476.70 USD Equity: 7 460.77 Margin: 1 138.90 Free margin: 6 321.87 Margin level: 655.09%</b>										<b>-15.93</b>

21. Terminal: Added new graphical objects: Button, Bitmap, Bitmap Label, Rectangle Label. These objects cannot be manually installed on the charts and are designed for organizing a user interface for MQL5 applications: Expert Advisors, scripts and indicators.
22. Terminal: Optimized work with [MQL5.community](#) in case of slow Internet connection. Now, you can work with [Trading Signals](#), [Code Base](#) library and the version [MQL5 Storage](#) conveniently even at low connection speed.
23. Terminal: Improved accuracy of the time displayed in the client terminal's Journal - milliseconds have been added.
24. Terminal: Fixed [One Click closing](#) of positions in the list of open orders and positions (by clicking Cross button) - now the value of the acceptable price deviation from the [trading settings](#) is used in Instant Execution mode.
25. Terminal: Added ability to copy to clipboard on Company tab.
26. Terminal: Fixed arranging chart windows by commands.
27. Terminal: Fixed saving profiles when switching between full-screen charts.
28. Terminal: Fixed display of a number of dialogs when using the Japanese interface.
29. Terminal: All client terminal components require support for SSE2 instruction set by PC's processor for efficient operation and high performance.
30. Terminal: Optimized scanning for the best access point.
31. Terminal: Fixed looping errors that occurred during the update in case of refusal from elevating the rights when using UAC.
32. Terminal: Fixed display of the news header and contents, as well as of internal e-mails using UTF-8 encoding.
33. Terminal: Fixed an error that in some cases could lead to incorrect display of Margin and Free Margin fields in the list of open orders and positions. This error might occur when a user declined to accept "One-Click Trading" agreement.
34. Terminal: Updated the list of countries in the appropriate dialog when opening a demo account.

35. Terminal: Updated and enhanced translation of the interface into Swedish, Hungarian, Japanese, Italian and Bulgarian.
36. Terminal: Fixed time in the bar dialog of the quotes Archive. Previously, bar time was displayed incorrectly when re-opening the bar dialog. The quotes Archive allows managing the terminal's price data - download, edit and delete bars of various timeframes.
37. Terminal: Fixed downloading history data via the quotes Archive. The error could occur if the number of characters on the current account is different from that of downloaded quotes (for example, if the current account's prices have four decimal places, while the prices downloaded from the archive have five ones).
38. Terminal: Fixed displaying hedged margin in the symbol specification dialog. Previously, values were always rounded up to the integer part:



39. Signals: Expanded information about a trading signal:
  - added information about the date of the latest trade and the number of trades for the last week.
  - added Equity chart of the trading signals source account.
  - added Distribution diagram displaying trading activity by symbols.

Free

<b>Growth:</b>	<b>2743.19%</b>
Initial Deposit:	10 000.00 USD
Profit:	255 436.97 USD
Deposits:	30 225.00 USD
Withdrawals:	212 000.00 USD
Balance:	83 661.97 USD
Equity:	83 495.97 USD

---

**Current drawdown:**

Currency:	-166.00 USD
Percents:	-0.20%

---

Abs. drawdown: 1 987.25 USD  
 Maximal: 109 290.00 USD  
 Relative: 83.43%

---

Broker: WindsorBrokers-REAL  
 Leverage: 1:100  
 Trading Mode: real

---

Started: 2013.05.13 14:09  
 Updated: 2013.11.18 16:37

Latest trade: 3 day(s)  
 Trades per week: 10.29

Author: [george papas](#)

### Trend Follower on H1

Follows trends in any direction based on Hourly and daily chart analysis and enters into positions by confirming with RSI and MACD.

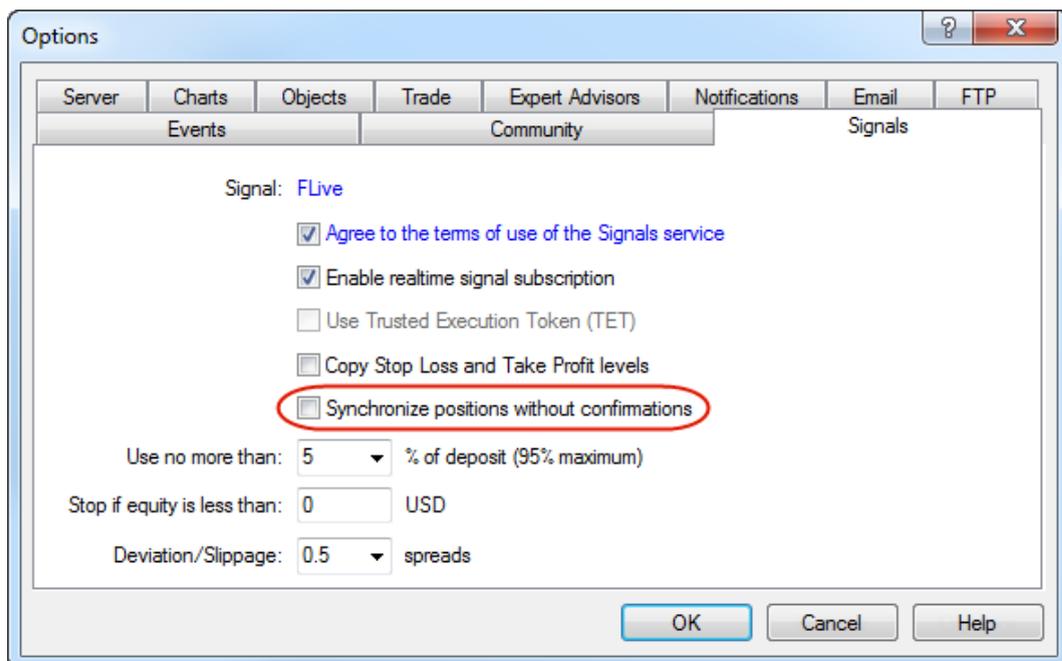
Growth
Balance
Equity
Distribution

Equity

Trades:	556	Profit Factor:	1.96	Expected Payoff:	459.42 USD
Profit Trades:	432	Gross Pr522	537.77 USD	Maximum conse47	(38 230.75 USD)
Loss Trades:	124	Gross L-267	100.80 USD	Maximum con17	(-109 290.00 USD)
Long Trades:	178	Average Pr1	209.58 USD	Maximal consec69	197.50 USD (41)
Short Trades:	378	Average L-2	154.04 USD	Maximal cons-	109 290.00 USD (17)
Best trade:	17 675.00 USD	Recovery Factor:	2.34	Monthly growth:	216.00%
Worst trad-	11 060.00 USD	Sharpe Ratio:	0.08	Annual Forecast:	2620.80%

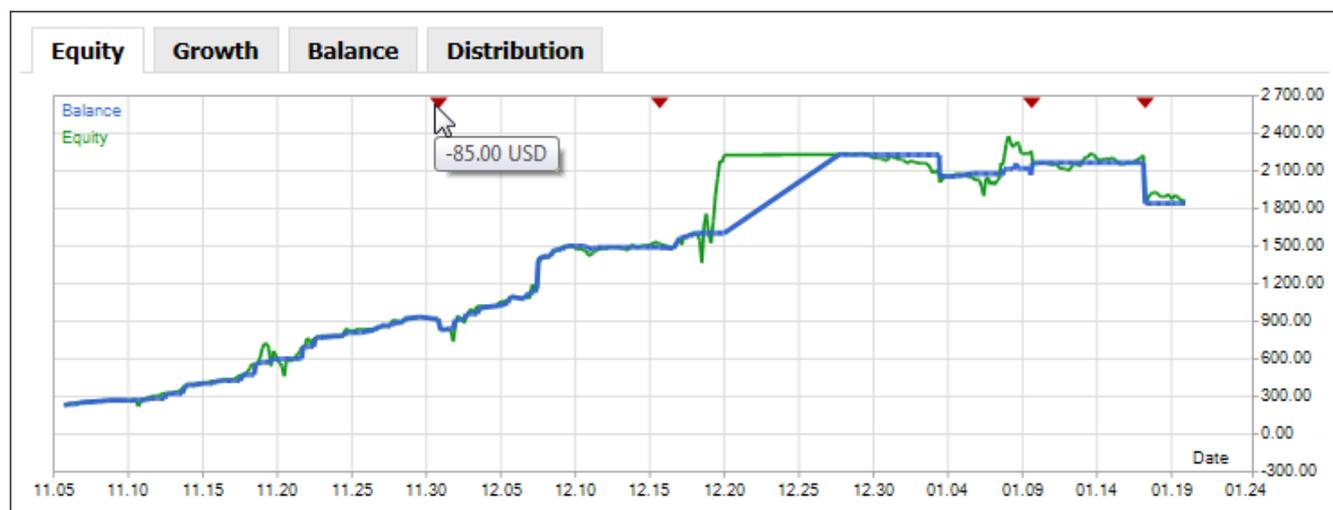
Positions
History
Reviews (8)

40. Signals: Added the option for unconditional synchronization of positions between a signal source and a subscriber's account. If enabled, synchronization of subscriber's and signal source's accounts (including closing open positions not related to the signal) is performed without subscriber's additional confirmation.



This option is necessary when using signals on VPS (Virtual Private Server). It can also be used to increase the synchronization reliability when working with the already selected signal.

41. Signals: Added legend for equity, growth, balance and distribution graphs. Also, marks displaying funds depositing and withdrawal have been added to the equity graph. When hovering the mouse cursor over the balance operation triangle, a tooltip with the operation sum is displayed:



42. Signals: Added display of the number of new trading signals submitted over the past 3 days on the signals tab.
43. Signals: Added consideration of the minimum allowed volume and minimum volume step during the initial synchronization of trading positions. If the size of the local position is less than the estimated volume at the signal's side, increase in the local position is not performed, as it would be necessary to re-open the position. This might lead to the loss of the spread.
44. Signals: Fixed opening and closing trade positions by symbols with the execution type being changed from Instant Execution to Request Execution when the threshold order volume value is exceeded.
45. Signals: When symbols of a signal source and a subscriber are compared and a number of appropriate symbols at the subscriber's side are found, permission to trade them is now considered.
46. Signals: Added History and Reviews tabs to the information about a trading signal.
47. Signals: Added signal's rating column to the signal list.
48. Signals: Removed display of tickets of signal's orders and positions from the information about a trading signal.
49. Signals: Fixed an error of trading signals synchronization that occasionally occurred in case of the large number of unidirectional positions having the same volume.
50. Signals: Added automatic conversion of trading symbols GOLD  $\Leftrightarrow$  XAUUSD and SILVER  $\Leftrightarrow$  XAGUSD.
51. Signals: Added prohibition to subscribe to trading signals for the accounts compatible with NFA FIFO regulations.
52. Signals: Fixed subscription to signals when working under Wine (Linux and Mac).
53. Signals: Removed "Show full list of signals" option from Signals tab of Symbols window. If this option is disabled, only signals provided from the accounts of the same broker are displayed in the the terminal's signal showcase. Now, all signals are displayed in the showcase, regardless of a broker. As a result, traders always have a wide selection of signals available for subscription allowing them to choose the best offers.
54. Tester: Fixed consideration of a tick spread set in the testing parameters when optimizing using a tick cache.
55. Tester: Fixed testing error that could occasionally lead to incorrect calculation of a trading deal commission. During the tests, charged commission settings are taken from the appropriate trading symbol parameters. Previously, if the commission value was charged as a percentage per annum, it was calculated incorrectly in the strategy tester.
56. MQL4: Fixed an error that occasionally prevented closing and opening trading positions from MQL4 apps. This error could occur when dealing with trading symbols of Market Execution type when sending big volume orders.
57. Fixed errors reported on the forum and in crash logs.

### MetaTrader 4 MultiTerminal

1. Fixed occasional incorrect update of the list of orders after their removal.
2. Fixed display of the current floating profit on an account - the profit value has not included accumulated swaps at open positions.
3. The accuracy of the last quote's time displayed in Market Watch and Popup Prices windows has been improved.
4. Fixed display of the news header and contents, as well as of internal e-mails using UTF-8 encoding.
5. Improved accuracy of the time displayed in the terminal's Journal - milliseconds have been added.
6. Fixed errors reported in crash logs.

### What's New in MQL4

MQL4 programming language has been completely revised and brought to the level of MQL5 - now you can develop trading robots in MQL4/5 using the unified MetaEditor development environment, single style, libraries and debugging tools.

MQL4 is popular among automated system developers due to the ease of learning and a huge amount of code generated for many years of using MetaTrader 4 terminal. However, the language also has some drawbacks arising from its main advantage - a simple programming language does not allow development of complex systems and hinders porting of debugged libraries from high-level languages. Therefore, we decided to implement in it the maximum possible amount of MQL5 language functions and features fully preserving MQL4 functionality. In

other words, all powerful MQL5 functions, including OOP and the native code compiler, will become available in MQL4.

To achieve this, we have developed a unified compiler that automatically supports both MQL4 and MQL5 languages. MetaEditor will also become a unified application both for MetaTrader 4 and MetaTrader 5 platforms. Thus, it will be possible to compile both MQL4 and MQL5 from any version. MQL5 Storage also becomes available for work.

Protection of MQL4 applications rises to MQL5 level. New EX4/EX5 files are provided with a serious and completely revised protection. This means that the Market of secure EX4 applications also becomes available to MetaTrader 4.

Besides, MQL4 now features new graphical objects and new functions for working with charts. MQL5 Standard Library is to be ported to MQL4 providing developers with new possibilities in creating their own graphical interfaces and trading libraries. Now, you can create full-fledged applications in MetaTrader 4 using the resources.

## Changes in MQL4 Language

- Added new char, short, long, uchar, ushort, uint, ulong and double data types. This will allow transferring codes from other C++ like languages. Data of various type is processed at different rates. Integer data is the fastest one to be processed. A special co-processor is used to handle the double-precision data. However, due to the complexity of the internal representation of floating-point data, it is processed slower than integer one. Typecasting has also been implemented.
- Strings are now presented in Unicode format, though they were in ANSI format (single byte ones) before. That should be considered if the program uses DLLs and passes string variables to them. When calling Windows API functions, Unicode versions of these functions should be used.
- Predefined [Volume](#) variable is now of long type. The time series for accessing the volumes also consist of long type arrays. It is recommended to use [explicit casting](#) of data having this type to the target type in old MQL4 programs to avoid type overflow error.
- Structures and classes, object pointers, [void](#) type and [this](#) key word allowing an object to receive a reference to itself have been added. All object-oriented programming standards are supported:
  - [Encapsulation and Extensibility of Types](#)
  - [Inheritance](#)
  - [Polymorphism](#)
  - [Overload](#)
  - [Virtual functions](#)
- OOP allows developing programs using classes. This facilitates debugging and development of large applications, as well as provides ability to reuse previously generated code multiple times due to inheritance. However, that does not mean that you cannot generate your MQL4 code in procedure-oriented style as before. You can develop your programs as you did before if you do not need the new features.
- `init()`, `deinit()` and `start()` predefined functions have remained for compatibility, however, `OnInit()`, `OnDeinit()`, `OnStart()`, `OnCalculate()` and `OnTick()` ones can now be used instead. Besides, new `OnTimer()`, `OnChartEvent()` and `OnTester()` predefined handler functions have been added. In the previous MQL4 versions, predefined functions could have any parameters and any return type. These functions were called by their names, not signatures. In the new MQL4, all predefined functions should exactly match their signatures. In other words, they should have a clearly defined set of parameters and a return type.
- Now, variable names cannot contain special characters and points, and new MQL4 language keywords cannot be used as names. Old MQL4 programs can be recompiled with the new compiler in order to easily correct all such errors while following the compiler's messages.
- The [Precedence rule](#) now matches C language standards. If you are unsure, you can insert parentheses in old MQL4 apps to clearly indicate the priority to increase reliability.
- Shortened conditions check is now used in logical operations, unlike the old MQL4 version where all expressions have been calculated and the check has been performed afterwards. Suppose there is a check of two conditions with the use of logical AND

```
if(condition1 && condition2)
{
    // some block of operations
}
```

If condition1 expression is **false**, calculation of condition2 expression is not performed, as **false** && **true** result is still equal to **false**.

- Changed `ArrayCopyRates()` behavior - in the old MQL4 version, this function copied price series to `double[][6]` array. Now, the array from `MqlRates` structure elements should be used in order to receive time series:

```
//Structure for storing data on prices, volumes and spread.
struct MqlRates
{
    datetime time;           // period start time
    double open;            // Open price
    double high;            // High price for the period
    double low;             // Low price for the period
    double close;           // Close price
    long tick_volume;       // tick volume
    int spread;             // spread
    long real_volume;       // exchange volume
};
```

The new format of the function also performs virtual copying. In other words, the actual copying is not performed. When the copied

values are appealed to, the price data is accessed directly.

```
int ArrayCopyRates(
    MqlRates& rates_array[], // MqlRates array passed by reference
    string symbol=NULL,     // symbol
    int timeframe=0        // timeframe
);
```

In order to maintain compatibility with old MQL4 applications, the old call format is also preserved. However, a real copying of data to double type array is now performed.

```
int ArrayCopyRates(
    void& dest_array[[]], // array passed by reference
    string symbol=NULL,  // symbol
    int timeframe=0      // timeframe
);
```

This means that the necessary data should be copied to dest\_array[[]] again when changing values in time series (adding new bars, restructuring or updating the last bar's Close price). In this case, the receiver array will be automatically distributed according to the necessary amount of the copied bars even if it has been declared statically.

- In file operations, the number of simultaneously open files can now reach 64 ones, while there could be no more than 32 ones in the old MQL4. Until recently, the files were always opened in FILE\_SHARE\_READ or FILE\_SHARE\_WRITE mode. Now, the necessary opening mode should be specified explicitly.
- Working with functions, scope of variables and memory release in local arrays has also been changed. Since the number of changes is large enough, the new **#property strict** property has been introduced to provide maximum compatibility with the previous approach to developing MQL4 programs. When creating new MQL4 application using MQL wizard, this property is always added to the template. The table below contains the differences between MQL4, new MQL4 without using strict and new MQL4 with specified strict compilation mode

```
#property strict
```

#### The table of differences between compilers

Old MQL4 compiler	New MQL4 compiler	New MQL4 with #property strict
init(), start() and deinit() entry points may have any parameters and any return type	init(), start() and deinit() left intact for the sake of compatibility and new OnInit(), OnStart(), OnCalculated(), OnTick(), OnTimer(), OnChartEvent(), OnTester() and OnDeinit() should exactly match their signatures	
Virtually any variable names (except for the reserved words) are possible, including special characters and points	Variable names cannot have special characters and points. The list of the reserved words has been expanded. Thus, such widespread words as short, long, const, etc. cannot be used as names	
Variable scope is from declaration (even in the nested block) to the function end		Variable scope is from declaration to the end of the block, in which the variable is declared
Implicit initialization of all the variables (both global and local ones) by zero		Only global variables are initialized. Only strings are initialized implicitly in local variables.
Local arrays are not released when exiting the function	Local arrays are released when exiting the function	Local arrays are released when exiting {} block
"Array out of range" does not cause a critical error *	Ditto, except for the arrays of structures and classes, for which this error is critical one	"Array out of range" is a critical error causing the program to stop
No structures and classes	Structures and classes are present. Additional data types are implemented	
Strings are single-byte. datetime is a 32-bit integer. Predefined Volume variable is of double type	Strings are unicode ones. datetime is a 64-bit integer. Predefined Volume variable is of long type	
ArrayCopyRates() performs virtual copying to double[][] array	ArrayCopyRates() performs virtual copying to MqlRates[] array. Copying to double[][][6] array has remained intact for the sake of compatibility, however, that copying is real, not virtual	
The functions may not return values even if they have a type. To do this, return(0) is automatically inserted by the compiler in the function end		Functions of any type should return a value
The number of simultaneously open files is 32	The number of simultaneously open files is 64	
The files are always opened in shared use mode **	Shared use mode should be specified explicitly using FILE_SHARE_READ and/or FILE_SHARE_WRITE	

\* Please pay special attention to "Array out of range" error - many old custom indicators will display this error in strict mode of the new

compiler when launched on the chart. It is recommended to find the cause and eliminate it.

\*\* In the new MQL4 and MQL5, FILE\_SHARE\_READ and FILE\_SHARE\_WRITE flags are responsible for the files shared use mode. There were no such flags in the old MQL4.