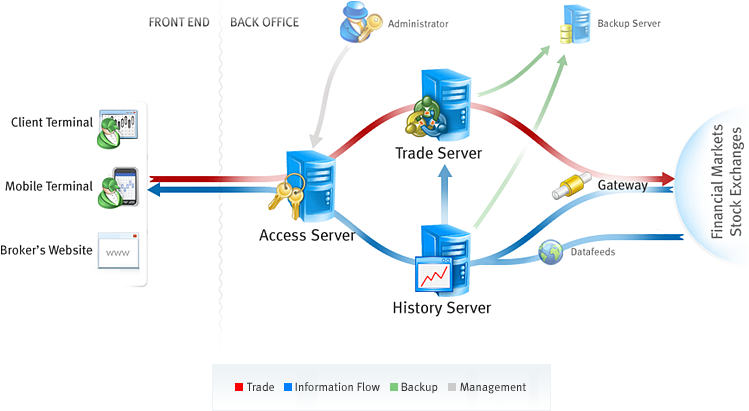
# Platform Components

The trading platform consists of the following components:

* **Main Trade Server**  
  Besides the tasks of maintaining trading, this sever is designed for managing the entire platform. It means that the administrator of the main trade server can control the entire system.
* **Trade Servers**  
  Besides the main trade server, the unlimited number of trade servers can be installed. They are intended for storing and managing all client and trade databases; trading activities are performed through them.
* **Access Server**  
  Access servers are proxy servers and firewalls of the system at the same time. They check clients' connections, collect authorization information and route clients' connections. Besides that access servers cache the largest part of all data transmitted to a client (including history data and terminal updates).
* **History Server**  
  History server receives, processes and stores price data, news and updates, and transmits them to other components of the system.
* **Backup Server**  
  Backup servers create backups of both history and trade servers.
* **Data Feeds**  
  Data feeds are special components of the platform that are implemented as separate executable files that enable the receipt of news and quotes from different providers.
* **Gateways**  
  Gateways are intended for integration of the MetaTrader 5 platform with ECNs and exchanges. Gateways allow bringing out trade operation to external systems as well as translating quotes and news from them.
* **Administrator Terminal**  
  This terminal enables the remote control of the entire MetaTrader 5 platform. It allows changing any platform settings, managing client and trade databases and perform other operation.
* **Manager Terminal**  
  This terminal is used for working with the broker's clients: managing databases, servicing trading operations, reporting and managing risks.
* **Client Terminal for Windows**  
  The trader's main workstation enabling traders to analyze quotes and to execute trading operations. It features algo trading and strategy testing facilities.
* **Mobile Terminal for iOS**  
  It enables traders to manage their trading accounts, to view symbol charts and to perform trading operations using iPhone or iPad.
* **Mobile Terminal for Android**  
  The terminal enables traders to manage their trading accounts, to view symbol charts and to perform trading operations using Android powered smartphones and tablets.
* **WebTerminal**  
  MetaTrader 5 WebTerminal allows trading in the financial markets via a web browser. It works in all operating systems and browsers requiring no additional software.
* **MetaTrader 5 API**  
  It is a toolkit which allow further expansion of the platform capabilities, integration with other trading systems and back-office components, as well as platform customization for specific business needs.

## Interaction of Components

The MetaTrader 5 trading platform is a distributed system. It is divided into separate components that are connected to each other via the main trade server. Configurations of all the platform components are stored on the main server. Other components refer to the main trade server, read the configurations and operate then in accordance with them.



The internal identification of servers is performed using IDs and passwords set during their installation. You can view or change identifiers and passwords in the "Network" section. The main trade server identifies a component that has referred to it by its ID and passes to it configurations set for this component.

|  |
| --- |
| The minimal working configuration of the platform is the installed main trade server and an access server. The access server allows the connection to the main server using terminals (client, manager or administrator terminal). |

# Trading Server

The system has two types of trading servers: one main server server and additional ones. The main trading server serves trading operations and manages the entire system configuration.

The trading server performs the following functions:

* Storing and management of clients' records.
* Authentication and authorization of client connections.
* Storing and management of trade records.
* Check, management and execution of trade requests.
* Management of the internal mailing system.

|  |
| --- |
| * One main trading server and the unlimited number of additional ones can be configured in the system. * For server configuration details, please see the "Network cluster" section. |

## Structure of Directories and Files

The trade server is installed to the *"Server"*folder. It contains the following executable files:

* **mt5srvupdater64.exe** — the executable file of the live update system of the trade server;
* **mt5Trade64.exe** — the executable file of the trade server;
* **mt5sendmail64.exe** — the executable file of an utility for sending reports via email.

The main server directory contains five folders: *archive*, *bases*, *config*, *logs*, *templates*.

The directory ***archive*** contains the database of accounts moved to the archive:

| **Files** | **Description** |
| --- | --- |
| **users\_archive\_yyyy.dat** | Databases of archive users by years. |
| **users\_archive\_yyyy.idx** | Index files of databases of archive users by years. |

The directory ***bases*** contains all databases of the trade server:

| **Files and folders** | **Description** | **Files** | **Description** |
| --- | --- | --- | --- |
| **deals\** | Database of deals. | **deals\_yyyy.mm.dat** | Databases of deals by months. |
| **deals\_yyyy.mm.idx** | Index files of databases of deals by months. |
| **history\** | Database of order history (filled, rejected, canceled and expired orders). | **history\_yyyy.mm.dat** | Databases of order history by months. |
| **history\_yyyy.mm.idx** | Index files of databases of order history by months. |
| **mail\** | Email database. | **mail\_yyyy.mm.dat** | Email databases by month. |
| **mail\_yyyy.mm.idx** | Index files of email databases by months. |
| **daily\** | The database of daily reports. | **daily\_yyyy.mm.dat** | Databases of daily reports by months. |
| **daily\_yyyy.mm.idx** | Index files of the daily report databases by months. |
| **crm\** | Client database. The client database itself, as well as the databases of documents, comments and files are stored in subdirectories. | | |
| **performance\** | Monthly trade server performance databases and data indexes, which are displayed on the Monitoring tab. | | |
| **confirmation.dat** | Database of verified phone numbers and emails. | | |
| **liveupdate.dat** | Database of live updates. | | |
| **orders.dat** | Database of active orders (placed orders). | | |
| **orders.idx** | The index file of the database of active orders. | | |
| **positions.dat** | Database of positions. | | |
| **positions.idx** | The index file of the database of positions. | | |
| **users.dat** | Database of accounts. | | |
| **daily.dat** | Database of daily reports. | | |
| **daily.idx** | Index file of the daily report database. | | |
| **certificates.dat** | Database of client certificates generated by the server for the extended authorization. The server stores only the public part of the certificate, while the private part is stored only at the client's side. | | |
| **certificates.idx** | Index file of the certificate database. | | |
| **executions.dat** | Database of trade executions. More detailed information is available in the MetaTrader 5 Gateway API user guide. | | |

The ***config*** directory contains all the server configurations in the form of encrypted \*.ini files:

| **Files** | **Description** |
| --- | --- |
| **access.ini** | Settings of access by IP addresses. |
| **common.ini** | Common server settings. |
| **feeders.ini** | Settings of data feeds. |
| **groups.ini** | Settings of groups. |
| **history\_sync.ini** | Settings of synchronization of history data. |
| **holidays.ini** | Settings of holidays. |
| **managers.ini** | Settings of managers. |
| **performance.ini** | Settings of the server performance display. |
| **requests.ini** | Settings of request routing rules. |
| **server.ini** | Individual server settings. |
| **servers.ini** | Settings of the internal network of servers. |
| **symbol\_groups.ini** | Individual settings of symbols for groups. |
| **symbols.ini** | Settings of symbols. |
| **time.ini** | Time settings. |
| **license.lic** | The license file. |

The ***logs*** directory contains the files of the server operation journal, as well as crash logs:

| **Folders and files** | **Description** |
| --- | --- |
| **Crash\crash.log.\*** | The /crash directory contains server crash files. These files are automatically sent to the software developing company for detecting reasons of the crash and eliminating them. |
| **yyyymmdd.log** | Journal files that contain all the information about events that occur on the trade server. Server logs are stored in separate files for each working day. Here yyyy — year, mm — month, dd — day. |
| **mt5srvupdater.log** | Journal files of the platform updates. |

The ***plugins*** directory contains DLL files of plugins for the trade server written using the MetaTrader 5 Server API.

The ***templates*** directory contains two folders: "Certificate", "Confirmation", "Greeting" and "Statement". They contain templates of emails for certificate confirmation, templates of welcoming letters and templates for daily and monthly reports.

| **Subfolders** | **Description** | **Files** | **Description** |
| --- | --- | --- | --- |
| **Default\** | Folder of default email templates. | **Default.html** | Templates of default emails, that are used if templates corresponding to the user's language were  not found. |
| **Users folder\** | Folder of templates of emails that can be sent to separate groups only according to their settings. | **language\_name.html** | templates of emails by languages that are sent in accordance with the language specified by the user. E.g. English.html. |

The ***reports*** directory contains the DLL files of reports created using MetaTrader 5 Report API.

The ***confirms*** directory is used to prepare HTML files of daily reports before sending them to clients.

The ***settings*** directory stores dashboard and filter settings used in manager terminals.

## Mail Templates

Several types of emails that are automatically sent to clients are implemented in the the MetaTrader 5 platform:

* Welcoming email at account opening;
* Email requesting the certificate confirmation;
* Daily report on trade activity;
* Monthly report on trade activity.

|  |
| --- |
| * All templates must be of the Unicode format (UTF-16/UCS-2 Little Endian). * In order to start using a new template, the main trade server must be restarted. |

### Welcoming Emails

Welcome messages are sent to clients through the internal mail system when an account is opened. It does not matter how the account is opened: by the trader via the client terminal or by the broker via the Manager/Administrator terminal. The welcome email contains the account number and password, as well as general information about the MetaTrader 5 platform.

Welcome email templates are located in the following directory:

* **Trader server directory\templates\greeting\default\\*.htm** — default templates used for groups;
* **Trader server directory\templates\greeting\*custom folder\*\*.htm** — templates of emails that can be sent only to separate groups with account of their settings;
* **Trader server directory\templates\greeting\preliminary\\*.htm** — templates of emails that are sent when a preliminary account is opened from the client terminal.

|  |
| --- |
| An unlimited number of folders with special templates can be create. The name of the folder with templates that will be used for a group are specified in the "Company" tab in its settings. |

### Certificate Confirmation Email

These letters are sent if the extended authorization and certificate confirmation mode is enabled for this group. Such an email contains information on how to confirm a certificate. Templates of such emails are stored in the following directory:

* **Trader server directory\templates\certificate\default\\*.htm** — default templates used for groups;
* **Trader server directory\templates\certificate\*custom folder\*\*.htm** — templates of emails that are sent only to separate groups with account of their settings.

### Phone and Email Verification

These emails\messages are sent to verify phone numbers and emails, specified during registration of demo and preliminary accounts from client terminals.

* **Trader server directory\templates\verify\_email** — email templates as HTM files.
* **Trader server directory\templates\verify\_phone** — SMS templates as text files. Do not use too long messages. If the allowed length is exceeded (depending on the provider), a message may be cropped or split into multiple SMS.

The <!--CONFIRMATION\_CODE--> macro is used for confirmation emails. This code adds the generated confirmation code to the text.

### Daily Report on Trade Activity

Daily reports on trade activity of clients are sent if the "Send daily statements by email" option is enabled in the group settings. Templates of such emails are stored in the following:

* **Trader server directory\templates\confirmation\default\\*.htm** — default templates used for groups;
* **Trader server directory\templates\confirmation\*custom folder\*\*.htm** — templates of reports that are sent only to separate groups with account of their settings.

### Monthly Report on Trade Activity

Monthly reports on trade activity of clients are sent if the "Send daily statements by email" option is enabled in the group settings. Templates of such emails are stored in the following:

* **Trader server directory\templates\statement\default\\*.htm** — default templates used for groups;
* **Trader server directory\templates\statement\*custom folder\*\*.htm** — templates of reports that are sent only to separate groups with account of their settings.

### Email templates in different languages

All template files should be named according to the language they are written in. A file name is used by the platform to determine which users the template is to be applied to. The user's language is defined in the account settings.

Languages are specified in the format standard for the family of Windows operating systems, though without the specification of regional dialect specifics. E.g. English.htm, Russian.htm and so on. The only exceptions are Chinese language templates. In case of simplified Chinese, a template should be named chinese.htm, while in case of traditional Chinese, it is named chinese\_traditional.htm.

If no template was found for the language specified in the account, the default.htm template is used. If a separate folder is specified for a group, the default template is first searched for in this folder, and if it is not found an email according to the default.htm template located in the "Default" folder is sent.

### The Format of Files

A template file can contain a plain text, HTML tags, as well as CSS design elements. To insert images, set them on your web server and specify the appropriate links in <img src="URL">. Besides, the templates can be equipped with special macros for inserting various data depending on an account an email is sent to.

* Common macros
* **<!--ACCOUNT-->**— account number.
* **<!--LOGIN-->** — account number.
* **<!--GROUP-->**— user group.
* **<!--PASSWORD-->** — account master password.
* **<!--INVESTOR-->** — account investor password.
* **<!--NAME-->**—  user first name (obsolete macro).
* **<!--USERNAME-->** — user first name (obsolete macro).
* **<!--FIRST\_NAME-->** — user first name.
* **<!--LAST\_NAME-->** — user last name.
* **<!--MIDDLE\_NAME-->** — user middle name.
* **<!--COUNTRY-->** — country of residence.
* **<!--CITY-->** — city of residence.
* **<!--STATE-->** — state (region) of residence.
* **<!--ZIPCODE-->** — zip code.
* **<!--ADDRESS-->** — residential address.
* **<!--PHONE-->** — phone number.
* **<!--EMAIL-->** — email address.
* **<!--COMMENT-->** — account comment.
* **<!--ID-->** — ID.
* **<!--STATUS-->** — residency status.
* **<!--PHONEPASS-->** — phone password.
* **<!--AGENT-->**— agent account associated with the user.
* **<!--CURRENCY-->**— account deposit currency.
* **<!--COMPANY-->**— company name from the account group settings.
* **<!--LEVERAGE-->** — recipient's current leverage.

#### Daily and monthly report macros

* **<!--DATE-->**— date the report is generated for, YYYY.MM.DD.
* **<!--TIME-->**— time the report is generated for, HH::MM:SS.
* **<!--FULLTIME-->**— date and time the report is generated for, YYYY.MM.DD HH::MM:SS.
* **<!--BALANCE-->**— balance at the time of the report generation.
* **<!--CREDIT-->**— credit funds at the time of the report generation.
* **<!--INTERESTRATE-->**— client's annual interest rate.
* **<!--COMMISSION\_DAILY-->**— amount of standard commissions charged from a client for the day the report is generated for.
* **<!--COMMISSION\_MONTHLY-->**— amount of standard commissions charged from a client for the month the monthly report is generated for (or for the current month in case of a daily report).
* **<!--AGENT\_DAILY-->**— agent commissions charged from a client for the day the report is generated for.
* **<!--AGENT\_MONTHLY-->**— agent commissions charged from client's operations for the month the monthly report is generated for (or for the current month in case of a daily report).
* **<!--PREV\_BALANCE\_DAILY-->**— client balance at the end of the previous trading day.
* **<!--PREV\_BALANCE\_MONTHLY-->**— client balance at the end of the previous month.
* **<!--PREV\_EQUITY\_DAILY-->**— client equity at the end of the previous trading day.
* **<!--PREV\_EQUITY\_MONTHLY-->**— client equity at the end of the previous month.
* **<!--PREV\_DATE-->** — the date of the last but one closure of the trading day.
* **<!--PREV\_TIME-->** — the time of the last but one closure of the trading day.
* **<!--PREV\_FULLTIME-->** — the date and time of the last but one closure of the trading day.
* **<!--MARGIN-->**— money required to cover open positions as of the end of the day/month.
* **<!--MARGIN\_FREE-->**— amount of free margin volume as of the end of the day/month.
* **<!--MARGIN\_LEVEL-->**— margin level as of the end of the day/month.
* **<!--FLOATING\_PROFIT-->**— floating profit/loss on all open positions at the time of the report generation.
* **<!--FLOATING\_STORAGE-->**— size of swaps charged for client's positions for a day, but not yet reflected in the balance.
* **<!--FLOATING\_COMMISSION-->**— floating client commission blocked on the account but not yet reflected in the balance at the time of the report generation.
* **<!--FLOATING\_EQUITY-->**— client equity volume at the time of the report generation.
* **<!--FLOATING\_PL-->**— total client's floating profit/loss at the time of the report generation. Calculated as <!--FLOATING\_PROFIT--> + <!--FLOATING\_STORAGE--> + <!--FLOATING\_COMMISSION-->.
* **<!--FLOATING\_LIABILITIES-->** — amount of client liabilities at the time of report generation. It is only used for the Exchange risk management model.
* **<!--FLOATING\_ASSETS-->** — amount of client assets at the time of report generation. It is only used for the Exchange risk management model.
* **<!--CLOSED\_PROFIT-->**— total closed profit/loss at all deals per day/month.
* **<!--CLOSED\_STORAGE-->**— size of swaps charged for a client's positions for a day/month.
* **<!--CLOSED\_DEPOSIT-->**— total funds deposited/withdrawn from the account per day/month.
* **<!--CLOSED\_CREDIT-->**—  credit funds deposited/withdrawn from the account per day/month.
* **<!--CLOSED\_CHARGE-->**— other depositions/withdrawals from the client's balance per day/month.
* **<!--CLOSED\_CORRECTION-->**— corrective balance operations performed on the client's account per day/month.
* **<!--CLOSED\_BONUS-->**— bonus funds deposited/withdrawn from the account per day/month.
* **<!--CLOSED\_COMMISSION\_INSTANT-->**— standard commissions withdrawn from a client's account per day/month instantly (during a trade).
* **<!--CLOSED\_COMMISSION\_ROUND-->**— commission by orders and positions accumulated during a day/month. Depending on the settings (specified for the group in the administrator terminal), preliminary commission calculation is performed during a day/month and the appropriate funds are blocked in the account and displayed here. Final commission calculation is performed at the end of a day/month and the appropriate sum is withdrawn from the account by the balance operation.
* **<!--CLOSED\_FEE-->**— total fees charged from the client for the day/month.
* **<!--CLOSED\_AGENT-->**— agent commissions charged from the client for the day the report is generated for.
* **<!--CLOSED\_INTEREST-->**— annual interest rate accruals per day/month the report is generated for.
* **<!--CLOSED\_DIVIDEND-->**— dividends received by the client per day/month the report is generated for.
* **<!--CLOSED\_TAX-->**— taxes charged per day/month the report is generated for.
* **<!--CLOSED\_PL-->**— total profit/loss at a client's account per day/month the report is generated for. Calculated as <!--CLOSED\_PROFIT--> + <!--CLOSED\_STORAGE--> + <!--CLOSED\_COMMISSION\_INSTANT-->.
* **<!--CLOSED\_ADDITIONAL-->**— financial result of other transactions conducted on the account for the day/month the report is generated for. These are additional charges, corrections, bonuses, agent commissions, annual interests, dividends and taxes
* **<!--CLOSED\_TOTAL-->**— total financial result of the client's account for the day/month the report is generated for. Calculated as the sum of all the above values with the <!--CLOSED\_\*--> prefix apart from <!--CLOSED\_PL--> and <!--CLOSED\_ADDITIONAL-->.
* **<!--CLOSED\_SO\_COMPENSATION-->** — the sum of balance operations connected with the negative balance compensation after Stop Out.
* **<!--CLOSED\_COST-->** — the total amount of costs for all deals for the day/month for which the report is generated. The value calculation does not depend on group settings. If a macro is included in a report template, its value will be calculated and substituted.
* **<!--PREV\_EQUITY\_DIFF\_PERC\_DAILY-->** — a change in equity as compared to the previous day value. Indicated as a percentage.
* **<!--PREV\_EQUITY\_DIFF\_PERC\_MONTHLY-->** — a change in equity as compared to the previous month value. Indicated as a percentage.

#### Macros of Trading Operations

Daily reports include blocks of trading operations performed by a client during a day/month. Each of these blocks begins with a macro corresponding to the operation type:

* **<!--MQTABLE=Closed Orders-->** — closed orders.
* **<!--MQTABLE=Closed Deals-->** — executed deals.
* **<!--MQTABLE=Positions-->** — current open positions.
* **<!--MQTABLE=Orders-->** — current open orders.

Each of these blocks ends with the **<!--MQTABLE-->** macro. Information about trading operations is displayed using macros inside the blocks. Depending on the block in which a macro is contained, it substitutes information about the appropriate operation type, i.e. order, deal or positions.

* **<!--Ticket-->**— the ticket of a trading operation.
* **<!--Type-->** — the type of a trading operation (buy, sell or a pending order).
* **<!--Size-->** — the volume of a trading operation. The initial and filled volume is additionally displayed for orders.
* **<!--Item-->** — the name of a trading instrument.
* **<!--ISIN-->** — International Securities Identification Number (ISIN).
* **<!--Price-->** — the price at which a trading operation was executed.
* **<!--SL-->** — the Stop Loss level.
* **<!--TP-->** — the Take Profit level.
* **<!--Open Time-->** — operation time.
* **<!--Close Time-->** — order or position closing time.
* **<!--State-->** — order status (filled, rejected, canceled).
* **<!--Comment-->** — a comment on the operation.
* **<!--Entry-->** — trade direction (in, out, in/out).
* **<!--Commission-->** — commission charged for the operation.
* **<!--Fee-->** — the amount of fees for the operation.
* **<!--Swap-->** — operation swap.
* **<!--Cost-->** — the amount of costs incurred when performing deals relative to the current mid-point spread cost. The value calculation does not depend on group settings. If a macro is included in a report template, its value will be calculated and substituted.
* **<!--Profit-->** — profit received from the operation.
* **<!--Market-->** — the market price of a trading instrument at the time of report generation.
* **<!--Color-->** — a macro for alternating the background color of even and odd rows (a row with a white background, the next one has a gray background, etc).

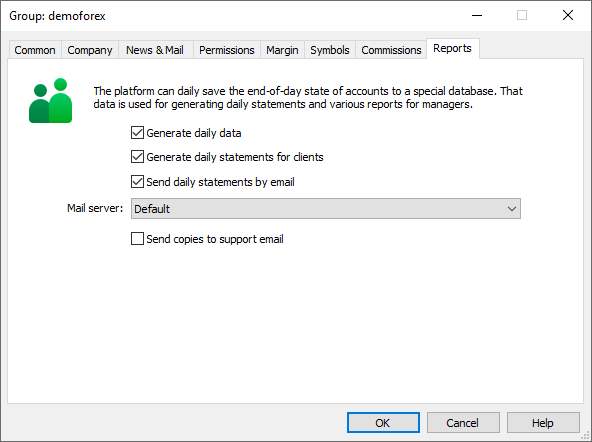
## Daily Reports

At the end of each working day and month, the trade server prepares reports on client deals and places them to [trade server directory]/confirms/YYYYMMDD", where YYYYMMDD is the current date). Each report is an HTML file having the name "login\_mail.htm" (for example, "123\_mail.htm"). After creating the reports, the trade server sends them using SendMail utility.

The database of all generated reports is saved in the file [trade server directory]/bases/daily.dat. The data from the database can be requested via the Manager terminal, for example, by using a report from Daily Report standard delivery.

### Enabling/Disabling Daily Reports

Generation of reports can be enabled/disabled for each client group separately.



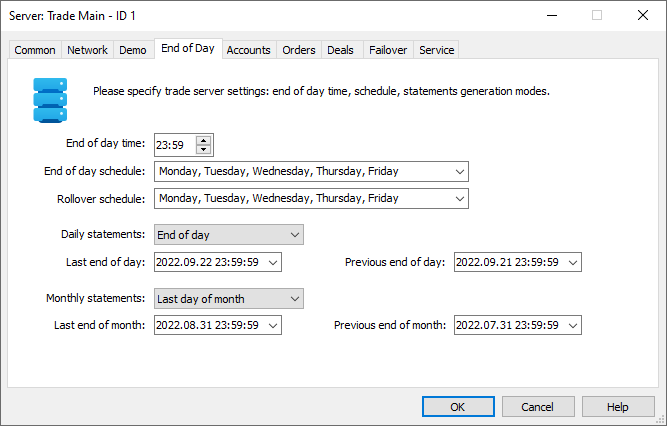
Here you can also configure sending reports to clients and (if necessary) the technical support via email.

### Customizing Report Appearance

Email templates with daily and monthly trading activity reports are located in the /templates/confirmation and /templates/statement folders of the trade server.

### Configuring Daily Report Generation Time

Daily and monthly report generation time is defined by the trade server settings:



The reports are affected by the following parameters:

* **End of day time** — report generation start time.
* **End of day schedule** — report generation days. The option does not affect monthly reports.
* **Daily statements** — daily report generation time: at the end of a trading day (before charging swaps, annual interest, and commissions) or at the start of a trading day (after charging swaps, annual interest, and commissions).
* **Monthly statements** — monthly report generation time: at the end of the last day of month (before charging swaps, annual interest, and commissions) or at the start of a first day of the next month (after charging swaps, annual interest, and commissions).

### Report Data

| **Name** | **Description** |
| --- | --- |
| Datetime | Daily report generation date and time. |
| Login | Initial key. The login of a client for whom the daily report is generated. |
| DatetimePrev | Daily report previous generation date and time. |
| Name | The name of a client in a daily report. |
| Group | Client group in a daily report. |
| Currency | Client's deposit currency in a daily report. |
| Company | The company serving the client in a daily report. |
| EMail | An email of a client in a daily report. |
| Balance | The size of a client's balance in a daily report. |
| Credit | The amount of a client's credit funds in a daily report. |
| InterestRate | The annual interest rate of a client in a daily report. |
| CommissionDaily | The amount of commissions charged from a client for a day in the report. |
| CommissionMonthly | The amount of commissions charged from a client for the current month in a report. |
| AgentDaily | The size of agent commissions charged for a client's trade operations for a day, from a daily report. |
| AgentMonthly | The amount of agent commissions charged for a client's trade operations for the current month. |
| BalancePrevDay | Client's balance as of the end of the previous day. |
| BalancePrevMonth | Client's balance as of the end of the previous trading month. |
| EquityPrevDay | A client's equity as of the end of the previous day. |
| EquityPrevMonth | The value of a client's equity as of the end of the previous trading month. |
| Margin | Size of a client's margin in a daily report. |
| MarginFree | A client's free margin in a daily report. |
| MarginLevel | The margin level of a client in the daily report. |
| MarginLeverage | The margin leverage of a client in the daily report. |
| Profit | The size of the current profit for all open positions of a client in a daily report. |
| ProfitStorage | The current size of swaps charged for a client's open positions for a day, but not yet reflected in the balance. |
| ProfitEquity | The amount of the current floating equity of a client in a daily report. |
| DailyProfit | The amount of a client's daily profit. |
| DailyBalance | The amount accrued to a client's balance during the reported day. |
| DailyCredit | The amount of credit given to a client during the reported day. |
| DailyCharge | The amount of other charges to the client's balance during the reported day. |
| DailyCorrection | The amount of corrective balance operations for a reported day. |
| DailyBonus | The amount of bonuses added to the client's balance for the reported day. |
| DailyStorage | The amount of swaps calculated for the client for the reported day. |
| DailyCommInstant | The amount of instant commissions charged from the client for a reported day. |
| DailyCommRound | The amount of turnover commissions charged from the client for a reported day. |
| DailyCommFee | The fee amount charged for the client's deals for the reported day. |
| DailyAgent | The size of agent commissions charged for a client's trade operations for the reported day. |
| DailyInterest | The amount accrued to a client as part of the annual interest rate for the reported day. |

## SendMail Utility

MetaTrader 5 SendMail is a command-line utility and a component of MetaTrader 5 platform. It is used by a trade server for sending reports and emails. This application can be used for sending reports manually.

### Report Generation

The trade server prepares reports on client deals at the end of a working day placing them to the separate directory ("*/MetaTraderServer/confirms/YYYYMMDD*", where *YYYYMMDD* is current date). Each report is an HTML file having the name *"login\_mail.htm"* (for example, *"123\_mail.htm"*). After generating all reports, the trade server saves the configuration file (*"mail.cfg"*) containing descriptions of account groups with e-mail settings in the same directory and then launches MetaTrader 5 SendMail utility.

### Operation Principles

After launching MetaTrader 5, SendMail reads the configuration file and starts sending emails to account groups. Description of each group contains data that is sufficient for authorization on the specified SMTP server. In case of an authorization error, the program simply moves to the next group sending the appropriate error message to the journal. If authorization is successful, reports are consistently sent to these group's accounts.

Two types of errors may occur when sending reports. First, connection to the SMTP server may be lost. In this case, an attempt to handle the next group is made. Second, the server may report that it is not able to send a report to the specified client address (for example, if the latter is incorrect). In this case, the report is marked as unsent and handling is moved to the next one.

After all MetaTrader 5 groups are handled, SendMail records the results in the same configuration file. Then all the reports and the configuration file are compressed into a ZIP file. ZIP file's name is based on the name of a low-level folder in the path to the reports (for example, if the path is *"C:\mt5\trade\_server\confirms\20130101"*, then ZIP file's name is *"20130101.zip"*). After the successful archiving, all source files are deleted.

|  |
| --- |
| Attention: We strongly recommend to use your own mail server (at least having the *mail.your\_company.domain* name) instead of a provider's or even a public one. That ensures sender's (your company's) authenticity and allows adjusting the mail server's parameters for efficient mailing. Currently, mail servers and providers introduce strict rules concerning mail sending and receiving. That often leads to some brokers having issues when performing mass mailing.  To simplify the task, you may configure your own mail server, so that it is able to accept any mail from the trade server's IP address without limitations. If you do not have access to your mail server's settings, ask your provider to configure it appropriately. |

### Sending Reports Manually

Reports may be sent manually in two modes: group and individual ones. To choose the mode, one of the keys should additionally be specified in the command line when launching MetaTrader 5 SendMail:

* mt5sendmail64.exe /mail:[path] — individual mailing.
* mt5sendmail64.exe /group:[path] — group mailing.
* /archive — specify this key after /mail:[path] or /group:[path] to force the SendMail archive the reports after sending.

#### Individual Mailing

MetaTrader 5 SendMail with "mail" key should be launched to send a report to a single user:

|  |
| --- |
| mt5sendmail64.exe /mail:"path" /archive |

Path is the path to the directory containing *folder.cfg* mailing configuration file (the file should have exactly the name mentioned above). Do not add a backslash at the end of the path:

|  |
| --- |
| mt5sendmail64.exe /mail: "C:\MetaTrader 5 Platform\MainTrade\confirms\2021.05.11.daily" /archive |

Here is an example of a configuration file structure:

|  |
| --- |
| from=abc@company.net name=ABC Company subject=Trade Report to=johnsmith@mail.net to\_name=John Smith charset=utf-8 body=D:\Reports\John\_Smith\mail.htm attachments=D:\Reports\John\_Smith\balance.jpg smtp\_srv=abc@company.net smtp\_login=mailer smtp\_pass=mailerpassword |

The following parameters are specified in the configuration file:

* **from** — e-mail address, from which the report is sent.
* **name** — sender's name.
* **subject** — email subject.
* **to** — recipient's email address.
* **to\_name** — recipient's name.
* **charset** — email's character set.
* **body** — path to the HTM file containing the email's contents.
* **attachments** — path to the file that is to be attached to the email. If you want to attach several files, specify paths to them divided by tab. The line length, including the parameter name, should not exceed 256 characters.
* **smtp\_srv** —  SMTP server address used for sending messages.
* **smtp\_login** — login for authorization on the mail server. In most cases, it is a mailbox, for example, *"your\_name@mail.ru"*.
* **smtp\_pass** — password for authorization on the mail server (mailbox password).

|  |
| --- |
| * Up to 8 attachments can be added to an email. * Attachment size may not exceed 4 MB. * The password for authorization on the SMTP server is stored in the clear. This is done in order to modify the configuration file easily. The administrator can change the configuration (for example, a password) and launch SendMail manually. For security purposes, it is recommended to limit access to report configuration files. |

#### Group Mailing

MetaTrader 5 SendMail with "group" key should be launched to send a report to a group of users:

|  |
| --- |
| mt5sendmail64.exe /group:"path" /archive |

Path is a path to *mail.cfg* file (the file should have exactly the name mentioned above) containing description of mailing settings.

|  |
| --- |
| Report files being sent should be in the same directory with *mail.cfg* configuration file. The report for each user should be in the form of an HTM file named *"login\_mail.htm"* (for example, *"123\_mail.htm"*). |

Here is an example of a configuration file structure:

|  |
| --- |
| <group> name=demoforex company=MetaQuotes Software Corp. email=reporter@metaquotes.ru subject=Trade Report smtp\_srv=mail.metaquotes.ru smtp\_login=reporter smtp\_pass=securepass      101   1   John Smith   jjohnsmith@mail.ru    102   1   Ivan Ivanov   ivan@mail.ru    103   2   Larisa Ivanovna   larisa@mail.ru </group>   <group> name=forever company=MetaQuotes Software Corp. email=mailer@metaquotes.ru smtp\_srv=mail.metaquotes.ru smtp\_login=mailer smtp\_pass=mailerpassword      151   0   Alisa   alisa@pole.ru    152   0   Brom   brom@fix.com    153   0   SirX   sirx@fix.com </group> |

Each target user group is described by a couple of <group> tags. Description of each group contains several mandatory fields:

* **name** — group name.
* **company** — company name that will be specified as a sender's name.
* **email** — e-mail address, on behalf of which the report is sent.
* **subject** — email subject.
* **smtp\_srv** —  SMTP server address used for sending messages.
* **smtp\_login** — login for authorization on the mail server. In most cases, it is a mailbox, for example, *"your\_name@mail.ru"*.
* **smtp\_pass** — password for authorization on the mail server (mailbox password).

|  |
| --- |
| The password for authorization on the SMTP server is stored in the clear. This is done in order to modify the configuration file easily. The administrator can change the configuration (for example, a password) and launch SendMail manually. For security purposes, it is recommended to limit access to report configuration files. |

The group's description is followed by the list of users that should receive the reports. The description and the list should be divided by an empty line used as a separator.

Each entry in the user list consists of the four fields: login, mailing status, recipient name and email. Tab character is used as a field separator. The first field may contain spaces before its contents. Let's examine a sample entry in more details:

|  |
| --- |
| 102   1   Ivan Ivanov   ivan@mail.ru |

Status ID may have the following values:

* **0** — report has not been sent or is on hold.
* **1** — report has been sent successfully.
* **2** — report delivery error, invalid recipient e-mail address.

After the reports have been sent, mailing status is updated in the same configuration file.

## Return Errors

When an attempt is made to perform invalid operations, a trade server returns errors to terminals. They are displayed on the "Journal" tab of the "Toolbox" window, as well as in journals of servers.

### Common Errors

| **Error** | **Description** |
| --- | --- |
| Common error | Error not included to any of categories listed below. |
| Invalid parameters | Incorrect parameters are specified at the attempt to change any of configurations. |
| Disk error | This error occurs when it is impossible to write information on a disk. |
| Memory error | This error indicates that there is not enough memory. |
| Network error | This error is shown when a failure occurs in the network interaction of the platform components. |
| Not enough permissions | This error occurs when a manager/administrator attempts to perform an action, for which he or she doesn't have enough permissions. |
| Operation timeout | The error means that the operation fulfillment waiting period is over. |
| No connection | Connection to the server couldn't be established. |
| Service is not available | This error can appear when information is requested from one of the platform components that is currently unavailable. |
| Too frequent requests | The error occurs if requests to the server are made to often. |
| Not found | Requested information was not found. |

### Authorization Errors

| **Error** | **Description** |
| --- | --- |
| Invalid terminal type | This error can occur at the attempt to connect using a manager group account via the client terminal or vice versa - connect from manager/administrator terminal using an account from a common group. |
| Invalid account | This error appears if invalid account number or incorrect password are specified during authorization. |
| Unknown account | This error appears if a user tries to authorize using an account that is created on a trade server, connection to which is prohibited for the access server. Due to it, the access server cannot identify the account and denies its connection. |
| Account disabled | The error appears at the attempt to connect using a disabled account. |
| Advanced authorization | This message indicates that the procedure of extended authorization must be performed for connecting. |
| Certificate required | This message is shown at the attempt to to connect using the account that requires a SSL certificate to be generated in order to connect. |
| Invalid certificate | This message appears at the attempt to connect using a certificate that is invalid for this account. For example, if it was reset on the server. |
| Certificate is not confirmed | The message is shown at the attempt to connect using an unconfirmed certificate. |
| Attempt to connect to non-access server | This error means that an attempt is made to connect to one of the platform components directly, bypassing the access server. |
| Invalid or fake server | The error indicates that the server, to which there was an attempt to connect, is invalid. |
| Only updates available | This error is shown when an attempt is made to connect to the history server at the time when only connecting to it for obtaining updates is allowed. This situation can appear during the platform update, when the server has been updated and is operating, but its configurations are not yet synchronized with the main trade server. |
| Old version | It indicates that the version of the component is old. |
| Account doesn't have manager config | The error occurs at the attempt to connect using the account for which no manager configurations have been created. |
| IP address unallowed for manager | The error occurs at the attempt of a manager connection from an unallowed IP address. |
| Group is not initialized | The error occurs at the attempt to create an account in the group that hasn't been initialized yet. In order to initialize a group after it has been created, the trade server needs to be restarted. |
| Certificate generation disabled | The error occurs if the terminal requests generation of a new SSL certificate, but the possibility of their automatic generation is disabled on the server. |

### Configuration Management Errors

| **Error** | **Description** |
| --- | --- |
| Last admin config deleting | The error appears at the attempt to delete the last manager configuration. |
| Last admin group cannot be deleted | The error appears at the attempt to delete the last administrator group. |
| Accounts or trades in group | The error appears at the attempt to delete a group that contains accounts or trade operations. |
| Invalid accounts or trades ranges | The error appears at the attempt to set the range of accounts, orders or deals for a trade server that coincide with the same ranges on other servers. |
| Account is not from manager group | The error occurs at the attempt to create a manager configuration based on the account that does not belong to the manager group. |
| Built-in protected config | The error occurs at the attempt to delete a built-in configuration of collection of the server performance parameters. |
| Configuration duplicate | This error is shown when an attempt is made to add the second main trade server or a history server in the "Network" section. |
| Configuration limit reached | This error appears at the attempt to create a new configuration, when the limit to their creation is reached. This can happen if the license was not activated. In this case the number of groups, for example, is limited to five. |
| Invalid network configuration | This error appears at the attempt to save such network settings that the administrator will not be able no connect to the main trade server. |

### Account Management Errors

| **Error** | **Description** |
| --- | --- |
| Last admin account deleting | This error appears at the attempt to delete the last manager account. |
| Logins range exhausted | The error appears at the attempt to create an account when the range of allowed accounts was exhausted. |
| Login reserved at another server | This error appears at the attempt to create an account via a manager or administrator terminal with the login that already exists on another server. |
| Account already exists | The error appears at the attempt to create an account with already existing login. |
| Attempt of self-deletion | The error appears if an administrator is trying to delete his or her own account via the administrator terminal. |
| Invalid account password | The error appears if incorrect password was specified during password verification. |
| Users limit reached | The error appears at the attempt to create a new account when their limit is over. The situation is possible when the license was not activated. |
| Account has open trades | The error appears at the attempt to delete an account with open positions. |
| Attempt to move account to different server | The error appears at the attempt to move an account to the group that belongs to another trade server. |
| Attempt to move account to different currency group | The error appears at the attempt to move an account to the group with a different deposit currency. |

### State of Trade Requests

| **Error** | **Description** |
| --- | --- |
| Request on the way | The request has been sent to a server but it hasn't accepted it yet. |
| Request accepted | The request has been accepted by the server and is waiting to be processed. |
| Request processed | The server has started to process the request. |
| Requote | Requoting as a reply to a client's request to execute a trade operation. |
| Prices | Sending prices. |
| Request rejected | The request is rejected by the server or dealer. |
| Request canceled | the request was canceled by the server, dealer or client. |
| Order placed | The order has been placed and is waiting till the specified processing conditions appear. |
| Request executed | The requested operation has been executed. |
| Request executed partly | The requested operation has been executed partially. E.g. a deal on the part of volume specified in the order is executed. |
| Request error | Error of request execution. |
| Request timeout | Time of request execution waiting is over. |
| Invalid request | The request didn't pass the common validation procedure. |
| Invalid volume | Incorrect volume is specified in the request. |
| Invalid price | Incorrect price is specified in the request. |
| Invalid stops | Incorrect Take Profit and Stop Loss levels are specified in the request. |
| Trade disabled | Trading is disabled for this account. |
| Market closed | The error appears at the attempt to execute trade operations beyond the allowed period (both common trading hours and trade sessions of separate symbols). |
| No money | Attempt to execute a trade operations, when there is not enough money for its execution. |
| Price changed | The error appears when the maximal deviation of the current price at the request price is exceeded. |
| No prices | The error appears when there is no thread of quotes. |
| Invalid expiration | The error occurs at the attempt to place an order with an incorrect expiration date. |
| Order has been changed already | The error appears at the attempt to modify an order, which has been simultaneously modified. |
| Too many trade requests | The error appears if trade requests are sent too often to the server. |
| AutoTrading disabled by server | This error indicates that trading of Expert Advisors on this account is prohibited. |
| AutoTrading disabled by client | This error indicates that trading of Expert Advisors is disabled in the client terminal. |

# Access Server

Access servers are proxy servers and the platform firewalls at the same time. They perform the following functions:

* Processing of incoming client connections.
* Packing authorization requests and sending them to the trade server.
* Checking activity of client connections protecting the trade server from attacks and overload (antiflood control).
* Saving history data, depth of market and news, and translate them to clients, thus reducing the load to the history server.
* Cashing and providing Live Update to terminals.
* Monitoring the operation of the history and trade servers.

The unlimited number of access servers can exist for each trade server. Terminals are switched between them automatically, depending on the priority settings.

|  |
| --- |
| For server configuration details, see the "Network cluster" section. |

## Structure of Directories and Files

The access server is installed to folder "*access\_server*". This folder contains the following executable files:

* **mt5srvupdater64.exe** — the executable file of the live update system of the access server;
* **mt5access64.exe** — the executable file of the access server.

The main directory of the access server contains five folders: *bases*, *config*, *history*, *liveupdate*, *logs*.

The ***bases*** directory contains the news databases, as well as data on the server performance.

| **Files** | **Description** |
| --- | --- |
| **performance\** | Monthly access server performance databases and data indexes, which are displayed on the Monitoring tab. |
| **news.dat** | Database of news sent to clients. |
| **news.idx** | The index file of the news database. |
| **performance.dat** | Data about the access server performance that are displayed on the "Monitor" tab are written to this file. |

The ***config*** directory contains configuration files:

| **Files** | **Description** |
| --- | --- |
| **server.ini** | Individual settings of the access server. |
| **servers.ini** | Settings of the internal network of servers. |
| **symbols.ini** | Configurations of symbols. |
| **time.ini** | Time settings. |

The ***history*** directory contains the base of history data by symbols, which was received from the history server:

| **Files** | **Description** |
| --- | --- |
| **yyyy.hsc** | Minute data for the symbol, divided by years. |
| **yyyymm.tkc** | Tick data for the symbol, divided by months. |
| **tickers.dat** | Data by tickers. |

The ***liveupdate*** directory contains the latest updates of the client, manager and administrator terminals:

| **Files** | **Description** |
| --- | --- |
| **mt5adm.build** | Live update of the administrator terminal. The build number is specified after the point. |
| **mt5clw.build** | Live update of the client terminal. |
| **mt5ckwide.build** | Live Update of MetaEditor. |
| **mt5clwmql.build** | Live Update of the MQL5 compiler. |
| **mt5man.build** | Live update of the manager terminal. |

The ***logs*** directory keeps files of the access server operation journal, as well crash logs:

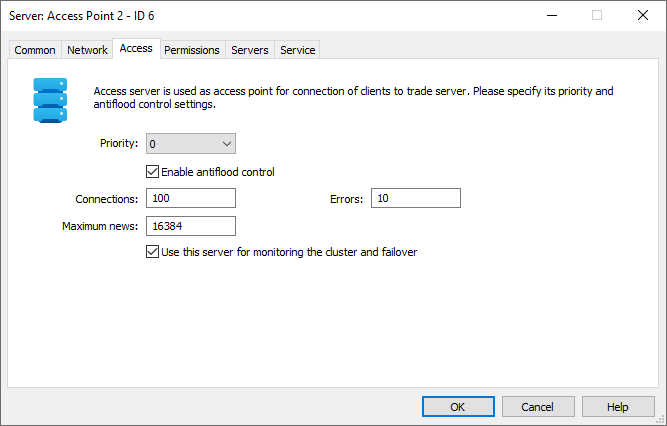
| **Files and folders** | **Description** |
| --- | --- |
| **Crash\crash.log.\*** | The /crash directory contains server crash files. These files are automatically sent to the software developing company for detecting reasons of the crash and eliminating them. |
| **yyyymmdd.log** | Journal files that contain all the information about events that occur on the access server. Server logs are stored in separate files for each working day. Here yyyy — year, mm — month, dd — day. |
| **mt5srvupdater.log** | Journal files of the platform updates. |

## Antiflood Control

The antiflood control system works on access servers. It allows protecting the trading platform from external harmful attacks. This protection system collects the database of users who send incorrect requests to the server (e.g. attempt to authorize with an incorrect login or password), as well as users who send too often requests.

|  |
| --- |
| Antiflood control works with all types of connections to the server including manager ones (via the manager terminal and Manager API). |

The number of connections and incorrect request per time unit can be set up on the "Access" tab of the access server:



### System of Operation

The antiflood control system monitors user activity. Users are identified by their IP addresses as well as by the ID linked to their computers and operating systems. The following two activity types are controlled:

* **The total number of connections from one user**  
  The system tracks if a user creates too many connections. When a user connects to the server, their connection counter is incremented by one. If the next connection occurs in less than 3 seconds, the counter is incremented. If the specified time interval is exceeded, the counter is reset. When the counter reaches the number specified in the "Connections" field, the user is blocked for 5 minutes. The blocking period increases if the connections limit is reached again. The maximum blocking period is one hour.
* **Number of invalid packets from one user**  
  The system blocks brute-force attacks by analyzing authentication errors, which imply multiple login attempts with incorrect data. Also, the system detects garbage flood packets, which can be sent to the server by third-party utilities in an effort to reduce the server performance (i.e. a DoS attack). When a user sends an invalid packet to the server, the user's error counter is incremented by one. If the next invalid packet is sent in less than 5 minutes, the counter is incremented. If the specified time interval is exceeded, the counter is reset. When the counter reaches the number specified in the "Errors" field, the user is blocked for 5 minutes. The blocking period increases if the connections limit is reached again. The maximum blocking period is one hour.

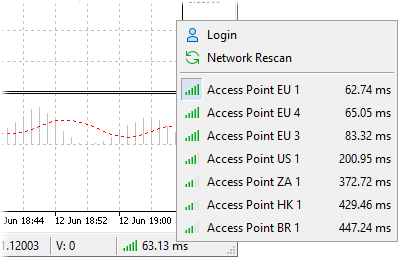
The following records appear in the access server operation Journal when a user is blocked:

* *IP* is blocked after *N* connections [intruder] — a user is blocked by IP if the number of connections is exceeded;
* *CID* is blocked after *N* errors [intruder] — a user is blocked by CID (computer ID) if the number of errors is exceeded.

|  |
| --- |
| * It is strongly recommended to keep the antiflood control system enabled. * IP addresses added to the "always allowed" list in the corresponding section, are not checked by the antiflood control system. |

## Priority

The preference of an access server for client terminals to connect to a trade server is defined by its priority and connection quality. The lower the value if priority is, the more preferable the access server is. A base priority (from 0 to 15) can be specified in its settings. It defines the server preference if all other conditions are equal. The final analysis of an access server is conducted upon the ping and the current priority, which depends on the basic priority and the number of current connections. Also, the quality of connection to the server is shown in the client, manager and administrator terminals based on the same data:



The current priority is calculated according to the following formula: *Current Priority = (Base Priority + Connections / 200)*,

where:

* **Current Priority** is the priority at the server current moment;
* **Base Priority** is the base priority set in its parameters;
* **Connections** — the number of current connections.

Every 200 client connections increase the current priority of a server by one. The value of the current priority of access servers is available on the "Network" tab.

# History Server

The history server processes price and news data. This server performs the following functions:

* Receiving and filtering price and news data from gateways and datafeeds.
* Packing price and news data.
* Storing and providing price history in the form of 1-minute bars and ticks to other components of the platform.
* Storing and providing the news thread.
* Receiving, checking and distributing Live Updates among the MetaTrader 5 platform components.

## Structure of Directories and Files

The history server is installed to folder "*history\_server*". It contains the following executable files:

* **mt5srvupdater64.exe** — the executable file of the live update system of the history server. This component has a number of console commands;
* **mt5history64.exe** — the executable file of the history server.

The main directory of the history server contains the following folders: bases, config, datafeed, gateway, history, liveupdate, logs, plugins.

The ***bases*** directory contains different data bases:

| **Files and folders** | **Description** | **Files and folders** | **Description** |
| --- | --- | --- | --- |
| **ecn\** | ECN data directory. | executions\ | Bases and indexes of ECN trade executions by trade servers. |
| history\ | Bases and indexes related to the history of client order execution in ECN, by months. |
| symbols\{Symbol}\matching.dat | Symbol databases related to client orders placed in ECN for matching. |
| symbols\{Symbol}\filling\_items.dat | Symbol databases of matching operations processed in ECN. |
| symbols\{Symbol}\books\yyyymmdd.book | ECN Market Depth journals by days. |
| filling\_orders.dat | Databases of internal ECN order which are executed on gateways. |
| **performance\** | Monthly history server performance databases and data indexes, which are displayed on the Monitoring tab. | | |
| **news.dat** | News data base. | | |
| **news.idx** | The index file of the news database. | | |

The ***config*** directory contains different configurations as \*.ini files:

| **Files** | **Description** |
| --- | --- |
| **common.ini** | Common History Server settings. |
| **ecn\_symbols.ini** | ECN symbol settings. |
| **mt5srvupdater.ini** | Update settings. |
| **history\_sync.ini** | Settings of history data synchronization. |
| **server.ini** | Individual settings of the access server. |
| **servers.ini** | Settings of the internal network of servers. |
| **symbol\_groups.ini** | Individual settings of symbols for groups. |
| **symbols.ini** | Symbol settings. |
| **time.ini** | Time settings. |

The ***datafeed*** directory contains files for working with data feeds:

| **Files** | **Description** |
| --- | --- |
| **[datafeed\_name]\logs\yyyymmdd.log** | Journal files in which records regarding data feed operation are stored. For each data feed which has been added through the relevant section of the Administrator terminal, a separate journal file is created. The file name is set in accordance with the data feed name. |
| **[datafeed\_name]\\*.dat** | Data files with data feed settings. |
| **\*.exe** | Data feed executables. It is not allowed to have several datafeed executable files with the same name in the history server directory. If you place several identical files in different subdirectories, this may lead to conflicts in the operation and display of modules in the Administrator terminal. |
| **MT5APIGateway.dll, MT5APIGateway64.dll** | Libraries for data feed operation. |

The ***gateway*** directory contains files for working with gateways:

| **Files and folders** | **Description** |
| --- | --- |
| **[gateway\_name]\[gateway configuration name]\logs\yyyymmdd.log** | Journal files in which gateway operation logs are stored. For each data feed which has been added through the relevant section of the Administrator terminal, a separate journal file is created. The file name is set in accordance with the data feed name. |
| **[gateway\_name]\[gateway configuration name]\\*.dat** | Data files with gateway settings. |
| **\*.exe** | Gateway executables. It is not allowed to have several datafeed executable files with the same name in the history server directory. If you place several identical files in different subdirectories, this may lead to conflicts in the operation and display of modules in the Administrator terminal. |
| **MT5APIGateway.dll, MT5APIGateway64.dll** | Libraries for gateway operation. |

The ***history*** folder contains history data divided by symbols:

| **Files** | **Description** |
| --- | --- |
| **yyyy.hsc** | History data on a symbol, divided by years. |
| **yyyymm.tkc** | Tick data on a symbol, divided by months. |

The ***liveupdate*** folder contains the latest updates of all the platform components:

| **Files** | **Description** |
| --- | --- |
| **mt5adm.build** | Live update of the administrator terminal. The build number is specified after the point. |
| **mt5as.build** | Live update of the access server. |
| **mt5bs.build** | Live update of the backup server. |
| **mt5clw.build** | Live update of the client server. |
| **mt5clwide.build** | Live Update of MetaEditor. |
| **mt5clwmql.build** | Live Update of the MQL5 compiler. |
| **mt5hs.build** | Live update of the history server. |
| **mt5hsu.build** | Live update of the update system of the history server. |
| **mt5man.build** | Live update of the manager server. |
| **mt5ts.build** | Live update of the trade server. |

The ***logs*** folder contains log files of the history server operation, as well as crash logs:

| **Files and folders** | **Description** |
| --- | --- |
| **Crash\crash.log.\*** | The /crash directory contains server crash files. These files are automatically sent to the software developing company for detecting reasons of the crash and eliminating them. |
| **yyyymmdd.log** | Journal files that contain all the information about events that occur on the history server. Server logs are stored in separate files for each working day. Here yyyy — year, mm — month, dd — day. |
| **mt5srvupdater.log** | Journal files of the platform updates. |

## Interaction with Quote Providers

In MetaTrader 5, gateways and data feeds can be used as providers of quotes. Their interaction with the history server is identical. This interaction can be analyzed in terms of the physical connection and at the level of quotes streaming.

### Physical Connection

The physical connection must be established for each source of quotes enabled in the appropriate settings of the platform. The physical connection details can be configured on the "Timeouts" tab of gateways and data feeds. Let's consider the following configuration example:

* Interval between reconnections = 5 seconds.
* Number of reconnection attempts = 10.
* Interval between series of reconnections = 60 seconds.

If a gateway/data feed loses connection with an external server, a reconnection attempt is made in 5 seconds. If it fails, another one is made in 5 seconds. The total number of attempts is 10. If unable to reconnect, a series of attempts is repeated after a pause of 60 seconds.

### Stream of Quotes

A stream of prices for several symbols goes through each physical connection to a quote provider.

At each point of time, the history server accepts the stream of prices for a certain symbol only from one quote provider, while other price streams of the same symbol are ignored. The source selected for the stream of prices for a symbol is considered a current (active) source for this symbol.

During operation the active source for an instrument may change. It is changed in accordance with priority settings. The priority of data feeds and gateways is determined by their position in the list.

|  |
| --- |
| The priority of gateways, if they are used as a source of quotes, is always higher than that of data feeds. |

The stream of prices switches to the source with a higher priority as soon as the first quotes for a symbol is received from that source.

It switches to the source with a lower priority by a timeout. In case no quotes are received from the active quote source during a certain time period (it is specified in the "Datafeeds timeout" parameter in history server settings), then the history server switches to a source with the lower priority that provides quotes for the same symbol.

|  |
| --- |
| The time to wait for a quote for a symbol is defined in the "Datafeeds timeout" parameter in history server settings. |

After a new quotes source is selected, it is considered active. All cases of server switching to streams from other sources are reflected in the journal in the form of the following entry:

|  |
| --- |
| 2011.03.10 10:11:05    Ticks    datafeed 4:  CHFJPY activation |

Here the entry means that for CHFJPY, a stream of quotes from the fourth data feed (a position in the list of data feeds at the moment the entry is made) is selected.

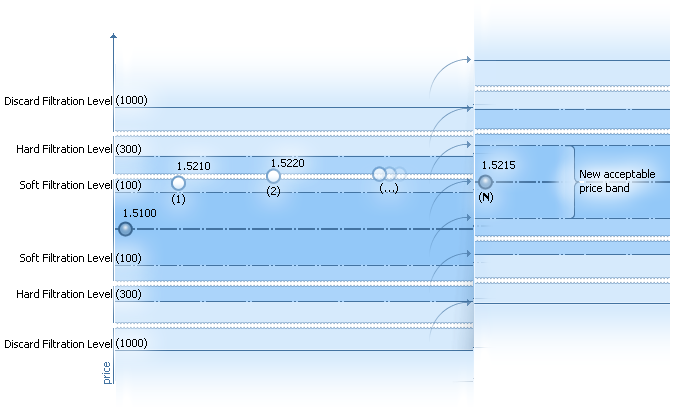
## Quotes Filtration

The filtration system is intended for controlling the correctness of quotes on financial symbols from data feeds. Filters can be set upon the "Quotes" tab of each symbol.

|  |
| --- |
| * Filters cannot be applied to instruments with the enabled Depth of Market **and** with the one of the following properties: Exchange calculation type (begins with "Exchange") **or** Last price based charting mode. Also, filtering is not applied to splice symbols. * To completely disable filtering, set 0 for all levels: mild, strict and exclusive. * The platform automatically filters out negative and zero Bid and Ask prices of OTC instruments, regardless of whether their Market Depth is enabled or not. OTC (over-the-counter or off-exchange) symbols include financial instruments whose calculation type does not start with "Exchange". * Analysis and filtration are performed separately for Bid and Ask. * Filters are not applied to the first tick after a break in the quotes stream: after platform restart, after a break in quoting sessions, after off-hours and after holidays. Filtration cannot be applied because it is not known in advance what was the price preceding the break. This rule does not apply to other price checks, including allowable spread. |

Three levels of quote filtration are available:

* **Soft filtration** — the soft filtration level is the first border of the channel of allowed symbol prices. If a new price (Bid or Ask) differs from the previous one by more than the specified value (in points), it is deleted from the thread translated to clients. However, if such a price difference appears again the number of times specified in the "Filter" field, the new price level is accepted, and the filtration level is shifted by the specified value. Such quotes will be translated again.
* **Hard filtration**— the hard filtration level is the second border of the price channel. If a received quote exceeds the level both of the soft and of the hard filtration, it is cut out from the thread translated to clients. For a new level of accepted price to be set, the quotes must be repeat the number of times specified in both levels;
* **Discard filtration level** — if the difference between prices of the previous and new quote exceed the specified value, such new prices are definitely removed from the thread.



| **Channel of allowed prices** | **— channel of allowed prices** | **Previous quote** | **— previous quote** |
| --- | --- | --- | --- |
| Soft filtration zone | — soft filtration zone | New quote | — new quote |
| Hard filtration zone | — hard filtration zone | **N** | — number of quotes (set in the "Filter" filed) required to move to a new level |
| Discard filtration zone | — discard filtration zone |  |

It is considered that there is a certain acceptable channel of price data fluctuation. This channel is limited by the set level of soft filtration. If the Bid or Ask price of the newly received quote differs from appropriate prices of the previous quote by the value of the specified soft filtration level, it is discarded. If several following quotes (the number is specified in the "Filter" parameter) also exceed the soft filtration level, the new price channel is set.

| **Example:** |
| --- |
| The last EURUSD quote was 1.50213/1.50231 (Bid/Ask). The soft filtration level is equal to 150, "Filter" parameter is equal to 3. The following quotes are received: 1.50373/1.50391, 1.50370/1.50388, 1.50372/1.50390, 1.50374/1.50392. In this case the first three quotes will be filtered away, because they exceed the previous one by more than 150 points. The last quote will be let in, and the new filtration level will be set 1.50374/1.50392 ± 150. |

The hard filtration level is an additional way to protect from incorrect quotes. If a new quote differs from the previous one by the value that is higher than the specified hard filtration level, it will be filtered away. The additional hard filtration protection is a more complicated mechanism of setting the new price channel. To confirm the new level, first the soft filter (specified number of quotes that exceed this value) must be passed, and then the hard one.

The discard filtration implies the unconditional filtering away of quotes that differ by this value. Such prices are deliberately considered incorrect.

### Filtration of Similar Quotes

The trading platform filters similar quotes received from data sources. If the platform receives the same quote as the previous one within a minute, it skips the quote. If the time interval is greater than one minute, the platform accepts the quote. The same quote is also accepted if the minute has changed. It allows plotting the charts correctly on a low liquidity market.

Example:

* 13:01:15 a quote is received
* 13:01:32 the same quote is received, it is not accepted
* 13:01:50 the same quote is received, it is not accepted
* 13:02:01 the same quote is received, it is accepted (the interval between quotes is less than 60 seconds, but the minute has changed)

### Spread Control

The Minimum Spread and Maximum Spread parameters in the Quotes tab are provided for additional protection. If the difference between the Bid and Ask prices in the incoming quote does not fall within the specified values, such a quote is removed from the stream.

The check only applies to over-the-counter (OTC) financial instruments with the floating spread. OTC instruments have one of the following calculation types: Forex, Futures, CFD, CFD Leverage or CFD Index.

# Backup Server

Servers of this type are used for creating backup copies of data that will be used in case of the trade or history server failure. They perform the following functions:

* They provide the real-time backuping of the trade server and the history server. Each server is associated with one or more separate backup server instances which can replace it at any time.
* They create backups of all databases every day, besides they perform regular backups of client and trade bases.
* When backuping a history server they create real time backups of data required for correct restoring of gateways: custom settings that can be stored (at developer's discretion) in the settings.dat file in a gateway work folder, and trade executions database.
* They provide automatic failover in case the primary server becomes unavailable.
* Using backup servers, you can easily migrate servers to new hardware.
* Backup servers enable you to quickly restore server operation in the manual mode even if the main trade server is unavailable, and connection to the platform via MetaTrader 5 Administrator is not possible.
* Replicate information to database managed by MySQL, MariaDB, PostgreSQL, Firebird, MSSQL or Oracle.

|  |
| --- |
| For server configuration details, please see the "Network cluster" section. |

## Backup Features

All critical trade and history server data is backed up in real time. Some non-critical trade server data, as well as history server data are backed up every hour. Critical and non-critical data are displayed in the table below:

| **Data** | **Trade server** | **History server** |
| --- | --- | --- |
| Critical (real time backup) | User base  Order base  Deal base  Configuration databases | Trade execution base (executions) and custom settings (settings.dat) of running gateways |
| Non-critical (backup every hour) | mt5sendmail64.exe and mt5trade64.exe files  Plugins (DLL, INI, DAT files including subdirectories)  Reports (DLL, INI files without subdirectories)  Templates (HTML, HTM files without subdirectories) | mt5history(64).exe file  Gateways and data feeds (EXE, DLL, INI, DAT, JAR, CMD, BAT, PS1,VBS and PY files including subdirectories)  Plugins (DLL, INI, DAT files including subdirectories) |
| User directories and files specified in the Folders section in backup server settings | |
| Non-critical (backup every five hours) | Archive databases  Mail database  Daily report database | Minute bar history |
| Non-critical (backup every 24 hours) |  | Tick history |

The installation directory of the backup server contains the same folders as the server, whose backups it creates, except for the *logs* folder that contains the journal entries and crash logs of the backup server itself. In addition, the following executable files are available in the directory:

* **mt5srvupdater64.exe** — the executable file of the live update system of the backup server;
* **mt5archive.exe** — the file archiver used to pack and unpack backup databases;
* **mt5backup64.exe** — the executable file of the backup server.

### Periodic file backup

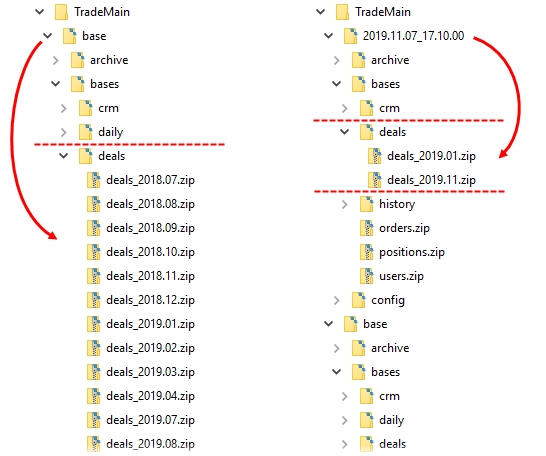
Apart from synchronizing with the main server in real time, the backup server creates database file copies on the disk for certain points in time. The path for saving files and the creation time are set in the backup server settings.

To save time and disk space, backups are performed incrementally:

* The first copy is made for all data of the backed up server. That copy is placed to ["Backups path"]\base\. The folder and file structure there fully matches the one of the backed up server.
* Each subsequent day (hour of 4 hours, depending on the "Additional backups"), an incremental copy of the data is created. Such a copy contains only the changes as compared to the first (base) copy. Such copies are placed to ["Backups path"]\[YYYY.MM.DD\_HH.MM.SS]\.

Thus, the server does not need to copy all the data during each backup.

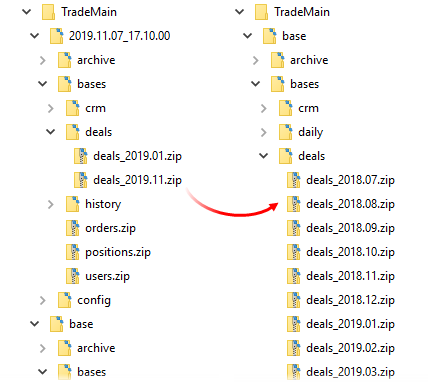
Let us consider an example with a trade database. Backup was enabled on 2019.11.06. The server created a database copy on this day. It is shown on the left in the figure below. The entire history of deals is contained under the 'base' directory. It is divided by months and is presented as ZIP files. During the next day, traders performed deals through the platform, as well as the administrator edited a trade which was executed in January 2019. The next day, the backup server created an incremental copy in the catalog 2019.11.07\_17.10.00 (on the right). You can see that only two ZIP archives are included in the incremental copy directory, i.e. those which have changed relative to the base copy:



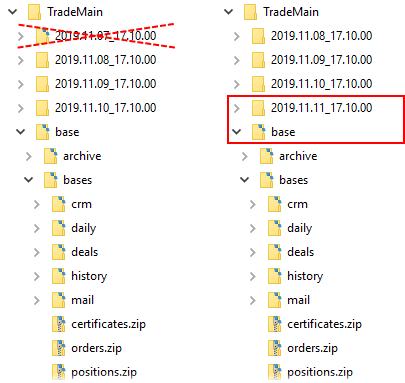
The server is able to automatically delete old copies to ensure that the backups do not take up all the free disk space over time. The storage depth is set by the "Keep backups" parameter in the backup server settings. When deleting an old backup copy, it is preliminarily applied to the main one. In fact, its files are combined with the files from the "base" directory. After that, the "base" directory will have the same status as the databases on the server at the time of creation of the deleted incremental copy. Subsequent incremental copies will already be created relative to the new base copy. This approach also reduces the amount of data copied during backup.

Let us consider this procedure using the previous example. Suppose, the "Keep backups" parameter is set to "3 days".

* The backup was enabled on 2019.11.06 and the server created a backup copy ("base" folder). the following three days, the server was creating incremental copies in separated directories.
* On 2019.11.11, when it is time to create the next copy, the server copies data from the incremental archive of 2019.11.07 into the base copy.



Then, the incremental archive of 2019.11.07 is deleted, while the incremental archive dated 2019.11.11 is created based on the updated base copy:



|  |
| --- |
| In addition to backup copies, the server saves the "mt5backupinfo.txt" file under the directory [backup server installation directory]\base\. The file contains backup service information. Do not delete or modify this file. The file is not required for database recovery, so there is no need to copy it. |

The "Enable backups" parameter allows you to completely disable creating backup copies on the disk. The backup server status synchronization with the main one is not disabled. The backup server still remains its full copy. Disabling creation of copies on the disk may be needed if the backup server is used only to export data to SQL.

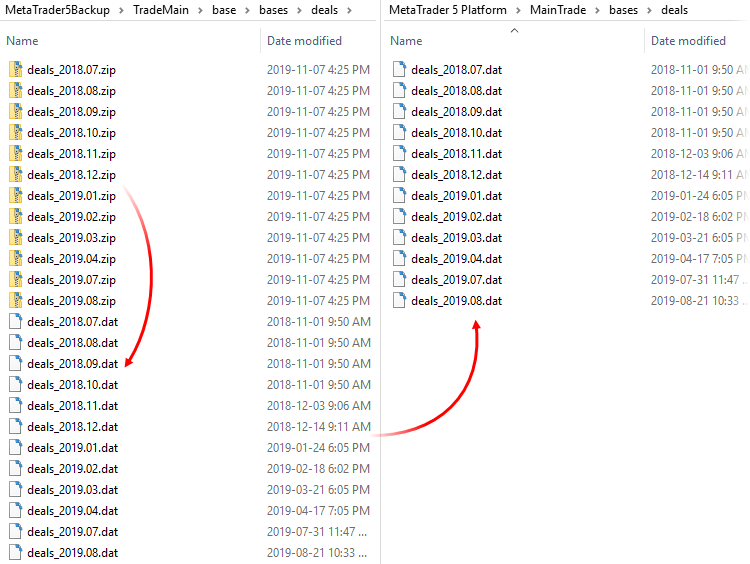
#### Working with file copies

If you want to save the database status as of a certain point in time (for example, in order to move it to the long-term archive on another server), simply copy the basic and incremental copies for the desired date. Together, they represent the complete database status of the backed up server as of the certain point in time.

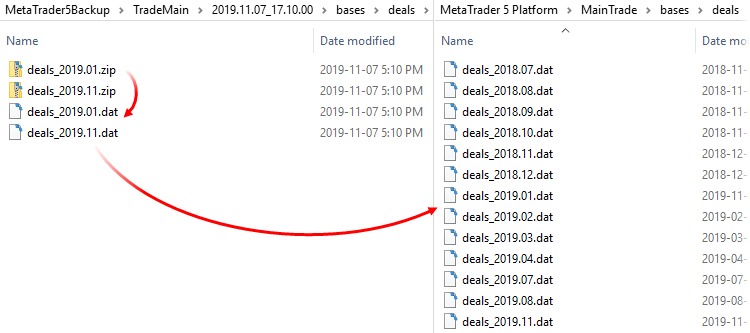
To request data from a file copy, go to the necessary section of MetaTrader 5 Administrator (Orders, Deals, Positions, Accounts) and select a copy for the desired date in the query line. The terminal collects and presents data from the basic and selected incremental copies. Such requests actually restore the state of the database as at the selected point in time.

#### Database recovery for a specific date

To restore server databases as on a specific date, stop all the servers within the cluster. Next, unzip and copy the necessary databases from the \base\ directory of the backup server to the corresponding directories of the primary server.



Then unzip and copy the required data from the incremental copy directory for the desired date (its format is [YYYY.MM.DD\_HH.MM.SS]) to the relevant directories of the main server.



|  |
| --- |
| It is not recommended to restore trading databases separately. Orders, deals and positions are interrelated and should match each other. |

### Moving the current day logs

| **Contents of \MainTrade\logs\** | **Contents of \Backup\logs\** |
| --- | --- |
| Current day log: 20170330.log   first entry:    10:56:16.265    Startup    service start initialized last entry: 10:57:55.421    Exit    shutdown finished | Current day log: 20170330.log   first entry:    10:56:16.265    Startup    service start initialized last entry: 10:56:17.888    192.168.0.131    Config: network config synchronized |
| Switching the servers, the backup server is active here now | Switching the servers, the trade server is active here now |
| The previous trade server log is renamed to 20170330.log.failover:   first entry:    10:56:16.265    Startup    service start initialized last entry: 10:57:55.421    Exit    shutdown finished  The new file of the backup server log: 20170330.log   first entry:    10:57:52.507    Startup    service start initialized last entry: 10:57:55.401    192.168.0.131    Config: network config synchronized | The new file of the trade server log 20170330.log:   first entry:    10:57:52.507    Startup    service start initialized last entry: 11:02:28.433    Exit    shutdown finished |
| Reverse switching, the trade server is active here again | Reverse switching, the backup server is active here again |
| 20170330.log.failover is renamed back to 20170330.log and the new trade server entries are added to it:   first entry:    10:56:16.265    Startup    service start initialized                   10:57:55.421    Exit    shutdown finished                   11:02:25.454    Startup    service start initialized last entry: 11:04:01.157    192.168.0.131    Config: network config synchronized | The previous trade server log is renamed to 20170330.log.failover:   first entry:    10:57:52.507    Startup    service start initialized last entry: 11:02:28.433    Exit    shutdown finished  The backup server resumes adding entries in its former 20170330.log file:   first entry:    10:56:16.265    Startup    service start initialized last entry: 11:04:01.157    192.168.0.131    Config: network config synchronized |

The backup server is a copy of the backed up one since it stores the same data and features the same file and directory structure. In particular, it stores its logs in directories and files with the same names (Logs\\*.log)

The special mechanism prevents mixing of the both servers' current day logs when switching to the backup server.

When switching the servers, the current day log file of the previous backed up server is renamed by adding the .failover extension to its name. The backup server to replace it starts keeping the current day log from scratch. The \*.failover file is generated only at the previous backed up server. The server, to which the switching is performed, simply creates its log file from scratch.

If a reverse switching is performed the same day, the .failover file is renamed back to .log and the backup server goes on keeping the log in it.

Let's consider an example of switching the main trade server to the backup one and vice versa.

### Emergency Situations During the Backup Process

Creation of backup copies by the server can be tracked by its journal. This section contains examples of journal entries describing emergency situations during backup.

|  |
| --- |
| 2012.06.13 17:19:57 TradeOrders: invalid index header timestamp |

Such a journal entry means that orders base index (order.idx) does not correspond to the orders base data (order.dat). A possible reason for such entry may be a backup server abnormal shutdown (for example, power cut or server manual shutdown).

This situation is not dangerous. The backup server will reconstruct the index and continue its operation. The data in the database remains fully intact. However, the reason of the abnormal server shutdown should be found out.

|  |
| --- |
| 2012.06.13 17:19:57 TradeDeals: deals\_2012.06.dat: base has invalid hash |

This entry indicates that the deals base on a backup server differs from the one on a trading server after synchronization. In this case, the backup server will remove the indicated base and synchronize it again.

Such backup server journal entries are permissible for deals base (deals\_\*) after weekends (when databases optimization compaction are performed on the main server). The reason is that a backup server cannot always synchronize a month deals base in saving mode (delete the entries that were deleted during a trading server optimization) after optimizing the base on a trading server. In that case, the backup server performs the full database synchronization.

For the rest of the databases such a situation is not acceptable in any time.

## Switching to Backup Server

The MetaTrader 5 platform can automatically monitor the availability of trade and history servers. The monitoring is performed by the backup server itself as well as by access servers. Each server monitors the availability of the main server and polls other monitoring servers to check whether the main server is available to them.

If the main server is unavailable for some time, the platform will automatically switch to the backup server. The downtime is minimal, while switching usually takes less than a minute.

In the platform, you can also switch to a backup server manually. In case of a failure of the history server or a non-main trade server, you can quickly switch to the backup server in the automated mode. The same procedure provides for an easy migration of servers to new hardware. You will only need to properly setup the backup server via MetaTrader 5 Administrator, use the fast deployment procedure and switch to the newly installed backup server afterwards.

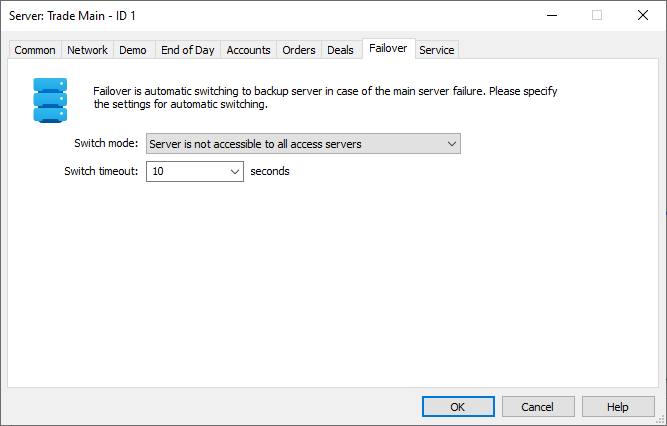
|  |
| --- |
| All critical trade and history server data is backed up in real time. Some non-critical trade server data, as well as history server data are backed up every hour. When switching to the backup server, critical data obtained during the procedure may be lost (the procedure itself usually takes less than a minute). Therefore, we strongly recommend that you switch to the backup server only outside of working hours. |

### Switching to the Backup Server Automatically

Automatic switching to a backup server allows minimizing the platform unavailability time in case of emergency situations. The platform automatically monitors the performance of its components and switches to backup servers if necessary.

The necessity to switch to the backup server is defined by the monitoring ("witness") servers. The backup server itself and access servers (with monitoring mode enabled) act as the monitoring ones. The backup server monitors the availability of the master server in real time mode and checks if it is available for the access servers as well.

Automatic switching can be enabled in the master server's settings (either a trade or a history one):



There are two scenarios for determining the unavailability of the main server:

* Server is not accessible to most access servers — the number of the monitoring servers unable to access the master server should exceed the ones able to access it at least by one for the switch to occur.  If you have configured five monitoring servers, the main server should be unavailable to at least three of them, or to four of six monitoring servers. If you are using two monitoring servers (the minimum allowed number), the main server should be unavailable for both of them.
* Server is not accessible to all access servers — the master server should be unavailable for all monitoring servers for the switch to occur.If you have configured five monitoring servers, the main server should be unavailable to all of them.

If several backup servers are used for one main server, then each of the backup servers will start switching to the master server mode in case of the main server failure. The last switched server will be used as the main server, while all the rest of them will switch back to backup mode.

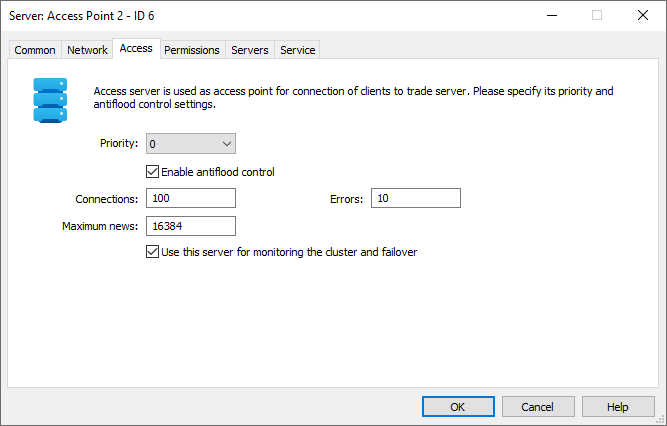
If a certain backup server should not be used for automatic switching, disable in its settings the following option: "Use this backup server for failover". This may be needed if the backup server is used only for exporting data to SQL database.

|  |
| --- |
| * The number of monitoring servers should not be less than 2. * It is recommended to install access servers on different computers (different data centers), separate from the main and history servers. This will provide the most relevant server monitoring data. |

In "Switch timeout" parameter, you can specify the time (in seconds) during which the server should be unavailable for monitoring servers to start switching to the back-up server. Also, after this time period, other platform components start their attempts to connect to the access points of the current backup server (trying to connect to it as to the master one).

|  |
| --- |
| When measuring the time of the master server's unavailability, its cause is considered. In case of a manual restart of the server, the unavailability time is increased according to the time required for restart.   * Unexpected failure — switching is performed after the specified timeout. * The server or the system service or the operating system is restarted by the administrator — switch time is increased by the time spent for restart. The minimum time is 30 seconds and the maximum time is 300 seconds. * Server restart during update (LiveUpdate) — switch time is increased by the time spent for restart. The minimum time is 180 seconds and the maximum time is 300 seconds. |

To make an access server a monitoring one, enable "Use this server for monitoring the cluster and failover" option in the server's settings:



|  |
| --- |
| * If the access server is unavailable for the backup one, that does not mean that the master server is also unavailable. In this case, the number of monitoring servers is reduced. * In case of the master and other servers' simultaneous failure, the master server is restored first. |

#### Features of Connecting to Monitoring Servers

The backup server uses the following algorithm for connecting to the monitoring servers:

* As soon as the main server becomes unavailable for the backup server, the backup server checks all monitoring servers one by one and trues to connect to them.
* First, the backup server tries to connect to the monitoring server via the local address if available. Connection to a local address is performed if listen addresses of the backup and access servers are located in 10.\*, 172.16.\* — 172.31.\* or 192.168\* subnet. The first three octets in their addresses should coincide. In that case, both servers are deemed to be located in a single subnet, and the backup server tries to connect directly to the listen address of the access server. Example: the backup server has 192.168.0.100:1951 listen address, while the access one - 192.168.0.105:1950.
* If connection via the local address has failed, the backup server uses public points of the access server.

|  |
| --- |
| In order for the switch to occur as fast as possible, all access servers should be available. There should be no disabled servers among the monitoring ones. The backup server spends 5 seconds trying to connect to a non-existent address. |

#### Using Several Backup Servers

If several backup servers are used for a single main one, then each of the backup servers starts switching to the master server mode in case of the main server's failure. The last switched server is used as the master one, while all the rest of them switch back to backup server mode.

#### Logging Monitoring Results

You can request the backup server's log using "Failover" keyword to control the process of monitoring the master server. Sample entries:

|  |
| --- |
| 2013.09.09 09:07:33     Failover        master server '1' - 'Trade Main' is available 2013.09.09 09:07:33     Failover        master server '1' - 'Trade Main' is available **for** witness server '2' - 'Access Point 1' 2013.09.09 09:07:34     Failover        master server '1' - 'Trade Main' is available **for** witness server '6' - 'Access Point 2' 2013.09.09 09:07:54     Failover        witness access server '7' is not available 2013.09.09 09:07:54     Failover        master server '1' - 'Trade Main' is available **for** witness server '11' - 'Access Point 3 2013.09.09 09:07:54     Failover        master server '1' - 'Trade Main' is available **for** 4 witnesses and unavailable **for** 0 witnesses [0 min] |

These entries mean as follows:

* Master server with identifier 1 is available.
* Master server with identifier 1 is available for monitoring server 2 named Access Point 1.
* Master server with identifier 1 is available for monitoring server 6 named Access Point 2.
* Monitoring server with identifier 7 is unavailable for the backup server.
* Master server with identifier 1 is available for monitoring server 11 named Access Point 3.
* Master server is available for 4 witnesses and not available to 0 witnesses. The time, during which the server has been unavailable, is shown in brackets.

#### Working after Switching to the Backup Server

After the backup server has switched to the trading one, the client terminals scan public access points of the access server in order to connect to it. The time of going through the access points depends on the following factors:

* Actual accessibility of the public point for a client. If an address is not available for the client (for example, a local IP address is specified in the settings as a public access point), the terminal spends 10 seconds trying to connect to it. An attempt to connect to the next access point is made only in 10 seconds.
* The number of the access server addresses unavailable for clients. For example, if two local addresses 192.168.0.100 and 192.168.0.101, as well as an external one - access.server.com (available external address of the server) are specified among the public access points, the client terminals will first spend 20 seconds trying to connect to local addresses before successfully connecting to the external one.
* Availability of the trade and history servers for the access one. The access server goes through the access points of the trade and history servers according to their network settings. Correctness of the network settings defines how quickly the access server becomes ready for work.

|  |
| --- |
| It is recommended that the access servers having local IP addresses in the list of public points are made available only for the administrators and managers working in the same local network. To do this, uncheck all options except "Allow administrator connection" and "Allow manager connection" ones in the Permissions tab of the access server. |

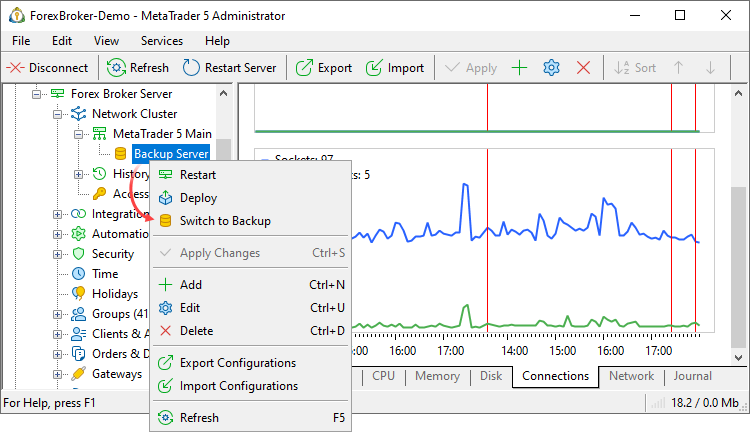
### Manual Switching to a Backup Server and Migration of Servers

In the platform, it is possible switch to a backup server from the main one manually. It is a quick and automatic procedure. In addition to emergency cases, the procedure can be used for migrating servers to new hardware.

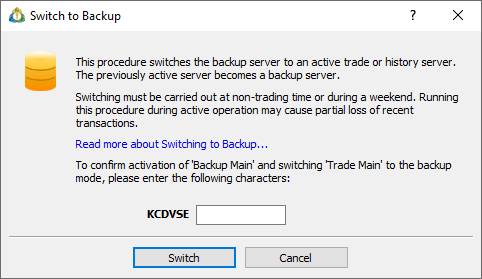
Install the backup server on the computer, to which you plan to migrate the server. After installing and launching the backup server, it is recommended to let it operate for a few days on the new machine to make sure it operates well.

|  |
| --- |
| * If you want to avoid the loss of data that is backed up every hour, the backup server should be restarted before switching to it. The server creates the latest data backup. While the server is busy backing up data, the following sign is displayed on its icon Backup in process (the icon itself has the following look Backup in process). Start switching to the backup server only after the backup process is complete. * Server migration must only be performed in non-trading hours. During the switching procedure, the main server continues to receive data, which will not be copied to the backup server. * In order to prevent important information from being lost, trading and changes in the client base are not allowed on the main server right after the start of switching to the backup server. The ban is valid for one minute. If the platform fails to switch to a backup server within this period, the ban is removed. |

Execute "Switch to backup server Switch to backup server" command:



To avoid accidental switching, the platform requires an additional confirmation. In the dialog that appears, enter the required characters and click "Switch".

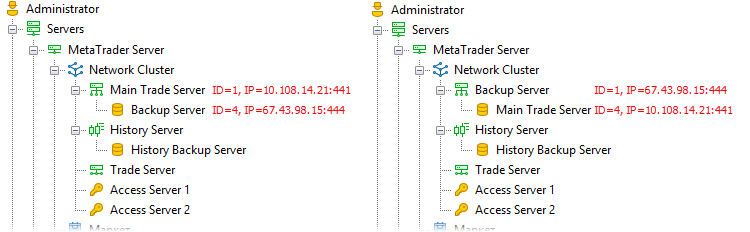


After the procedure is complete, you will see that the trade/history server has changed places with the backup server in Network section.

### The Features of the Switching Procedure

On the backup server's side, the switching process is performed as follows:

* The backup server stops Windows service.
* It also updates the network settings of the platform. The servers exchange roles, while the main server becomes the backup one, and the backup server starts operating as the main trade server:
  + The network settings (IP addresses, outgoing address, public points) and the name of the former backup server are set for the main server in the platform configuration.
  + The IP address and the name of the former main server are set for the backup server in the platform configuration.
  + On the servers, only their internal IDs change. The ID of the former main server is set for the new backup server, and the ID of the former backup server is set for the new main one.



* The master server's Windows service is installed
* Notification is sent to the master server, waiting for confirmation.
* The master server's Windows service is launched.
* The former backup server's Windows service is deleted.

If the backup server has been active during the switch, it is switched to the backup mode:

* The master server's Windows service is stopped.
* Network settings are updated similar to how it is done for the backup server.
* The backup server's Windows service is installed and launched.
* The former master server's Windows service is deleted.

|  |
| --- |
| * If the former master server was inactive during the switch to the backup one (for example, the server computer was shut down) and the copy restored from backup is already working by the moment the server is restored, the former master server switches to the backup mode automatically. * Physically (on the hard drive), the new trade/history server operates from the former backup server's directory, while the new backup server operates from the trade/history server's one. |

## Restoring Server

If the main trade server fails, you will not be able to switch to the backup server manually using the MetaTrader 5 Administrator, because you will not be able to connect to the server. Managing other cluster components will also be impossible. The system of automatic switching to the backup server allows avoiding such a situation. However, if the system was not enabled, you will need to recover the server manually:

1. Go to the computer where the main trade server is installed, and stop the system service if it runs. The default name of the main trade server service is mt5tmsrv. It can be stopped in the Control Panel — Administrative Tools — Services, as well as using the command line:

|  |
| --- |
| net stop mt5tmsrv or D:\MetaTrader 5 Platform\Main Trade\mt5trade64.exe /stop. |

If the computer is down (is unavailable), skip this step. However, once the computer is back up, please make sure that the old trade server service has not been restarted.

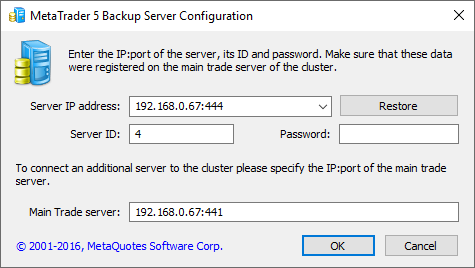
2. Go to the computer where the backup server is installed, and stop the system service. The default name of the backup server service is mt5bsrv. It can also be stopped using the Control Panel or the command line:

|  |
| --- |
| net stop mt5bsrv or D:\MetaTrader 5 Platform\Backup Main Trade\mt5backup64.exe /stop. |

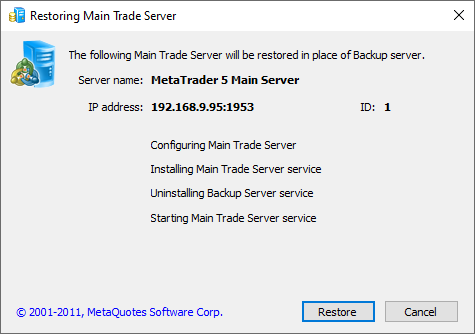
3. Run the backup server file from the command line with the /gui parameter. For example:

|  |
| --- |
| D:\MetaTrader 5 Platform\Backup Main Trade\mt5backup64.exe /gui. |

Click Restore in the window that appears.



After that, the recovery process is started. Other components of the cluster will automatically switch to the new main server.



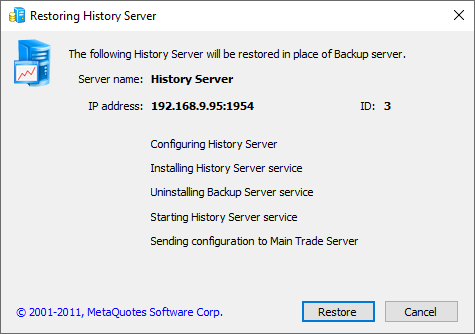
The process of restoring is run automatically in several steps:

* **Configuring Main Trader Server**  
  At this stage, the IP address and port of the main server are replaced with those of the backup server, which are displayed in the upper part of the window. The password and identifier are not changed.
* **Installing Main Trade Server service**
* **Uninstalling Backup Server service**
* **Starting Main Trade Server service**

|  |
| --- |
| * Before you launch the backup server, make sure the main server is stopped. Otherwise, your clients may start working with different servers. For example, your main server and access server are located at the same provider. The provider has issues with the internet connection and the servers became unavailable. The backup server is deployed in this case. After a while, connection to the main server is restored causing two servers to work simultaneously. In this case, contact your provider and request the immediate disabling of the main server. * If one of the stages cannot be completed, all the changes made during restoring will be rolled back. * As soon as restoring is finished, it is recommended to setup a new backup server. |

After the recovery of the main trade server, you will be able to connect to the cluster via MetaTrader 5 Administrator. Other components can be switched to backup servers in a regular way through the interface. For example, if the history server was installed on the same machine, you can switch to the backup server by running the appropriate command in its context menu.

The described procedure can also be used to restore other servers, including additional trade servers and history server. When restoring servers, new configuration data will be sent to the main trade server. A restored server will connect to the cluster without additional settings, and you will be able to control it via MetaTrader 5 Administrator.



# SQL Export

The MetaTrader 5 trading platform provides standard options for the real-time data export to MySQL, Microsoft SQL Server, FireBird, Oracle, MariaDB and PostgreSQL databases. This option enables the quick and easy deployment of data export to an external DBMS for using the platform data in any popular programming language and third-party applications. Thus, it is possible to create an intermediate layer between a trade server and broker's program services that regularly access trading data. This reduces the load on the trade server when it receives the current trading data.

The export function is enabled by simple specification of settings for connection to DBMS via MetaTrader 5 Administrator. After that, the backup server will immediately perform an initial synchronization of data from the DBMS. Further, new data will be exported to the DBMS in real time. The following data is exported:

* Information about clients (general information and trading status)
* Current active orders and positions
* The history of orders and deals
* Current prices
* Virtually all settings of the trading platform (except for the working time, synchronization and spreads)

The description of installation and setup of popular databases is provided in appropriate subsections:

* MySQL Server 5.7 (supported versions: 5.1, 5.5, 5.6, 5.7, 8.0)
* MariaDB 10.2 (all version supported)
* FireBird 3.0 (supported versions: 2.0, 2.1, 2.5, 3.0)
* Microsoft SQL Express 2012 (supported versions: 2005, 2008, 2008 R2, 2012, 2014, 2016, 2017, 2019RC)
* Oracle Database Express Edition 11g (supported versions: 11g/11g Express Edition, 12c)
* PostgreSQL (supported versions: 8.4 — 14.5)

## Operation Principle

Export mechanism has been implemented on the side of backup servers replicating data from the trade servers.

Below are the general steps of synchronization in a backup server:

* The backup server connects to the appropriate trade server and is synchronized with it.
* After synchronization with the trade server is successfully complete, the backup server synchronizes an external DBMS.
* The backup server applies changes to its databases and the external DBMS via transactions from the trade server.

The backup server performs initial synchronization based on time stamps (Timestamp field in the tables) during each connection to DBMS:

* All logs with different time stamps in the external DBMS are replaced with backup server logs.
* All entries that are absent at the backup server are removed from the external DBMS.

|  |
| --- |
| A time stamp is used for checking the identity of the backup and the external DBMS logs. If a time stamp on the backup is similar to the on at the external DBMS, a backup server considers that all other log fields are similar and the log update is not required. |

After databases are synchronized, the backup server applies change transactions on the external DBMS. Besides, prices and profit values for active orders (mt5\_orders), positions (mt5\_positions) and related trading accounts (mt5\_accounts) are also periodically updated.

A detailed description of exported tables can be found in the following subsections:

* **mt5\_symbols** — symbols' configurations.
* **mt5\_symbols\_sessions** — symbols' trade and quotation sessions.
* **mt5\_groups** — groups' configurations.
* **mt5\_groups\_symbols** — individual symbol settings for groups.
* **mt5\_commissions** — commission settings for groups.
* **mt5\_commissions\_tiers** — commission level settings.
* **mt5\_managers** — manager accounts.
* **mt5\_clients** — client database.
* **mt5\_documents** — database of client documents.
* **mt5\_users** — account database.
* **mt5\_orders** — open order database.
* **mt5\_positions** — position database.
* **mt5\_orders\_history** — closed order database.
* **mt5\_deals** — deal database.
* **mt5\_accounts** — trade account state database.
* **mt5\_prices** — database of prices.
* **mt5\_daily** — database of daily reports.
* **mt5\_holidays** — holiday configurations.
* **mt5\_network** — general settings of servers.
* **mt5\_network\_access\_servers** — access servers settings.
* **mt5\_network\_history\_servers** — history server settings.
* **mt5\_network\_trade\_servers** — trade servers settings.
* **mt5\_network\_backup\_servers** — backup servers settings.
* **mt5\_network\_backup\_folders** — backed up custom folders.
* **mt5\_firewall** — firewall settings.
* **mt5\_routing** — settings of routing rules.
* **mt5\_routing\_dealers** — settings of dealers/gateways in routing rules.
* **mt5\_routing\_conds** — additional conditions in routing rules
* **mt5\_feeders** — settings of data feeds.
* **mt5\_feeder\_translates** — conversion settings on data feeds.
* **mt5\_feeder\_params** — additional settings of data feeds.
* **mt5\_reports** — report settings.
* **mt5\_report\_params** — additional settings of reports.
* **mt5\_plugins** — plugin settings
* **mt5\_plugin\_params** — additional settings of plugins.
* **mt5\_time** — platform trading time settings.
* **mt5\_time\_weekdays** — working time schedule of the platform, by days.
* **mt5\_gateways** — gateway settings.
* **mt5\_gateways\_params** — additional gateway settings.
* **mt5\_gateways\_translates** — translation settings in gateways.

## Your own data in MetaTrader 5 tables

In the platform, you can create your own tables and databases, add your own fields in mt5\_\* tables, which are used for data export, as well as create stored procedures and triggers.

* The backup server does not recreate data, but only adds or updates existing records. A table entry is only deleted if the appropriate entry is deleted on the platform side. For example, if a user is added to the mt5\_users tables, the appropriate user record will exist in the database until the user is deleted from MetaTrader 5.
* The backup server only works with its own tables and fields.
* The backup server does not create default indexes. Each index is an extra load on the database, which can slow down exports and information updates.
* When you create your own fields in mt5\_\* tables, do not forget to set default values for them (or allow NULL). Otherwise, the backup server will not be able to add new entries to the tables.

# mt5\_symbols

Symbols' configurations are exported to the table. The table contains the following fields:

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| Symbol\_ID | Integer | Primary key. Unique symbol ID for more efficient request of the symbol data from the database. Assigned automatically during the export. |
| Timestamp | Integer | Unique record within the table. It is used for internal purposes of MetaTrader 5 servers. If the timestamp is changed for a record, it means that the record has been changed. |
| Symbol | String | Symbol name. |
| Path | String | Path to a symbol. |
| ISIN | String | International Securities Identification Number (ISIN) of a symbol. |
| Description | String | Symbol description. |
| International | String | The international symbol name. |
| Category | String | The name of the category or sector to which the symbol belongs. |
| Exchange | String | The name of the exchange in which the security is traded. |
| CFI | String | Instrument classification in accordance with the ISO 10962 standard. |
| Sector | Integer | The economic sector the instrument belongs to. Passed as a value of the EnSectors enumeration. |
| Industry | Integer | The industry branch the instrument belongs to. Passed as a value of the EnIndustries enumeration. |
| Country | String | The country of the company whose shares are traded on the stock exchange. |
| Basis | String | The underlying asset of a derivative financial instrument. |
| Source | String | The name of the source symbol whose quotes are used for the current financial instrument. |
| Page | String | The address of the web page of a symbol. |
| CurrencyBase | String | The base currency of a symbol. |
| CurrencyBaseDigits | Integer | The accuracy of conversion into the base currency. |
| CurrencyProfit | String | The profit currency for a symbol. |
| CurrencyProfitDigits | Integer | The accuracy of conversion into the profit currency. |
| CurrencyMargin | String | The symbol margin currency. |
| CurrencyMarginDigits | Integer | The accuracy of conversion into the margin currency. |
| Color | Integer | The color of the symbol in the "Market Watch" window of the terminals. |
| ColorBackground | Integer | The color of the symbol background in the "Market Watch" window of the terminals. |
| Digits | Integer | The number of decimal places in the price of the symbol. |
| Point | Float | Point size. |
| Multiply | Float | The value to multiply the price to, to get the number of points. |
| TickFlags | Integer | Options for working with tick data. Passed as a value of the EnTicksFlags enumeration (sum of values of appropriate flags). |
| TickBookDepth | Integer | The range of the Depth of Market. |
| FilterSoft | Integer | The soft level of price filtering. |
| FilterSoftTicks | Integer | The value of the ticks counter for the soft filtering. |
| FilterHard | Integer | The hard level of price filtering. |
| FilterHardTicks | Integer | The value of the ticks counter for the hard filtering. |
| FilterDiscard | Integer | The discard level of price filtering. |
| FilterSpreadMax | Integer | The maximum allowed spread value. |
| FilterSpreadMin | Integer | The minimum allowed spread. |
| SubscriptionsDelay | Integer | The delivery delay for the quotes provided by subscription. Indicated in minutes. |
| TradeMode | Integer | The symbol trading mode. Passed in a value of the EnTradeMode enumeration. |
| CalcMode | Integer | The mode of margin and profit calculation. Passed in a value of the EnCalcMode enumeration. |
| ExecMode | Integer | Execution mode of a symbol. Passed in a value of the EnExecutionMode enumeration. |
| GTCMode | Integer | Types of orders that can be set for the symbol. Passed as a value of the EnGTCMode enumeration (sum of values of appropriate flags). |
| FillFlags | Integer | Types of filling allowed for the symbol. Passed as a value of the EnFillingFlags enumeration (sum of values of appropriate flags). |
| ExpirFlags | Integer | Available types of order expiration for a symbol. Passed as a value of the EnExpirationFlags enumeration (sum of values of appropriate flags). |
| Spread | Integer | Symbol spread size. |
| SpreadBalance | Integer | Symbol spread balance. Spread balance is set a shift from the equal distribution of the spread value between Bid and Ask prices. For example, if the spread is equal to 10 and it is distributed as -5 Bid/+5 Ask, then the spread balance value is 0. The -6 Bid/+4 Ask ratio corresponds to value -1, ratio -4 Bid/+6 Ask corresponds to value 1. |
| SpreadDiff | Integer | Symbol spread difference. This parameter returns the base value of the spread, which is actually equal to 0. To work with spread difference of a particular group, the corresponding parameter of the group should be used. |
| SpreadDiffBalance | Integer | Spread balance difference. This parameter returns the base value of the balance of spread difference, which is actually equal to 0. To work with the balance of spread difference of a certain group, the corresponding parameter of the group should be used. |
| TickValue | Float | The price of one tick of a symbol. |
| TickSize | Float | The size of one tick of a symbol. |
| ContractSize | Float | The contract size for the symbol. |
| StopsLevel | Integer | The price band, within which placing stop orders is not allowed. |
| FreezeLevel | Integer | The price band, within which it is not allowed to modify orders and positions. |
| QuotesTimeout | Integer | The time to wait for quotes in seconds, after which trading is automatically disabled for the symbol. |
| VolumeMin | Integer | The minimum volume of trade operations for a symbol. One unit corresponds to 1/10000 lot. |
| VolumeMinExt | Integer | The minimum volume of trade operations for the symbol for the group with extended accuracy. One unit corresponds to 1/100000000 lot. |
| VolumeMax | Integer | The maximum volume of trade operations for a symbol. One unit corresponds to 1/10000 lot. |
| VolumeMaxExt | Integer | The maximum volume of trade operations for the symbol for the group with extended accuracy. One unit corresponds to 1/100000000 lot. |
| VolumeStep | Integer | The volume change step for trade operations for a symbol. One unit corresponds to 1/10000 lot. |
| VolumeStepExt | Integer | The volume change step allowed for trade operations for the symbol, with extended accuracy. One unit corresponds to 1/100000000 lot. |
| VolumeLimit | Integer | The maximum allowed aggregate volume of positions and orders for a symbol in one direction. One unit corresponds to 1/10000 lot. |
| VolumeLimitExt | Integer | The maximum aggregate volume (with extended accuracy) of positions and orders for the symbol in one direction. One unit corresponds to 1/100000000 lot. |
| MarginFlags | Integer | Additional margin checking modes. Passed in a value of the EnMarginFlags enumeration. |
| MarginInitial | Float | The size of the initial margin. |
| MarginMaintenance | Float | The size of the maintenance margin. |
| MarginInitialBuy | Float | The initial margin rate for market Buy orders. |
| MarginInitialSell | Float | The initial margin rate for market Sell orders. |
| MarginInitialBuyLimit | Float | The initial margin rate for Buy Limit orders. |
| MarginInitialSellLimit | Float | The initial margin rate for Sell Limit orders. |
| MarginInitialBuyStop | Float | The initial margin rate for Buy Stop orders. |
| MarginInitialSellStop | Float | The initial margin rate for Sell Stop orders. |
| MarginInitialBuyStopLimit | Float | The initial margin rate for Buy Stop Limit orders. |
| MarginInitialSellStopLimit | Float | The initial margin rate for Sell Stop Limit orders. |
| MarginMaintenanceBuy | Float | The maintenance margin rate for market Buy orders. |
| MarginMaintenanceSell | Float | The maintenance margin rate for market Sell orders. |
| MarginMaintenanceBuyLimit | Float | The maintenance margin rate for Buy Limit orders. |
| MarginMaintenanceSellLimit | Float | The maintenance margin rate for Sell Limit orders. |
| MarginMaintenanceBuyStop | Float | The maintenance margin rate for Buy Stop orders. |
| MarginMaintenanceSellStop | Float | The maintenance margin rate for Sell Stop orders. |
| MarginMaintenanceBuyStopLimit | Float | The maintenance margin rate for Buy Stop Limit orders. |
| MarginMaintenanceSellStopLimit | Float | The maintenance margin rate for Sell Stop Limit orders. |
| MarginHedged | Float | The hedged margin value. |
| SwapMode | Integer | The swap calculation mode for a symbol. Passed in a value of the EnSwapMode enumeration. |
| SwapLong | Float | The swap size for long positions. |
| SwapShort | Float | The swap size for short positions. |
| SwapYearDay | Integer | The number of days in a year used in calculating swap percent. Passed by the EnSwapDays enumeration value. |
| SwapFlags | Integer | Additional swap settings. Passed by the EnSwapFlags enumeration value. |
| SwapRateSunday | Float | Swap multiplier for Sundays. |
| SwapRateMonday | Float | Swap multiplier for Mondays. |
| SwapRateTuesday | Float | Swap multiplier for Tuesdays. |
| SwapRateWednesday | Float | Swap multiplier for Wednesdays. |
| SwapRateThursday | Float | Swap multiplier for Thursdays. |
| SwapRateFriday | Float | Swap multiplier for Fridays. |
| SwapRateSaturday | Float | Swap multiplier for Saturdays. |
| TimeStart | Integer | The start date of trading for a symbol. It is considered that there is no time limitation for trading by a symbol if both TimeStart and TimeExpiration are equal to 0. |
| TimeExpiration | Integer | The date of trading expiration for a symbol. It is considered that there is no time limitation for trading by a symbol if both TimeStart and TimeExpiration are equal to 0. |
| REFlags | Integer | The request execution flags. Passed as a value of the EnRequestsFlags enumeration (sum of values of appropriate flags). |
| RETimeout | Integer | Time in seconds during which the price issued by a dealer in the request execution mode is valid. |
| IECheckMode | Integer | Check mode for instant execution. Passed in a value of the EnInstantMode enumeration. |
| IETimeout | Integer | The maximum allowed difference between the time of arrival of the price, at which the client places an order, and the time of the last price. |
| IESlipProfit | Integer | The maximum allowed slippage in the profitable direction during instant execution. |
| IESlipLosing | Integer | The maximum allowed slippage in the loss direction during instant execution. |
| IEVolumeMax | Integer | The maximum volume of a trade operation that can be executed in the instant execution mode. One unit corresponds to 1/10000 lot. |
| IEVolumeMaxExt | Integer | The maximum volume (with extended accuracy) of a trade operation that can be executed in the instant execution mode. One unit corresponds to 1/100000000 lot. |
| PriceSettle | Float | The clearing price of the previous session. |
| PriceLimitMax | Float | The maximum allowed price of the symbol. |
| PriceLimitMin | Float | The minimum allowed price of the symbol. |
| TradeFlags | Integer | The trade flags of the symbol. Passed in a value of the EnTradeFlags enumeration. |
| OrderFlags | Integer | The flags of order types that are allowed for the symbol. Passed in a value of the EnOrderFlags enumeration (sum of values of appropriate flags). |
| MarginRateLiquidity | Float | The liquidity rate of the symbol. It determines the amount of the current value of an asset for the specified financial instrument, which will be taken into account as collateral (accounted for in client's equity). |
| MarginRateCurrency | Float | The margin currency rate (rate change radius of the currency, a futures contract is denominated in, relative to the Russian ruble). |
| FaceValue | Float | The face value of a bond. |
| AccruedInterest | Float | The accrued interest of a bond. |
| SpliceType | Integer | The futures contract splicing type. |
| SpliceTimeType | Integer | The date of splicing of the futures contracts. |
| SpliceTimeDays | Integer | The offset of splicing of the futures contracts. |
| OptionMode | Integer | Option type and style:   * 0 means a European call option * 1 means a European put option * 2 means an American call option * 3 means an American put option |
| PriceStrike | Float | The price, at which an option gives the right to buy or sell an asset (the strike price). |
| FilterGap | Integer | The difference between the previous and the next quote, starting from which a gap is considered to be formed. |
| FilterGapTicks | Integer | The number of ticks for disabling the gap mode. If no new gap occurs within the specified number of quotes, the mode is disabled. |
| TickChartMode | Integer | The mode of creation of the symbol chart: 0 — using the Bid price, 1 — using the Last price. |

# mt5\_symbols\_sessions

Data on symbols' trade and quoting sessions is exported to the table. The table contains the following fields:

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| Session\_ID | Integer number | Primary key. Unique session ID. Assigned automatically during the export. |
| Symbol\_ID | Integer number | Unique ID of the symbol, to which the session is applied. |
| Type | Integer number | Session type: 0 - quoting, 1 - trade. |
| Day | Integer number | Day of the week from 0 to 6. 0 - Sunday, 6 - Saturday. |
| Open | Integer number | The opening time of a trading or quoting session of a symbol in minutes elapsed since 00:00. For example, 100 denotes 01:40. |
| Close | Integer number | The closing time of a trading or quoting session of a symbol in minutes elapsed since 00:00. |

mt5\_groups

Groups' configurations are exported to the table. The table contains the following fields:

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| Group\_ID | Integer | Primary key. Unique group ID for more efficient request of the group data from the database. Assigned automatically during the export. |
| Timestamp | Integer | Unique record within the table. It is used for internal purposes of MetaTrader 5 servers. If the timestamp is changed for a record, it means that the record has been changed. |
| Group | String | The name of a group, including a path to it in accordance with the hierarchy. |
| Server | Integer | The ID of the trade server, to which the group is linked. |
| PermissionFlags | Integer | Flags of group permissions. Passed as a value of the EnPermissionsFlags enumeration (sum of values of appropriate flags). |
| AuthMode | Integer | Authorization mode for accounts in the group. Passed in a value of the EnAuthMode enumeration. |
| AuthPasswordMin | Integer | The minimum password length for accounts in the group. |
| Company | String | Name of the company that services the group. |
| CompanyPage | String | The website address of the company that services the group. |
| CompanyEmail | String | The email address of the company that services the group. |
| CompanySupportPage | String | The technical support website address of the company that services the group. |
| CompanySupportEmail | String | The technical support email address of the company that services the group. |
| CompanyCatalog | String | The name of the subdirectory that stores the templates of reports, emails, etc. for the company that services this group. |
| Currency | String | The group deposit currency. |
| CurrencyDigits | Integer | The number of digits after the decimal point in the group deposit currency. |
| ReportsMode | Integer | Report generation modes. Passed in a value of the EnReportsMode enumeration. |
| ReportsFlags | Integer | Report sending options. Passed as a value of the EnReportFlags enumeration (sum of values of appropriate flags). |
| ReportsEmail | String | The mail server used for sending reports to clients from the group. |
| ReportsSMTP | String | Address of SMTP server for sending reports. The field is obsolete and is not updated. |
| ReportsSMTPLogin | String | A login for the authorization on the SMTP server that is used for sending reports. The field is obsolete and is not updated. |
| NewsMode | Integer | The mode of news sending to the clients from the group. Passed in a value of the EnNewsMode enumeration. |
| NewsCategory | String | The categories of news received by the group. Use the backslash character "\" to specify subcategories. |
| NewsLangs | Array of integer numbers | The array of languages, in which the group receives news. The language is specified in the LANGID format used in the MS Windows (value from Prim.lang.identifier). |
| MailMode | Integer | The mode of operation of the internal mail system for the group. Passed in a value of the EnMailMode enumeration. |
| TradeFlags | Integer | Trade options of the group. Passed as a value of the EnTradeFlags enumeration (sum of values of appropriate flags). |
| TradeInterestrate | Float | The annual interest rate on deposits of the group accounts. |
| TradeVirtualCredit | Float | The amount of additional funds that a brokerage company can provide to a client for opening a position with a volume larger than allowed by the client's current funds. |
| MarginFreeMode | Integer | The mode of using of floating profit/loss in the free margin. Passed in a value of the EnFreeMarginMode enumeration. |
| MarginSOMode | Integer | The mode of checking the levels of Stop Out and Margin Call. Passed in a value of the EnStopOutMode enumeration. |
| MarginCall | Float | The level of Margin Call. Units are determined by the MarginSOMode parameter. |
| MarginStopOut | Float | The level of Stop Out. Units are determined by the MarginSOMode parameter. |
| MarginFreeProfitMode | Integer | The mode of using the profit/loss fixed during a trade day in the free margin. |
| MarginMode | Integer | The risk management model of the group. |
| MarginFlags | Integer | Margin calculation flags. |
| DemoLeverage | Integer | The default credit leverage for demo accounts opened in the group. |
| DemoDeposit | Float | The default amount of deposit for demo accounts opened in the group. |
| LimitHistory | Integer | The maximum number of days, for which the group can request data on conducted trade operation. Passed in a value of the EnHistoryLimit enumeration. |
| LimitOrders | Integer | The maximum number of orders that can be simultaneously placed by an account from this group. |
| LimitSymbols | Integer | The maximum number of symbols, for which an account can simultaneously receive quotes. |
| LimitPositions | Integer | The maximum number of open positions which the client can have on the account at the same time. |
| LimitPositionsVolume | Float | Currently the field is not used. |
| TradeTransferMode | Integer | The mode of transferring funds between accounts. |

mt5\_groups\_symbols

Individual symbol settings for groups are exported to the table. The table contains the following fields:

| **Name** | **Type** | **Description** | |
| --- | --- | --- | --- |
| Symbol\_ID | Integer | Primary key. Unique symbol ID for the group. | |
| Group\_ID | Integer | ID of the group, to which symbol settings are applied. | |
| Timestamp | Integer | Unique record within the table. It is used for internal purposes of MetaTrader 5 servers. If the timestamp is changed for a record, it means that the record has been changed. | |
| Path | String | The path to a symbol or group of symbols that are subject to the special group settings. | |
| TradeMode | Integer | The symbol trading mode for the group. Passed in a value of the EnTradeMode enumeration. | |
| ExecMode | Integer | The symbol execution mode for the group. Passed in a value of the EnExecutionMode enumeration. | |
| FillFlags | Integer | Types of filling allowed for the symbol in this group. Passed as a value of the EnFillingFlags enumeration (sum of values of appropriate flags). | |
| ExpirFlags | Integer | Types of order expiration allowed for the symbol in this group. Passed as a value of the EnExpirationFlags enumeration (sum of values of appropriate flags). | |
| SpreadDiff | Integer | Difference between the symbol spread for the group and the default spread. | |
| SpreadDiffBalance | Integer | The balance of spread difference set for the group. The balance of spread difference is set a shift from the equal distribution of the spread difference value between Bid and Ask prices. For example, if the spread difference is equal to 4 and it is distributed as -2 Bid/+2 Ask, then the balance of spread difference value is 0. The -3 Bid/+1 Ask ratio corresponds to value -1, ratio -1 Bid/+3 Ask corresponds to value 1. | |
| StopsLevel | Integer | The price band, within which the group is not allowed to place stop orders for a symbol. | |
| FreezeLevel | Integer | The price band, within which it is not allowed to modify orders and positions for the group. | |
| VolumeMin | Integer | The minimum volume of trade operations for a symbol for the group. One unit corresponds to 1/10000 lot. | |
| VolumeMinExt | Integer | The minimum volume (with extended accuracy) of trade operations for a symbol for the group. One unit corresponds to 1/100000000 lot. | |
| VolumeMax | Integer | The maximum volume of trade operations for a symbol for the group. One unit corresponds to 1/10000 lot. | |
| VolumeMaxExt | Integer | The maximum volume (with extended accuracy) of trade operations for a symbol for the group. One unit corresponds to 1/100000000 lot. | |
| VolumeStep | Integer | The step of change of trade operations volume for a symbol for the group. One unit corresponds to 1/10000 lot. | |
| VolumeStepExt | Integer | The step of change of trade operations volume (with extended accuracy) for a symbol for the group. One unit corresponds to 1/100000000 lot. | |
| VolumeLimit | Integer | The maximum allowed aggregate volume of positions and orders for a symbol in one direction for this group. One unit corresponds to 1/10000 lot. | |
| VolumeLimitExt | Integer | The maximum aggregate volume (with extended accuracy) of positions and orders for a symbol for this group. One unit corresponds to 1/100000000 lot. | |
| MarginFlags | Integer | The additional modes of symbol margin checking for the group. Passed in a value of the EnMarginFlags enumeration. | |
| MarginInitial | Float | The size of initial symbol margin for the group. | |
| MarginMaintenance | Float | The size of symbol maintenance margin for the group. | |
| MarginLong | Float | The group margin ratio for long positions and orders for a symbol. | |
| MarginShort | Float | The group margin ratio for short positions and orders for a symbol. | |
| MarginLimit | Float | The group margin ratio of limit orders for a symbol. | |
| MarginStop | Float | The group margin ratio of stop orders for a symbol. | |
| MarginStopLimit | Float | The group margin ratio for stop-limit orders for a symbol. | |
| MarginHedged | Float | The hedged margin value. | |
| SwapMode | Integer | The swap calculation mode for a certain symbol for the group. Passed in a value of the EnSwapMode enumeration. | |
| SwapLong | Float | The long position swap for a symbol for the group. | |
| SwapShort | Integer | The short position swap for a symbol for the group. | |
| Swap3Day | Integer | A day of charging a triple swap for a symbol for this group. | |
| SwapYearDays | Integer | The number of days in a year used in calculating swap percent for a given group. Passed by the EnSwapDays enumeration value. | |
| SwapFlags | Integer | Additional swap settings by symbol for the given group. Passed by the EnSwapFlags enumeration value. | |
| SwapRateSunday | Float | Sunday swap multiplier in symbol settings for the given group. | |
| SwapRateMonday | Float | Monday swap multiplier in symbol settings for the given group. | |
| SwapRateTuesday | Float | Tuesday swap multiplier in symbol settings for the given group. | |
| SwapRateWednesday | Float | Wednesday swap multiplier in symbol settings for the given group. | |
| SwapRateThursday | Float | Thursday swap multiplier in symbol settings for the given group. | |
| SwapRateFriday | Float | Friday swap multiplier in symbol settings for the given group. | |
| SwapRateSaturday | Float | Saturday swap multiplier in symbol settings for the given group. | |
| RETimeout | Integer | Time in seconds during which the price issued by a dealer in the request execution mode is valid. | |
| IECheckMode | Integer | The mode of checking during instant execution set for a group. Passed in a value of the EnInstantMode enumeration. | |
| IETimeout | Integer | The maximum allowed difference between the time of arrival of the price, at which the client places an order, and the time of the last price. The timeout is specified in seconds. | |
| IESlipProfit | Integer | The maximum allowed slippage in the profitable direction during instant execution. | |
| IESlipLosing | Integer | The maximum allowed slippage in the loss direction during instant execution. | |
| IEVolumeMax | Integer | The maximum volume of a trade operation that can be executed in the instant execution mode. One unit corresponds to 1/10000 lot. | |
| IEVolumeMaxExt | Integer | The maximum volume (with extended accuracy) of a trade operation that can be executed in the instant execution mode. One unit corresponds to 1/100000000 lot. | |
| OrderFlags | Integer | The flags of order types that are allowed for the symbol. Passed in a value of the EnOrderFlags enumeration (sum of values of appropriate flags). | |
| MarginRateLiquidity | Float | The liquidity rate of the symbol for the group. It determines the amount of the current value of an asset for the specified financial instrument, which will be taken into account as collateral (accounted for in client's equity). | |
| REFlags | Integer | The flags of request execution for the group. | |
| NULL value in the fields beginning from TradeMode means that the appropriate setting is inherited from the base symbol. | | |

mt5\_commissions

Commission settings for groups are exported to this table. The table contains the following fields:

| **Name** | **Type** | **Purpose** |
| --- | --- | --- |
| Commission\_ID | Integer | Initial key. A unique commission setting ID. |
| Group\_ID | Integer | ID of the group, a commission setting belongs to. |
| Name | String | Commission setting name (up to 64 characters). |
| Description | String | Commission setting description (up to 64 characters). |
| Path | String | Path to a symbol or a group of symbols covered by a commission setting. |
| Mode | Integer | Commission type: 0 — standard, 1 — agent. |
| ModeRange | Integer | Commission levels type:   * 0 — volume * 1 — turnover in money * 2 — turnover in volume |
| ModeCharge | Integer | Commission charge mode:   * 0 — daily * 1 — monthly * 2 — instant |
| TurnoverCurrency | String | Currency, in which the money turnover is calculated. |
| ModeEntry | Integer | Commission calculation mode depending on the trade direction:   * 0 — all trades regardless of direction. * 1 — only entry deals. * 2 — only exit deals. |
| ModeAction | Integer | Commission calculation mode depending on the trade type:   * 0 — all trades regardless of type. * 1 — only Buy deals. * 2 — only Sell deals. |
| ModeProfit | Integer | Commission calculation modes depending on the deal profit:   * 0 — all deals. * 1 — only profitable deals. * 2 — only losing deals. |
| ModeReason | Integer | Commission calculation modes depending on the reason for the deal.   * 0x00000000 — no commission will be charged for any trades. * 0x00000001 — the deal was performed by the client manually via the client terminal. * 0x00000002 — the deal was performed by the client using an Expert Advisor. * 0x00000004 — the deal was performed by a dealer via the Manager terminal. * 0x00000008 — the deal was performed from an external trading system. * 0x00000010 — the deal was performed via the MetaTrader 5 mobile terminal for Android or iPhone. * 0x00000020 — the deal was performed via the web terminal. * 0x00000040 — the deal was performed as a result of copying of a trading signal, in accordance with the subscription, in the client terminal. |

mt5\_commissions\_tiers

Commission levels are exported to this table. The table contains the following fields:

| **Name** | **Type** | **Purpose** |
| --- | --- | --- |
| Tier\_ID | Integer | Initial key. Unique commission level ID. |
| Commission\_ID | Integer | ID of the commission setting the level belongs to. |
| Mode | Integer | Commission calculation unit:   * 0 — account currency * 1 — base currency * 2 — profit currency * 3 — margin currency * 4 — points * 5 — percentage * 6 — specified currency |
| Type | Integer | Commission charge type:   * 0 — per trade * 1 — per volume |
| Value | Fraction | Commission sum. Commission units depend on the commission calculation method (Mode). |
| RangeFrom | Fraction | The minimum deal volume (turnover), from which the commission will be charged. |
| RangeTo | Fraction | The maximum trade volume (turnover), from which the commission will be charged. |
| Minimal | Fraction | The minimum amount of commission. The value is specified in the group deposit currency. |
| Currency | String | Commission calculation currency (if "Specified currency" is selected in the Mode field). |

mt5\_managers

Data about manager accounts are exported to this table. The table contains the following fields:

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| Login | Integer | Initial key. The user login, based on which the manager account is created. |
| Timestamp | Integer | A unique values within the table. Used by MetaTrader 5 servers for internal purposes. If the Timestamp of a record has changed, it means that the record has changed. |
| Name | The line | The name of the manager. |
| Mailbox | The line | The name of the manager's mailbox in the internal mailing system. |
| Server | Integer | The ID of the trade server to which the manager belongs. |
| RequestLimitLogs | Integer | The time period of system logs that are available to a manager:   * 0 — unlimited * 1 — 1 month * 2 — 3 months * 3 — 6 months * 4 — 1 year * 5 — 2 years * 6 — 3 years |
| RequestLimitReports | Integer | The time period of reports that are available to a manager:   * 0 — unlimited * 1 — 1 month * 2 — 3 months * 3 — 6 months * 4 — 1 year * 5 — 2 years * 6 — 3 years |
| Groups | The line | The list of groups processed by the manager. The groups are separated by commas, for example: "demo\forex-netting,demo\forex-netting". |
| Access | The line | The list of IP addresses, from which a manager is allowed to connect to the platform. Example: 192.168.0.1-192.168.0.10,192.168.0.12-192.168.0.20. |
| Further fields in the table describe manager permissions. A value of 1 means that the right is granted to the manager, 0 means no permission. | | |
| Right\_Admin | Integer | Connection using the administrator terminal. |
| Right\_Manager | Integer | Connection using the manager terminal. |
| Right\_Cfg\_Servers | Integer | Network configuration |
| Right\_Cfg\_Access | Integer | Configuration of the list of IP access. |
| Right\_Cfg\_Time | Integer | Configuration of the server working time. |
| Right\_Cfg\_Holidays | Integer | Configuration of holidays. |
| Right\_Cfg\_Groups | Integer | Configuration of groups. |
| Right\_Cfg\_Managers | Integer | Configuration of manager rights. |
| Right\_Cfg\_Requests | Integer | Configuration of the routing table. |
| Right\_Cfg\_Gateways | Integer | Configuration of gateways. |
| Right\_Cfg\_Plugins | Integer | Configuration of plugins. |
| Right\_Cfg\_Datafeeds | Integer | Configuration of data feeds. |
| Right\_Cfg\_Reports | Integer | Configuration of reports. |
| Right\_Cfg\_Symbols | Integer | Configuring the symbols. |
| Right\_Cfg\_Hst\_Sync | Integer | Configuration of synchronization. |
| Right\_Cfg\_ECN | Integer | ECN configuration. |
| Right\_Cfg\_VPS | Integer | Configuring Sponsored VPS for traders. |
| Right\_Cfg\_Web\_Services | Integer | Configuring integration with web services: SSL certificates and addresses for callback requests. |
| Right\_Cfg\_Funds | Integer | Configuring investment funds in the Administrator terminal |
| Right\_Cfg\_Messengers | Integer | Configuring integration with SMS providers and messengers in the Administrator terminal |
| Right\_Cfg\_KYC | Integer | Configuring integration with KYC services in the Administrator terminal |
| Right\_Cfg\_Automations | Integer | Configuring automatic actions for specified scenarios in the Administrator terminal. |
| Right\_Cfg\_Allocations | Integer | Accessing the Allocations section of the Administrator terminal. The section allows configuring groups, in which traders are able to open demo and preliminary real accounts directly from client terminals. |
| Right\_Cfg\_Payments | Integer | Configuring integration with payment systems (in development). |
| Right\_Srv\_Journals | Integer | Access to server journals. |
| Right\_Srv\_Reports | Integer | Receiving automatic server reports. |
| Right\_Charts | Integer | Editing history data on the server. |
| Right\_Email | Integer | Sending internal emails. |
| Right\_News | Integer | Permission to send newsletters. An administrator or manager can only send newsletters if his or her account belongs to a group created on the main trade server. |
| Right\_Export | Integer | Permission to export data. |
| Right\_Techsupport | Integer | Access to the technical support tab in the administrator and manager terminals. The permission is obsolete and is no longer used. |
| Right\_Market | Integer | Permission to access the Market of applications in the MetaTrader 5 Administrator. |
| Right\_Accountant | Integer | Permission to work with funds on accounts. |
| Right\_Acc\_Read | Integer | Access to accounts. |
| Right\_Acc\_Technical | Integer | When combined with the "Enable visibility for regular managers" permission, provides greater convenience when working with various testing and technical accounts. Disable the "Enable visibility for regular managers" permission for all technical accounts, and then disable access to technical accounts for the managers who do not configure the platform. Otherwise, such technical accounts can be confusing for managers working with clients. |
| Right\_Acc\_Details | Integer | Access to personal details of accounts. |
| Right\_Acc\_Manager | Integer | Account editing. |
| Right\_Acc\_Delete | Integer | Deleting client accounts via the administrator and manager terminals, and via the Manager API. The Right\_Acc\_Manager permission is required in order to enable this permission. |
| Right\_Acc\_Online | Integer | Getting the current client connections. |
| Right\_Confirm\_Actions | Integer | By default the manager terminal displays a confirmation dialog when performing balance operations on client accounts and closing multiple orders. A manager needs to enter a randomly generated sequence of characters in order to confirm the appropriate action. If this permission is disabled, the above actions will be performed immediately without any confirmation. |
| Right\_Notifications | Integer | Permission to send push notifications to clients' mobile devices from the manager terminal. Messages are sent based on MetaQuotes ID, which is a unique user identifier. To obtain the ID, a user needs to install MetaTrader 5 Mobile for iPhone and Android. For more information please read the MetaTrader 5 Manager user guide. |
| Right\_Trades\_Read | Integer | Viewing trading orders, deals and positions. This right affects the possibility to enable Right\_Trades\_Manager and Right\_Trades\_Dealer permissions. |
| Right\_Trades\_Manager | Integer | Changing any fields of orders, deals and positions in the administrator terminal, and changing position open prices in the manager terminal. |
| Right\_Trades\_Delete | Integer | Deleting any orders, deals and positions via the administrator and manager terminals, and via the Manager API. The Right\_Trades\_Manager permission is required in order to enable this permission. |
| Right\_Trades\_Dealer | Integer | The possibility to perform trading and dealing operations in the manager terminal. |
| Right\_Trades\_Supervisor | Integer | This right allows the manager to view the entire queue of requests received from groups of clients available to the manager, as well as to track processing of requests by other dealers in the manager terminal. Manager works in the "Supervisor" mode without connecting as a dealer in the manager terminal. After connecting as a dealer, the manager will only see the requests that are forwarded to him or her for processing in accordance with the routing rules. |
| Right\_Quotes\_Raw | Integer | If this permission is enabled, the "Show raw quotes" command will appear in the context menu of the Market Watch window in the manager terminal. With this permission, the manager can view quotes without considering spread difference settings set for the manager group. |
| Right\_Quotes | Integer | Permission to throw in quotes |
| Right\_Symbol\_Details | Integer | Permission to change spread and execution mode. |
| Right\_Risk\_Manager | Integer | Permission to receive information about client's aggregate positions and company's coverage positions. |
| Risk\_Group\_Margin | Integer | Permission to configure margin for groups in MetaTrader 5 Manager. |
| Risk\_Group\_Commission | Integer | Permission to configure commissions for groups in MetaTrader 5 Manager. |
| Right\_Reports | Integer | Permission to request and receive various reports on client operations. |
| Right\_Finteza\_Websites | Integer | View Finteza data relating to websites in the Analytics section of the Manager terminal. |
| Right\_Finteza\_Campaigns | Integer | View Finteza data relating to marketing campaigns in the Analytics section of the Manager terminal. |
| Right\_Finteza\_Reports | Integer | The permission is currently not used. |
| Right\_Clients\_Access | Integer | Access to the Clients section in the Administrator and Manager terminals. |
| Right\_Clients\_Create | Integer | Permission to create new client records manually. |
| Right\_Clients\_Edit | Integer | Permission to edit client data, except for documents. |
| Right\_Clients\_Delete | Integer | Permission to delete client records. |
| Right\_Documents\_Access | Integer | Permission to view client documents. |
| Right\_Documents\_Create | Integer | Permission to add general information about documents in client records. |
| Right\_Documents\_Edit | Integer | Permission to edit general information about documents in client records. |
| Right\_Documents\_Delete | Integer | Permission to delete general information about documents from client records. |
| Right\_Documents\_Files\_Add | Integer | Permission to add document files in client records. |
| Right\_Documents\_Files\_Delete | Integer | Permission to delete document files from client records. |
| Right\_Comments\_Access | Integer | Permission to read comments to clients and their documents. |
| Right\_Comments\_Create | Integer | Permission to write comments to clients and their documents. |
| Right\_Comments\_Delete | Integer | Permission to delete comments to clients and their documents. |
| Right\_Admin\_Computer | Integer | Access to the server machine administration menu in the Network section of the Administrator terminal |
| Right\_Subscriptions\_View | Integer | Permission to view existing settings in the Subscriptions section in the Administrator terminal, as well as access to subscription statistics. |
| Right\_Subscriptions\_Edit | Integer | Permission to create, edit and remove settings in the Subscriptions section of the Administrator terminal. |

mt5\_clients

Data about clients is exported to this table. The table contains the following fields:

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| ClientID | Integer | Initial key. Unique entry ID. |
| Timestamp | Integer | A unique values within the table. Used by MetaTrader 5 servers for internal purposes. If the Timestamp of a record has changed, it means that the record has changed. |
| ClientType | Integer | Client type:   * 0 — Not specified * 1 — Individual * 2 — Corporate * 3 — Fund |
| ClientStatus | Integer | Client status:   * 0 — Not registered * 1 — Registered * 2 — not interested * 3 — Not completed * 4 — Completed * 5 — Information * 6 — Rejected * 7 — Approved * 8 — Financed * 9 — Active * 10 — Inactive * 11 — Suspended * 12 — Closed * 13 — Deleted |
| AssignedManager | Integer | The login of the assigned manager. |
| Comment | String | A comment to the client. |
| ComplianceApprovedBy | Integer | The login of the manager by whom the client was approved. |
| ComplianceClientCategory | String | Client compliance category. Currently not used. |
| ComplianceDateApproval | DateTime | The date when the client was approved, in the format of YYYY-MM-DD HH:MM:SS |
| ComplianceDateTermination | DateTime | The date when the provision of services to the client was discontinued, in the format of YYYY-MM-DD HH:MM:SS |
| LeadCampaign | String | The website from which the client came (lead source). |
| LeadSource | String | The name of the marketing campaign, as a result of which the client came (lead campaign). |
| Introducer | String | The login of the user by whom the client was introduced. |
| PersonTitle | String | Client's title. |
| PersonName | String | First name and last name |
| PersonMiddleName | String | Middle name. |
| PersonBirthDate | DateTime | Date of birth, in the format of YYYY-MM-DD HH:MM:SS |
| PersonCitizenship | String | Citizenship. |
| PersonGender | Integer | Gender:   * 0 — Not specified * 1 — male * 2 — Female |
| PersonTaxID | String | Client's Tax ID. |
| PersonDocumentType | String | Document type of document: passport, driver's license, etc. |
| PersonDocumentNumber | String | Document number. |
| PersonDocumentDate | DateTime | Document issue date, in the format of YYYY-MM-DD HH:MM:SS |
| PersonDocumentExtra | String | Additional document information. |
| PersonEmployment | Integer | Employment status:   * 0 — Unemployed * 1 — Employed * 2 — Entrepreneur or self-employed * 3 — Retired * 4 — Student * 5 — Other |
| PersonIndustry | Integer | Employment area:   * 0 — Not specified * 1 — Agriculture, Food and Natural Resources * 2 — Architecture and Construction * 3 — Business Administration and Management * 4 — Art, Audio/Video Technology and Communication * 5 — Education and Training * 6 — State and Administrative Management * 7 — Health * 8 — Tourism and Hospitality * 9 — Information Technology * 10 — Legal and Public Safety, Correction and Protection Services * 11 — Manufacturing * 12 — Marketing and Sales * 13 — Science and Technology * 14 — Engineering and Mathematics * 15 — Transportation, Distribution and Logistics * 16 — other |
| PersonEducation | Integer | Education:   * 0 — Not specified * 1 — Secondary * 2 — Bachelor's degree or equivalent * 3 — Master's degree or equivalent * 4 — PhD or equivalent * 5 — other |
| PersonWealthSource | Integer | Source of income:   * 0 — Employment/business activity * 1 — Savings or investments * 2 — Gift or inheritance * 3 — other |
| PersonAnnualIncome | Fractional number | Annual income. |
| PersonNetWorth | Fractional number | Net assets. |
| PersonAnnualDeposit | Fractional number | Annual deposit. |
| CompanyName | String | Company name. |
| CompanyRegNumber | String | Company registration number. |
| CompanyRegDate | String | company registration date. |
| CompanyRegAuthority | String | Company registration authority. |
| CompanyVat | String | VAT number. |
| CompanyLei | String | LEI number for EMIR reports. |
| CompanyLicenseNumber | String | Company license number. |
| CompanyLicenseAuthority | String | Licensing authority. |
| CompanyCountry | String | Country of incorporation. |
| CompanyAddress | String | Company's legal address. |
| CompanyWebsite | String | Company's webiste. |
| ContactPreferred | Integer | Preferred method of communication:   * 0 — Not specified * 1 — Email * 2 — Telephone * 3 — SMS * 4 — Messenger |
| ContactLanguage | String | Client's language. |
| ContactEmail | String | Client's email. |
| ContactPhone | String | Phone number. |
| ContactMessengers | String | Messengers. |
| ContactSocialNetworks | String | Accounts in social networks. |
| ContactLastDate | DateTime | Last contact date, in the format of YYYY-MM-DD HH:MM:SS |
| AddressCountry | String | Client's country. |
| AddressPostcode | String | Client's postal code. |
| AddressStreet | String | Client's address. |
| AddressState | String | State/region of residence. |
| AddressCity | String | City. |
| ExperienceFX | Integer | Forex trading experience, number of years. |
| ExperienceCFD | Integer | CFD trading experience, number of years. |
| ExperienceFutures | Integer | Futures trading experience, number of years. |
| ExperienceStocks | Integer | Stock trading experience, number of years. |
| ClientOrigin | Integer | How the client record was created:   * 0 — manually * 1 — based on a demo account * 2 — based on a contest account * 3 — based on a preliminary account * 4 — based on a real account |
| ClientOriginLogin | Integer | The number of the account, based on which the client record was created. |

mt5\_documents

Data about client documents is exported to this table. The table contains the following fields:

| **Name** | **Type** | **Description** | |
| --- | --- | --- | --- |
| DocumentID | Integer | Initial key. Unique document ID. | |
| Timestamp | Integer | A unique values within the table. Used by MetaTrader 5 servers for internal purposes. If the Timestamp of a record has changed, it means that the record has changed. | |
| RelatedClient | Integer | The ID of the client to whom the document belongs. Corresponds to Client ID from the mt5\_clients table. | |
| ApprovedDate | DateTime | Document approval date, in the format of YYYY-MM-DD HH:MM:SS | |
| ApprovedBy | Integer | The login of the manager by whom the document was approved. | |
| DateIssue | DateTime | Document issue date, in the format of YYYY-MM-DD HH:MM:SS | |
| DateExpiration | DateTime | Document expiration date, in the format of YYYY-MM-DD HH:MM:SS | |
| DocumentType | Integer | Document type:   * 0 — Other * 1 — Proof of identity * 2 — Proof of address * 3 — Registration address * 4 — CEO's ID document * 5 — Certificate of Registration * 6 — Certificate of Directors * 7 — Certificate of good standing | |
| DocumentName | String | Document name. | |
| DocumentComment | String | Comment to the document. | |
| DocumentStatus | Integer | Document status:   * 0 — New * 1 — Approved * 2 — Rejected * 3 — Archived * 4 — Deleted | |
| Only document data is exported to SQL. Document files themselves are not exported. | | |

mt5\_users

Data on account database is exported to the table. The table contains the following fields:

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| Login | Integer | Primary key. The login of a user. |
| Timestamp | Integer | Unique record within the table. It is used for internal purposes of MetaTrader 5 servers. If the timestamp is changed for a record, it means that the record has been changed. |
| TimestampTrade | Integer | Unique record within the table. It is used for internal purposes of MetaTrader 5 servers. If the timestamp is changed for a record, it means that the record has been changed. |
| Group | String | User group. |
| CertSerialNumber | Integer | The number of a last used certificate for user authorization. |
| Rights | Integer | Flags of the users permissions. Passed using a value of the EnUserRights enumeration (sum of values of appropriate flags). |
| Registration | DateTime | Time of a client record generation in the YYYY-MM-DD HH:MM:SS format. |
| LastAccess | DateTime | The date of the last connection using an account in the YYYY-MM-DD HH:MM:SS format. This field is not updated in real time, in order to save traffic and reduce the load on the platform. The value is only updated when the user connects to the platform, if more than 24 hours have passed since the previous connection. |
| LastPassChange | DateTime | The date of the last password change. |
| LastIP | String | The IP address from which the user last connected to the server. |
| Name | String | The name of the user. Obsolete field. |
| FirstName | String | The first name of the client. |
| LastName | String | The last name of the client. |
| MiddleName | String | The middle name of the client. |
| Company | String | The name of user's company. |
| Account | String | The number of a user's account in an external bank. |
| Country | String | The user's country of residence. |
| Language | Integer | User's language in the format LANGID used in MS Windows (value from Prim.lang.identifier). |
| ClientID | Integer | The identifier of the client, to whom the trading account corresponds. |
| City | String | The user's city of residence. |
| State | String | The user's state (region) of residence. |
| ZIPCode | String | The user's zip code. |
| Address | String | The address of the user. |
| Phone | String | The user's phone number. |
| EMail | String | The email address of the user. |
| ID | String | The number of a user's identity document. |
| Status | String | Client's status. |
| Comment | String | A comment to the user. |
| Color | Integer | The color of the user. This is the color of the user's requests shown when handling the requests via the manager terminal. |
| PhonePassword | String | The user's phone password. |
| Leverage | Integer | The size of a user's leverage. |
| Agent | Integer | Agent account number of the user. |
| Balance | Float | The current balance of a user. |
| Credit | Float | The current amount of funds credited to the user. |
| InterestRate | Float | The amount accrued for the current month calculated based on the annual interest rate. |
| CommissionDaily | Float | The amount of commissions charged from the user for a day. |
| CommissionMonthly | Float | The total amount of commissions charged from the user for the current month. |
| BalancePrevDay | Float | The value of the user's balance as of the end of the previous day. |
| BalancePrevMonth | Float | The value of a user's balance as of the end of the previous trading month. |
| EquityPrevDay | Float | The user's equity as of the end of the previous day. |
| EquityPrevMonth | Float | The value of the user's equity as of the end of the previous trading month. |
| TradeAccounts | String | Account numbers in external trading systems and gateway identifiers used for working with that systems. The string format is: gateway\_ID=account\_number|gateway\_ID=account\_number... |
| MQID | String | MetaQuotes ID of the user. |
| LeadCampaign | String | Name of the marketing campaign a client was attracted by. |
| LeadSource | String | Lead source (address of the website a client has come from). |
| ApiData | String | User data which can be added via MetaTrader 5 API. Sample user data entry:  [{pos:0,app\_id:1,valInt:500,valUInt:500,valDbl:0.00000000}]. It specifies the user data index, the ID of the application that added it, as well as the data of three types: Int, UInt and double. The string may contain up to 16 such entries. |
| LimitOrders | Integer | The maximum number of active (placed) pending orders allowed on the account. |
| LimitPositions | Integer | Maximum value of open positions allowed on the account. |

## Enumerations

To pass information about users the following enumerations are used:

* EnUsersRights
* EnUsersPasswords
* EnUsersConnectionTypes
* EnSoActivation

### EnUsersRights

Permissions that can be given to a user are enumerated in EnUsersRights.

| **ID** | **Value** | **Description** |
| --- | --- | --- |
| USER\_RIGHT\_NONE | 0x0000000000000000 | No permissions. |
| USER\_RIGHT\_ENABLED | 0x0000000000000001 | The user is allowed to connect. |
| USER\_RIGHT\_PASSWORD | 0x0000000000000002 | The user is allowed to change the password. |
| USER\_RIGHT\_TRADE\_DISABLED | 0x0000000000000004 | Trading is disabled for the user. |
| USER\_RIGHT\_INVESTOR | 0x0000000000000008 | Service value for internal use. |
| USER\_RIGHT\_CONFIRMED | 0x0000000000000010 | User's certificate is confirmed. |
| USER\_RIGHT\_TRAILING | 0x0000000000000020 | The user is allowed to use trailing stop. |
| USER\_RIGHT\_EXPERT | 0x0000000000000040 | The user is allowed to use Expert Advisors. |
| USER\_RIGHT\_OBSOLETE | 0x0000000000000080 | The flag is obsolete and is not used. |
| USER\_RIGHT\_REPORTS | 0x0000000000000100 | The user is allowed to receive daily reports. If the permission is not enabled, daily reports are neither generated nor sent for the account. |
| USER\_RIGHT\_READONLY | 0x0000000000000200 | Service value for internal use. |
| USER\_RIGHT\_RESET\_PASS | 0x0000000000000400 | The user must change password during the next connection. |
| USER\_RIGHT\_OTP\_ENABLED | 0x0000000000000800 | The user can use OTP authentication. |
| USER\_RIGHT\_SPONSORED\_HOSTING | 0x0000000000002000 | Brokers can pay the virtual hosting fee for their customers. The service is extremely important for traders, and the opportunity to receive a VPS for free can give them a good reason to choose your company over competitors. The availability of a broker-sponsored VPS is controlled at the individual account level. Only if this flag is enabled, the appropriate payment plan will be shown to the trader in the client terminal. For more details, please read the appropriate section. |
| USER\_RIGHT\_API\_ENABLED | 0x0000000000004000 | The user is allowed to connect via the Web API. |
| USER\_RIGHT\_TECHNICAL | 0x0000000000010000 | Permission for convenient work with technical accounts. Disable it for testing accounts to hide them from all managers who do not have special access to technical accounts. Such technical accounts can be confusing for the managers working with clients, in which case hiding them can be useful.  This permission affects the account visibility in the general account list in the Administrator and Manager terminals, as well as in the list of online accounts in the Manager terminal. |
| USER\_RIGHT\_EXCLUDE\_REPORTS | 0x0000000000020000 | Allows excluding an account from server reports. Like the previous permission, it is used for convenient management of technical accounts. |

### EnUsersPasswords

Types of passwords are enumerated in EnUsersPasswords.

| **ID** | **Value** | **Description** |
| --- | --- | --- |
| USER\_PASS\_MAIN | 0 | The master password. |
| USER\_PASS\_INVESTOR | 1 | The investor password. |
| USER\_PASS\_API | 2 | API password. |

### EnUsersConnectionTypes

Types of client connections are enumerated in EnUsersConnectionTypes:

| **ID** | **Value** | **Description** |
| --- | --- | --- |
| USER\_TYPE\_CLIENT | 0 | Connection from a client terminal. |
| USER\_TYPE\_CLIENT\_WINMOBILE | 1 | Connection from a mobile terminal for Windows Mobile (under development). |
| USER\_TYPE\_CLIENT\_WINPHONE | 2 | Connection from a mobile terminal for Windows Phone 7 (under development). |
| USER\_TYPE\_CLIENT\_API\_WEB | 3 | Connection via the client Web API. |
| USER\_TYPE\_CLIENT\_IPHONE | 4 | Connection from a mobile terminal for iPhone. |
| USER\_TYPE\_CLIENT\_ANDROID | 5 | Connection from a mobile terminal for Android (under development). |
| USER\_TYPE\_CLIENT\_BLACKBERRY | 6 | Connection from a mobile terminal for BlackBerry (under development). |
| USER\_TYPE\_ADMIN | 32 | Connection from an administrator terminal. |
| USER\_TYPE\_MANAGER | 33 | Connection from a manager terminal. |
| USER\_TYPE\_MANAGER\_API | 34 | Connection via the manager interface of the Manager API. |
| USER\_TYPE\_ADMIN\_API | 36 | Connection via the administrator interface of the Manager API. |
| USER\_TYPE\_MANAGER\_API\_WEB | 37 | Connection via the manager Web API. |

### EnSoActivation

The account status as per the minimum amount of funds on the account required to maintain trading positions are enumerated in EnSoActivation.

| **ID** | **Value** | **Description** |
| --- | --- | --- |
| ACTIVATION\_NONE | 0 | None. |
| ACTIVATION\_MARGIN\_CALL | 1 | Margin call. |
| ACTIVATION\_STOP\_OUT | 2 | Stop out. |

# mt5\_orders

Data on open orders is exported to the table. The table contains the following fields:

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| Order | Integer number | Primary key. Order ticket. |
| Timestamp | Integer number | Unique record within the table. It is used for internal purposes of MetaTrader 5 servers. If the timestamp is changed for a record, it means that the record has been changed. |
| ExternalID | String | The order ID in external trading systems. |
| Login | Integer number | The login of the client, to whom the order belongs. |
| Dealer | Integer number | The login of a dealer, who has processed an order. |
| Symbol | String | The symbol of an order. |
| Digits | Integer number | The number of decimal places in the price of an order. |
| DigitsCurrency | Integer number | The number of decimal places the deposit currency of the client who has placed the order. |
| ContractSize | Fractional number | The contract size of the symbol, for which an order was placed. |
| State | Integer number | The current state of an order. Passed as a value of the EnOrderMode enumeration. |
| Reason | Integer number | The reason for placing the order. Passed as a value of the EnOrderReason enumeration. |
| TimeSetup | DateTime | Order placement time in the YYYY-MM-DD HH:MM:SS format. |
| TimeSetupMsc | DateTime | Order placement time in milliseconds in the YYYY-MM-DD HH:MM:SS.MS format. |
| TimeExpiration | DateTime | Order expiration time in the YYYY-MM-DD HH:MM:SS format. |
| TimeDone | DateTime | Order execution time in the YYYY-MM-DD HH:MM:SS format. |
| TimeDoneMsc | DateTime | Order execution time in milliseconds in the YYYY-MM-DD HH:MM:SS.MS format. |
| Type | Integer number | Order type. Passed as a value of the EnOrderType enumeration. |
| TypeFill | Integer number | Order filling type. Passed as a value of the EnOrderFilling enumeration. |
| TypeTime | Integer number | Order expiration type. Passed in a value of the EnOrderTime enumeration. |
| PriceOrder | Fractional number | Order price. |
| PriceTrigger | Fractional number | The order triggering price. |
| PriceCurrent | Fractional number | The current price of the symbol, for which an order has been placed. |
| PriceSL | Fractional number | The Stop Loss level of an order. |
| PriceTP | Fractional number | The Take Profit level of an order. |
| VolumeInitial | Integer number | The initial order volume. One unit corresponds to 1/10000 lot. |
| VolumeInitialExt | Integer number | The initial order volume with an extended accuracy. One unit corresponds to 1/100000000 lot. |
| VolumeCurrent | Integer number | The current unfilled volume of an order. One unit corresponds to 1/10000 lot. |
| VolumeCurrentExt | Integer number | The current unfilled order volume with an extended accuracy. One unit corresponds to 1/100000000 lot. |
| ExpertID | Fractional number | The ID of the Expert Advisor that has placed the order. |
| PositionID | Fractional number | The position ID (ticket) set in the order. |
| PositionByID | Fractional number | The opposite position ID (ticket) set in the order. The property is set for Close By operations (OP\_CLOSE\_BY). The ticket of the position that is closed by the opposite one is set in PositionID. |
| Comment | String | A comment to an order. |
| ActivationMode | Integer number | Order activation type. Passed in a value of the EnOrderActivation enumeration. |
| ActivationTime | DateTime | Order activation time in the YYYY-MM-DD HH:MM:SS format. |
| ActivationPrice | Fractional number | The price, at which the order was activated. |
| ActivationFlags | Integer number | Order activation flags. Passed as a value of the EnTradeActivationFlags enumeration (sum of values of appropriate flags). |
| RateMargin | Fractional number | The rate of conversion of the margin currency of the symbol to the deposit currency of the user, which is used when calculating the margin requirements for the order. |
| ApiData | String | User data which can be added via MetaTrader 5 API. Sample user data entry:  [{pos:0,app\_id:1,valInt:500,valUInt:500,valDbl:0.00000000}]. It specifies the user data index, the ID of the application that added it, as well as the data of three types: Int, UInt and double. The string may contain up to 16 such entries. |

## Enumerations

To pass information about orders the following enumerations are used:

* EnOrderType
* EnOrderFilling
* EnOrderTime
* EnOrderState
* EnOrderActivation
* EnOrderReason
* EnTradeActivationFlags

### EnOrderType

Types of trade orders are listed in EnOrderType.

| **ID** | **Value** | **Description** |
| --- | --- | --- |
| OP\_BUY | 0 | A Buy order. |
| OP\_SELL | 1 | A Sell order. |
| OP\_BUY\_LIMIT | 2 | A Buy Limit order. |
| OP\_SELL\_LIMIT | 3 | A Sell Limit order. |
| OP\_BUY\_STOP | 4 | A Buy Stop order. |
| OP\_SELL\_STOP | 5 | A Sell Stop order. |
| OP\_BUY\_STOP\_LIMIT | 6 | A Buy Stop Limit . |
| OP\_SELL\_STOP\_LIMIT | 7 | A Sell Stop Limit order. |
| OP\_CLOSE\_BY | 8 | A close by order — closing two oppositely directed positions at a single symbol. This type of operations is used only for the hedging position accounting system (MARGIN\_MODE\_RETAIL\_HEDGED). |

### EnOrderFilling

Types of order filling are listed in EnOrderFilling.

| **ID** | **Value** | **Description** |
| --- | --- | --- |
| ORDER\_FILL\_FOK | 0 | Fill or Kill. The order must be filled completely or canceled. This type of filling is automatically set for the instant and request execution. |
| ORDER\_FILL\_IOC | 1 | Immediate or Cancel. An order can be filled partially and the residual volume is canceled. This type of filling is only available for the stock and market execution. |
| ORDER\_FILL\_RETURN | 2 | Return the remainder to the queue. This mode is intended only for pending orders. |

### EnOrderTime

Types of order expiration are listed in EnOrderTime.

| **ID** | **Value** | **Description** |
| --- | --- | --- |
| ORDER\_TIME\_GTC | 0 | Good till Canceled. |
| ORDER\_TIME\_DAY | 1 | Intraday. |
| ORDER\_TIME\_SPECIFIED | 2 | Specified time. |
| ORDER\_TIME\_SPECIFIED\_DAY | 3 | Specified day. An order expires at 00:00 of the specified day or the nearest trading time. |

### EnOrderState

Possible staes of orders are listed in EnOrderState.

| **ID** | **Value** | **Description** |
| --- | --- | --- |
| ORDER\_STATE\_STARTED | 0 | Started. |
| ORDER\_STATE\_PLACED | 1 | Placed. |
| ORDER\_STATE\_CANCELED | 2 | Canceled. |
| ORDER\_STATE\_PARTIAL | 3 | Partially filled. |
| ORDER\_STATE\_FILLED | 4 | Filled. |
| ORDER\_STATE\_REJECTED | 5 | Rejected. |
| ORDER\_STATE\_EXPIRED | 6 | Expired. |
| ORDER\_STATE\_REQUEST\_ADD | 7 | The order passed (by the gateway) to be placed. This state is used for notifying that a request for placing the order is being already processed. |
| ORDER\_STATE\_REQUEST\_MODIFY | 8 | The order passed (by the gateway) to be modified. This state is used for notifying that a request for modifying the order is being already processed. |
| ORDER\_STATE\_REQUEST\_CANCEL | 9 | The order passed (by the gateway) to be deleted. This state is used for notifying that a request for deleting the order is being already processed. |

### EnOrderActivation

Types of order activation are listed in EnOrderActivation.

| **ID** | **Value** | **Description** |
| --- | --- | --- |
| ACTIVATION\_NONE | 0 | Not activated. |
| ACTIVATION\_PENDING | 1 | Activation of a pending order. |
| ACTIVATION\_STOPLIMIT | 2 | Activation of a Stop Limit order. |
| ACTIVATION\_EXPIRATION | 3 | Cancellation of an order upon expiration. |
| ACTIVATION\_STOPOUT | 4 | Order is being removed because of a stop out. |

### EnOrderReason

Types of reasons for order placing are listed in EnOrderReason.

| **ID** | **Value** | **Description** |
| --- | --- | --- |
| ORDER\_REASON\_CLIENT | 0 | Order placed by a client manually through the client terminal. |
| ORDER\_REASON\_EXPERT | 1 | Order placed by a client with using an Expert Advisor. |
| ORDER\_REASON\_DEALER | 2 | Order placed by a dealer through the manager terminal. |
| ORDER\_REASON\_SL | 3 | Order placed as a result of Stop Loss activation. |
| ORDER\_REASON\_TP | 4 | Order placed as a result of Take Profit activation. |
| ORDER\_REASON\_SO | 5 | Order placed when the client reached the Stop-Out level. |
| ORDER\_REASON\_ROLLOVER | 6 | Order placed when reopening a position for charging swaps. |
| ORDER\_REASON\_EXTERNAL\_CLIENT | 7 | Order placed by a client from an external trading system. |
| ORDER\_REASON\_VMARGIN | 8 | Order placed for accruing variation margin. |
| ORDER\_REASON\_GATEWAY | 9 | Order placed by a MetaTrader 5 gateway that had connected to the trading platform. |
| ORDER\_REASON\_SIGNAL | 10 | Order placed as a result of copying a trade signal  according to a subscription in the client terminal. |
| ORDER\_REASON\_SETTLEMENT | 11 | Order placed as a result of performing operations connected with the settlement of a futures contract/option. Not used at the moment. |
| ORDER\_REASON\_TRANSFER | 12 | Order placed due to transferring a position at the settlement price to a new symbol with the same underlying asset. Not used at the moment. |
| ORDER\_REASON\_SYNC | 13 | Order placed as a result of synchronization of an account's trade state with an external system. |
| ORDER\_REASON\_EXTERNAL\_SERVICE | 14 | Order placed from an external trading system for technical reasons (for example, to correct the trade state of a client). |
| ORDER\_REASON\_MIGRATION | 15 | Order created as a result of importing trade operations from a MetaTrader 4 server. |
| ORDER\_REASON\_MOBILE | 16 | The order was created via the MetaTrader 5 mobile terminal for Android or iPhone. |
| ORDER\_REASON\_WEB | 17 | The order was created via the web terminal. |
| ORDER\_REASON\_SPLIT | 18 | The order was created as a result of a symbol split. |

### EnTradeActivationFlags

Types of activation flags that can be assigned to orders upon forming a trade execution are listed in EnTradeActivationFlags:

| **ID** | **Value** | **Description** |
| --- | --- | --- |
| ACTIV\_FLAGS\_NO\_LIMIT | 0x01 | Do not handle reaching of the Limit level. |
| ACTIV\_FLAGS\_NO\_STOP | 0x02 | Do not handle the reaching of the stop level. |
| ACTIV\_FLAGS\_NO\_SLIMIT | 0x04 | Do not handle reaching of the Stop-Limit level. |
| ACTIV\_FLAGS\_NO\_SL | 0x08 | Do not handle activation upon Stop Loss. |
| ACTIV\_FLAGS\_NO\_TP | 0x10 | Do not handle activation upon Take Profit. |
| ACTIV\_FLAGS\_NO\_SO | 0x20 | Do not handle activation upon Stop-Out. |
| ACTIV\_FLAGS\_NO\_EXPIRATION | 0x40 | Do not handle order cancellation upon expiration. |
| ACTIV\_FLAGS\_NONE | 0x00 | No flags. |

Flags of orders are inherited by positions created as a result of their execution.

# mt5\_orders\_history

Data on closed orders is exported to the table. If "Export history orders and deals into separate tables by years" option is enabled in settings, closed orders for each year are exported to a separate table. A year is specified in the table heading, for example, mt5\_orders\_2012. If the option is disabled, all closed orders are exported to a single mt5\_orders\_history table.

The table contains the following fields:

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| Order | Integer number | Primary key. Order ticket. |
| Timestamp | Integer number | Unique record within the table. It is used for internal purposes of MetaTrader 5 servers. If the timestamp is changed for a record, it means that the record has been changed. |
| ExternalID | String | The order ID in external trading systems. |
| Login | Integer number | The login of the client, to whom the order belongs. |
| Dealer | Integer number | The login of a dealer, who has processed an order. |
| Symbol | String | The symbol of an order. |
| Digits | Integer number | The number of decimal places in the price of an order. |
| DigitsCurrency | Integer number | The number of decimal places the deposit currency of the client who has placed the order. |
| ContractSize | Fractional number | The contract size of the symbol, for which an order was placed. |
| State | Integer number | The current state of an order. Passed as a value of the EnOrderMode enumeration. |
| Reason | Integer number | The reason for placing the order. Passed as a value of the EnOrderReason enumeration. |
| TimeSetup | DateTime | Order placement time in the YYYY-MM-DD HH:MM:SS format. |
| TimeSetupMsc | Integer | Order placement time in milliseconds in the YYYY-MM-DD HH:MM:SS.MS format. |
| TimeExpiration | DateTime | Order expiration time in the YYYY-MM-DD HH:MM:SS format. |
| TimeDone | DateTime | Order execution time in the YYYY-MM-DD HH:MM:SS format. |
| TimeDoneMsc | Integer | Order execution time in milliseconds in the YYYY-MM-DD HH:MM:SS.MS format. |
| Type | Integer number | Order type. Passed as a value of the EnOrderType enumeration. |
| TypeFill | Integer number | Order filling type. Passed as a value of the EnOrderFilling enumeration. |
| TypeTime | Integer number | Order expiration type. Passed in a value of the EnOrderTime enumeration. |
| PriceOrder | Fractional number | Order price. |
| PriceTrigger | Fractional number | The order triggering price. |
| PriceCurrent | Fractional number | The current price of the symbol, for which an order has been placed. |
| PriceSL | Fractional number | The Stop Loss level of an order. |
| PriceTP | Fractional number | The Take Profit level of an order. |
| VolumeInitial | Integer number | The initial order volume. One unit corresponds to 1/10000 lot. |
| VolumeInitialExt | Integer number | The initial order volume with an extended accuracy. One unit corresponds to 1/100000000 lot. |
| VolumeCurrent | Integer number | The current unfilled volume of an order. One unit corresponds to 1/10000 lot. |
| VolumeCurrentExt | Integer number | The current unfilled order volume with an extended accuracy. One unit corresponds to 1/100000000 lot. |
| ExpertID | Fractional number | The ID of the Expert Advisor that has placed the order. |
| PositionID | Fractional number | The position ID (ticket) set in the order. |
| PositionByID | Fractional number | The opposite position ID (ticket) set in the order. The property is set for Close By operations (OP\_CLOSE\_BY). The ticket of the position that is closed by the opposite one is set in PositionID. |
| Comment | String | A comment to an order. |
| ActivationMode | Integer number | Order activation type. Passed in a value of the EnOrderActivation enumeration. |
| ActivationTime | DateTime | Order activation time in the YYYY-MM-DD HH:MM:SS format. |
| ActivationPrice | Fractional number | The price, at which the order was activated. |
| ActivationFlags | Integer number | Order activation flags. Passed as a value of the EnTradeActivationFlags enumeration (sum of values of appropriate flags). |
| RateMargin | Fractional number | The rate of conversion of the margin currency of the symbol to the deposit currency of the user, which is used when calculating the margin requirements for the order. |
| ApiData | String | User data which can be added via MetaTrader 5 API. Sample user data entry:  [{pos:0,app\_id:1,valInt:500,valUInt:500,valDbl:0.00000000}]. It specifies the user data index, the ID of the application that added it, as well as the data of three types: Int, UInt and double. The string may contain up to 16 such entries. |

# mt5\_positions

Data on positions is exported to the table. The table contains the following fields:

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| Position\_ID | Integer | Primary key. Internal position identifier used by the backup server to export data. Use the Position field to identify positions in your reports ans databases. The Position field contains the real identifier (ticket) assigned to the position in MetaTrader 5. The Position\_ID field is only used for internal purposes and is not related with trading operation tickets. |
| Timestamp | Integer | Unique record within the table. It is used for internal purposes of MetaTrader 5 servers. If the timestamp is changed for a record, it means that the record has been changed. |
| Login | Integer | The login of the client, to whom the trade position belongs. |
| Symbol | String | The symbol of a trade position. |
| Action | Integer | Position type. Passed in a value of the EnPositionAction enumeration. |
| Digits | Integer | The number of decimal places in the price of a position. |
| DigitsCurrency | Integer | The number of decimal places the deposit currency of the client who has opened the position. |
| Reason | Integer | The reason for position opening. Passed in a value of the EnPositionReason enumeration. |
| ContractSize | Float | The contract size of the symbol, for which a position is opened. |
| Position | Integer | The ticket (unique identifier) of a trade position in a MetaTrader 5 platform. |
| ExternalID | String | The position ticket (unique number) in an external trading system. |
| TimeCreate | DateTime | Time of position creation, in seconds that have elapsed since 01.01.1970. |
| TimeUpdate | DateTime | Time of the last modification of a position, in seconds that have elapsed since 01.01.1970. The modification time of a position is the time of the last modification of its volume. Virtually, it is the time of the last deal performed by the financial instrument that corresponds to that position. |
| TimeCreateMsc | DateTime | Position generation time in the YYYY-MM-DD HH:MM:SS.MS format. |
| TimeUpdateMsc | DateTime | Time of a trading position last change in the YYYY-MM-DD HH:MM:SS.MS format. The modification time of a position is the time of the last modification of its volume. Virtually, it is the time of the last deal performed by the financial instrument that corresponds to that position. |
| PriceOpen | Float | The weighted average open price of a position. Calculated by the following formula: (price of deal 1 \* volume of deal 1 1 + ... + price of deal N \* volume of deal N) / (volume of deal 1 + ... + volume of deal N). |
| PriceCurrent | Float | The current price of the symbol, for which a trade position has been opened. |
| PriceSL | Float | The Stop Loss level of a trade position. |
| PriceTP | Float | The Take Profit level of a trade position. |
| Volume | Integer | The volume of a trade position. One unit corresponds to 1/10000 lot. |
| VolumeExt | Integer | The trade position volume with an extended accuracy. One unit corresponds to 1/100000000 lot. |
| Profit | Float | Returns of a trade position in deposit currency. |
| Storage | Float | The swap size for a position in deposit currency. |
| RateProfit | Float | The exchange rate of the profit currency of a position to the deposit currency of a client group. |
| RateMargin | Float | The exchange rate of the margin currency of a position to the client's deposit currency. |
| ExpertID | Integer | The ID of the Expert Advisor that has opened the position. |
| ExpertPositionID | Integer | Position ID. |
| Comment | String | A comment to a position. |
| Dealer | Integer | The login of a dealer, who has processed an order, which opened the position. |
| ActivationMode | Integer | Position activation type. Passed in a value of the EnActivation enumeration. |
| ActivationTime | DateTime | Position activation time in the YYYY-MM-DD HH:MM:SS.MS format. |
| ActivationPrice | Float | Position activation price. |
| ActivationFlags | Integer | Position activation flags. Passed as a value of the EnTradeActivationFlags enumeration (sum of values of appropriate flags). |
| ApiData | String | User data which can be added via MetaTrader 5 API. Sample user data entry:  [{pos:0,app\_id:1,valInt:500,valUInt:500,valDbl:0.00000000}]. It specifies the user data index, the ID of the application that added it, as well as the data of three types: Int, UInt and double. The string may contain up to 16 such entries. |

## Enumerations

To pass information about positions the following enumerations are used:

* EnPositionAction
* EnActivation
* EnTradeActivationFlags
* EnPositionReason

### EnPositionAction

Types of positions are listed in EnPositionAction.

| **ID** | **Value** | **Description** |
| --- | --- | --- |
| POSITION\_BUY | 0 | Buy. |
| POSITION\_SELL | 1 | Sell. |

### EnActivation

Types of position activation are listed in the EnActivation enumeration.

| **ID** | **Value** | **Description** |
| --- | --- | --- |
| ACTIVATION\_NONE | 0 | None. |
| ACTIVATION\_SL | 1 | Stop Loss. |
| ACTIVATION\_TP | 2 | Take Profit |
| ACTIVATION\_STOPOUT | 3 | Stop Out. |

### EnTradeActivationFlags

Flags of trade position activation are listed in the EnTradeActivationFlags enumeration:

| **ID** | **Value** | **Description** |
| --- | --- | --- |
| ACTIV\_FLAGS\_NO\_LIMIT | 0x01 | Do not handle reaching of the Limit level. |
| ACTIV\_FLAGS\_NO\_STOP | 0x02 | Do not handle the reaching of the stop level. |
| ACTIV\_FLAGS\_NO\_SLIMIT | 0x04 | Do not handle reaching of the Stop-Limit level. |
| ACTIV\_FLAGS\_NO\_SL | 0x08 | Do not handle activation upon Stop Loss. |
| ACTIV\_FLAGS\_NO\_TP | 0x10 | Do not handle activation upon Take Profit. |
| ACTIV\_FLAGS\_NO\_SO | 0x20 | Do not handle activation upon Stop-Out. |
| ACTIV\_FLAGS\_NO\_EXPIRATION | 0x40 | Do not handle order cancellation upon expiration. |
| ACTIV\_FLAGS\_NONE | 0x00 | No flags. |

Activation flags are inherited from the orders, as a result of which the position is created.

### EnPositionReason

Reasons for position opening are enumerated in EnPositionReason:

| **ID** | **Value** | **Description** |
| --- | --- | --- |
| POSITION\_REASON\_CLIENT | 0 | Position opened manually by a client from the client terminal. |
| POSITION\_REASON\_EXPERT | 1 | Position opened by a client using an Expert Advisor. |
| POSITION\_REASON\_DEALER | 2 | Position opened by a dealer through the manager terminal. |
| POSITION\_REASON\_SL | 3 | Not used for positions. |
| POSITION\_REASON\_TP | 4 | Not used for positions. |
| POSITION\_REASON\_SO | 5 | Not used for positions. |
| POSITION\_REASON\_ROLLOVER | 6 | Position reopened to charge swaps. |
| POSITION\_REASON\_EXTERNAL\_CLIENT | 7 | Position opened from an external trading system. |
| POSITION\_REASON\_VMARGIN | 8 | Not used for positions. |
| POSITION\_REASON\_GATEWAY | 9 | Position opened by a MetaTrader 5 gateway connected to the platform. |
| POSITION\_REASON\_SIGNAL | 10 | Position opened as a result of copying a trading signal according to the subscription in the client terminal. |
| POSITION\_REASON\_SETTLEMENT | 11 | Position opened as a result of operations associated with a futures/option delivery date. It is currently not used. |
| POSITION\_REASON\_TRANSFER | 12 | Position opened as a result of transferring a position with a calculated price to a new symbol with the same underlying asset. |
| POSITION\_REASON\_SYNC | 13 | Position opened while synchronizing a trading account state with an external system. |
| POSITION\_REASON\_EXTERNAL\_SERVICE | 14 | Position opened in the external trading system for service purposes (e.g. to correct a trading state). |
| POSITION\_REASON\_MIGRATION | 15 | Position opened as a result of import of clients' trading operations from the MetaTrader 4 server. |
| POSITION\_REASON\_MOBILE | 16 | Position opened via the MetaTrader 5 mobile terminal for Android or iPhone. |
| POSITION\_REASON\_WEB | 17 | Position opened via the web terminal. |
| POSITION\_REASON\_SPLIT | 18 | Position opened as a result of a symbol split. |

# mt5\_deals

Data on deals is exported to the table. If "Export history orders and deals into separate tables by years" option is enabled in settings, deals for each year are exported to a separate table. A year is specified in the table heading, for example, mt5\_deals\_2012. If the option is disabled, all deals are exported to a single mt5\_deals table.

The table contains the following fields:

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| Deal | Integer number | Primary key. The ticket of a deal. |
| Timestamp | Integer number | Unique record within the table. It is used for internal purposes of MetaTrader 5 servers. If the timestamp is changed for a record, it means that the record has been changed. |
| Login | Integer number | The login of the client, to whom the deal belongs. |
| Dealer | Integer number | The login of a dealer, who has processed a deal. |
| Order | Integer number | The ticket of the order, as a result of which a deal was executed. |
| Action | Integer number | Type of action performed with a deal. Passed in a value of the EnDealAction enumeration. |
| Entry | Integer number | Deal direction. Passed in a value of the EnEntryFlags enumeration. |
| Digits | Integer number | The number of decimal places in the price of a deal. |
| DigitsCurrency | Integer number | The number of decimal places the deposit currency of the client who has executed the deal. |
| ContractSize | Float | The contract size of the symbol, for which a deal was executed. |
| Time | DateTime | Trade execution time in the YYYY-MM-DD HH:MM:SS format. |
| Symbol | String | The symbol, for which a deal is executed. |
| Price | Float | The price of the deal. |
| PriceSL | Float | The Stop Loss level of a deal. Stop Loss values for entry and reversal deals are set in accordance with the Stop Loss of orders, which initiated these deals. The Stop Loss values ​​of appropriate positions as of the time of position closing are used for exit deals. |
| PriceTP | Float | Take Profit values for entry and reversal deals are set in accordance with the Take Profit of orders, which initiated these deals. The Take Profit values ​​of appropriate positions as of the time of position closing are used for exit deals. |
| Volume | Integer number | The deal volume. One unit corresponds to 1/10000 lot. |
| VolumeExt | Integer number | The deal volume with an extended accuracy. One unit corresponds to 1/100000000 lot. |
| VolumeClosed | Integer number | The position volume that was closed by the deal. One unit corresponds to 1/10000 lot. |
| VolumeClosedExt | Integer number | The extended accuracy volume of a position that was closed by this deal. One unit corresponds to 1/100000000 lot. |
| Profit | Float | Profit from a deal. |
| Storage | Float | The swap size for a deal. |
| Commission | Float | The amount of commission charged for a deal. |
| Fee | Float | Fee per deal. |
| RateProfit | Float | The exchange rate of the profit currency of a deal to the deposit currency of a client group. |
| RateMargin | Float | The exchange rate of the margin currency of a deal to the client's deposit currency. |
| ExpertID | Integer number | The ID of the Expert Advisor that has executed a deal. |
| PositionID | Float | The position identifier (ticket) for a deal. |
| Comment | String | Comment to a deal. |
| ProfitRaw | Float | The amount of return resulting from a deal. Return is specified in the profit currency of the symbol, for which the deal is executed. |
| PricePosition | Float | The price of the position closed with this deal. |
| TickValue | Float | The tick price for a deal. |
| TickSize | Float | The tick size for a deal. |
| Flags | Integer number | The common flags of a deal. This parameter is reserved for future use. |
| Reason | Integer number | The reason for performing a deal. Passed in a value of the EnDealReason enumeration. |
| Gateway | String | The ID of a gateway, using which a deal was performed. |
| PriceGateway | Float | The price that was actually used for performing a deal through a gateway in an external trading system without taking in consideration its price transformation settings of the gateway. |
| MarketBid | Float | The market Bid price as at the time of deal execution by the server. The field is only filled for the deals which were created after the platform was updated to build 2890 or higher. For earlier deals, the value will be zero. |
| MarketAsk | Float | The market Ask price as at the time of deal execution by the server. The field is only filled for the deals which were created after the platform was updated to build 2890 or higher. For earlier deals, the value will be zero. |
| MarketLast | Float | The market Last price as at the time of deal execution by the server. The field is only filled for the deals which were created after the platform was updated to build 2890 or higher. For earlier deals, the value will be zero. |
| TimeMsc | DateTime | Trade execution time in the YYYY-MM-DD HH:MM:SS.MS format. |
| ApiData | String | User data which can be added via MetaTrader 5 API. Sample user data entry:  [{pos:0,app\_id:1,valInt:500,valUInt:500,valDbl:0.00000000}]. It specifies the user data index, the ID of the application that added it, as well as the data of three types: Int, UInt and double. The string may contain up to 16 such entries. |

## Enumerations

To pass information about deals the following enumerations are used:

* EnDealAction
* EnEntryFlags
* EnDealReason

### EnDealAction

Types of actions performed by a deal are enumerated in EnDealAction.

| **ID** | **Value** | **Description** |
| --- | --- | --- |
| DEAL\_BUY | 0 | A Buy deal. |
| DEAL\_SELL | 1 | A Sell deal. |
| DEAL\_BALANCE | 2 | Balance operation. |
| DEAL\_CREDIT | 3 | Credit operation. |
| DEAL\_CHARGE | 4 | Additional charges/withdrawals. |
| DEAL\_CORRECTION | 5 | Correcting operations. |
| DEAL\_BONUS | 6 | Bonuses. |
| DEAL\_COMMISSION | 7 | Commission. |
| DEAL\_COMMISSION\_DAILY | 8 | Daily commission. |
| DEAL\_COMMISSION\_MONTHLY | 9 | Monthly commission. |
| DEAL\_AGENT\_DAILY | 10 | Daily agent commission. |
| DEAL\_AGENT\_MONTHLY | 11 | Daily agent commission. |
| DEAL\_INTERESTRATE | 12 | Accrual of annual interest. |
| DEAL\_BUY\_CANCELED | 13 | A canceled Buy deal. Using the IMTExecution::TE\_DEAL\_CANCEL trade execution, the Gateway API can notify the platform about the cancellation of a previously executed deal in the external trading system. In this case the type of the earlier executed Buy trade is replaced with this one. The profit/loss of a trade is cleared. Then the client's position is recalculated and the appropriate profit/loss is added/subtracted as a separate balance operation. Deal cancellation does not change the client's order history. Deal cancellation does not entail changes in client's orders history. A deal of the DEAL\_BUY\_CANCELED type is not included into the calculation of the financial state of account and is not taken into account in recalculated positions. |
| DEAL\_SELL\_CANCELED | 14 | A canceled Sell deal. Using the IMTExecution::TE\_DEAL\_CANCEL trade execution, the Gateway API can notify the platform about the cancellation of a previously executed deal in the external trading system. In this case the type of the earlier executed Buy trade is replaced with this one. The profit/loss of a trade is cleared. Then the client's position is recalculated and the appropriate profit/loss is added/subtracted as a separate balance operation. Deal cancellation does not change the client's order history. Deal cancellation does not entail changes in client's orders history. A deal of the DEAL\_SELL\_CANCELED type is not included into the calculation of the financial state of account and is not taken into account in recalculated positions. |
| DEAL\_DIVIDEND | 15 | Dividend operations. |
| DEAL\_DIVIDEND\_FRANKED | 16 | Franked (non-taxable) dividend operations (tax is paid by a company, not a client). |
| DEAL\_TAX | 17 | Charging a tax. |
| DEAL\_AGENT | 18 | Charging an agent commission. Used in case of an instant commission charge to an agent (every time the agent's client performs a deal). |
| DEAL\_SO\_COMPENSATION | 19 | An operation connected with the compensation of a negative account after the Stop Out event. |

### EnEntryFlags

Types of actions performed by a deal with respect to positions are enumerated in EnEntryFlags.

| **ID** | **Value** | **Description** |
| --- | --- | --- |
| ENTRY\_IN | 0 | Entering the market or adding the volume. |
| ENTRY\_OUT | 1 | Exit from the market or partial closure. |
| ENTRY\_INOUT | 2 | Reversal. |
| ENTRY\_OUT\_BY | 3 | Close by — a simultaneous closure of two opposite positions of the sane financial instrument. This operation type is only used in the hedging mode. |

### EnDealReason

Types of reasons for order placing are listed in EnDealReason.

| **ID** | **Value** | **Description** |
| --- | --- | --- |
| DEAL\_REASON\_CLIENT | 0 | Deal performed by a client manually through the client terminal. |
| DEAL\_REASON\_EXPERT | 1 | Deal performed by a client with using an Expert Advisor. |
| DEAL\_REASON\_DEALER | 2 | Deal performed by a dealer through the manager terminal. |
| DEAL\_REASON\_SL | 3 | Deal performed as a result of Stop Loss activation. |
| DEAL\_REASON\_TP | 4 | Deal performed as a result of Take Profit activation. |
| DEAL\_REASON\_SO | 5 | Deal performed when the client reached the Stop-Out level. |
| DEAL\_REASON\_ROLLOVER | 6 | Deal performed when reopening a position for charging swaps. |
| DEAL\_REASON\_EXTERNAL\_CLIENT | 7 | Deal performed by a client from an external trading system. For this type of deals the commission is charged as distinct from DEAL\_REASON\_EXTERNAL\_SERVICE. |
| DEAL\_REASON\_VMARGIN | 8 | Deal performed for accruing variation margin. |
| DEAL\_REASON\_GATEWAY | 9 | Deal performed by a MetaTrader 5 gateway that had connected to the trading platform. |
| DEAL\_REASON\_SIGNAL | 10 | Deal performed as a result of copying a trade signal  according to a subscription in the client terminal. |
| DEAL\_REASON\_SETTLEMENT | 11 | Deal performed to compulsory close a position due to the settlement of a futures contract/option. |
| DEAL\_REASON\_TRANSFER | 12 | Deal performed due to transferring a position at the settlement price to a new symbol with the same underlying asset. |
| DEAL\_REASON\_SYNC | 13 | Deal performed as a result of synchronization of an account's trade state with an external system. |
| DEAL\_REASON\_EXTERNAL\_SERVICE | 14 | Deal performed from an external trading system for technical reasons (for example, to correct the trade state of a client). For this type of deals the commission is not charged. |
| DEAL\_REASON\_MIGRATION | 15 | Deal created as a result of importing trade operations from a MetaTrader 4 server. |
| DEAL\_REASON\_MOBILE | 16 | The deal is conducted via the MetaTrader 5 mobile terminal for Android or iPhone. |
| DEAL\_REASON\_WEB | 17 | The deal is conducted via the web terminal. |
| DEAL\_REASON\_SPLIT | 18 | The deal is conducted as a result of a symbol split. |

# mt5\_accounts

Data on the state of trading accounts is exported to the table. The table contains the following fields:

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| Login | Integer | Primary key. The login of the client, to whom the trading account belongs. |
| CurrencyDigits | Integer | The number of digits after the decimal point in the account deposit currency. |
| Balance | Float | The balance of a trade account. |
| Credit | Float | The current amount of credit given to an account. |
| Margin | Float | The current margin of the account. |
| MarginFree | Float | The free margin of an account. |
| MarginLevel | Float | The margin level as a percentage. It is calculated as a percentage of the current account equity (Equity) to the margin volume (Margin); |
| MarginLeverage | Integer | Margin leverage. |
| MarginInitial | Float | The current size of the initial margin of positions on a trading account. |
| MarginMaintenance | Float | The current size of the maintenance margin of positions on a trading account. |
| Profit | Float | The size of the current profit for all open positions. |
| Storage | Float | The current size of swaps charged for open positions on the account. |
| Floating | Float | The size of floating profit/loss of open positions on the account. The floating profit/loss is calculated as the sum of Profit, Storage and Commission of open positions on the account. |
| Equity | Float | The account equity calculated as a sum of Balance, Credit and Floating. |
| BlockedCommission | Float | The amount of the standard commission locked on the account, which has been accumulated during the day/month. |
| BlockedProfit | Float | The amount of intraday profit locked on the account. |
| Assets | Float | The current total amount of assets on a trading account. |
| Liabilities | Float | The current total amount of liabilities on a trading account. |

# mt5\_prices

Price data of financial instruments is exported to the table. The table contains the following fields:

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| Price\_ID | Integer number | Primary key. Unique symbol ID for more efficient request of the data from the database. Assigned automatically during the export. |
| Symbol | String | Symbol name. |
| Digits | Integer number | The number of decimal places in the price. |
| Time | DateTime | The time of quotes coming in the format YYYY-MM-DD HH:MM:SS. |
| BidLast | Fractional number | The Bid price. |
| BidLow | Fractional number | The lowest bid price for the current day . |
| BidHigh | Fractional number | The highest bid price for the current day. |
| BidDir | Integer number | The direction of change of the bid price relative to its previous state (0 — unchanged, 1 — upwards, 2 — downwards). |
| AskLast | Fractional number | The Ask price. |
| AskLow | Fractional number | The lowest ask price for the current day. |
| AskHigh | Fractional number | The highest ask price for the current day. |
| AskDir | Integer number | The direction of change of the ask price relative to its previous state (0 — unchanged, 1 — upwards, 2 — downwards). |
| LastLast | Fractional number | The price of the last committed transaction. |
| LastLow | Fractional number | The lowest price, at which a deal was executed during the current day. |
| LastHigh | Fractional number | The highest price, at which a deal was executed during the current day. |
| LastDir | Integer number | The change direction of the price of the last deal relative to its previous state (0 — unchanged, 1 — upwards, 2 — downwards). |

# mt5\_daily

Daily reports generated for clients are exported to the table. The table contains the following fields:

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| Datetime | DateTime | The date and time of the daily report generation. |
| Login | Integer | The login of the client for whom the daily report is generated. |
| Timestamp | Integer | Unique record within the table. It is used for internal purposes of MetaTrader 5 servers. If the timestamp is changed for a record, it means that the record has been changed. |
| DatetimePrev | DateTime | Date and time of the previous daily report generation in the YYYY-MM-DD HH:MM:SS. |
| Name | String | The name of a client in the daily report |
| Group | String | The group of a client in a daily report. |
| Currency | String | The client's deposit currency in a daily report. |
| Company | String | The company serving the client in a daily report. |
| EMail | String | An email of a client in a daily report. |
| Balance | Float | The size of a client's balance in a daily report. |
| Credit | Float | The amount of a client's credit funds in a daily report. |
| InterestRate | Float | The size of the annual interest rate of a client in a daily report. |
| CommissionDaily | Float | The amount of a client's commissions for a day in a report. |
| CommissionMonthly | Float | The total amount of a client's commissions for the current month in a report. |
| AgentDaily | Float | The amount of agent commission charged for a client's trade operations for a reported day. |
| AgentMonthly | Float | The amount of agent commission charged for a client's trade operations for the current month. |
| BalancePrevDay | Float | The value of a client's balance as of the end of the previous day. |
| BalancePrevMonth | Float | The value of a client's balance as of the end of the current month. |
| EquityPrevDay | Float | The value of a client's equity as of the end of the previous day. |
| EquityPrevMonth | Float | The value of a client's equity as of the end of the previous trading month. |
| Margin | Float | The size of a client's margin in a daily report. |
| MarginFree | Float | A client's free margin in a daily report. |
| MarginLevel | Float | The margin level of a client in a daily report. |
| MarginLeverage | Integer | The margin leverage of a client in a daily report. |
| Profit | Float | The size of the current profit for all open positions of a client in a daily report. |
| ProfitStorage | Float | The current size of swaps charged for a client's open positions for a day, but not yet reflected in the balance. |
| ProfitEquity | Float | The amount of the current floating equity of a client in a daily report. |
| ProfitAssets | Float | The current amount of a client's assets in a daily report. It is only used for the Exchange risk management model. |
| ProfitLiabilities | Float | The current amount of a client's liabilities in a daily report. It is only used for the Exchange risk management model. |
| DailyProfit | Float | The amount of a client's daily profit. |
| DailyBalance | Float | The amount accrued to a client's balance during the reported day. |
| DailyCredit | Float | The amount of credit given to a client during the reported day. |
| DailyCharge | Float | The amount of other charges to the client's balance during the reported day. |
| DailyCorrection | Float | The amount of corrective balance operations for a reported day. |
| DailyBonus | Float | The amount of bonuses added to the client's balance for the reported day. |
| DailyStorage | Float | The amount of swaps calculated for the client for a reported day. |
| DailyCommInstant | Float | The amount of instant commissions charged from the client for a reported day. |
| DailyCommRound | Float | The amount of turnover commissions charged from the client for a reported day. |
| DailyCommFee | Float | The fee amount charged for the client's deals for the reported day. |
| DailyDividend | Float | The amount of dividends accrued to the client for a reported day. |
| DailyTaxes | Float | The amount of taxes withheld from the client for the reported day. |
| DailySOCompensation | Float | The amount of negative balance compensation accrued to the client for a reported day. |
| DailyAgent | Float | The amount of agent commission charged for a client's trade operations for a reported day. |
| DailyInterest | Float | The amount accrued to a client as part of the annual interest rate for the reported day. |

# mt5\_holidays

Data about holidays are exported to this table. The table contains the following fields:

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| Year | Integer | The year of a holiday. 0 stands for yearly holidays. |
| Month | Integer | Month of a holiday (1 — January, 12 — December). |
| Day | Integer | Day of a holiday. |
| From | Integer | Holiday start time. Specified in a number of minutes from 00:00. For example, 600 corresponds to 10:00. |
| To | Integer | Holiday end time. Specified in a number of minutes from 00:00. For example, 1200 corresponds to 20:00. |
| Description | String | Holiday description (no more than 128 symbols). |
| Timestamp | Integer | A unique value within the table. Used by MetaTrader 5 servers for internal purposes. If the Timestamp of a record has changed, it means that the record has been changed. |
| Mode | Integer | Holiday mode. 0 — holiday disabled, 1 — enabled. |
| Symbols | String | The list of financial instruments or their groups, for which a holiday is valid. Instruments and groups are separated by comma, for example: "EURUSD,CFD\\*". |

# mt5\_network

Data on the platform servers' general settings are exported to this table. The table contains the following fields:

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| Login | Integer | Server ID. |
| Timestamp | Integer | A unique value within the table. Used by MetaTrader 5 servers for internal purposes. If the Timestamp of a record has changed, it means that the record has been changed. |
| Type | Integer | Server type. |
| Name | String | Server name. |
| Address | String | Server address. |
| Port | String | Server port. |
| Adapter | String | Name of the currently used network controller. |
| ServiceTime | Integer | Service time (time of optimization) when various operations aimed at increasing the performance and reliability of the platform are conducted. Specified in a number of minutes from 00:00. For example, 600 corresponds to 10:00. |
| FailoverMode | Integer | Automatic failover modes:   * 0 — failover disabled * 1 — server is unavailable for most access servers * 2 — server is unavailable for all access servers |
| FailoverTimeout | Integer | Time in seconds, during which the server should be unavailable for monitoring servers to start switching to the backup server. |
| Adapters | String | List of all available network controllers on PC (comma-separated). |
| Addresses | String | List of available addresses for outgoing connections from this server (comma-separated). |
| Binds | String | List of listen addresses (comma-separated). |
| Points | String | List of public access points, via which connections are to be accepted. |
| Version | Integer | Server version. |
| Build | Integer | Server build. |
| BuildDate | String | Server build date. |
| SysConnection | Integer | Status of a server connection to the main trade server. |
| SysLastBoot | DateTime | Time of the last server boot in the YYYY-MM-DD HH:MM:SS.MSC format. |
| SysOsName | String | Operating system of the computer running the server. |
| SysCpuName | String | Processor type of the computer that is running the server. |
| SysCpuNumber | Integer | Number of CPU cores. |
| SysBits | Integer | Operating system bits:   * 32 — 32 bits * 64 — 64 bits * 0 — other |
| SysMemoryTotal | Integer | The total amount of RAM in megabytes. |
| SysMemoryFree | Integer | The amount of free memory in megabytes. |
| SysMemoryCritical | Integer | The critical amount of free memory in megabytes. |
| SysHddSize | Integer | Total volume of a disk in megabytes. |
| SysHddFree | Integer | Free memory on the disk in megabytes. |
| SysHddCritical | Integer | Critical amount of free memory on the disk in megabytes. |
| SysHddFragmentation | Integer | The current level of fragmentation of the server files in percentage. |
| SysHddFragCritical | Integer | The critical level of fragmentation of the server files in percentage. |
| SysDefragRecommend | Integer | The flag indicating that the operating system recommends defragmenting the disk:   * 0 — no recommendation * 1 — recommendation is present |
| SysHddReadSpeed | Integer | The current speed of data reading from the disk in megabytes per second. |
| SysHddReadCritical | Integer | The critical speed of data reading from the disk in megabytes per second. |
| SysHddWriteSpeed | Integer | The current speed of saving data to the disk in megabytes per second. |
| SysHddWriteCritical | Integer | The critical speed of saving data to the disk in megabytes per second. |
| PerfConnectsMax | Integer | The maximum number of simultaneous connections to a server that has been achieved during the day. |
| PerfConnectsCritical | Integer | Critical number of simultaneous connections to the server. |
| PerfCpuMax | Integer | Get the maximum level of CPU usage in percentage for the current day. |
| PerfCpuCritical | Integer | Get the critical level of CPU usage in percentage. |
| PerfMemoryMin | Integer | The minimum size of free random access and virtual memory in megabytes per day. |
| PerfMemoryCritical | Integer | The critical amount of free memory in megabytes. |
| PerfMemBlockMin | Integer | The minimum value of the maximum memory block in megabytes per day. |
| PerfMemBlockCritical | Integer | The critical value of the maximum memory block in megabytes per day. |
| PerfNetworkMax | Integer | The maximum level of network usage in kilobytes per second for the current day. |
| PerfNetworkCritical | Integer | The critical level of network usage in kilobytes per second. |
| PerfSocketsMax | Integer | The maximum number of active sockets per day. |
| PerfSocketsCritial | Integer | The critical number of active sockets. |

# mt5\_network\_access\_servers

Data on the access servers settings are exported to this table. The table contains the following fields:

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| Login | Integer | Server ID. |
| Priority | Integer | Base priority of the access server from 0 to 15. The special priority 255 (idle) designed to create backup access servers is possible as well. |
| AntifloodEnable | Integer | Antiflood control: 0 — disabled, 1 — enabled. |
| AntifloodConnects | Integer | The maximum number of connections from one IP address for a certain period of time, after which the address is temporarily blocked. |
| AntifloodErrors | Integer | The maximum number of incorrect connections, after which the IP address is temporarily blocked. |
| NewsMaxCount | Integer | The maximum number of news that can be stored on the access server. |
| BalancingConnections | Integer | The current number of connections. |
| BalancingPriority | Integer | The current priority of the access server. |
| AccessMask | Integer | The allowed types of connection to the access server. Set as a sum of flag values:   * 1 — client connections. * 2 — manager connections. * 4 — administrator connections. * 8 — connections via the Client API. * 16 — connections via the Manager API. * 32 — connections via the Web API.   For example, the value of 63 means that all connection types are allowed. |
| AccessFlags | Integer | Additional server accessing flags:   * 1 — the access server is hidden from all terminals but is available for connection. |
| Servers | String | IDs of trade servers (comma-separated), the connection to which is implemented through this access server. |

# mt5\_network\_history\_servers

Data on the history server settings are exported to this table. The table contains the following fields:

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| Login | Integer | Server ID. |
| DatafeedsTimeout | Integer | Timeout of data feeds before switching to other ones. Specified in seconds. |
| NewsMax | Integer | The maximum number of news that can be stored on the history server. |

# mt5\_network\_trade\_servers

Data on the trade servers settings are exported to this table. The table contains the following fields:

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| Login | Integer | Server ID. |
| DemoMode | Integer | Mode of demo account allocation:   * 0 — creation of demo accounts is disabled. * 1 — prolong the period of demo accounts after connection. * 2 — demo accounts with a fixed expiration date. |
| DemoPeriod | Integer | The validity of demo accounts. |
| OvernightMode | Integer | The overnight mode. |
| OvernightTime | Integer | The time of transition to the next day in minutes after 00:00. |
| OvernightTimeLast | DateTime | The time of the last transition to the next day in the YYYY-MM-DD HH:MM:SS.MSC format. |
| OvernightTimePrev | DateTime | The time of the penultimate transition to the next day in the YYYY-MM-DD HH:MM:SS.MSC format. |
| OvernightDays | Integer | Schedule of operations related to the trading day closure. Set as a sum of flags:   * 0x00000001 — Sunday * 0x00000002 — Monday * 0x00000004 — Tuesday * 0x00000008 — Wednesday * 0x00000010 — Thursday * 0x00000020 — Friday * 0x00000040 — Saturday |
| OvermonthMode | Integer | The overmonth mode:   * 0 — on the last day of the month. * 1 — on the first day of the month. |
| OvermonthTimeLast | DateTime | The time of the last transition to the next month in the YYYY-MM-DD HH:MM:SS.MSC format. |
| OvermonthTimePrev | DateTime | The time of the penultimate transition to the next month in the YYYY-MM-DD HH:MM:SS.MSC format. |
| TotalUsers | Integer | The total number of client accounts on the trade server. |
| TotalUsersReal | Integer | The total number of real clients on the trade server. |
| TotalDeals | Integer | The total number of deals executed on the trade server. |
| TotalOrders | Integer | The total number of active orders placed on the trade server. |
| TotalOrdersHistory | Integer | The total number of orders in the history on the trade server. |
| TotalPositions | Integer | The total number of positions on the trade server. |
| LoginsRange | String | Account ranges on the trade server. For example, 1000-1000000,2000001-2999001. |
| LoginsRangeUsed | String | Ranges of actually used logins from LoginsRange.  For example, 1000-1004,0-0. |
| OrdersRange | String | Order ticket ranges on the trade server. For example, 1-1000000,2000001-2999001. |
| OrdersRangeUsed | String | Ranges of actually used order tickets from OrdersRange. For example, 1-53,0-0. |
| DealsRange | String | Trade ticket ranges on the trade server. For example, 1-1000000,2000001-2999001. |
| DealsRangeUsed | String | Ranges of actually used trade tickets and DealsRange. For example, 1-56,0-0. |

# mt5\_network\_backup\_servers

Data on the backup servers settings are exported to this table. The table contains the following fields:

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| Login | Integer | Server ID. |
| PairrServer | Integer | ID of the server to backup. |
| BackupFlags | Integer | Backup settings. The settings are specified as a sum of flags:   * 1 — enables backup. * 2 — enables the backup of tick data. * 4 — enables the possibility to use the server for the automatic Failover, with which the system switches to this backup server. * 8 — enables synchronization of logs with the primary server. |
| BackupPath | String | The path to save backups. |
| BackupPeriod | Integer | Backup frequency:   * 0 — no periodic backups * 1 — every 15 minutes * 2 — every 30 minutes * 3 — every hour * 4 — every 4 hours * 5 — every day |
| BackupTtl | Integer | Period to keep backups:   * 1 — one day * 2 — three days * 3 — one week * 4 — one month * 5 — three months * 6 — six months |
| BackupTimeFull | Integer | Time of creating full backup copies in minutes since 00:00. |
| BackupLastStartup | DateTime | The last backup copy creation time when launching the server in the YYYY-MM-DD HH:MM:SS.MSC format. |
| BackupLastFull | DateTime | The last full backup copy creation time in the YYYY-MM-DD HH:MM:SS.MSC format. |
| BackupLastArchive | DateTime | The last increment backup copy creation time in the YYYY-MM-DD HH:MM:SS.MSC format. |
| BackupLastSync | DateTime | The time of the last successful synchronization with the backed up server in the YYYY-MM-DD HH:MM:SS.MSC format. |
| SqlMode | Integer | Mode of exporting data to an SQL database:   * 0 — export disabled * 1 — export to Microsoft SQL Server * 2 — export to FireBird * 3 — export to MySQL * 4 — export to Oracle |
| SqlServer | String | The address of the server the database is installed on. |
| SqlFolder | String | The name of the SQL database the data is exported to. |
| SqlFlags | Integer | Additional settings of data export to an SQL database. Defined by the sum of flags:  1 — export trade history to separate tables.  2 — do not export accounts and trade operations of demo groups  For example, the value of 3 means that both settings are enabled. |
| SqlPeriod | Integer | The frequency of price and profit data export. |
| SQLExportLastSync | Integer | The time of the last full synchronization of databases and platform configurations with the SQL database. Specified in seconds since 01.01.1970. If data export is disabled or synchronization is in progress, the value is 0. The full synchronization is launched when a backup server is started or after connection to the SQL database or to the trading server is lost. After synchronization, the SQL database is updated in real time in accordance with the transactions of changes in the platform databases. |

# mt5\_network\_backup\_folders

Data on the backed up custom directories are exported to this table. The table contains the following fields:

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| Folder\_ID | Integer | Unique entry ID. |
| Login | Integer | Backup server ID. |
| Folder | String | The path to the backed up folder relative to the backup server installation directory. |
| Masks | String | List of copied files (comma-separated). Files can be specified by masks. |
| Filter | String | List of ignored files (comma-separated). Files can be specified by masks. |

# mt5\_firewall

Data on the firewall settings are exported to this table. The table contains the following fields:

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| Timestamp | Integer | A unique value within the table. Used by MetaTrader 5 servers for internal purposes. If the Timestamp of a record has changed, it means that the record has been changed. |
| Action | Integer | The type of actions undertaken in accordance with the firewall rule:   * 0 — block * 1 — allow * 2 — always allow |
| From | String | Beginning of the range of the IP addresses the firewall rule is applied to. |
| To | String | End of the range of the IP addresses the firewall rule is applied to. |
| Comment | String | A comment to the firewall rule. |

# mt5\_routing

Data about trade request routing rules are exported to this table. The table contains the following fields:

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| Name | String | The name of a routing rule. |
| Timestamp | Integer | A unique value within the table. Used by MetaTrader 5 servers for internal purposes. If the Timestamp of a record has changed, it means that the record has been changed. |
| Mode | Integer | The state of a routing rule: 0 — disabled, 1 — enabled. |
| Request | Integer | Types of requests for which the rule is applicable. The types are passed using the EnRouteFlags enumeration (as a sum of flags). |
| Type | Integer | Types of orders for which the rule is applicable. The types are passed using the EnTypeFlags enumeration (as a sum of flags). |
| Flags | Integer | Currently not used. |
| ActionType | Integer | The value type for Action. Based on this field, it is possible to determine the field in which the Action value is contained: ActionValueInt, ActionValueUInt, ActionValueFloat or ActionValueString.  Possible values:   * 0 — the current parameter does not have values (for example, ACTION\_CLEAR\_TP) * 1 — the value is located in the ActionValueString field, and its type is string * 2 — the value is located in the ActionValueInt field, and its type is int * 3 — the value is located in the UInt field, and its type is uint * 4 — the value is located in the ActionValueFloat field, and its type is float |
| Action | Integer | The type of action that is applied to a request in accordance with a rule. Passed as a value of the EnRouteAction enumeration. |
| ActionValueInt | Integer | An int value for the action applied to the rule. For example, for the rule "pass to online dealers", the 0 value means that the additional option "skip this rule if no dealers online" is disabled. |
| ActionValueUInt | Integer | An uint value for the action applied to the rule. |
| ActionValueFloat | Fraction | A float value for the action applied to the rule. |
| ActionValueString | String | A string value for the action applied to the rule. |

## Enumerations

The following enumerations are used for passing information about trade routing rules:

* EnRouteFlags
* EnTypeFlags
* EnRouteAction
* EnRouteCondition
* EnConditionRule

### EnRouteFlags

Conditions for applying a rule based on the request type are enumerate in EnRouteFlags.

| **ID** | **Value** | **Description** |
| --- | --- | --- |
| REQUEST\_NONE | 0x00000000 | Conditions by the request type are not specified. |
| REQUEST\_PRICE | 0x00000001 | Price request (for request execution). |
| REQUEST\_REQUEST | 0x00000002 | Confirmation of order execution ar a dealer's type in the request execution mode (with the order confirmation option enabled). |
| REQUEST\_INSTANT | 0x00000004 | Placing an order in the instant execution mode. |
| REQUEST\_MARKET | 0x00000008 | Placing an order in the market execution mode. |
| REQUEST\_EXCHANGE | 0x00000010 | Placing an order in the exchange execution mode. |
| REQUEST\_PENDING | 0x00000020 | Placing a pending order. |
| REQUEST\_SLTP | 0x00000040 | Modification of Stop Loss and Take Profit of a position. |
| REQUEST\_MODIFY | 0x00000080 | Modification of a pending order. |
| REQUEST\_REMOVE | 0x00000100 | Deleting a pending order. |
| REQUEST\_ACTIVATE | 0x00000200 | Activation (triggering) of a pending order. |
| REQUEST\_STOPLIMIT | 0x00000400 | Activation of a Stop Limit order. |
| REQUEST\_SL | 0x00000800 | Triggering of a Stop Loss order. |
| REQUEST\_TP | 0x00001000 | Triggering of a Take Profit order. |
| REQUEST\_STOPOUT\_ORDER | 0x00002000 | A request to delete a pending order in case of reaching the stop-out level (if margin requirements are set for pending orders) |
| REQUEST\_STOPOUT\_POSITION | 0x00004000 | A request to close a position when reaching stop-out. |
| REQUEST\_EXPIRATION | 0x00008000 | Cancellation of an order upon expiration. |
| REQUEST\_DEALER\_POS\_EXECUTE | 0x00010000 | Position opening and closing by a dealer. |
| REQUEST\_DEALER\_ORD\_PENDING | 0x00020000 | Placing of a pending order by a dealer. |
| REQUEST\_DEALER\_POS\_MODIFY | 0x00040000 | Position modification by a dealer. |
| REQUEST\_DEALER\_ORD\_MODIFY | 0x00080000 | Order modification by a dealer. |
| REQUEST\_DEALER\_ORD\_REMOVE | 0x00100000 | Order deletion by a dealer. |
| REQUEST\_DEALER\_ORD\_ACTIVATE | 0x00200000 | Order activation by a dealer. |
| REQUEST\_DEALER\_ORD\_SLIMIT | 0x00400000 | Activation of a Stop Limit order by a dealer. After this action is performed, the order turns into a limit order. |
| REQUEST\_DEALER\_CLOSE\_BY | 0x00800000 | Close By. An operation of closing two oppositely directed positions at a single symbol performed by a dealer. |
| REQUEST\_CLOSE\_BY | 0x01000000 | Close By. An operation of closing two oppositely directed positions at a single symbol performed by a client. |

### EnTypeFlags

Conditions for applying a rule based on the order type are enumerate in EnTypeFlags.

| **ID** | **Value** | **Description** |
| --- | --- | --- |
| TYPE\_NONE | 0x0000 | No conditions by the order type. |
| TYPE\_BUY | 0x0001 | A Buy order. |
| TYPE\_SELL | 0x0002 | A Sell order. |
| TYPE\_BUY\_LIMIT | 0x0004 | A limit Buy order. |
| TYPE\_SELL\_LIMIT | 0x0008 | A limit Sell order. |
| TYPE\_BUY\_STOP | 0x0010 | A stop Buy order. |
| TYPE\_SELL\_STOP | 0x0020 | A stop Sell order. |
| TYPE\_BUY\_STOP\_LIMIT | 0x0040 | A limit Buy Stop order. |
| TYPE\_SELL\_STOP\_LIMIT | 0x0080 | A limit Sell Stop order. |

### EnRouteAction

Types of actions that are applied to requests are listed in EnRouteAction.

| **ID** | **Value** | **Description** |
| --- | --- | --- |
| ACTION\_DELAY\_TIME | 0 | Delay request execution by the specified number of milliseconds. After applying this action to a request, its execution continues in accordance with the created rules located below in the list. The delay is indicated in a separate parameter. |
| ACTION\_DELAY\_TICK | 1 | Delay request execution by the specified number of ticks. After applying this action to a request, its execution continues in accordance with the created rules located below in the list. The delay is indicated in a separate parameter. |
| ACTION\_CLEAR\_TP | 2 | Clear the Take Profit level set in the order. |
| ACTION\_CLEAR\_SL | 3 | Clear the Stop Loss level set in the order. |
| ACTION\_CLEAR\_SLTP | 4 | Clear the Stop Loss and Take Profit levels set in the order. |
| ACTION\_DEALER | 1001 | Enqueue the request to be processed by the specified dealer. The flag of action omission in case there are no dealers online, is specified by an additional parameter. |
| ACTION\_DEALER\_ONLINE | 1002 | Pass the request to dealers that are currently online. The flag of action omission in case there are no dealers online, is specified by an additional parameter. |
| ACTION\_REJECT | 1003 | Reject a request. |
| ACTION\_REQUOTE | 1004 | Send current market prices in response to the request. |
| ACTION\_CONFIRM\_CLIENT | 1005 | Confirm the execution of an order at a price requested in it. |
| ACTION\_CONFIRM\_MARKET | 1006 | Confirm the execution of an order at the current market price. |
| ACTION\_CANCEL\_ORDER | 1007 | Cancel a pending order during its activation or modification. For example, if a pending order has triggered, but the client has already reached the maximum position volume and a new position cannot be opened, the routing rule will remove this order. Otherwise, the order would have continued to trigger on each new tick. When removing an order by this rule, "deleted [by dealer]" is added to the order comment. An entry about the routing rule that canceled the order is also added to the server journal. The server returns error code MT\_RET\_REQUEST\_REJECT\_CANCEL.  The action can only be applied during pending order activation or modification (including modification buy a dealer). The rule does not affect other trade requests. |

### EnRouteCondition

Additional conditions for activation of a routing rule are listed in IMTConCondition::EnRouteCondition.

| **ID** | **Value** | **Description** |
| --- | --- | --- |
| CONDITION\_DATETIME | 0 | Using this parameter you can compare date and time of a request with that specified in the "Value" field. |
| CONDITION\_SYMBOL | 1 | This parameter is used for specifying a symbol or a group of symbols requests for which will be subject to the routing rule. |
| CONDITION\_VOLUME | 2 | Deal volume requested in an order (in lots). This parameter is used for configuring a rule depending on the request volume, e.g. automatic processing of requests less than 1 lot. |
| CONDITION\_MARKET\_DEVIATION | 3 | This condition is applicable only with the instant execution mode. It takes into account the difference between the price of a client's request and the current market price. For Buy trades the deviation is calculated as ("Current Ask price" - "Client's request price"), for Sell trades it is equal to ("Client's request price" - "Current Bid price"). For example, if a client wants to buy at 1.2000, and the current Ask is 1.2008, then the deviation is equal to 1.2008 - 1.2000 = 8 points. |
| CONDITION\_TIME | 4 | This parameter can be used for comparing the time or a request arrival (in minutes since 00:00) with the value specified in the "Value" field. |
| CONDITION\_WEEKDAY | 5 | This parameter allows to route requests depending on a day of the week. |
| CONDITION\_COMMENT | 6 | This parameter allows to compare a request comment with a specified one. If "=" condition is specified, exact match of a comment is checked. If ">" or ">=" conditions are set, a specified substring is searched in a comment string. If "<" or "<=" conditions are set, comment substring is searched in a specified string. |
| CONDITION\_EXPERT | 7 | This parameter allows to route requests placed by MQL5 programs. |
| CONDITION\_SIGNAL | 8 | All operations copied by the client terminal in accordance with the subscription to a trading signal are marked with a special flag. This parameter allows routing trade requests created by trade signals. If this condition is enabled, the rule will trigger for all signal operations. |
| CONDITION\_DEALER\_LOGIN | 9 | This parameter allows applying the routing rules depending on a dealer (or gateway) identifier specified in an order or position. The dealer identifier is specified in an order after it has been confirmed (processed) by the dealer/gateway. Due to it, this rule can be applied only when modifying/deleting an order, and not for newly created orders as they do not have a dealer identifier.. For positions, the dealer identifier is specified according to the dealer identifier of the order, whose execution resulted in the position opening. This parameter can be used when processing trade operations for a symbol through several gateways simultaneously. An order or position created through a specific gateway must be further processed through the same gateway. |
| CONDITION\_SOURCE\_LOGIN | 10 | The parameter allows routing requests by a login of a dealer who set a request on a client's behalf. |
| CONDITION\_MARKET\_DEVIATION\_SPR | 11 | This condition works when executing orders in the Instant or Market modes, as well as when pending orders and Stop Loss/Take Profit orders are triggered. It takes into account the difference between the price of a client's request and the current market price. During a market execution, when a client does not set a price in the order, the difference between the market price during the request and the current market price is taken into account.  The deviation is set in spreads. For floating-spread symbols, the current spread valid during the request check is used. For fixed-spread symbols, a spread value from the symbol settings is used.  For Buy trades the deviation is calculated as ("Current Ask price" - "Client's request price"), for Sell trades it is equal to ("Client's request price" - "Current Bid price"). For example, if a client wants to buy at 1.2000, and the current Ask is 1.2008, then the deviation is equal to 1.2008 - 1.2000 = 8 points. The current spread is divided by this value and the result is compared with the value in the rule.  When setting the condition, keep in mind that if the deviation is positive, opening at the request price is performed in the client's favor, if the deviation is negative, opening is performed against the client. Another example: if we set < -1, the condition corresponds to the buy requests where a request price exceeds the current price by more than 1 spread. |
| CONDITION\_GAP | 12 | This parameter allows processing trade requests in a special way under the market conditions that differ from normal ones. For example, after a gap, client requests can be rejected or requoted during a certain number of subsequent ticks. The gap mode is defined separately for each symbol according to its settings.  The parameter may take two values — true or false (enabled/disabled). If the gap mode is active when checking a request according to the selected symbol routing rule, actions set in this rule are applied to it.  The gap mode is checked by an instrument's Bid and Ask prices. If a gap is detected at least on one of the prices, the rule is triggered. |
| CONDITION\_LOGIN | 1000 | The number of a client's account. This parameter allows creating individual rules for accounts. |
| CONDITION\_GROUP | 1001 | The group to which the client's account is included. This parameter is used for configuring rules for separate account groups. |
| CONDITION\_COUNTRY | 1002 | In this parameter a client's country can be specified. The specified rule will be applied to all clients living in this country. |
| CONDITION\_CITY | 1003 | Use this parameter to apply the rule to all clients living in a specified city. |
| CONDITION\_COLOR | 1004 | Use this parameter to apply the rule for clients that are marked with the specified color. |
| CONDITION\_LEVERAGE | 1005 | Use this parameter to apply the rule for clients with the specified leverage. |
| CONDITION\_COMMENT\_CLIENT | 1006 | Use this parameter to apply the rule for clients with the specified comment. |
| CONDITION\_MARGIN | 2000 | Use this parameter to set up rule application depending on the margin volume that is currently reserved (in the deposit currency). |
| CONDITION\_MARGIN\_LEVEL | 2001 | This parameter allows using rules depending on the current margin level (in percents). |
| CONDITION\_MARGIN\_FREE | 2002 | This parameter allows using rules depending on the current amount of free margin (in the deposit currency). |
| CONDITION\_EQUITY | 2003 | This parameter allows using rules depending on the current equity on a client's account (in the deposit currency). |
| CONDITION\_BALANCE | 2004 | This parameter allows using rules depending on the current balance of a client (in the deposit currency). |
| CONDITION\_PROFIT | 2005 | This parameter allows using rules depending on the current floating profit of a client. |
| CONDITION\_DAILY\_DEALS | 3000 | This parameter allows using rules depending on the number of deals of a client for the current and previous days (including weekends and holidays). |
| CONDITION\_DAILY\_DEALS\_PERIOD | 3001 | The frequency of deals for a day. Calculated on the basis of the last 8 deals (the average time between deals). |
| CONDITION\_DAILY\_PROFIT | 3002 | The profit of the client, whose request is being handled, for the current and previous days (including weekends and holidays). |
| CONDITION\_POSITION\_VOLUME | 4000 | The current volume of a position for the symbol, for which a request has arrived. |
| CONDITION\_POSITION\_PROFIT | 4001 | The current profit of a position for the symbol, for which a request has arrived. |
| CONDITION\_POSITION\_AGE | 4002 | Using this parameter you can specify time in seconds elapsed since position opening for the symbol a request for which is currently being handled. This parameter allows to track positions based on the time they have been held. |
| CONDITION\_POSITION\_MODIFY\_TIME | 4003 | Using this parameter you can specify time in seconds elapsed since the last modification of a position for the symbol a request for which is currently being handled. Position modification means increase of its volume, partial closure, and modification of Stop Loss and Take Profit levels. This parameter allows to prevent evasion of the previous rule through manipulating one position, increasing or reducing its volume. |
| CONDITION\_POSITION\_AVERAGE\_TIME | 4004 | This parameter allows to track positions based on the time the average age of the position for the symbol for which a request is being handled. The average position age is calculated as follows: Current time — ((Open time + Modification time)/2). |
| CONDITION\_POSITION\_TOTAL | 4005 | This parameter allows tracking the total number of a client's open positions on all symbols. For example, you can set the platform to reject trade requests to open new positions, if the client has reached the specified limit. |
| CONDITION\_POSITION\_TOTAL\_SYMBOL | 4006 | The parameter allows tracking the number of positions on the symbol that is specified in the current trade request. For example, if a client has placed an order on EURUSD, this condition will check the current number of the client's open positions on EURUSD. |
| CONDITION\_ORDER\_TOTAL | 4007 | This parameter allows tracking the total number of a client's orders on all symbols. All orders are taken into account, including pending and history orders. Each opening and closing of a position (including partial closure), as well as placing of a pending order increases this counter. |
| CONDITION\_ORDER\_TOTAL\_SYMBOL | 4008 | The parameter allows tracking the number of orders on the symbol that is specified in the current trade request. For example, if a client has placed an order on EURUSD, this condition will check the current number of the client's orders (both active and history) on EURUSD. |
| CONDITION\_POSITION\_SL\_TOUCHED | 4009 | The condition is triggered when the market price touches the stop loss of a position (while the stop loss may not be activated yet). Possible values — true or false. |
| CONDITION\_POSITION\_TP\_TOUCHED | 4010 | The condition is triggered when the market price touches the take profit of a position (while the take profit may not be activated yet). Possible values — true or false. |
| CONDITION\_ORDER\_SL\_TOUCHED | 4011 | The condition is triggered when the market price touches the stop loss of a pending order. Possible values — true or false. In combination with the REQUEST\_ACTIVATE condition, it allows you to track the simultaneous breakthrough of a pending order (trigger) and its stop loss level. This may occur during a release of important news or after a weekend when a large price gap is formed. |
| CONDITION\_ORDER\_TP\_TOUCHED | 4012 | The condition is triggered when the market price touches the stop loss of a pending order. Possible values — true or false. In combination with the REQUEST\_ACTIVATE condition, it allows you to track the simultaneous breakthrough of a pending order (trigger) and its take profit level. This may occur during a release of important news or after a weekend when a large price gap is formed. |

### EnConditionRule

Types of parameter and value comparison are enumerated in IMTConCondition::EnConditionRule.

| **ID** | **Value** | **Description** |
| --- | --- | --- |
| RULE\_EQ | 0 | Condition of equality. |
| RULE\_NOT\_EQ | 1 | Condition of inequality. |
| RULE\_GREATER | 2 | Condition of "greater than". |
| RULE\_NOT\_LESS | 3 | Condition of "not less than". |
| RULE\_LESS | 4 | Condition of "less than". |
| RULE\_NOT\_GREATER | 5 | Condition of "not greater than". |

# mt5\_routing\_dealers

Data about dealers/gateways to whom trade requests are forwarded for processing in accordance with a routing rule, are exported to this table.

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| Login | Integer | The login of a dealer/gateway ID. |
| RoutingName | String | The name of the routing rule in which the dealer/gateway is specified. |
| Name | String | The name of the dealer/gateway. |

# mt5\_routing\_conds

Data about additional conditions specified in a routing rule are exported to this table.

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| Condition\_ID | Integer | The unique identifier of the condition. |
| Name | String | The name of the routing rule to which the additional condition applies. |
| Condition | Integer | The type of the additional condition for the rule. Passed using the EnRouteCondition enumeration. |
| Rule | Integer | A method for comparing a condition with the specified value. Passed using the EnConditionRule enumeration. |
| Type | Integer | The type of value of the additional condition for a routing rule. Based on this field, it is possible to determine the field in which the Condition value is contained: ValueInt, ValueUInt, ValueFloat or ValueString.  Possible values:   * 0 — the current parameter does not have values (currently not used) * 1 — the value is located in the ValueString field, and its type is string * 2 — the value is located in the ValueInt field, and its type is int * 3 — the value is located in the ValueUInt field, and its type is uint * 4 — the value is located in the ValueFloat field, and its type is float |
| ValueInt | Integer | An int value for the condition. |
| ValueUInt | Integer | An uint value for the condition. |
| ValueUInt | Integer | An uint value for the condition. Used for volume with extended accuracy. |
| ValueFloat | Fraction | A float value for the condition. |
| ValueString | String | A float value for the condition. For example, for the "Symbols" condition, the names of symbol (symbol groups) will be specified in this field. |

# mt5\_feeders

Data about data feed settings are exported to this table. The table contains the following fields:

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| Name | String | Get and set the data feed name. |
| Timestamp | Integer | A unique value within the table. Used by MetaTrader 5 servers for internal purposes. If the Timestamp of a record has changed, it means that the record has been changed. |
| Module | String | The name of the data feed module. |
| GatewayServer | String | The addresses at which the data feed will accept connections from the history server. |
| FeedServer | String | The addresses of the server to which the data feed is connected. |
| Enable | Integer | The data feed configuration status: 0 — disabled, 1 — enabled. |
| Mode | Integer | Data feed operation mode. Passed using the EnFeedersMode enumeration as a sum of flags. For example, 1 means that the data feed receives news, 9 — receives news while working in the "remote datafeed" mode. |
| Timeout | Integer | Timeout of a data feed before reconnecting. |
| TimeoutReconnect | Integer | Timeout between attempts to reconnect to the source server. |
| TimeoutSleep | Integer | Timeout between the series of reconnections to the source server. |
| AttemptsSleep | Integer | The number of attempts in the series of reconnections to the source server. |
| Symbols | String | The list of symbols for which the data feed provides quotes. |
| SysConnection | Integer | The status of the data feed connection to a source server. 0 — no connection, 1 — connected. |
| SysLastTime | DateTime | The time of the last reconnection to the source server in the YYYY-MM-DD HH:MM:SS.MSC format. |
| Company | String | The name of the company who signed the executable file of the data feed. |
| Issuer | String | The certification authority that issued the certificate of the above company. |
| TickStatsCount | Integer | The amount of price statistics received by the data feed from an external data source during the current session. |
| TicksCount | Integer | The number of price changes received by the data feed from an external data source during the current session. |
| BooksCount | Integer | The number of Market Depth changes received by the data feed from an external data source during the current session. |
| NewsCounts | Integer | The number of news items received by the data feed from an external data source during the current session. |
| BytesReceived | Integer | The volume of traffic (in bytes) received by the data feed during the current session. |
| BytesSent | Integer | The volume of traffic (in bytes) sent by the data feed during the current session. |
| StateFlags | Integer | Flags of states. |

## Enumerations

The following enumerations are used for passing information about data feed configurations:

* EnFeederFlags
* EnFeedersFieldFlags

### EnFeederFlags

Flags of predefined data feed settings are listed in EnFeederFlags.

| **Identifier** | **Value** | **Description** |
| --- | --- | --- |
| FEED\_FLAG\_QUOTES | 1 | The data feed sends quotes. |
| FEED\_FLAG\_NEWS | 2 | The data feed sends news. |
| FEED\_FLAG\_REMOTE | 8 | The data feed is running on a remote computer. |

### EnFeedersFieldFlags

Flags of editable fields are listed in EnFeedersFieldFlags.

| **Identifier** | **Value** | **Description** |
| --- | --- | --- |
| FEED\_FIELD\_SERVER | 1 | The "Server" field. |
| FEED\_FIELD\_LOGIN | 2 | The "Login" field. |
| FEED\_FIELD\_PASS | 4 | The "Password" field. |
| FEED\_FIELD\_PARAM | 8 | The "Parameters" field. |

# mt5\_feeder\_translates

Data about conversion settings of data feeds are exported to this table. The table contains the following fields:

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| Symbol | String | The name of the symbol in the trading platform. |
| Feeder | String | The name of the data feed, to which the conversion setting applies. |
| Source | String | The name of the symbol on the source server. |
| BidMarkup | Integer | Markup for the symbol's Bid price received from the data feed. |
| AskMarkup | Integer | Markup for the symbol's Ask price received from the data feed. |
| Digits | Integer | The number of digits after the decimal point in the price of the symbol that receives quotes. |

# mt5\_feeder\_params

Data about additional settings of data feeds are exported to this table. The table contains the following fields:

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| ParamID | String | The unique identifier of the parameter. |
| Feeder | String | The name of the data feed, to which the setting applies. |
| Type | Integer | Parameter type:   * 0 — string * 1 — integer * 2 — floating-point number * 3 — time * 4 — date * 5 — date and time * 6 — list of groups * 7 — list of symbols |
| Name | String | The name of the parameter. |
| Value | String | The value of the parameter. |

# mt5\_reports

Data about report settings are exported to this table. The table contains the following fields:

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| Name | String | The name of the report configuration. |
| Server | Integer | The ID of the trade server, for which the report is configured. |
| Timestamp | Integer | A unique value within the table. Used by MetaTrader 5 servers for internal purposes. If the Timestamp of a record has changed, it means that the record has been changed. |
| Module | String | The name of the report module. |
| Mode | Integer | Report operation mode: 0 — disabled, 1 — enabled. |

# mt5\_report\_params

Data about additional report settings are exported to this table. The table contains the following fields:

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| ParamID | String | The unique identifier of the parameter. |
| Report | String | The name of the report, to which the setting applies. |
| Server | Integer | The ID of the trade server, for which the report is configured. |
| Type | Integer | Parameter type:   * 0 — string * 1 — integer * 2 — floating-point number * 3 — time * 4 — date * 5 — date and time * 6 — list of groups * 7 — list of symbols |
| Name | String | The name of the parameter. |
| Value | String | The value of the parameter. |

# mt5\_plugins

Data about plugin settings are exported to this table. The table contains the following fields:

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| Name | String | The name of the plugin configuration. |
| Server | Integer | The ID of the trade server, for which the plugin is configured. |
| Timestamp | Integer | A unique value within the table. Used by MetaTrader 5 servers for internal purposes. If the Timestamp of a record has changed, it means that the record has been changed. |
| Module | String | The name of the plugin module. |
| Enable | Integer | Plugin operation mode: 0 — disabled, 1 — enabled. |
| Flags | Integer | Plugin operation flags:   * 0 — no flags * 1 — permission to configure the plugin from a manager terminal * 2 — the profiling mode enabled |

# mt5\_plugin\_params

Data about additional plugin settings are exported to this table. The table contains the following fields:

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| ParamID | String | The unique identifier of the parameter. |
| Plugin | String | The name of the plugin, to which the setting applies. |
| Server | Integer | The ID of the trade server, for which the plugin is configured. |
| Type | Integer | Parameter type:   * 0 — string * 1 — integer * 2 — floating-point number * 3 — time * 4 — date * 5 — date and time * 6 — list of groups * 7 — list of symbols |
| Name | String | The name of the parameter. |
| Value | String | The value of the parameter. |

# mt5\_time

Platform trading time settings are exported to this table. The table contains the following fields:

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| TimeZone | Integer number | The time zone of a server in minutes from GMT. For example: 0 = GMT, -60 = GMT - 1, 60 = GMT + 1. |
| Timestamp | Integer number | A unique value within the table. Used by MetaTrader 5 servers for internal purposes. If the Timestamp of a record has changed, it means that the record has changed. |
| TimeServer | String | The address of the current time synchronization server. |
| Daylight | Integer number | Daylight Saving Time mode: 0 — off, 1 — on. |
| DaylightState | Integer number | The presence of the daylight saving time in the platform time zone. 0 means no daylight saving time is applied in the platform time zone. Otherwise, any non-zero value is used. |

# mt5\_time\_weekdays

Platform's working time schedule by days is exported to this table. The table contains the following fields:

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| TimeZone | Integer number | The time zone of a server in minutes from GMT. For example: 0 = GMT, -60 = GMT - 1, 60 = GMT + 1. Corresponds to the TimeZone value in the mt5\_time table. |
| Day | Integer number | The ordinal number of the day of the week. For example, 0 is Sunday, 6 is Saturday. |
| 00 | Integer number | Flag of working time in the period from 00:00 to 00:59. 0 — non-working time, 1 — working time. |
| 01 | Integer number | Flag of working time in the period from 01:00 to 01:59. 0 — non-working time, 1 — working time. |
| ... | Integer number | Further similar fields apply for each hour of the day. |
| 23 | Integer number | Flag of working time in the period from 23:00 to 23:59. 0 — non-working time, 1 — working time. |

# mt5\_gateways

Data on the gateway settings is exported to this table. The table contains the following fields:

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| Name | String | Gateway configuration name. |
| Timestamp | Integer | A unique value within the table. Used by MetaTrader 5 servers for internal purposes. If the Timestamp of a record has changed, it means that the record has been changed. |
| Module | String | Gateway module name. |
| GatewayServer |  | The address at which the gateway accepts connections from the history and trade servers. |
| TradingServer |  | Address of the server to which the gateway connects. |
| Enable | Integer | Gateway operation mode: 0 — disabled, 1 — enabled. |
| Flags | Integer | Gateway operation flags:   * 0x00000001 — gateway works as a remote application. * 0x00000002 — gateway is allowed to import symbol settings. * 0x00000004 — do not broadcast quotes from the gateway in the system. * 0x00000008 — gateway can manage clients' balances using IMTExecution::TE\_BALANCE\_CHANGE and IMTExecution::TE\_BALANCE\_CORRECT trade executions. * 0x00000010 — if enabled, the gateway log receives additional operation data, including the results of measuring the trading operations handling speed. A more detailed information on extended logging is provided in the Journal of Gateways section. * 0x00000020 — gateway supports requesting the state of external trading system positions. The request is made from the Positions tab of the gateway. * 0x00000040 — collect advanced metrics related to request processing by the gateway. * 0x00000100 — gateway is running in demo mode. The mode is checked based on the license at the time the module is loaded. * 0x00000200 — gateway is integrated into the platform. |
| Gateway | String | Gateway module name. |
| TimeoutReconnect | Integer | Timeout between attempts to reconnect to an external server in seconds. |
| TimeoutSleep | Integer | Timeout between the series of reconnections to an external server in seconds. |
| AttempsSleep | Integer | Number of attempts in a series of reconnections to an external server. |
| ID | Integer | The gateway ID. |
| Symbols | Array | The list of symbols for which the gateway provides quotes and processes trading operations. |
| SysConnection | Integer | The state of gateway connection to an external trading system: 0 — connected, 1 — disconnected. |
| SysLastTime | Integer | The time of the last successful gateway connection to an external trading system in the YYYY-MM-DD HH:MM:SS.MS format. |
| Company | String | The company by which the gateway executable is signed. |
| Issuer | String | The certification authority that issued the certificate to the above company. |
| TickStatsCount | Integer | The number of price statistics changes received by the gateway from the external system for the current session. |
| TicksCount | Integer | The number of price changes received by the gateway from the external system for the current session. |
| BooksCount | Integer | The number of Market Depth changes received by the gateway from an external trading system for the current session. |
| TradeAverageTime | Integer | Average time spent by the gateway to process one trading operation in milliseconds. |
| TradeRequestsCount | Integer | The number of trading operations processed by the gateway during the current session. |
| BytesReceived | Integer | Traffic volume received by the gateway during the current session. |
| BytesSent | Integer | Traffic volume sent by the gateway during the current session. |
| StateFlags | Integer | Flags of states. Currently not used. |

# mt5\_gateways\_params

Data about additional report settings is exported to this table. The table contains the following fields:

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| ParamID | String | The unique identifier of the parameter. |
| GatewayName | String | The name of the gateway configuration the setting applies to. |
| Type | Integer | Parameter type:   * 0 — string * 1 — integer * 2 — floating-point number * 3 — time * 4 — date * 5 — date and time * 6 — list of groups * 7 — list of symbols |
| Name | String | Parameter name. |
| Value | String | Parameter value. |

# mt5\_gateways\_translates

Data about price translation settings in the gateway is exported to this table. The table contains the following fields:

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| Symbol | String | The name of the symbol in the trading platform. |
| GatewayName | String | The name of the gateway configuration the setting applies to. |
| Server | Integer | The ID of the trade server, for which the plugin is configured. |
| Source | String | The symbol name in the data feed, to which the gateway connects. |
| BidMarkup | Integer | Correction for the Bid price received for a symbol from the data source, to which the gateway connects. |
| AskMarkup | Integer | Correction for the Ask price received for a symbol from the data source, to which the gateway connects. |
| Digits | Integer | The number of digits after the decimal point in the price of the symbol that receives quotes. |