



QUIK Server

Installation manual

Version 15.0



Contents

1. General statements	4
1.1 Components of QUIK system	4
1.2 License management on QUIK servers	5
2. Server installation and configuration	6
2.1 Technical requirements	6
2.2 Installation on Windows	6
2.3 Installation on Linux	11
2.4 Server configuration	19
2.5 Server migration from Windows to Linux	37
2.6 Server removal	45
2.7 QUIK Administrator installation and configuration	47
2.8 QUIK Workstation banner configuration	68
2.9 Message output configuration.....	70
2.10 File transfer settings	70
3. Running QUIK server	71
3.1 QUIK server launch parameters.....	72
4. Password encryption	75
5. Appendix.....	75
5.1 Format of positions	75
5.2 INI file conversion rules when migrating software from Windows to Linux.....	77
5.3 OpenSSL encryption configuration file format.....	79
5.4 MP encryption configuration file format	80
5.5 CryptoPro configuration.....	86
5.6 Signal-COM configuration	97
5.7 Stop order rollover	107
5.8 Daily maintenance of QUIK server	109
5.9 List of QUIK server error codes.....	115



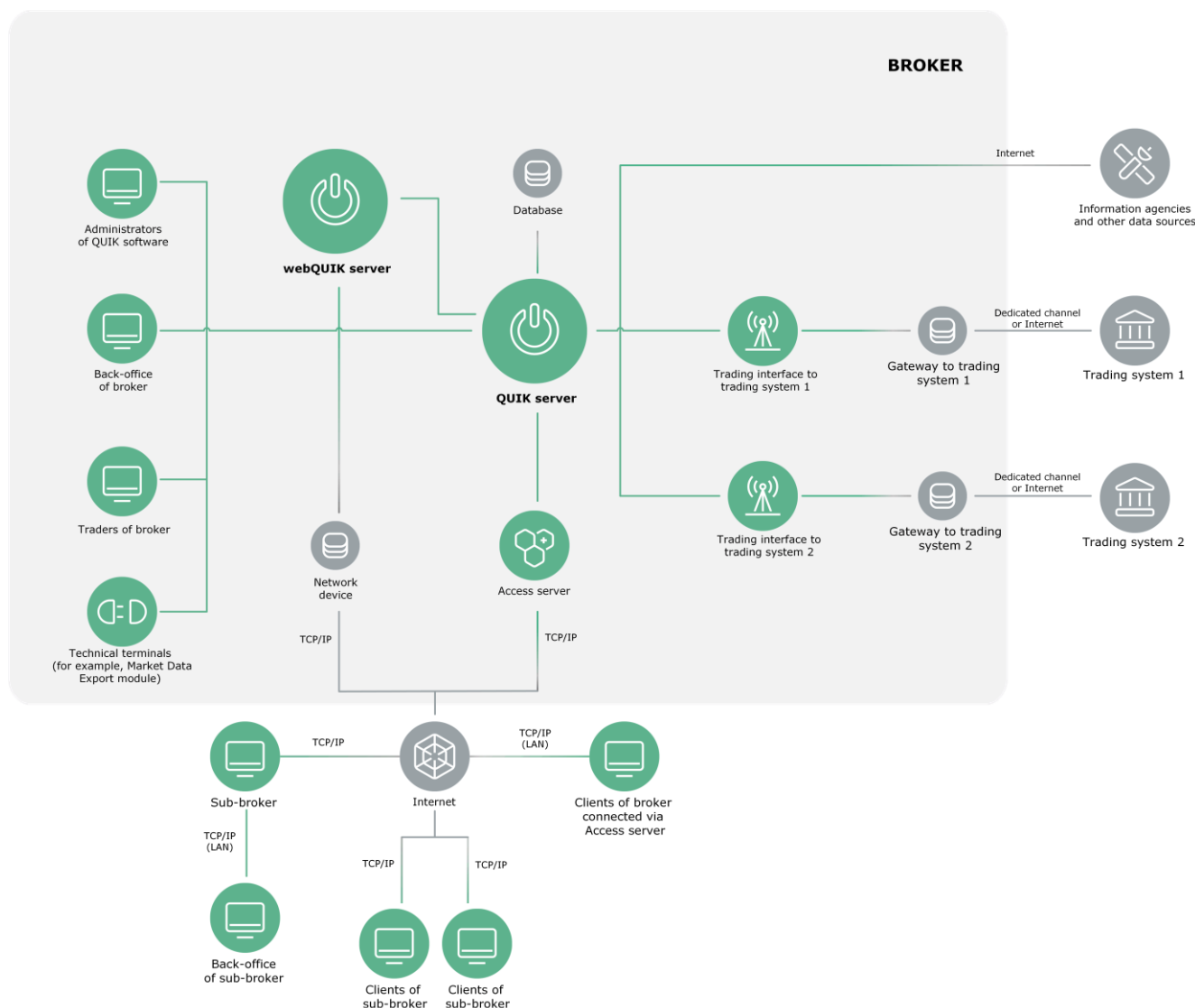
You may email your comments on this Manual to: quiksupport@arqatech.com



1. General statements

1.1 Components of QUIK system

QUIK software is implemented on the basis of 'client-server' technology. Data exchange between the server and user terminals is carried out by means of the Internet and/or a local network. The server provides the connection of information sources (trading systems, news flows and other) and processing of client orders. Client applications are designed for displaying the information and entering of customers' orders.



The central part of QUIK software is **QUIK Server** having the logical core and **interfaces** to connected elements (exchange interfaces and other information sources, data storage). The modular concept of connection allows accumulating the system's capacities when necessary by equipping the basic configuration by new features.

The interaction with **database** is realized through a special interface to ease changing of the used type of **DBMS**. The database is used to store information with a storage period of more than one day - registration data of users and rights to work with the system, as well as logs of executed operations.

Information received from the exchange trading systems is processed in RAM and not entered to the database. For a quick recovery in situation of failure the received data the cached on the disk in form of binary file is used. All work with the database is carried out through the dedicated database server.

The **access server** is connected to the principal QUIK server, receives data and distributes it to users according to their rights. The **access server** is used to handle a large number of user connections as it allows reducing the load on the principal system server.

Administration of the complex is carried out from the **QUIK Administrator** program unifying all the settings used by the logic core of server. For monitoring over the current state of the server and client connections the terminal **QMonitor** is used.

1.2 License management on QUIK servers

The licensing includes the right to connect one copy of the application (plugin, technologic or client module) to a QUIK server. Applications of one type are included in a license pool.

If the broker uses several QUIK servers, the total number of licenses within the license pool, which is distributed among them using the data synchronization service (the Replicator program), is controlled.

After the maximum number of available licenses is reached, a corresponding diagnostic is written to a server log file. If no licenses are available for the plugin, the message "Not enough licenses!" appears in a QUIK workstation.

The applications for which license management is supported:

- Q2Q adapter;
- FIX adapter program interface (accounts);
- FIX adapter program interface (transactions);
- FIX Drop Copy program interface;
- FIX Client Connector program interface;
- Market Data Export Module;
- Level 2 Quotes export module;
- QUIK workstation for mobile devices (PocketQUIK);
- CoLibri SM risk manager terminal module;
- CoLibri FM risk manager terminal module;
- CoLibri FX risk manager terminal module;
- Specialized multi-broker service workstation;
- FX Convertor;



- BasketTrading module;
- Option analytics module;
- Asset manager terminal module (TrustManager);
- webQUIK workstation;
- QORT2QUIK adapter;
- Position management program interface.

2. Server installation and configuration

2.1 Technical requirements

The system and software requirements table is available on the official [website of QUIK](#) for the following:

- QUIK Server;
- Trading interfaces;
- Market data interfaces;
- Service modules;
- Workstation of the administrator QUIK Administrator;
- Monitoring workstation QMonitor;
- Other modules and interfaces to other exchanges.

The minimum requirements are listed on the website. It is recommended to consult with the QUIK technical support on the technical requirements, for example, if you intend to:

- Serve a large number of users;
- Connect a large number of interfaces;
- Use the QUIK server for the market maker tasks or high-frequency trading.

2.2 Installation on Windows

2.2.1 DBMS configuration

1. Create a database on **SQL** server to be used by QUIK server. To create the DB, the scripts (located in the distribution) of the creating DB is used:

- _ For MS SQL: full.sql;
- _ For PostgresPro run sequentially the following command files:
 - _ Server_create_pg_db_from_windows.cmd;
 - _ Server_db_fill_windows.cmd.

To obtain the actual scripts for creating database, contact QUIK technical support: quiksupport@argatech.com.



2. Open to edit **quik.ini** file in **quik\server** catalog with QUIK Server settings.

The file must be in Windows-1251 encoding.

In the [DB] section of this file configures the following parameters instead of the **file** parameter:

```
server=<name_of_SQL_server>,<port>
base=<DB_name_QUIK>
```

Example of the configured connection to SQL Server instance:

```
[DB]
server=SqlServerInstance,1234
base=QUIKDB
```

3. Configure the following parameters in section [DB]:

- _ **db_connect_timeout** – time of waiting for connection of QUIK Server to the DB server (in seconds). Range of valid values: from 1 to 600. Value by default: 60.
- _ **AuthMode** – used type of authentication on MS SQL server. Valid values:
 - _ login (by default) – use login and password specified in 'user' and 'password' parameters (the password can be [encrypted](#));
 - _ domain – use domain authentication.

If AuthMode=domain, the 'user' and 'password' in section [DB] are ignored and may be not specified.

- _ **Driver** – name of a driver which is used to connect to the QUIK server database. Valid values:
 - _ ODBC Driver 17 for SQL Server (by default) – used to operate MS SQL Server 2014, 2016 or a newer DBMS (from the list of supported ones). Link to download the driver installation file from the official Microsoft website:
<https://go.microsoft.com/fwlink/?linkid=2223304>;
 - _ PostgreSQL ANSI(x64) – used to operate PostgresPRO. Link to download the driver installation file from the official ARQA Technologies website:
ftp://ftp.quik.ru/public/updates/odbc/odbc_pg_win_13.2.0.zip;

When operating MS SQL Server, ODBC Driver 17 for SQL Server must be installed. When operating PostgresPRO, the psqlODBC driver version 13.2 or newer must be installed.



Example:

```
[DB]
server=<SQL server name without "tcp:" prefix>,<port>
Driver=ODBC Driver 17 for SQL Server
AdditionalParameters=MultiSubnetFailover=Yes;Network=dbmssocn
db_connect_timeout=120
```

- **AdditionalParameters** – setting to specify additional parameters for connecting to the DBMS. Value by default: "" (empty).

When connecting to the MS SQL cluster, set MultiSubnetFailover=Yes and Network=dbmssocn.

```
AdditionalParameters=MultiSubnetFailover=Yes;Network=dbmssocn
```

To connect to the PostgresPRO DBMS using a port other than a default one (5432), specify it in the AdditionalParameters parameter. For example:

```
AdditionalParameters=port=5431
```

- **SSLMode** – secure connection mode. Supported only when configuring a connection to the PostgresPRO DBMS. Valid values:
 - "" (empty, by default) – connection without encryption;
 - require;
 - verify-ca;
 - verify-full;
 - disable;
 - allow;
 - prefer.

For the description of secure connection modes, see the official website of PostgreSQL: www.postgresql.org/docs/current/libpq-ssl.html.

If the parameter is not set, the CertFileName, CertPKeyFileName, CertPass and CertCAFileName parameters are ignored.

- **CertCAFileName** – path to the file container of certificates of certification authorities. Value by default: "" (empty). Supported only when configuring a connection to the PostgresPRO DBMS.
- **CertFileName** – path and name of the secure connection certificate file. Value by default: "" (empty). Supported only when configuring a connection to the PostgresPRO DBMS.
- **CertPKeyFileName** – path and name of the secure private key file. Value by default: "" (empty). Supported only when configuring a connection to the PostgresPRO DBMS.



- **CertPass** – password for the private key file of the secure connection. Value by default: "" (empty). Supported only when configuring a connection to the PostgresPRO DBMS. The password must not start with \$ and #. The password can be [encrypted](#).
4. In section [DB], configure the possibility of receiving unread messages of the previous trading days:
- **messages_delivery_period** – number of days for which messages are loaded. Value by default: 0 (load only messages for the current trading day).
5. In section [db_module], configure the database provider being used:
- **driver** – the name of the executable file of the database provider. Value by default: "" (empty). Valid values:
 - For MS SQL – set the value "quik_db_ms.dll";
 - For PostgresPRO – set the value "quik_db_pg.dll".

Example of sections for MS SQL DBMS:

```
[DB]
server=MSSQL_SRV
base=QUIK_DB
user=sa
password=password
db_connect_timeout=30
AuthMode=login
Driver=ODBC Driver 17 for SQL Server
AdditionalParameters=MultiSubnetFailover=No;ApplicationIntent=ReadWrite

[db_module]
module=dstubrc.dll
driver=quik_db_ms.dll
```

Example of sections for PostgresPRO DBMS:

```
[DB]
server=PostgreSQL_SRV
base=QUIK_DB
user=sa
password=password
Driver=PostgreSQL ANSI (x64)
SSLMode=verify-full
CertCAFileName=Certs/root.crt
CertFileName=Certs/user.crt
CertPKeyFileName=Certs/user.key
CertPass=secret
```



```
[db_module]
module=dstubrc.dll
Driver=quik_db_pg.dll
```

2.2.2 Copying files

1. Create the catalog for installation of QUIK system (hereinafter <quik_dir>).
2. Copy there **keygen** and **server** catalogs from <inst_dir> (directory with files for installation).
3. Copy the <inst_dir>/front catalog to the directory that is intended to store the current distribution of the QUIK Workstation (see [2.4.3](#)).
4. Copy the <inst_dir>/qadminsrv catalog to an arbitrary directory. This catalog contains the necessary files for QUIK Administrator installation (see [2.7](#)).

2.2.3 Creating a key for QUIK server

Using the **KeyGen** program create public and secret keys that will be later used by QUIK server (see keygen.pdf – User’s manual for keygen.exe).

1. To configure the parameter ‘Name of key holder’ type the name of organization in format “Name (identifier)”, for example:

```
Test-Invest (testinvest)
```

2. The identifier in parentheses will be then used to configure the system, call it <name_of_quik_server>.
3. Remember the password provided to protect the key. Call it <password_of_server_key>.
4. Copy the secret key of server (program KeyGen.exe saves file a:\secreg.tsk by default) to catalog <quik_dir>\server. For convenience in further work rename it to secrserv.tsk.
5. Copy the public key of server (program KeyGen.exe creates file a:\pubring.tsk by default) to catalog <quik_dir>\keygen. For convenience in further work rename it to <name_of_quik_server>.tsk. This key will be used by all users of the server with the software package **KeyGen**.
6. Open to edit file <quik_dir>\keygen\crypto.cfg. Specify as value of parameters:
 - «IMPORT» – identifier of organization;
 - «Import Keys» – names of application system keys to be added to the public key.

For example:

```
IMPORT=<name_of_QUIK Server>
Import Keys=Test Invest (testinvest)
```



7. Using the **KeyGen** program create public and secret keys for user of QUIK server (see keygen.pdf – User’s manual for keygen.exe). In step 3, the program prompts you to select file with public key of the system – select <name_of_quik_server>.txk file. Select the public key of the server in the list of keys (in the example that is “Test Invest (testinvest)”). Select the checkbox “Record and use skipping this step” to allow your users to avoid unnecessary operations subsequently when forming keys.

2.2.4 Registration in Windows services

1. Install the QUIK Server and the interface to trading system as services. For this, type the following in the command line:

```
<quik_dir>\server\quik.exe -install
```

2. Open **controlpanel\services** and check if there are records identifying services of QUIK Server (by default **Quik Server**).
3. Create a user-administrator and give it a name (for example <quikserv>) and password (for example <passwd>).
4. To use the domain authentication on MS SQL server, specify name of the domain user and password in startup parameters of the services.

2.3 Installation on Linux

Users to operate with service under Linux

The following users are used to operate and administrate the service:

- The platform administrator:

To denote this user, the root name is used. This user, who knows the root password, or is added to the root group, or has access to wide list commands to execute from the root name via the sudo utility.

This type of user can install and remove, and also can administrate the service: update, run and stop.

- The user, on whose behalf the service is operating:

To denote this user, the arqasrv name is used. When installing the service, the user is created with this name by default.

This type of user also can administrate the service: update, run and stop – is applied if administration by the root user is unavailable.

Therefore, the service always operates as the arqasrv user. Installation and removing service can be executed only by the root user. The service can be administrated by the root or arqasrv user.



Installation procedure

The service installation must be performed by the root user who will be called the run file.

The service is installed to the **/opt/arqasrv** directory and on behalf of the **arqasrv** system user by default. The **<srvname>** service name for servers and the **quik** executable file name are similar by default.

When installing server, the service name is specified in lowercase only.

All specified parameters can be overridden with the corresponding launch parameters of the run file.

The initial actions of the service installation (see 1 – 5) must be performed by the **root** user. Follow the steps below to install:

1. Copy the `install_server-xx.xx.xx.run` file to the directory where current user has permissions to perform operations. For example, home directory: (**/home/<user>** or **~/**).
2. Add execution permissions to the run file by running the command:

```
chmod +x install_server-xx.xx.xx.run
```

3. Install the service by running the command:

```
sudo ./install_server-xx.xx.xx.run -i
```

It is required to enter the current user's password to execute the command. The following operations will be performed during the execution of the run file:

- If there is no **arqasrv** system user, a user with default permissions is created and included in the group of the same name. The service to be installed will work on behalf of this user. The **/opt/arqasrv** directory will be set as a home directory for this user.
- A subdirectory with the **quik** name is created in the **/opt/arqasrv/quik** directory. This is the service installation directory into which the distribution files of the installed service are unpacked.
- The **arqasrv** user is owner of the all files and subdirectories in the service directory.
- A service named **quik** will be registered in the **systemd** service administration system and run as the **arqasrv** system user.
- A file named **quik** is created in the **/etc/arqasrv** directory to fix the service installation parameters.
- The following commands files are generated in the service directory:
 - **q_start.sh** – start the service, including launch parameters. Intended to be performed by the **arqasrv** and **root** users;



- _ **q_stop.sh** – stop the service. Intended to be performed by the **arqasrv** and **root** users;
- _ **q_status.sh** – output of the current service status. Intended to be performed by the **arqasrv** and **root** users;
- _ **uninstall.sh** – remove service. Intended to be performed by the **root** user;
- _ **postmigration.sh** – convert configuration file names to lower case during migration. Intended to be performed by the **arqasrv** and **root** users.

If the service is successfully installed, the following message will be displayed in the console: “Installation of service ‘quik’ has been successfully completed”. In case of another diagnostic output, please contact the QUIK technical support: quiksupport@arqatech.com.

4. Install libjemalloc2 package from the official Astra Linux repository. To do this, run the following command:

```
sudo apt install libjemalloc2
```

This library implements RAM handling mechanisms that are more appropriate for server scenarios than those built into the system by default.

5. Configure the connection to PostgreSQL DBMS:

- _ ODBS driver is used to operate with DBMS. For driver operation, it is required to install the libodbc1 and unixodbc packages by running the commands:

```
sudo apt install libodbc1
sudo apt install unixodbc
```

- _ Use link to download installation packages from the official ARQA Technologies website:

ftp://ftp.quik.ru/public/updates/odbc/odbc_pg_linux_16.0.5.zip

- _ Unpack the archive and install deb packages:

```
sudo dpkg -i libpq5_16.3-1.pgdg100+1_amd64.deb
sudo dpkg -i odbc-postgresql_16.00.0005-1.pgdg100+1_amd64.deb
```

- _ Make sure that PostgreSQL ANSI driver used by default is specified in the /etc/odbcinst.ini file:

```
[PostgreSQL ANSI]
Description=PostgreSQL ODBC driver (ANSI version)
```



```
Driver=psqlodbc.so
Setup=libodbcpsqlS.so
Debug=0
CommLog=1
UsageCount=1
```

If the record was not added, then add it by yourself.

6. If the administration of the installed service is expected to be performed by the `arqasrv` user, execute the following steps:

- To prevent the `arqasrv` from making changes to the service administration scripts, on which the `arqasrv` user will be granted root permissions, make them immutable by running the following commands:

```
chattr +i /opt/arqasrv/quik/quik/q_start.sh
chattr +i /opt/arqasrv/quik/quik/q_stop.sh
```

- Grant the `arqasrv` user permissions to execute the service administration scripts as root by running the following steps:
- Run the command:

```
sudo visudo
```

- Add the string in the vi-like editor that opens:

```
arqasrv ALL=(root) /opt/arqasrv/quik/quik/q_start.sh,
/opt/arqasrv/quik/quik/q_stop.sh
```

This string permits the `arqasrv` user to execute the listed scripts on current computer as root.

The following steps of this instruction can be performed by the `arqasrv` and root users.

7. Check that the following parameters are specified in the `crypto.cfg` server file:

```
SECRING=secrserv.txk
PUBRING=quik.dwq

[auth]
sign=libsecprov.so
look=crypto.cfg,db_look
```



```
[db_look]
do=_QXKeyDB
use=libdwqx.so

[CryptoProviders]
placebo=1

[placebo]
Module=libplacebo_pr.so
```

8. If the quik.ini and crypto.cfg current configuration files are missing in the service directory, the following operations should be executed:

- _ Create the minimal quik.ini and crypto.cfg configuration files to the server operation based on the quik.ini.tmpl and crypto.cfg.tmpl templates, which after installation are located in the service directory.
- _ Rename the templates by running the commands:

```
[sudo] cp quik.ini.tmpl quik.ini
[sudo] cp crypto.cfg.tmpl crypto.cfg
```

Specifying sudo is required if the service is updated by the root user.

- _ Make sure that received files are encoded in win1251.
- _ Set the sections with settings in the quik.ini file:
 - _ Set port to connect QMonitor program to the server, name of the server key and password to this key in the [ctl_module] section.

```
[ctl_module]
module=libdwqctl.so
port=15121
name=broker
password=
```

- _ Set port to connect interfaces to the server, name of the server key and password to this key in the [data_ip_module] section.

```
[data_ip_module]
module=libdtwipsrv.so
port=17100
```



- _ Set name or IP address of the DBMS server, database name of the server, the user, on whose behalf the connection to the DBMS and this user password.

```
[db]
server=
base=
user=
password=
driver=PostgreSQL ANSI
```

When connecting to the Postgres DBMS by using a port other than a default one (5432), specify it in the AdditionalParameters parameter. For example:

```
AdditionalParameters=port=5431
```

- _ Set the path to the all Limits calculation library settings in the [dealer-library] section. If paths specified with .dll file in the current settings file, then it requires to change to .ini.

```
[dealer-library]
ALGO=./dealer/algo.ini
SMS=./dealer/sms.ini
```

- _ Set system name, port to connect users to the server, server key name and this key password.

```
[usend_module]
module=libdwsend.so
sysname=QUIK "broker" information and trading system
name=broker
password=
port=15100
```

- _ If the replication is configured, set server identifier in the [stop_sync] server.

```
[stop_sync]
serverid="ID_SERV1"
```



- _ Create the crossrate.ini file, ([cur_module] section) that including the currency descriptions are used in the server:

```
[cur_module]
class_cfg=crossrate.ini
```

- _ Minimal currency description in the crossrate.ini file:

```
[crossrate]
RUR=Ruble, Ruble, 3

[crossrate_rate]
RUR=1
```

The crossrate.ini file must be saved in win1251 encoding to display correctly the currency names, for example, in the QUIK Workstation.

9. For service installed by default, run it for the first time with -clean launch parameter from directory with installed service by running the command:

```
sudo ./q_start.sh -clean
```

or from any directory other than the service directory by running the command:

```
sudo /opt/arqasrv/quik/quik/q_start.sh -clean
```

10. Configure automatic daily start and stop of the service. To do this, use in the automatic scripts the **q_start.sh** and **q_stop.sh** command files located in the directory of installed service.

The server requires to be restarted every day. Server must be started 3-5 minutes before the first interface start. Server must be stopped 3-5 minutes after interfaces are stopped.

Directory structure is important. Replacing or renaming service files can cause an installation error.

2.3.1 Launch parameters run file of the distribution

The run file supplied in the distribution supports:



-i [<srvname>]

Executing with this launch parameter is available only for the root user.

Installing the service named <srvname>. The <srvname> parameter is optional.

If the <srvname> parameter is missing, the default service name is used, equal to the name of the executable file of the software being installed. When a <srvname> parameter is specified, its value will be converted to lowercase, and the service will be installed with a name in lowercase.

Installing two services with the same name is not allowed, even in different root directories. If you try to do this, the installation will fail with an error.

-d <rootdir>

The launch parameter used when installing the service (with the **-i** launch parameter combination) to specify the root directory for installing the service other than the **/opt/arqasrv** default directory. This is the directory in which the **<softwaredir>/<srvname>** sub-directory is created into which the executable and configuration files of the installed software will be unpacked.

-n <srvusername>

The launch parameter used only when installing the service (with the **-i** launch parameter combination) to specify the <srvusername> user who will be the files owner of the service and on whose behalf the installed service will run. The user name by default: **arqasrv**.

-u [<srvname>]

Executing with this launch parameter is available for the arqasrv and root users.

Updating the service named <srvname>. The <srvname> parameter is optional.

If the <srvname> parameter is missing, the default service name is used, equal to the name of the executable file of the software being installed.

The service location and the user-owner name of the server are not required, because it is saved in the **/etc/arqasrv/<srvname>** directory when installing the service.

-e

Executing with this launch parameter is available for any user.

Unpacking the full archive including configuration file templates. Unpacking is performed to a subdirectory with a name equal to the name of the run file (without .run files), which is created next to the run file in case of absence.

-h, -?

Executing with this launch parameter is available for any user.

Output a summary of the launch parameters.

For example:



Running the run file to install the service:

```
sudo ./install_server-xx.xx.xx.run -i quikserver -d /opt/customdir
```

Running the run file to update the service:

```
[sudo] ./install_server-xx.xx.xx.run -u quikserver
```

Specifying sudo is required if the service is updated by the root user.

Running the run file to unpack:

```
./install_server-xx.xx.xx.run -e
```

2.4 Server configuration

2.4.1 Currency and cross rates configuration

1. On the computer where the QUIK server is installed, open the **quik.ini** file in the **quik\server** directory and edit it as follows:

- add a string (if not present) to [natural_level_section] section:

```
module=cur_module
```

- add strings in the end of the **quik.ini** file:

```
[cur_module]
module=dwcurg.dll
class_cfg=crossrate.ini
```

2. Open the **crossrate.ini** file in the **quik\server** directory and edit it as follows:

- specify for each currency code in [crossrate] section:
 - currency full name;
 - currency short name;
 - accuracy of cross rate value;
 - accuracy of cash position parameter values. Possible values are in the range [0, 8]. If the value is not explicitly specified, it is set as 2 by default.



Example:

```
[crossrate]
...
RUR=Rouble,Rouble,2
SUR=Rouble,Rouble,2
JPY=Japanese yen,Yen,6,0
```

NOTE:

If the accuracy of cash position parameter values for a currency differs from 2, all the interfaces, which are connected to the server, that transmit:

- _ this currency;
- _ instruments that are traded in this currency;
- _ currency pairs containing this currency;
- _ any monetary values expressed in this currency;

must support the transmission of monetary values with an accuracy that differs from 2, and an accuracy of a corresponding currency must be the same as se on the QUIK server. Otherwise, correct function of the system cannot be guaranteed.

For information about the transmission of monetary values with an accuracy that differs from 2 by a specific interface, contact QUIK technical support: quiksupport@argatech.com.

- _ specify each currency conversion rate in [crossrate_rate] section. As a rate value a free number can be used with the number of decimal places which does not exceed the **accuracy of cross rate value**, specified in the [crossrate] section (see as above).

For example:

```
[crossrate_rate]
RUR=1
SUR=1.00
JPY=0.627686
```

If you use cross rate for currencies, which are not specified in the [crossrate_rate] section, the corresponding error message will be displayed in the QMonitor program.

- _ specify in [main_currency] section:
 - _ currency code. Value by default: RUR;
 - _ currency symbol code in Unicode standard. Value by default: 0.



Currency codes are available by link:

https://www.unicode.org/charts/nameslist/n_20A0.html

- string with HTML code of a currency symbol in the format <Unicode-currency code>. If the code is specified in the 16-ary system, then add symbol 'x' before it. Value by default: "" (empty).

- Example:

```
[main_currency]
code=RUB
ucode=20BD
htmlcode=8381
```

2.4.2 QUIK server parameters configuration

QUIK server parameters are configured in the following files:

- [crypto.cfg](#);
- [quik.ini](#);
- [trade_dates.ini](#);
- [settlement_dates.ini](#).

crypto.cfg file

For more information about the crypto.cfg file format, see the User's manual of the Software for creating and managing keys (KeyGen).

Open **crypto.cfg** file in catalog <quik_dir>\server\ and edit it as follows:

1. In [CryptoProviders] section specify the encrypting methods that are used (0 – disabled, 1 – enabled), for example:

```
[CryptoProviders]
openssl=1
```

2. In section, which corresponds to the encrypting method, selected on the previous step, specify full or relative paths to the executable and configuration files, for example:

```
[openssl]
Module=openssl_pr.dll
IniFile=openssl_pr.ini
```



3. Specify the following settings in the [auth] section:

```
[auth]
sign=secprov.dll
sign_provider=openssl      ; encryption method that is used for digital signature
```

quik.ini file

Open **quik.ini** file in catalog <quik_dir>\server\ and edit it as follows:

1. In section [dealer-library], configure the following parameters:

```
<code_of_your_firm_in_trading_system>=<relative path to setting file>
```

Example:

```
[dealer-library]
TEST_FIRM=.\dealer\testfirm.ini
```

2. In section [DB], configure the following parameters to access the database:

- _ **Server** – name of a server. Value by default: QDB1.
- _ **Base** – name of a database. Value by default: QUIK_DB.
- _ **User** – name of a database user.

A user must be included in the quik_server role in the QUIK server database.

- _ **Password** – password of a database user. A password must not begin with \$ and #.
The password can be [encrypted](#).

Stop orders of users partially or completely blocked in QUIK Administrator (authorization is expired, the “Blocked” attribute is selected, authorization for markets is expired) can be also loaded from the QUIK server database at startup. Checking for blocked users and setting a period are managed by the quik.ini file settings:

- _ **stoporders.load_when_expired_la** – indicates of loading of stop orders for users with a general authorization expired as well as expired authorization for markets.
Valid values:
 - _ 0 (by default) – stop orders of the users with expired authorization are not loaded;
 - _ 1 – stop orders of the users with expired authorization are loaded.
- _ **stoporders.mark_expired_as_foreign** – indicates of marking stop orders of the users with expired authorization by the “Foreign” attribute. The setting is used if stoporders.load_when_expired_la=1. Valid values;



- _ 0 – stop orders of the users with expired authorization are not marked by the “Foreign” attribute;
- _ 1 (by default) – stop orders of the users with expired authorization are marked by the “Foreign” attribute.
- _ **stoporders.expired_la_period** – number of days for loading stop orders of blocked users. Value by default: 14.

3. In section [Files], configure the following parameters:

- _ **EventsLogFileName** * – path and name of the main log file. Value by default: events.log.
- _ **ErrorsLogFileName** * – path and name of the errors log file. Value by default: errors.log.
- _ **DeallibLogFileName** * – path and name of the Limits calculation library log file. Value by default: Deallib.log.
- _ **DataFilesFolderName** ** – path to data files (.ik files). Value by default: .\.
- _ **LogFilesFolderName** – path to log files (events.log, errors.log, DB provider, deallib.log). Value by default: .\.

(*) If the parameter contains only log file name, the path specified in the LogFilesFolderName parameter is used.

() The DataFilesFolderName parameter contains only path to folder that will create and store .ik files. This folder must be created before the server running. When this folder is missing, the server will be stopped.**

4. In section [usend_module], configure the following parameters:

```
module=dwsend.dll
sysname=Information and Trading System QUIK "TestInvest" ; name of system displayed
in the informational window when connecting to server
name=testinvest ; server key
password=secret ; password to access the server key
port=15100 ; port number for user terminals connection
crypto_provider_ini_file=OpenSSL_Pr.ini; full or relative path to the encryption
settings file
connection_acceptance_tmo_ms ; timeout in milliseconds to wait for a new
connection. Valid values: from 0 to 60000. Value by default: 500
```

5. In section [ctl_module], configure the following parameters:

```
module=dwwqctl.dll
port=15120 ; port number for QMonitor connection
```



```
name=testinvest      ; server key
password=secret      ; password to access the server key
```

6. In section [time-set], specify the **hourshift** parameter if the server time is different from the interface time:

```
hourshift=3 ; difference between the QUIK server time and the time of data
transmitted by the interface, in hours
```

7. In section [mail], configure the following parameters of access to SMTP server:

```
Server=smtp.quik.ru ; Name or IP address of SMTP server
Port=25000 ; Port of SMTP server. Value by default: 25
User=test_user ; User name for authentication on SMTP server
Password=secret ; Password for authentication on SMTP server. A password must not
begin with $ and #. A password can be encrypted
FromAddress ; Email address from which QUIK server letters are sent (for example:
two-factor authentication, password retrieval, limits load and etc.)
```

8. In section [limits_module], configure the following parameters:

_ for mail distribution:

```
loadlimits_mail_to=test1@post.ru ; Addresses of message receivers separated by commas
sign_settings_file=.\\qloadlimitx.ini ; Path to the QLoadLimitsX program settings
file for setting up clients' positions that QUIK server uses to load the limits file
signed by electronic digital signature
```

_ **calclimit_old_stoporders** – attribute of blocking funds under stop orders of previous days. Valid values:

_ 0 (by default) – do not block funds;

_ 1 – block funds.

For example:

```
calclimit_old_stoporders=0
```

The setting must not be used simultaneously with the Do not reserve actives for stop-orders on all of classes setting of Limits calculation library (for the setting description, see the Administrator's manual of Limits calculation library settings).



—
9. In section [trans_module], configure the parameters:

- **Change_wrong_single_clientcode** – the algorithm of client code change. The parameter is responsible for algorithm of substitution if the user has the only active client code. Valid values:
 - 0 – transaction is rejected with diagnostics 'Incorrect client code' if the client code does not match the user rights (Client specified the only code that does not match the rights. Or the client has two codes, one of them has an authorization expired and the client code in the submitted transaction does not match rights);
 - 1 (by default) – transaction is not rejected.

Example:

```
change_wrong_single_clientcode=1
```

- — **trans_handling_privileged_users** – the list of users whose transactions are processed by the server regardless of the processing stage of these users' previous transactions.

The setting is applied after saving the configuration file, the server reboot is not required.

Format:

```
trans_handling_privileged_users=<list of UIDs, separated by commas>
```

Example:

```
trans_handling_privileged_users=100,200,300
```

10.In section [data_module], configure the parameter.

- **inactive_stops_lifetime** – maximum possible time in days from the last activation of stop order after which the stop order is cancelled. Values in range 1 to 13 are rounded to the minimum possible value – 14 days.
 - 0 – the setting is disabled;
 - 14 (by default) – minimum possible number of days (setting enabled).

Example:

```
inactive_stops_lifetime=14
```



- **news_days_keep** – duration of news storage on the server (in days). When the server is started, the news, the storage time of which exceeds the value specified in the parameter, are deleted from the storage. Value by default: 2.

If the replacement settings are enabled in the section (firm_substitute = 1**, **client_codes_substitute = 1**), and the **codes.ini** or **firms.ini** files are empty or missing, the server crashes with an error:**

Error reason	Diagnostics
codes.ini replacement file is missing	Configuration file "path-to-codes.ini" not found
firms.ini replacement file is missing	Configuration file "path-to-firms.ini" not found
codes.ini replacement file is empty	Configuration file "path-to-codes.ini" is empty
firms.ini replacement file is empty	Configuration file "path-to-firms.ini" is empty

11.In section [Position], configure correspondence between alias of the firm (an alias used to replace a firm code with any other code to combine several firms into one within a unified cash position) from the Trading interface and firm code on the QUIK server with the unified cash position. The setting is relevant if the firm code of the trading interface and the unified cash position firm coincide. Alias is set by the trading interface when transmitting data on futures restrictions and positions of the unified cash position firm. Using the alias defines a type of account the data from the trading interface is intended for.

Format:

```
<alias firm with the unified cash position of Trading interface>=<code of firm with the unified cash position on QUIK server>
```

Example:

```
UNIFIED_FIRM=NC0038900000
```

Additional alias configuration is also required in section [Position] of MOEX Derivatives trading interface. Alias length must not exceed 12 characters.

12.In the [data_pipe_module] section, configure the inbound / outbound replication of the date / time of the last login using the following settings:

- **disable_last_login_replication_in** – accept / not receive information about the date / time of the last login from the replicator. Default value is 0 (accept).
- **disable_last_login_replication_out** – send / not send information about the date / time of the last login to the replicator. Default value is 0 (send).



13. If you use the Module for forming cross currency rates, then server rejects connections of the others Interfaces before connecting to this Module. To disable this mode, use the setting in the [data_pipe_module] section:

- _ **WaitforCrossrate** – cross rates waiting mode. Valid values:
 - _ 0 – disable mode.
 - _ 1 (by default) – enable mode.

When the server uses the integrated Module for forming cross currency rates, the setting value is ignored.

The CROSSRATE class must be selected in the Q2Q service settings in section of the separate connection for correct work of the slave server.

For example:

```
[pt_connect_crossrate]
connection_transport=tcp
server_ip_address=127.0.0.1
port=17103
check_tmo=150
Classes=|CROSSRATE|
```

14. Set up two-factor authentication according to the section 2 of the Administrator's manual for the QUIK two-factor authentication.

15. Perform the setting of long order entry prohibition to prohibit or allow the user to submit orders with the "Good till date" / "Good till cancel" validity period to the trading system.

- _ Specify the following parameters in the [gtc-orders] section:
 - _ **markets** – list of markets in random order, separated by commas.
 - _ The **<Market>_classes** view parameters – classes of the market, which is specified in the parameter name, in any order separated by commas.
 - _ The **<Market>_expiration_mark_field** view parameters – name of the expiration attribute field for the market.
 - _ The **<Market>_expiration_date_field** view parameters – name of the expiration date field for the market.

An empty list of markets or classes in these markets means that the prohibition on entering long orders is disabled.

- _ In the **max_days_to_expire_order** parameter of the [calc-limits] section specify the maximum validity period of long orders in days, which will be valid for all classes of instruments listed in the **<Market>_classes** view parameters. Submission of an order that exceeds the **max_days_to_expire_order** limit will be terminated with an error.



Example:

```
[calc_limits]
max_days_to_expire_order=<number>

[gtc-orders]
markets=MARKET1, MARKET2
MARKET1_classes=XXXXX,NNNNN
MARKET2_classes=YYYYY,MMMMM
MARKET1_expiration_mark_field=UseExpDate
MARKET1_expiration_date_field=ExpDate
MARKET2_expiration_mark_field=IMMCANCEL
MARKET2_expiration_date_field=EXPIRYDATE
```

If the “GTD/GTC orders” attribute is enabled in the user settings in the QUIK Administrator program, the max_days_to_expire_order parameter value will not affect the order validity period.

If the max_days_to_expire_order parameter value specified in the server is greater than the limits specified in the gateway (for example: Algorithmic trading module, Moscow Exchange Derivative Market Trading Interface), then the gateway will reject this parameter.

16.In the [behavior] section, the hyperlink can be set, which is displayed in the QUIK Workstation in the **Information on instrument** table, to open the site with the detailed description of the instrument:

- _ security.hyperlink – set hyperlink format. You can use the metacharacters to create a unique hyperlink for each instrument.
- _ security.show_hyperlinks – set the attribute that description instrument hyperlink must be displayed in the QUIK Workstation in the **Information on instrument** table. Valid values:
 - _ 0 (by default) – not displayed in the QUIK Workstation.
 - _ 1 – displayed in the QUIK Workstation.

For example:

```
[behavior]
security.hyperlink=https://www.moex.com/ru/issue.aspx?code=<SECCODE>
security.show_hyperlinks=1
```

17.The following parameters are set in the [trade_dates] section:



- **ClearingTime** – session change time on the QUIK server in <HHMMSS> format. This point in time must be within the period when the QUIK server is stopped. Value by default: 000000.
- **no_weekend** – attribute of the QUIK server operation in the **no days off** mode (Saturday and Sunday are considered working days). Valid values:
 - 0 (by default) – the QUIK server operation mode is determined by the standard calendar of working (weekdays) and non-working (weekends) days;
 - 1 – the QUIK server operates in the 7 days a week mode.

trade_dates.ini file

Open the **trade_dates.ini** file in the <quik_dir>\server\ directory and edit it.

In the [trade_date] section set up the exception dates for the calendar of trading and non-trading days used on the QUIK server. By default, from Monday through Friday are considered by the QUIK Server as trading days, and Saturday and Sunday as non-trading days. A day is considered to be a trading day if on this day there are trades in this QUIK server on any of the trading venues:

- If trading is planned on a weekday at least on one venue available on the QUIK server, then this date does not need to be added to the [trade_date] section.
- If on a weekday trading is not scheduled on any of the venue, then you need to add this date with the **holiday** attribute to the [trade_date] section.
- If trading is planned on a weekend on at least one venue available on the QUIK server, then you need to add this date with the **workday** attribute to the [trade_date] section.
- Exclusion dates from the standard calendar, according to which the QUIK server is launched in the working or non-working day mode, in the format:

```
<YYYYMMDD>=holiday
<YYYYMMDD>=workday
```

Where:

- **holiday** is non-working day;
- **workday** is working day.

See [note](#) for an example of how to fill in a section.

When starting the QUIK server with the [-holiday](#) or [-workday](#) launch parameters, section values have a lower priority.

settlement_dates.ini file

Open the **settlement_dates.ini** file in the <quik_dir>\server\ directory and edit it.

In the [settlement_date] section the exception dates are set for the calendar of settlement and non-settlement days used on the main trading venue. By default, from Monday through Friday



are considered by the QUIK Server as settlement days, and Saturday and Sunday as non-settlement days.

If a weekday is considered a non-settlement day on the main trading venue, then it must be added to the [settlement_date] section with the **holiday** attribute, and if a weekend is considered a settlement day, then it must be added with the **workday** attribute.

The described exception dates are applied to all trading modes, but have a lower priority than calendars set in the QUIK Administrator program. If a date is set in the [settlement_date] section as a holiday, but in the QUIK Administrator calendar as a working day, it will be considered as a working day.

The presence/absence of holidays results in the shift of the settlement date, which is indicated in the increase/decrease of calendar days between the trade execution and settlement when transactions with settlement codes other than T0 are performed, for example, T1, T2.

Filling in the [settlement_date] section has an impact on:

- Correct accounting of commission volume;
- Specification of the limit chain to which the blocking, crediting and debiting are applied;
- Identification of the last settlement period in the transaction limit chain for reports/trades for execution of the second part of REPO;
- Calculation of REPO term for non-negotiated and negotiated REPO modes and REPO with CCP;
- Accounting of ACI when calculating the blocked order volume on limits for REPO-M and REPO with CCP;
- Calculation of the trading system commission on deliverable SWAPs in the FX market which determines the start date of the first part of REPO;
- Specification of life period of GTD/GTC orders, including OMS orders, within which the maximum order volume is calculated considering dynamically adjusted ACI and bond face value, which affects the volume of blocked funds.

Section format:

```
<YYYYMMDD>=holiday  
<YYYYMMDD>=workday
```

Where:

- _ **holiday** is non-working day;
- _ **workday** is working day.

Examples for the [trade_date] and [settlement_date] sections:

Example 1

There are two Interfaces on the server: to the Securities market of the Moscow Exchange and the Foreign securities market of the SPB Exchange. November 4, 2022 is a holiday that



is a non-settlement day on the Securities market of the Moscow Exchange, but trading on the Foreign securities market of the SPB Exchange.

You need to set the following settings.

```
[settlement_date]
20221104=holiday
```

Example 2

There is one Interface on the server to the Securities market of the Moscow Exchange. November 4, 2022 is a holiday that is a non-settlement day on the Securities market of the Moscow Exchange. Thus, there is not a single trading mode on the server, in which trading will take place on this day.

You need to set the following settings:

```
[settlement_date]
20221104=holiday

[trade_date]
20221104=holiday
```

Example 3

There is one Interface on the server to the Securities market of the Moscow Exchange. March 5, 2022 (Saturday) is a trading and settlement day.

You need to set the following settings:

```
[settlement_date]
20220305=workday

[trade_date]
20220305=workday
```

2.4.3 User distribution configuration

The <inst_dir>\front catalog contains the current QUIK Workstation distribution and intended for distribution to customers. When a user connects to the server, the user QUIK Workstation version is compared with the version stored in directory with current QUIK Workstation distribution. If versions do not match, then user will be offered an update.

Copy the contents of the <inst_dir>/front catalog to the directory that is intended to store QUIK Workstation distribution.



The server uses as directory with QUIK Workstation distribution by default:

- <quik_dir>/server/areas/frontend/x64/release – for QUIK Workstation version 8.0.0 and newer;
- <quik_dir>/server/areas/frontend/release – for QUIK Workstation version older than 8.0.0.

The directory of the QUIK Workstation installation kit can be overridden by the **frontend** setting in the [areas] section in the quik.ini configuration file of the QUIK server. In this setting, only path to common directory that storing both distributions is specified. Value by default: **frontend=areas/frontend**.

For example:

```
[areas]
frontend=clientfront
```

In this example, the QUIK Workstation distribution is stored in the following directories:

- <quik_dir>/server/clientfront/x64/release – for QUIK Workstation version 8.0.0 and newer;
- <quik_dir>/server/clientfront/release – for QUIK Workstation version older than 8.0.0.

2.4.4 Proxy configuration to connect users

The server supports the HAProxy protocol version 1 when processing user connections. When setting up the proxy, it is required to specify the server port that is specified in the [usend_module] section.

2.4.5 Block encryption configuration

Open to edit the file in catalog <quik_dir>\server\quik.ini and configure the following parameters in section [usend_module]:

- **enable_block_crypt** – indicates of block encryption. Valid values:
 - _ 0 (by default) – block encryption is not used;
 - _ 1 – block encryption is used.
- **block_crypt_size** – size of encryption block. Valid values: from 100 to 8192 (inclusively). Value by default: 1024.

2.4.6 Setting Limits calculation library asynchronous usage

If the parallel pre-trade and post trade control of operations with different orders is required to be set up, contact QUIK technical support: quiksupport@argatech.com for further instructions.



Firms to which asynchronous Limits calculation library usage is applied should be listed in the <quik_dir>\server\quik.ini file in the [dealer-library-async] section. For example:

```
[dealer-library-async]
NC0038900000
SPBFUT389
```

The following parameters can be set up in the [behavior] section:

- **numberofexecutors** – full number of queues of parallel processing of Limits calculation library algorithms for all the firms which use Limits calculation library asynchronously.
- **deallib.temppath** – file path to the directory for temporary storage of DLL copies for Limits calculation library.

For example:

```
[behavior]
deallib.temppath=.\temp
numberofexecutors=3
```

2.4.7 Exclusions for antivirus software

This section contains files and folders of the QUIK server that must be excluded from checking by antivirus software, as they are used during the server operation.

Files and folders to be excluded:

1. Folder with the QUIK server.
2. Folders and files specified in the server settings in quik.ini:
 - paths to log files and .ik files that are specified in the [Files] section in the **LogFilesFolderName** and **DataFilesFolderName** parameters, for example:

```
LogFilesFolderName=.\logs
DataFilesFolderName=.\data
```

- paths to the files of substitutions that are specified in the [data_module] section in the **firm_substitute_file** and **client_codes_substitute_file** parameters, for example:

```
firm_substitute_file=.\firms.ini
client_codes_substitute_file=.\codes.ini
```



- paths to the archives that are specified in the [archive_module] section in the **ttp_dir** and **minutes_dir** parameters, for example:

```
ttp_dir=.\archive\minutes_ttp
minutes_dir=.\archive\minutes
```

- path to the cross-rate settings that is specified in the [cur_module] section in the **class_cfg** parameter, for example:

```
class_cfg=crossrate.ini
```

- path to the crypto provider settings that is specified in the [usend_module] section in the **crypto_provider_ini_file** parameter, for example:

```
crypto_provider_ini_file=.\openssl_pr.ini
```

- paths to the templates that are specified in the [quik_auth] section in the **TemplateRuRetrievingNotSupportedFileName** and **TemplateRuUnknownErrorFileName** parameters, for example:

```
TemplateRuRetrievingNotSupportedFileName=templates\ru\retrievingnotsupported
templaterruunknownerrorfilename=templates\ru\unknownerror
```

3. Folders with the settings of crypto providers in the crypto.cfg file that are specified in the **IniFile** parameter, for example:

```
IniFile=openssl_pr.ini
```

The net ports that are specified in the settings of server and access servers in the quik.ini file in the port parameter must be added to the exclusions.

2.4.8 Configuration of notification text about expiration of attorney power

To configure the text of the sending notifications about the imminent expiration of attorney power of the connected user, set the following settings in the crypto.cfg:

- Set the day number before expiration of attorney power which the user will receive daily notifications (each time to connect) in the [register] section. Value by default: 3.

Example:

```
[register]
```



```
license_warning_days=4
```

- Set text description of the messages in the [auth] section. The key is the template string of the view:

```
lic_exp_<d>_day_<lang>
```

Where:

- _ <d> – day number before expiration of attorney power. Valid values: 1, 2, 3, <...> and etcetera;
- _ lic_exp_today_<lang> – correspond with current day;
- _ n – all the other days;
- _ <lang> – language code of the workstation. Valid values:
 - _ 25 – Russian;
 - _ 9 – English.

The lic_exp_today_<lang> template string is using to install text description, when the power of attorney expires today for client.

The lic_exp_n_days_<lang> template string is using to install text description by default.

Example:

```
[auth]
lic_exp_n_days_25=The attorney power will expire soon
lic_exp_1_day_25=One day left before the expiration of attorney power
lic_exp_today_25= The attorney power expires today
```

The text description supports the identifiers of the format that allows to enter the date of termination of attorney power on identifier location in the message. Supports the following identifiers:

- _ <> – number of days before expiration of attorney power;
- _ <d> – day number in the month when expiration of attorney power;
- _ <m> – month number in the year when expiration of attorney power;
- _ <y> – year, when expiration of attorney power.

Example:

```
[auth]
```



```
lic_exp_n_days_25=Attorney power will expire after <> days
```

Example:

```
[auth]
lic_exp_n_days_25= Attorney power will expire <d>.<m>.<y>
```

2.4.9 Position rolling

The QUIK server can be launched in the mode of rolling clients' positions by money and instruments (stock market).

Before configuring position rolling, contact QUIK technical support for additional instructions: quiksupport@argatech.com.

To roll positions, do the configuration:

1. In the **quik.ini** file located in the <quik_dir>\server\ directory configure the following parameters:

- [data_module] section:
 - **rolling.rotation_depth** – rotation depth of files with positions located in the Backup directory. The server stores no more than the number of files specified in the setting for each type of position by money and by instrument. When the maximum value of the setting is reached, old files are deleted. Default value: 7.
 - **rolling.CPLimitTag** – tags separated by commas, which are used by the QUIK REPO Module to maintain limits for counterparties when rolling money positions. Only positions with settlement term T0 with the specified tags are considered. Default value: "" (empty).
- [calc-limits] section:
 - **UnifiedLeverageOnClient** – indicates that leverage of the same size is used in all cash positions of a client for whom **By discounts** or **MD+** margin lending scheme is used. Set the value 1. Default value is 0.

2. Start the server with the [-rolling \(/ro\)](#) launch parameter.

3. Connect the interface that will activate the positions after rolling.

The information of the transferred positions is updated only after the interfaces (trade interfaces or the Holiday Information interface) are connected.



2.5 Server migration from Windows to Linux

When migrating the service running on Windows OS to the computer with the Astra Linux OS, require to remember the following Linux version limits compared on Windows version:

- MP crypto provider is not supported.
- Migrating software that runs on Windows OS to the computer with the Astra Linux OS:
 - The version of the software that is migrated from Windows must be equal to the version of the software that will be installed on Astra Linux.
 - The software on Astra Linux OS must be used as DBMS PostgreSQL.
 - The migration must be performed during strictly non-trading times: after the end of trading and stopping the service, and before the start of trading, before the first launch of the service.

Service migration is executed by two steps:

- Service installation on Astra Linux OS (executed by **root** user).
- Migrate configuration with Windows OS (executed by **root** or **arqasrv** user).

To install software on Astra Linux OS, execute the following operations:

1. Copy the `install_server-xx.xx.xx.run` file to the directory where current user has permissions to perform operations. For example, home directory: (**/home/<user>** or **~/**).
2. Add execution permissions to the run file by running the command:

```
chmod +x install_server-xx.xx.xx.run
```

3. Install the service by running the command:

```
sudo ./install_server-xx.xx.xx.run -i
```

Require to enter the current user's password to execute the command. The following operations will be performed during the execution of the run file:

- If there is no **arqasrv** system user, a user with default permissions is created and included in the group of the same name. The service to be installed will work on behalf of this user. The **/opt/arqasrv** directory will be set as a home directory for this user.
- A subdirectory with the **quik** name is created in the **/opt/arqasrv/quik** directory. This is the service installation directory into which the distribution files of the installed software are unpacked.
- The **arqasrv** user is owner of all files and subdirectories in the service directory.
- A service named **quik** will be registered in the systemd service administration system and run as the **arqasrv** system user.
- A file named **quik** is created in the **/etc/arqasrv** directory to fix the service installation parameters.



- _ The following command files are generated in the service directory:
- _ **q_start.sh** – start the service, including with launch parameters. Intended to be performed by the **arqasrv** and **root** users;
- _ **q_stop.sh** – stop the service. Intended to be performed by the **arqasrv** and **root** users;
- _ **q_status.sh** – output of the current service status. Intended to be performed by the **arqasrv** and **root** users;
- _ **uninstall.sh** – remove service. Intended to be performed by the **root** user;
- _ **postmigration.sh** – convert configuration file names to lower case during migration. Intended to be performed by the **arqasrv** and **root** users.

If the service is successfully installed, the following message is displayed in the console: "Installation of service 'quik' has been successfully completed". In case of another diagnostic output, please contact the QUIK technical support: quiksupport@argatech.com.

4. Install libjemalloc2 package from the official Astra Linux repository. To do this, run the following command:

```
sudo apt install libjemalloc2
```

This library implements RAM handling mechanisms that are more appropriate for server scenarios than those built into the system by default.

5. Configure the connection to PostgreSQL DBMS:

- _ ODBS driver is used to operate with DBMS. For driver operation require to install the libodbc1 and unixodbc packages by running the commands:

```
sudo apt install libodbc1
sudo apt install unixodbc
```

- _ Use link to download installing file of the driver from the official ARQA Technologies website:

ftp://ftp.quik.ru/public/updates/odbc/odbc_pg_linux_16.0.5.zip

- _ Unpack the archive and install deb packages:

```
sudo dpkg -i libpq5_16.3-1.pgdg100+1_amd64.deb
sudo dpkg -i odbc-postgresql_16.00.0005-1.pgdg100+1_amd64.deb
```

- _ Make sure that driver record was added in the /etc/odbcinst.ini file:



```
[PostgreSQL ANSI]
Description=PostgreSQL ODBC driver (ANSI version)
Driver=psqlodbc.so
Setup=libodbcpsqlS.so
Debug=0
CommLog=1
UsageCount=1
```

If the record was not added, then add it yourself.

6. If the administration of the installed service is expected to be performed by the arqasrv user, execute the following steps:
 - _ Prevent modifications by the arqasrv user in the scripts of the service administration, on which the arqasrv user will be granted permission to execute as the root user. Make them immutable by running the following commands:

```
chattr +i /opt/arqasrv/quik/quik/q_start.sh
chattr +i /opt/arqasrv/quik/quik/q_stop.sh
```

- _ Give to arqasrv user permissions for executing scripts of the service administration by root name, by running the following steps:
- _ Run the command:

```
sudo visudo
```

- _ Add the string in the vi-like editor that opens:

```
arqasrv ALL=(root) /opt/arqasrv/quik/quik/q_start.sh,
/opt/arqasrv/quik/quik/q_stop.sh
```

This string permits the arqasrv user to execute the listed scripts on current computer as the root name.

To migrate configuration with Windows OS, execute the following operations:

1. Copy from the Windows service directory to /opt/arqasrv/quik/<srvname> Linux service directory the following files:
 - _ Main configuration file: quik.ini;
 - _ General file of the crypto provider settings: crypto.cfg;
 - _ Files of the Limits calculation library settings: Dealer/ directory;



- _ Files of the templates-messages for two factor authentications: Templates directory;
- _ Cross rate settings file: crossrate.ini;
- _ Company substitution settings file: areas/firms.ini;
- _ File of banned UIDs: restu.cfg;
- _ Client code replacement settings file: codes.ini;
- _ Other .ini settings files, provider certificate directories:
 - _ The **secring.txk**, **pubring.txk** keys to set up the crypto32 crypto provider;
 - _ The **certs** catalog to set up the OpenSSL_pr crypto provider;
 - _ The **signalcom** catalog to set up the Signalcom_pr crypto provider;
 - _ The catalog is including the chart archives (value by default: "minutes"), if they are stored locally, near the server.

2. Convert the names of the files and directories copied in the previous step to lower case.
For this, run the command in the /opt/arqasrv/quik/<srvname> directory:

```
sudo ./postmigration.sh
```

3. Change the settings in the replaced files:

- _ All mentions of dynamic libraries (.dll) must be converted to the corresponding name on Linux. For this, the prefix "lib" is added to the name and the extension is replaced with .so, for example:

```
module=dwdata.dll
replace with
module=libdwdata.so
```

- _ quik.ini file sections list where dynamic libraries names which must be converted are specified:

```
[data_module]
module=dwdata.dll
replace with
module=libdwdata.so

[conn_module]
module=dwxs.dll
replace with
module=libdwxs.so

[trans_module]
module=dwtrans.dll
replace with
module=libdwtrans.so
```




```
[auth_module]
module=dwqx.dll
replace with
module=libdwqx.so

[limits_module]
module=dwlimits.dll
replace with
module=libdwlimits.so

[usend_module]
module=dwsend.dll
replace with
module=libdwsend.so

[ctl_module]
module=dwwqctl.dll
replace with
module=libdwwqctl.so

[msg_module]
module=dwqtalk.dll
replace with
module=libdwqtalk.so

[archive_module]
module=dwarch.dll
replace with
module=libdwarch.so

[banner_module]
module=dwbanner.dll
replace with
module=libdwbanner.so

[data_ip_module]
module=dtwipsrv.dll
replace with
module=libdtwipsrv.so

[report_module]
module=dwrep.dll
replace with
module=libdwrep.so
```



```
[data_pipe_module]
module=dtwsrv.dll
replace with
module=libdtwsrv.so

[db_module]
module=dstubrc.dll
driver=quik_db_pg.dll
replace with
module=libdstubrc.so
driver=libquik_db_pg.so

[cur_module]
module=dwcurg.dll
replace with
module=libdwcurg.so
```

- Replace dynamic libraries names in the `crypto.cfg` file. Sections where dynamic libraries names are specified:

```
[auth]
sign=secprov.dll
replace with
sign=libsecprov.so
```

For **look** setting, it is required to find the specified section, for example, if the **db_look** is set:

```
[auth]
look=crypto.cfg,db_look

[db_look]
use=dwqx.dll
replace with
use=libdwqx.so
```

- Change dynamic libraries names and configuration file register in the settings of the `crypto.cfg` file crypto providers (when there are the list of the sections):

```
[openssl]
Module=OpenSSL_Pr.dll
IniFile=OpenSSL_Pr.ini
replace with
```

```
[openssl]
Module=libopenssl_pr.so
IniFile=openssl_pr.ini

[SignalComMessageAndSSLPro]
Module=signalcom_pr.dll
IniFile=signalcom_pr.ini
replace with
[SignalComMessageAndSSLPro]
Module=libsignalcom_pr.so
IniFile=signalcom_pr.ini

[placebo]
Module=Placebo_Pr.dll
replace with
[placebo]
Module=libplacebo_pr.so
```

If, in addition to the listed crypto providers, there are other configured crypto providers in the `crypto.cfg` file, then the corresponding sections of the settings must be commented out by adding the ";" character (semicolon) to the beginning of the line. It is also necessary comment out the declaration in the `[CryptoProviders]` section, for example, disabling the MP provider:

```
[CryptoProviders]
;mp=1

; [mp]
;Module=MP_Pr.dll
;IniFile=MP_Pr_server.ini
```

- Change the name and letter case of the configuration file in the list of the loaded settings of the Limits calculation library of the **quik.ini** file, for example:

```
[dealer-library]
ALGO=./dealer/algo.ini
```

If the paths to files with the `.dll` extension were previously configured in this section, the extension must be changed to `.ini`, for example:

```
[dealer-library]
ALGO=.\Dealer\ALGO.dll
```



```
replace with  
[dealer-library]  
ALGO=./dealer/algo.ini
```

- All references to file and directory paths must be converted to the format accepted in the Linux file system – backslashes ("\") must be replaced with forward slashes ("/"), for example:

```
crypto_provider_ini_file=.\openssl_pr.ini  
replace with  
crypto_provider_ini_file=./openssl_pr.ini
```

- quik.ini settings list that contain path to files or directories with example values:

```
[ctl_module]  
rest_users_file=restu.cfg  
  
[usend_module]  
suspect_data_file=stats/suspect.txt  
  
[key_management]  
key_info_file=key_info.ini  
  
[areas]  
frontend=areas/frontend  
msg_priv=areas/msg_priv  
public=areas/public  
company=areas/company  
private=areas/private  
unrestricted=areas/unrestricted  
hasharchives=hasharchives  
hasharchivesttp=hasharchivesttp  
  
[archive_module]  
minutes_dir=minutes  
ttp_dir=ttp  
  
[auth_module]  
settings_file=qrypto.cfg
```

- qrypto.cfg settings list that contain paths to files or directories with example values:

```
[auth]
```



```
mail_settings_file=qrypto.cfg
arw_log=stats/arw.log
```

- Edit the path to the banner file, if it is configured. The banner configuration file is the first .ini file found by the server in the /areas/banners directory. For example:

```
[url_set]
image=./banner.bmp
```

- The network drives are usually mounted to local directories therefore it is required to check with network administrators for the mount point and specify the path to it. For example:

```
[areas]
Archives=\\minutes\qminreplicator\archive\
replace with
[areas]
Archives=/mnt/qminreplicator/archive
```

4. Configure automatic daily start and stop of the service. To do this, use in the automatic scripts the **q_start.sh** and **q_stop.sh** command files located in the directory of installed service.

2.6 Server removal

Removing server installed on Windows

To remove the service, execute the following operations:

1. Stop the server service. Stop it by Service application that opened as administrator name, or by Service tab of the task manager, also opened as administrator name.
2. Execute the server service removal by running the command in the console is running by administrator name:

```
quik.exe -remove
```

3. Execute the removal of the server database.
4. Execute the removal of the server directory.

Removing server installed on Linux

The service removal is performed by the root user to call the uninstall.sh script that placed in the directory of the corresponding service.



To remove the service, execute the following operations:

1. Stop the QUIK server service from the directory with installed service by running the command:

```
sudo ./q_stop.sh
```

or from any other directory that is different from service directory by running the command:

```
sudo /opt/arqasrv/quik/quik/q_stop.sh
```

2. If the administration of the installed service is expected to be performed by the arqasrv user, execute the following steps:

- _ Cancel the arqasrv user's permissions to execute scripts of the service administration as a root name by running the following steps:
 - _ Run the command:

```
sudo visudo
```

- _ Delete the string in the vi-like editor that opens:

```
arqasrv ALL=(root) /opt/arqasrv/quik/quik/q_start.sh,  
/opt/arqasrv/quik/quik/q_stop.sh
```

- _ Disable the attribute of immutability of scripts of service administration by running the following commands:

```
chattr -i /opt/arqasrv/quik/quik/q_start.sh  
chattr -i /opt/arqasrv/quik/quik/q_stop.sh
```

3. Run the uninstall.sh command file in the /opt/arqasrv/quik/quik directory of the installed service by running the command:

```
sudo ./uninstall.sh
```

or from any other directory that is different from service directory by running the command:



```
sudo /opt/arqasrv/quik/quik/uninstall.sh
```

Require to enter the current user's password to execute the command. The following operations will be performed during the execution of the run file:

- The **quik** service registration is deleted in the **systemd** service administration system.
- The **/etc/arqasrv/quik** file in which the installation service parameters were fixed is deleted.
- The **/opt/arqasrv/quik/quik** directory in which the service is installed, changes the name to the **/opt/arqasrv/quik/quik.backup_ YYYY.MM.DD_hh.mm.ss**.

If the service is successfully removed, the following message is displayed:
"Removing of service 'quik' has been successfully completed".

If the message has a different diagnostic, please, contact the QUIK technical support: quiksupport@argatech.com.

2.7 QUIK Administrator installation and configuration

2.7.1 Installation and configuration of QUIK Administrator server part

1. To configure the database connection, install ODBC drivers for the OS.

For Windows OS:

- When connecting to MS SQL DBMS, the ODBC Driver 17 for SQL Server ODBC driver must be installed on the computer where the qadminsrv service is installed. To check the driver availability, in the Windows menu enter the text "ODBC data sources" in the search box and open the appropriate window. Make sure that the Drivers tab in the list contains the driver with the following name: **ODBC Driver 17 for SQL Server**.

If there is no such driver, download the driver installation file from the official Microsoft website (<https://go.microsoft.com/fwlink/?linkid=2223304>) and install it on the computer where the service will be running.

- When connecting to PostgresPRO DBMS, the PostgreSQL ANSI(x64) ODBC driver version 13.02 or newer must be installed on the computer where the qadminsrv service is installed. To check the driver and its version, run the odbcad32 utility in the Windows menu. Make sure that the driver with the **PostgreSQL ANSI(x64)** name and version 13.02 or newer is listed in the Drivers tab.

If there is no such driver, or the version is lower than expected, download the driver installation file from the ARQA Technologies official website (ftp://ftp.quik.ru/public/updates/odbc/odbc_pg_win_13.2.0.zip) and install it on the computer where the service will be running.

For Linux OS:

1. **Driver installation commands must be executed by the root user.**
2. **Internet access is required.**



- To work with ODBC, install the libodbc1 and unixodbc packages using the following commands:

```
apt-get install libodbc1
apt-get install unixodbc
```

- Configure drivers for DBMS.

When connecting to the PostgreSQL DBMS, the libpq5 library must be updated to version 13.0 or newer on the computer where the qadminsrv service is installed.

- Run the following commands as a user with administrator permissions:

```
echo "deb http://apt.postgresql.org/pub/repos/apt buster-pgdg main" >
/etc/apt/sources.list.d/pgdg.list
wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | apt-key
add -
apt-get update
apt remove libpq5
apt remove odbc-postgresql
apt-get install libpq5/buster-pgdg
apt-get install odbc-postgresql/buster-pgdg
```

If installation of packages from the official repository is unavailable, then use link to download installing file of the driver from the official ARQA Technologies website:

ftp://ftp.quik.ru/public/updates/odbc/odbc_pg_linux_16.0.5.zip

Update the contents of the astra repository so that the package cache is consistent:

```
apt-get update
```

Remove old versions (if any):

```
apt remove libpq5
apt remove odbc-postgresql
```

These commands may terminate with an error, this does not affect further installation.

Install the downloaded packages:

```
dpkg -i libpq5_15.6-1.pgdg100+1_amd64.deb
```




```
dpkg -i odbc-postgresql_16.00.0000-1.pgdg100+1_amd64.deb
```

- Make sure that a driver record has been added to the /etc/odbcinst.ini file:

```
[PostgreSQL ANSI]
Description=PostgreSQL ODBC driver (ANSI version)
Driver=psqlodbc.so
Setup=libodbcpsqlS.so
Debug=0
CommLog=1
UsageCount=1
```

2. Install the qadminsrv service.

For Windows OS:

- Copy all files from catalog <inst_dir>\qadminsrv\server to an arbitrary directory. The given catalog provides the **qadminsrv.ini** file containing the program settings.
- The <inst_dir>\qadminsrv\client\ catalog contains the client application of the program.
- To install the program, log in as a user with administrator permissions.
- Set the service name in the **qadminsrv.ini** file. For this, specify the appropriate values of the **name** and **displayname** parameters in the [Service] section.
- Run the command from the command line:

```
qadminsrv -install
```

For Linux OS:

The service is installed by the root user via the console by calling the run file <install_qadminsrv-xx.xx.xx.run>, where xx.xx.xx is the product version.

- Copy the install_qadminsrv-xx.xx.xx.run file to the directory that the current user has permissions to operate in, such as the home directory (/home/<user> or ~/).
- Execute the command to add execution rights to the run file:

```
chmod +x ./install_qadminsrv-xx.xx.xx.run
```

- Execute the service installation command (qadminsrv – default name of the service to be installed):



```
sudo ./install_qadminsrv-xx.xx.xx.run -i [qadminsrv] [-d <rootdir>] [-n  
<srvusername>]
```

The service is installed to the /opt/arqasrv/quik/ directory. The following operations will be performed during the execution of the run file:

- _ If there is no system user arqasrv, it will be created with default permissions and included in the group with the same name. The /opt/arqasrv directory will be set as a home directory for this user. The service to be installed will work on behalf of this user.
- _ A subdirectory with the service name will be created in the /opt/arqasrv/quik directory, the distribution files of the software to be installed will be unpacked into this subdirectory.
- _ The arqasrv user will be assigned as the owner of all files and subdirectories in the service directory.
- _ A service named qadminsrv and running on behalf of the arqasrv system user will be registered in the systemd service management system.
- _ The .actions.log file will be created in the /opt/arqasrv/quik/qadminsrv directory, where the service installation process is logged.
- _ A file named qadminsrv will be created in the /etc/arqasrv directory to record the service installation parameters.
- _ Command files will be generated in the service directory to:
 - _ q_start.sh – start the service, including launch parameters. Intended to be performed by the arqasrv and root users;
 - _ q_stop.sh – stop the service. Intended to be performed by the arqasrv and root users;
 - _ q_status.sh – output of the current service status to the console. Intended to be performed by the arqasrv and root users;
 - _ uninstall.sh – remove service. Intended to be performed by the root user;
 - _ postmigration.sh – convert configuration file names to lower case during migration. Intended to be performed by the arqasrv and root users.

If the service is successfully installed, the following message is displayed in the console: "Installation of service 'qadminsrv' has been successfully completed".

In case of another diagnostic output, please contact the QUIK technical support: quiksupport@argatech.com.

Do not change the directory structure – moving or renaming may cause errors.

- _ If the administration of the installed service is expected to be performed by the arqasrv user, execute the following steps:
- _ To prevent the arqasrv from making changes to the service administration scripts, on which the arqasrv user will be granted root permissions, make them immutable by running the following commands:



```
chattr +i /opt/arqasrv/quik/qadminsrv/q_start.sh
chattr +i /opt/arqasrv/quik/qadminsrv/q_stop.sh
```

- _ Grant the arqasrv user permissions to execute scripts of the service administration as root by running the following steps:

- _ Run the command:

```
sudo visudo
```

- _ Add the string in the vi-like editor that opens:

```
arqasrv ALL=(root) /opt/arqasrv/quik/qadminsrv/q_start.sh,
/opt/arqasrv/quik/qadminsrv/q_stop.sh
```

This string allows the arqasrv user to execute the listed scripts on current computer as root.

The following steps of this instruction can be performed by the arqasrv and root users.

3. Configure the following settings in the **qadminsrv.ini** file:

- _ In section [Instances], specify configuration files of database connections in format:

```
<Port used for the given DB>=<Name of configuration file>
```

Example:

```
[Instances]
15301=cfg_broker1.ini
15302=cfg_broker2.ini
```

For Linux OS, perform conversions for the files that are added to the [Instances] section of the qadminsrv.ini configuration file according to the [INI file conversion rules when migrating software from Windows to Linux](#).

Parameters of connection with every database are described in a separate file.

- _ In section [Files], in the **Logs** parameter set the name of file for logging, if necessary.



- _ In section [General], edit the following settings, if necessary:
 - _ **MaxDeallibSize** – the maximum file size of the Limits calculation library settings to be saved to the Audit log. Value by default: 10485760 (the value in bytes corresponds to 10 MB).
 - _ **MaxDeallibSize_<Instance>** – the maximum file size of the Limits calculation library settings to be saved to the Audit log for the specified port (<Instance>). The value of the **MaxDeallibSize** setting is used as default.
 - _ **PathRelease** – path to the directory with update files of the QUIK Administrator client part. The path may be absolute or relative. Value by default for Windows OS: frontend\release\, for Linux OS: frontend/release/ (relative path from the QUIK Administrator server part directory).
 - _ **InactiveUserDisconnectTimeout** – indicates that a connection between the server and client parts of the QUIK Administrator program breaks when the user inactivity period in seconds, which is set in this setting, expires. If the value is 0, the setting is disabled. Default value: 900.
 - _ **DisconnectWhenSessionLock** – indicates that a connection between the server and client parts of the QUIK Administrator program breaks when a session is blocked or a remote session is disconnected. Valid values:
 - _ 0 – a connection should not be broken;
 - _ 1 (by default) – a connection should be broken.

4. Configure the following settings in **<name of configuration file>.ini** server configuration file.

For Linux OS, perform conversions for all sections of the configuration file according to the [INI file conversion rules when migrating software from Windows to Linux](#).

- _ In section [DB] configure the parameters of access to the database:
 - _ **Server** – name of server.
 - _ **Base** – name of database.
 - _ **User** – name of a database user.

A user must be included in the quik_rbo and quik_auditor roles in the QUIK server database.

- _ **Password** – password of a database user. A password must not begin with \$ and #. The password can be [encrypted](#).
- _ **UsePersonalLogin** – to use personal logins for connection to the database specify the value 1.
- _ **AuthMode** – used type of authentication on MS SQL server. Valid values:



- _ login (by default) – use login and password specified in 'user' and 'password' parameters (the password can be [encrypted](#));
- _ domain – use domain authentication.

If AuthMode=domain, the 'user' and 'password' in section [DB] are ignored and may be not specified.

- _ **Driver** – name of ODBC driver which is used to connect to the QUIK server's database. Valid values:

For Windows OS:

- _ ODBC Driver 17 for SQL Server (by default) – used when using MS SQL Server 2014, 2016 or a newer DBMS (from the list of supported ones);
- _ PostgreSQL ANSI(x64) – used when using PostgresPRO;
If no setting value is specified, the "ODBC Driver 17 for SQL Server" is used.

For Linux OS:

- _ PostgreSQL ANSI (by default) – used when using Linux OS.

When operating MS SQL Server, ODBC Driver 17 for SQL Server must be installed. When operating PostgresPRO, the psqlODBC driver version 13.2 or newer must be installed.

- _ **Port** – port to connect to the PostgresPRO server. Value by default: 5432.
- _ **AdditionalParameters** – setting for specifying additional parameters of the DBMS connection. Value by default: "" (empty).
- _ **SSLMode** – secure connection mode. Supported only when configuring a connection to the PostgresPRO DBMS. Valid values:
 - _ "" (empty, by default) – connection without encryption;
 - _ require;
 - _ verify-ca;
 - _ verify-full;
 - _ disable;
 - _ allow;
 - _ prefer.

For the description of secure connection modes, see the official website of PostgreSQL: www.postgresql.org/docs/current/libpq-ssl.html.

If the parameter is not set, the CertFileName, CertPKeyFileName, CertPass and CertCAFileName parameters are ignored.



- **CertCAFileName** – path to the file container of certificates of certification authorities. Value by default: "" (empty). Supported only when configuring a connection to the PostgresPRO DBMS.
- **CertFileName** – path and name of the secure connection certificate file. Value by default: "" (empty). Supported only when configuring a connection to the PostgresPRO DBMS.
- **CertPKeyFileName** – path and name of the secure private key file. Value by default: "" (empty). Supported only when configuring a connection to the PostgresPRO DBMS.
- **CertPass** – password for the private key file of the secure connection. Value by default: "" (empty). Supported only when configuring a connection to the PostgresPRO DBMS. The password must not start with \$ and #. The password can be [encrypted](#).

MS SQL example:

```
[DB]
Server=QSERV
Base=QBROKER
User=login
Password=secret
UsePersonalLogin=1
AuthMode=login
Driver=ODBC Driver 17 for SQL Server
AdditionalParameters=MultiSubnetFailover=No;ApplicationIntent=ReadWrite
```

PostgreSQL/PostgresPRO example:

```
[DB]
Server=QSERV
Base=QBROKER
User=login
Password=secret
Driver=PostgreSQL ANSI
SSLMode=verify-full
CertCAFileName=certs/root.crt
CertFileName=certs/user.crt
CertPKeyFileName=certs/user.key
CertPass=secret
```

- Set personal logins in section of [DataBaseLogin_<random text>] view if parameter **UsePersonalLogin**=1. Any number of sections may be specified.
 - **User** – login to work with database;
 - **Password** – password to work with database. A password must not begin with \$ and #. The password can be [encrypted](#);



- **Users** – list of identifiers of users who can use the login. To establish a connection with server part of QUIK Administrator identifier must be also set in the configuration file of sub-broker access.

When using sub-administrator mode to connect to the QUIK Administrator server, the identifier must also be specified in the subbroker_template.tpl file intended for setting up the sub-broker mode in the QUIK Administrator program. For the subbroker_template.tpl file description, see sub-section 1.2 of the Administrator's manual for QUIK Administrator (for sub-broker).

- In section [Connect] configure the parameters of authorization:
 - **Name** – key identifier;
 - **Password** – password for the key. A password must not begin with \$ and #.
The password can be [encrypted](#);

Example:

```
[Connect]
Name=testinvest
Password=secret
```

- In section [QuikOptions] specify the following parameters:
 - **RSAKeysInTransReport** – the mode of displaying RSA keys of user in report on transactions. Valid values:
 - 0 – RSA keys of user are not displayed in report on transactions;
 - 1 – all RSA keys of user are displayed in report on transactions;
 - 2 – only RSA keys not revoked for the moment of transaction registration are displayed in report on transactions.
 - **ClientCodeDelimiter** – delimiter used when setting up the "Code on the dealer's side" list in the rights by classes in the user editing dialog. Value by default: "," (comma).
 - **FirmId** – list of firms (separated by commas) available for selection in the Firm code field in the rights by classes in the user editing dialog.
 - **UserExpireCheckDays** – number of days before the user's authorization expires, when the notification starts. Value by default: 5.
 - **ServiceUserExpireCheckDays** – number of days before the expiration of service user's authorization, when the system notification starts. Value by default: current value of **UserExpireCheckDays**.
 - **Title** – additional title of the QUIK Administrator program. Displayed after User UID.
 - **PINCodesInTransReport** – feature of displaying PIN codes used for the Quik two-factor authentication type in the transaction report. Valid values:



- _ 0 (by default) – PIN codes are not displayed in the transaction report;
- _ 1 – PIN codes are displayed in the transaction report.
- _ **EnableMultiple2FaLogins** – attribute of using the same login for multiple users for the Active Directory two-factor authentication type. Logins are set in the Logins for another authorization system dictionary in QUIK Administrator. Valid values:
 - _ 0 (by default) – a login can only be used by one user;
 - _ 1 – the same login can be used for multiple users.
- _ **UsersCountThreshold** – the number of entries in the Users dictionary of the QUIK Administrator program at which the dictionary is split into pages containing the number of entries set in the **PageSize** parameter. Value by default: 10000.
- _ **PageSize** – the number of entries on a page in the Users dictionary of the QUIK Administrator program. This parameter is used, if the number of dictionary entries set in the **UsersCountThreshold** parameter is reached. Value by default: 100.

5. Configure the following settings of **crypto.cfg** file in first who strings without the section title:

- _ **PUBRING** – value: quik.dwq;
- _ **SECRING** – path to file with secret key of the server.

Example:

```
PUBRING=quik.dwq
SECRING=secrserv.txk
```

6. To create a QUIK Administrator new account and assign a user with administrator's permissions, run QUIK Administrator server part using the key **-set-admin-right**. For details, see [2.7.5](#).

2.7.2 Crypto provider settings in QUIK Administrator

Configuring crypto providers in the QUIK Administrator program using the MP and OpenSSL cryptographic libraries.

The MP crypto provider is used only for Windows OS.

1. In **qadmin.ini** (for client) and **qadminsrv.ini** (for server), specify settings of the used crypto providers (for authentications on QUIK Administrator server and for checking the digital signature):

- _ In the [CryptoProviders] section, specify available providers:
 - _ **openssl** – indicates that OpenSSL provider is used. Valid values:
 - _ 0 – provider is disabled;
 - _ 1 – provider is enabled;
 - _ **mp** – indicates that MP provider is used. Valid values:
 - _ 0 – provider is disabled;



- _ 1 – provider is enabled.
- _ **cryptopro** – indicates that CryptoPro provider is used. Valid values:
 - _ 0 – provider is disabled;
 - _ 1 – provider is enabled.
- _ **signalcom** – indicates that Signal-COM provider is used. Valid values:
 - _ 0 – provider is disabled;
 - _ 1 – provider is enabled.

For example:

```
[CryptoProviders]
openssl=1
mppr=1
cryptopro=1
signalcom=1
```

- _ When the openssl provider is used, specify its settings in the [openssl] section:
 - _ **Module** – path to the provider’s module. Value by default for Windows OS: openssl_pr.dll. Value for Linux OS: libopenssl_pr.so.
 - _ **IniFile** – path to the provider’s settings file. Value by default: openssl_pr.ini.
- _ When the MP provider is used, specify its settings in the [mp] section:
 - _ **Module** – path to the provider’s module. Value by default: mp_pr.dll.
 - _ **IniFile** – path to the provider’s settings file. Value by default: mp_pr.ini.
- _ When the CryptoPro provider is used, specify its settings in the [cryptopro] section:
 - _ **Module** – path to the provider’s module. Value by default: cpcsp_pr.dll. Value for Linux OS: libcpcsp_pr.so.
 - _ **IniFile** – path to the provider’s settings file. Value by default: cpcsp_pr.ini.
- _ When the Signal-COM provider is used, specify its settings in the [signalcom] section:
 - _ **Module** – path to the provider’s module. Value by default: signalcom_pr.dll. Value for Linux OS: libsignalcom_pr.so.
 - _ **IniFile** – path to the provider’s settings file. Value by default: signalcom_pr.ini.

For Windows OS:

```
[openssl]
Module=.\openssl_pr.dll
IniFile=.\openssl_pr.ini

[mppr]
Module=.\mp_pr.dll
IniFile=.\mp_pr.ini
```



```
[cryptopro]
Module=.\cpcsp_pr.dll
IniFile=.\cpcsp_pr.ini

[signalcom]
Module=.\signalcom_pr.dll
IniFile=.\signalcom_pr.ini
```

For Linux OS:

```
[openssl]
Module=./libopenssl_pr.so
IniFile=./openssl_pr.ini

[cryptopro]
Module=./libcpcsp_pr.so
IniFile=./cpcsp_pr.ini

[signalcom]
Module=./libsignalcom_pr.so
IniFile=./signalcom_pr.ini
```

- _ In the [CryptoProvidersSettings] section, in the **CipherCryptoProvider** setting, specify the provider used login/password authentication or for authentication in ActiveDirectory domain (not supported for Linux OS). Valid values:
 - _ **openssl** – default provider – openssl_pr;
 - _ **mp** – default provider – mp_pr (used for Windows OS only).

If the key is absent the qcrypto32 provider is used by default.

To validate the digital signature, it is required only the root certificate of certification authority available in the crypto provider setting.

When you work on Windows OS, for **mp_pr** and **cpcsp_pr** crypto providers, the CA root certifications must be installed in the system store of root certificates (**CryptoPro** has its own utility) and must be accessible to the user on whose behalf the qadminsrv server is started.

When you work on Linux OS, root certificates must be installed via **CryptoPro** utility. The command is performed by user on whose behalf the qadminsrv server is started (by default: **arqasrv**).



```
/opt/cprocsp/bin/amd64/certmgr -install -store root -file certnew.p7b
```

Where:

- _ «/opt/cprocsp/bin/amd64/certmgr» – certificates manager.
- _ «-install» – command of certificates installation.
- _ «-store root» – trusted root certification authorities.
- _ «-file certnew.p7b» – name of the container file containing the root certificates.

For **signalcom_pr** and **openssl_pr** crypto providers, the root certificates are copied to the corresponding CA directory specified in the configuration file.

For **signalcom_pr** crypto provider which has the **Use dictionary for verification of signature** option, a list of client certificates similar to the QUIK server is required.

When the QUIK Administrator server and QUIK server are located on the same computer, it is allowed to use .ini files that are used to configure crypto providers.

If one QUIK Administrator server has settings to use two or more instances of the QUIK Administrator server, then they all use the same crypto providers settings.

2. Specify the setting of provider for client in openssl_pr.ini or mp_pr.ini depending on provider in use: in the [TLS Connection Settings] section in the **UsedHandShakeScheme** parameter set up the authentication scheme number. Valid values:

- _ 2 – authentication scheme by domain login and password through the Domain Controller (supported only if the server part runs under Windows OS);
- _ 4 – authentication scheme by login and password.

When multiple authentication schemes are supported, the setting value is set as a sum of values corresponding to the supported schemes.

3. Specify the settings of provider for server:

- _ If you use OpenSSL in the openssl_pr.ini file in the [TLS Connection Settings] section in the **UsedHandShakeScheme** parameter set up the authentication scheme number. To select both schemes at once (2 and 4), specify 6.
- _ If you use MP specify its settings in the mp_pr.ini file:
 - _ In the [Certificate Stores] section set up certificate storage. For details, see [5.4.2](#).
 - _ In the [TLS Connection Settings] section set up the following settings:
 - _ **UsedHandShakeScheme** – authentication scheme number in correspondence with the rules of authentication schemes compatibility at server and client;
 - _ **CSPUseLocalMachineStore=0** – the key is located in the current user's storage;
 - _ **CheckRevocation=0** – the certificate check for revocation is not performed;
 - _ **CSPCertHash** – hash of the server's certificate.

For details about the settings see [5.4.4](#).



2.7.3 Working with qadminsrv service on Linux

The qadminsrv service is managed via the systemctl system service. To register the qadminsrv service, a special configuration file with the .service file extension is created. A shortcut is created in the systemd system directory, and the configuration file is located in the directory of the corresponding service. The name of the configuration file corresponds to the name of the service. For example, for the qadminsrv service, a configuration file named qadminsrv.service is created in the /opt/arqasrv/quik/qadminsrv directory.

The service operation is controlled by the following commands on behalf of a user with administrator permissions:

- Get the current status of the service:

```
sudo systemctl status qadminsrv
```

- Start the service:

```
sudo systemctl start qadminsrv
```

- Stop the service:

```
sudo systemctl stop qadminsrv
```

You can start, stop and get the service status using the following scripts: q_start.sh, q_stop.sh and q_status.sh. The scripts should be launched using the sudo utility, for example:

```
sudo ./q_start.sh
```

Distribution

The qadminsrv service is distributed for installation on computers with Astra Linux OS as a <install_qadminsrv-xx.xx.xx.run> file format, where xx.xx.xx – product version (hereinafter run file).

The run file is used to install and update services in the system.

The run file supports the following set of launch parameters:

- -i [<srvname>]
Install the service with the <srvname> name. The <srvname> parameter is optional. If it is missing, the default service name is used, that matches the name of the executable file of the service – qadminsrv.
Installation of two services with the same name is prohibited.



- **-d <rootdir>**
Launch parameter applied only when installing the service (combined with -i) to specify a root directory for installing the service other than the default directory /opt/arqasrv.
- **-n <srvusername>**
Launch parameter applied only when installing the service (combined with -i) to override the service owner (default username: arqasrv) user who will be the owner of the service files and on whose behalf the installed service will run.
- **-u [<srvname>]**
Update the service with the <srvname> name. The <srvname> parameter is optional. If it is missing, the default service name is used, that matches the name of the executable file of the Access server – qadminsrv.
The service location and the name of the server owner user are not required to be specified, as they are stored in /etc/arqasrv/<srvname>.
- **-e**
Unpack the full contents of the archive, including configuration ini files. It is unpacked into a subdirectory with a name equal to the name of the run file (without .run), which in case of missing is created next to the executable file.
- **-h, -?**
Display a summary of the launch arguments.

Example of executing the run file to install the service:

```
sudo ./install_qadminsrv-xx.xx.xx.run -i qadminsrv -d /opt/customdir
```

Example of executing the run file to update the service:

```
sudo ./install_qadminsrv-xx.xx.xx.run -u qadminsrv
```

Example of executing the run file for unpacking:

```
./install_qadminsrv-xx.xx.xx.run -e
```

Updating service

The service is updated by the root user via the console by calling the run file.

When updating, you must specify the service name. If you are updating the service with the default name – qadminsrv, then the name does not need to be specified.

Follow the steps below to update the service:

1. Stop the qadminsrv service from the directory with the installed service by the following command:



```
sudo ./q_stop.sh
```

or from any directory other from the service directory by the following command:

```
sudo /opt/arqasrv/quik/qadminsrv/q_stop.sh
```

2. Copy the `install_qadminsrv-xx.xx.xx.run` file to the directory where the current user has permissions to operate, for example, the home directory (`/home/<user>` or `~/`).
3. Run the command to add execution permissions to the run file:

```
chmod +x install_qadminsrv-xx.xx.xx.run
```

4. Update the service by running the command :

```
sudo ./install_qadminsrv-xx.xx.xx.run -u
```

The current user's password may be required to execute the command.

As a result of the command execution, the following actions are performed:

- Automatic backup is performed. In the `/opt/arqasrv/quik/qadminsrv` directory where the service is installed, in the `.backup` subdirectory an `qadminsrv_YYYY.MM.DD_hh.mm.ss.tar.gz` archive will be created, where all files of the updated service will be placed, except for log files and storage files, relevant only within a day.
- The previous files will be replaced with a new set of files in the directory of the service to be updated.
- The **arqasrv** user will be assigned as the owner of all updated files and subdirectories in the service directory.
- Conversion of settings in configuration files will be performed if required.

If the service is successfully updated, the following message is displayed in the console:
"Update of service "qadminsrv" has been successfully completed".

In case of another diagnostic output, please contact the QUIK technical support:

quiksupport@arqatech.com.

Postponed files from previous installations can accumulate, so it is recommended to periodically delete archives from the `.backup` directory.



Removing service

The service is removed by the root user by calling the `uninstall.sh` script which is located in the directory of the corresponding service.

Further, we assume that the service with the **qadminsrv** default name is being deleted.

To delete a service, follow these steps:

1. Stop the qadminsrv service from the directory with the installed service by the following command:

```
sudo ./q_stop.sh
```

or from any directory other than the service directory by the following command:

```
sudo /opt/arqasrv/quik/qadminsrv/q_stop.sh
```

2. If the installed service was administrated by the **arqasrv** user, follow the steps:

- _ Cancel the arqasrv user permissions to execute scripts of the service administration as a root name by running the following steps:
 - _ Run the command:

```
sudo visudo
```

- _ In the opened vi-like editor, delete the line:

```
arqasrv ALL=(root) /opt/arqasrv/quik/qadminsrv/q_start.sh,  
/opt/arqasrv/quik/qadminsrv/q_stop.sh
```

- _ Disable the attribute of immutability of the service administration scripts by running the following commands:

```
chattr -i /opt/arqasrv/quik/qadminsrv/q_start.sh  
chattr -i /opt/arqasrv/quik/qadminsrv/q_stop.sh
```

3. Run the `uninstall.sh` command file in the `/opt/arqasrv/quik/qadminsrv` installed service directory by the following command:



```
sudo ./uninstall.sh
```

or from any directory other than the service directory by the following command:

```
sudo /opt/arqasrv/quik/qadminsrv/uninstall.sh
```

The current user's password may be required to execute the command. As a result of the command execution, the following actions are performed:

- The **qadminsrv** service registration is deleted from the **systemd** service administration system.
- The `/etc/arqasrv/qadminsrv` file with the service installation parameters is deleted.
- The `/opt/arqasrv/quik/qadminsrv` directory where the service is installed changes its name to the `/opt/arqasrv/quik/qadminsrv.backup_YYYY.MM.DD_hh.mm.ss`. The directory is renamed in case you need data from it. If it is not required, delete the directory manually.

If the service is successfully removed, the following message is displayed in the console:

"Removing of service 'quik' has been successfully completed".

In case of another diagnostic output, please contact the QUIK technical support:

quiksupport@arqatech.com.

2.7.4 Migrating qadminsrv service from Windows to Linux

When migrating software from Windows OS to Astra Linux OS, the following restrictions of the Linux version compared to the Windows version should be considered:

- Only PostgresPRO DBMS is supported for Astra Linux OS. When using the QUIK server database on a DBMS other than the specified one, you must first migrate the database to PostgresPRO. For instructions on database migration to PostgresPRO, please contact QUIK technical support: quiksupport@arqatech.com.
- MP crypto provider is not supported.
- To migrate software running on Windows OS to a computer with Astra Linux OS:
 - The software version that is migrated from Windows OS must be equal to the software version that will be installed on Astra Linux OS.
 - Migration must be performed at non-trading time: after the end of trading and stopping the service and before the start of trading, before the first start of the service.

Before migrating the service from Windows to Linux, it is required to update all used plugins for QUIK Administrator to the versions listed below:

- EES 2.30.100;
- QME Qme\2.2.100;
- InvAdvDistributor 3.11.100;
- Instructions 20.7.100;



- BrokerQuote 4.48.100;
- OMS OMS\6.x\6.2.100;
- POService 1.9.100;
- QuikAuctions 1.1.100;
- QChatGate V_1.14.100;
- QuikREPO 1.2.100;
- SecDescDistributor 3.4.100;
- SMS 3.19.100;
- SOR 4.8.100.

To obtain the corresponding update, please contact QUIK technical support:
quiksupport@arqatech.com.

Service migration to Astra Linux OS should be performed by the root user .

For Linux OS, conversions must be performed for all migrated files according to the [INI file conversion rules when migrating software from Windows to Linux](#).

To migrate software from Windows OS to Astra Linux OS, follow the steps below:

1. Install the service by following the steps from sub-section [2.7.1](#).
2. Copy the following files from the Windows service directory to the Linux service directory /opt/arqasrv/quik/qadminsrv:
 - _ .ini configuration files;
 - _ .cfg crypto provider configuration files;
 - _ Key files (.txk), provider certificates catalogs:
 - _ The **seccring.txk**, **pubring.txk** keys to set up the qcrypto32 crypto provider;
 - _ The **certs** catalog to set up the OpenSSL_pr crypto provider;
 - _ The **signalcom** catalog to set up the Signalcom_pr crypto provider.
3. Convert the names of the files and directories copied in the previous step to lower case. To do this, run the command in the /opt/arqasrv/quik/qadminsrv directory:

```
sudo ./postmigration.sh
```

4. Start the qadminsrv service from the service directory with the command:

```
sudo ./q_start.sh
```



or:

```
sudo systemctl start qadminsrv
```

5. Check that the service has been successfully started by the command (from the service directory):

```
sudo ./q_status.sh
```

or:

```
sudo systemctl status qadminsrv
```

In case of problems, please contact the QUIK technical support: quiksupport@argatech.com.

2.7.5 QUIK Administrator account creation

To create a QUIK Administrator new account and assign a user with administrator permissions, run QUIK Administrator server part using the **-set-admin-right** launch parameter. Using the **KeyGen** program, create public and secret keys that will be later used by QUIK server (see [keygen.pdf](#) – User’s manual for keygen.exe).

The format:

```
qadminsrv.exe -set-admin-right -instance=[<instance>] -UID=[<UID>] -QUIK-key-  
filename=[<QUIK-key-filename>]
```

Where:

- <instance> – database identifier. The database identifiers are listed in [Instances] section of the qadminsrv.ini file. If the QUIK server uses a single DB, then the parameter is not required.
- <UID> – identifier of a user to assign an RSA key. If the parameter is not specified, and empty database, a new user will be created.
- <QUIK-key-filename> – path and name of the file containing one RSA public key. The relative or absolute path can be specified. If the name and / or path contains spaces, then the value is quoted, for example, for Windows OS: "c:\path\to\file\somefile.txk", for Linux OS: "/path/to/file/somefile.txk". If the file contains more than one RSA public key, then the first key in the file is used. If the <UID> parameter is specified and there is an RSA key for a user, then this parameter is not required.

Example for Windows OS:



```
qadminsrv.exe -set-admin-right -UID=350 -QUIK-key-  
filename="c:\path\to\file\somefile.txk"
```

Example for Linux OS:

```
./qadminsrv -set-admin-right -UID=350 -QUIK-key-filename=/path/to/file/somefile.txk
```

2.7.6 QUIK system user registration

1. Run the QUIK Administrator.
2. Select the **QUIK server/Organizations** menu item. Add your organization to the dictionary by filling in all fields.
3. Select the **QUIK server / Users** menu item. Add the account of the QUIK system administrator to the user dictionary.

To do this, select the name of your organization in the dictionary, enter full name of the manager and establish authorization's validity period.

The list of classes contains the names of security classes specified in settings. Select each of them sequentially and click the **Manager** button to set permissions to work with this class.

4. Specify the identifier of the trading participant in trading system in the **Firm code** field (the **Classes permissions** tab in the user edit window). The **Client code on the dealer's side** field is not filled in for manager.
5. Set up user authentication:
 - To log in using keys. Using the **KeyGen** program, create public and secret keys of user (see [2.2.3](#), item 7). Click the **Edit** button on the **Security** tab in the **QUIK keys** field. Load from these the key by pressing button **Add**.
 - To use the login authentication mechanism, user logins must be set in the program:
 - To log in by domain login and password – specify login in the **Logins for another authentication system** dictionary.
 - When authenticating, it is allowed to specify a login both with and without a domain. If domain is specified, full domain and login matching is checked. If domain is not specified, login check is executing for domain by default.
 - To log in by login and password – specify login in the user edit window on the **Security** tab.
6. Similarly, add other users of your system installing the necessary parameters of access to the system using templates rights. Complete the field **Client code on the dealer's side** by the symbolic code that will be subsequently used by manager for setting limits and authentication of users when working with the system.

Detailed description of **QUIK Administrator** work is presented in the Administrator's manual.



QUIK system is ready to work!

Run the QUIK Workstation using the access keys and credentials of your user.

2.8 QUIK Workstation banner configuration

In QUIK workstation, a banner may be displayed on the toolbar, clicking on which opens a website (for example, a broker's website).

2.8.1 Configuration of banner

To configure banner, follow the steps below:

1. Copy a banner image file to the directory **QUIK\Server\Areas\Banners**. The file requirements are:
 - _ file format – .bmp;
 - _ file extension – any. Recommended: 468*60 pixels;
 - _ color depth – 8 bit.
2. In the directory **QUIK\Server\Areas\Banners**, in the **banners.ini** file configure the settings in section [url_set]:
 - _ **url** – URL address.
 - _ **image** – name of the banner image file.

Example:

```
[url_set]
url=http://www.broker-website-example.ru
image=broker.bmp
```

3. On the computer where a QUIK server is run, in the file **QUIK\Server\quik.ini** configure the following settings:

```
[natural_level_section]
module=banner_module
[banner_module]
module=dwbanner.dll
```

The banner settings for the server and the access server are the same

2.8.2 Deactivation of banner

To disable receiving banners from the access server, do the following:



1. Make sure that the following line is present in the [natural_level_section] section in the quik.ini access server settings file:

```
module=banner_module
```

2. Make sure quik.ini file of access server has the following section:

```
[banner_module]
module=dwbanner.dll
```

3. Make sure the default value ("1") of the following key is displayed in the [banner_module] section of quik.ini file of access server:

```
ignore_up_dwell=1
```

```
[banner_module]
module=dwbanner.dll
ignore_up_dwell=1
```

4. Make sure the dwbanner.dll module is present and then start the access server.

After applying such settings, the banner won't be sent from the server to the access server and users of the access server won't see it.

2.8.3 Replacement of banner

To replace the banner, configure the scheme as follows:

1. The .\Areas\Banners\ directory has to contain the .ini file with the settings as following:

```
[url_set]
url=c:\
image=test.bmp
```

where:

- _ Value of the url key – url, that opens in QUIK Workstation;
 - _ image – name of the file in .\Areas\Banners\ directory.
2. Change the old file to new one with the name that differs from the current banner name (for example test2.bmp) in .\Areas\Banners\ directory.
 3. Change the value of the image key to a new one in .ini file – test2.bmp.



URL can be changed without changing the image.

Also, without changing image and url, you can add another .ini file with a different url and image. In this case, the banner from the new .ini file is used.

2.9 Message output configuration

The settings allow you to specify the path to the directory of the greeting messages for users that the server will send when they connect:

```
[areas]
msg_all=./areas/msg_all;
msg_priv=./areas/msg_priv/;
```

Where:

- msg_all – message root directory for all users. Value by default: ./areas/msg_all.
- msg_priv – individual message root directory for users with a specific identifier. Value by default: ./areas/msg_priv/.

The example of the file content of the greeting message for all users:

- _ Location: ./areas/msg_all/hello.txt.
- _ Content: Welcome to our server!

The example of the file content of the greeting message for user with "42" identifier:

- _ Location: ./areas/msg_priv/42/hello_42.txt.
- _ Content: Until 31.01.2025 you can make transactions with a reduced fee.

If the user is assigned the greeting messages from msg_priv directory, then messages from msg_all directory are not sent to this user.

Greeting messages must be stored in the text file format. The maximum message text size is 799 bytes.

2.10 File transfer settings

In the QUIK system, you can transfer files from the server to QUIK workstations (via personal file areas to the QUIK server). This makes possible to organize sending of reports to customers, also reports can be signed with digital signature.

To send files to clients that are not intended for updating the QUIK workstation, the files must be placed in the specified directory:

- server\areas\private\ – for sending to specific users.
- server\areas\public\ – for sending to all users.



To place documents and files intended for the employees of the company, you should use the server\areas\company\ directory.

After that, you need to use the **System / Receiving files** dialog box:

3. Running QUIK server

QUIK server is run as a service. Market data, as well as information about orders and trades, is loaded from the repository files and is available to all users connected to the QUIK server. It allows, for example, unloading information about trades executed on the previous trading day. The data for the previous trading session is cleared when the first trading interface connects to the QUIK server.

The QUIK server can also be run using the [launch parameters](#).

To upload positions to the QUIK server when it starts, copy the prepared .lim file to the computer where the QUIK server is installed. The QUIK server service is started with the launch parameters indicating the positions to be loaded from this file. For example:

```
-clean -loadlimits <file_with_positions>
```



When the interfaces are connected to the QUIK server, the uploaded positions will be displayed in the QUIK Workstation.

If you have any questions, contact QUIK technical support:
quiksupport@argatech.com.

3.1 QUIK server launch parameters

If the configuration file of Limits calculation library is empty or is not available the QUIK server start is interrupted with conclusion of the relevant diagnostic to log file.

3.1.1 -new or /n

Start a QUIK server in the loading mode with new data loading mode in which:

- market data and orders and trades are requested from the trading system,
- client positions are calculated based on the received information considering trades of the current session and position adjustments,
- all files with .ik extension are cleared except the following:
 - _ news file (news.ik),
 - _ file of server QPILE portfolios (cstructp.ik),
 - _ cash positions file (mlimit.ik)
 - _ instruments positions file (dlimit.ik).

3.1.2 -clean or /c

Start a QUIK server in the loading mode with full cleaning of data as well as cleaning all files with .ik extension.

If specify -new and -clean launch parameters simultaneously, then -clean launch parameter will be used.

The key can be used in the combinations that include:

- [-loadlimits](#);
- [-rolling](#) and [-loadlimits](#).

3.1.3 -old or /o

Start a QUIK Server in mode of loading with old data without cleaning storages.



3.1.4 -holiday or /h

Starting a QUIK server in holiday mode on a day that is a working day according to the standard calendar or according to the settings of the [\[trade_date\]](#) section of the trade_dates.ini file.

3.1.5 -workday or /w

Starting a QUIK server in working day mode on a day that is a day off according to the standard calendar or according to the settings of the [\[trade_date\]](#) section of the trade_dates.ini file.

3.1.6 -loadlimits or /l

Start a QUIK server with loading positions from a file. Name of file with positions and its path must be typed in the command line after the key. Field formats of positions to be loaded are given in Appendix (see [5.1](#)). The information of the positions loaded from the file is updated only after connecting the interfaces (trading interfaces or the Holiday Information interface).

The parameter can be used in the combination that includes **-clean** and **-rolling** parameters (see [3.1.14](#)).

3.1.7 -install or /i

Install a QUIK Server as a service of Windows.

3.1.8 -remove or /r

Delete a QUIK Server as a service of Windows.

3.1.9 -removestops or /rs

Stopping processing of conditional orders belonging to this QUIK server. It is executed by forcibly setting an accessory identifier of the stop orders to the **Other** value in terms of the **Stop orders table** of the QUIK Workstation or **foreign** in terms of the **Stop Orders** table of the Program interface QMonitor. The command requires user confirmation. For details, see [5.7](#).

After the start, users must make stop orders own on any other server so that these orders could trigger.

The **Remove all stoporders** command or run with **-removestops** launch parameter – it is an emergency for the server, after which the server loses information about belonging of existing stop orders. Therefore, it is necessary to create conditions for full replication of the current labels. For example, regular restart (scheduled) on the next day, Replicator connection and all trading interfaces in all circuit. It is better to return stop order on the next day, before the start of trading or after trading completion.

The **-removestops** launch parameter can be used together with **-new** and **-clean** parameters.

3.1.10-text or /t

Exporting data of a particular type to a text file.



Format:

```
quik.exe -text [<FILE>] [<TYPE>] [tsv]
```

where:

- <FILE> – exported file with orders and trades.
- <TYPE> – data type. Valid values:
 - _ orders – export orders from file (order.ik);
 - _ trades – export trades from file (trade.ik).

If names of files with orders and trades are different from order.ik and trade.ik, <TYPE> parameter is required.

- tsv – convert file to tsv format.

For example:

```
quik_02.exe -text trade.ik trades tsv > trade.txt
```

3.1.11-removelimits or /rl

Start a QUIK server with cleaning data on client cash and securities positions, and restrictions of accounts at derivatives market. Other data (e.g., orders, trades, data from the Quotes table, etc.) are not deleted. The launch parameter can be used with the **-loadlimits** launch parameter.

3.1.12-rolling or /ro

Start a QUIK server with rolling positions. The information of the transferred positions is updated only after the interfaces (trade interfaces or the Holiday Information interface) are connected.

The launch parameter can be used in the combination that includes **-clean** and **-loadlimits** launch parameters (see [3.1.14](#)).

3.1.13-clean -loadlimits

Start a QUIK server in the loading mode with full cleaning of data as well as cleaning all files with .ik extension and loading positions from file. Positions are recalculated considering trades of the current session but not considering position adjustments during the trading day (adjustments must be made again).



3.1.14-clean -loadlimits -rolling

Start of a QUIK in the loading mode with full cleaning of data as well as cleaning all files with .ik extension. The file that contains positions on the moment of the previous day trading end is used as a source for rolling.

For the field formats of loaded positions, see Appendix (sub-section [5.1](#)).

4. Password encryption

Passwords specified in the Module settings can be encrypted. The password must not begin with "\$" and "#". If you use a password that begins with any of the above characters, change it.

To activate password encrypting functionality, add "\$" before the password in the configuration file.

For example:

```
Password=$secret
```

When attempting to use a password, the server encrypts the password and saves it to the configuration file in an encrypted form. The encrypted password is marked by "#" character at the start of a string.

When transferring the program to another computer or reinstalling the operational system, the encrypted passwords get unavailable for decryption. In this case, the appropriate passwords must be reset.

5. Appendix

5.1 Format of positions

Parameters obligatory for configuration are ticked by "*".

Format of instrument positions:

Parameter	Description	Size, accuracy (in symbols)
* DEPO:	Title of line with expressed instrument position	



Parameter	Description	Size, accuracy (in symbols)
* FIRM_ID	Identifier of trading participant	12
* SECCODE	Instrument code	12
* TRDACCID	Trading account	12
* CLIENT_CODE	Client code	12
* OPEN_LIMIT	Opening limit	30
* OPEN_BALANCE	Client position	30
CURRENT_LIMIT	Current limit	30
WA_POSITION_PRICE	WA position price	30, 6
WA_PRICE_CURRCODE	WA position price currency	4
LIMIT_KIND	Settlement period. Valid values: _ 0 – for current limits; _ <N> – for limits on positions with T+<N>; _ value less than zero – for technological limits; _ <YYYYMMDD> – for limits on positions in calendar days	32, 0

Format of cash position:

Parameter	Description	Size, accuracy (in symbols)
* MONEY:	Title of line with expressed cash position	
* FIRM_ID	Identifier of trading participant	12
* CURR_CODE	Currency code	4
* TAG	Settlement tag	4
* CLIENT_CODE	Client code	12
OPEN_LIMIT	Opening limit	30, 2
* OPEN_BALANCE	Client position	30, 2



Parameter	Description	Size, accuracy (in symbols)
CURRENT_LIMIT	Current limit	30, 2
LIMIT_KIND	Settlement period. Valid values: _ 0 – for current limits; _ <N> – for limits on positions with T+<N>; _ value less than zero – for technological limits; _ <YYYYMMDD> – for limits on positions in calendar days	32, 0
* LEVERAGE	Leverage. Valid values: _ Non-negative values; _ -1 – no leverage	30, 2

Example of records in positions file:

```
MONEY: FIRM_ID = NC0038900000; TAG = EQTS; CURR_CODE = SUR ; CLIENT_CODE = 004;
OPEN_BALANCE = -46.52; OPEN_LIMIT = 0.00;
MONEY: FIRM_ID = NC0038900000; TAG = EQTS; CURR_CODE = SUR ; CLIENT_CODE = 004;
OPEN_BALANCE = -46.52; LEVERAGE = -1;
MONEY: FIRM_ID = NC0038900000; TAG = EQTV; CURR_CODE = SUR ; CLIENT_CODE = 004;
OPEN_BALANCE = 1048.51; OPEN_LIMIT = 450000.00; LIMIT_KIND = 1;
DEPO: FIRM_ID = NC0038900000; SECCODE = HYDR; CLIENT_CODE =10004/10004; OPEN_BALANCE
= 113955; OPEN_LIMIT = 0; TRDACCID = L01-00000F00;
DEPO: FIRM_ID = NC0038900000; SECCODE = IRAO-004D; CLIENT_CODE =10006/10006;
OPEN_BALANCE = 738467; OPEN_LIMIT = 0; TRDACCID = L01-00000F00; LIMIT_KIND = 1;
```

5.2 INI file conversion rules when migrating software from Windows to Linux

The following conversions are required for files and paths:

- File names and paths must be in lower case.
- All mentions of dynamic libraries (.dll) must be converted to the corresponding name on Linux. For this, the prefix "lib" is added to the name and the extension is replaced by .so, for example:

```
module=openssl_pr.dll
replace with
module=libopenssl_pr.so
```

- Path delimiters – backslash "\" should be replaced by a forward slash "/".



Settings that contain paths to files located on the QUIK server should be changed so that the path is correct in Linux semantics.

If the QUIK server is installed on a separate computer, it is required to connect the network resource by a command on behalf of the root user, it is necessary to have an installed cifs-utils package included in the AstraLinux distribution kit.

For example, the QuikIni configuration of the [DealerLibraryOptions] section for Windows is as follows:

```
[DealerLibraryOptions]
QuikIni=\\QUIKSERVER\DRIVE_D\QUIK1\Server\quik.ini
```

The resource DRIVE_D is available on the computer where the QUIK server with the QUIKSERVER network name is installed.

Connection of a network resource is performed with the command:

```
sudo mount.cifs //QUIKSERVER/DRIVE_D/ /mnt/q1/ -o uid=arqasrv,user=winuser
```

Where:

- QUIKSERVER – computer name where the QUIK server is installed;
- DRIVE_D – network resource name;
- /mnt/q1/ – network disk connection path (the directory must exist);
- arqasrv – Linux user on whose behalf the QUIK Administrator service is launched;
- winuser – Windows user who is allowed to connect to the network resource.

After connecting the network resource, the configuration is as follows (the path is case sensitive):

```
[DealerLibraryOptions]
QuikIni=/mnt/q1/QUIK1/Server/quik.ini
```

The Limits Calculation Library files listed in the [dealer-library] section must be available for recording by the user on whose behalf the network resource is connected.

For more information about connecting network resources in Linux, please contact your system administrator.



5.3 OpenSSL encryption configuration file format

In the OpenSSL_Pr.ini file the settings of encryption by OpenSSL cryptographic library is configured (www.openssl.org). OpenSSL application does not require saving certificates in OS register on the platform with the QUIK server. It makes certificate installation and servicing easier.

5.3.1 [General] section

- **CertStoreFile** – certificate storage file name. Value by default: certs\store.pem.
- **ServerCertificateKey** – password of server's certificate key. The password can be encrypted (see 4).
- **NewServerCertHash** – new server's certificate hash.
- **NewServerCertDomainName** – new server's certificate domain.

NewServerCertHash and NewServerCertDomainName settings specify the parameters of a new certificate when planned certificate replacing at the server. After certificate replacement the values of NewServerCertHash and NewServerCertDomainName settings are recorded to ServerCertHash and ServerCertDomainName parameters of OpenSSL_Pr.ini file in the QUIK Workstation.

- **ClientCertificatesDir** – path to the directory that contains certificates for checking digital signature. Value by default: certs\client.

5.3.2 [TLS Connection Settings] section

The following connection parameters are specified in the section:

- **ServerHandShakeScheme** – the supported authentication scheme. Possible values:
 - _ 1 – authentication by certificate;
 - _ 2 – authentication in domain;
 - _ 4 – authentication by login and password;
 - _ 7 (by default) – all the schemes are supported besides 8 (password recovery);
 - _ 8 – password recovery;
 - _ 32 – authentication within the domain (by domain name of the user).

When a few authentication schemes are supported, the specified value of the field is the sum of values, corresponding to the supported schemes.

- **UsedHandShakeSchemePrefer** – the preferred authentication scheme. It must be selected if a client had chosen scheme 0 (is defined by server). Possible values:
 - _ 1 – authentication by certificate;
 - _ 2 – authentication in domain;
 - _ 4 (by default) – authentication by login and password.



- **SSLProtocols** – the list of TLS protocol versions available for connection, separated by a comma. Possible values:
 - _ TLSv1.0;
 - _ TLSv1.1;
 - _ TLSv1.2.
- **CA** – path to the directory that contains root certificates. Value by default: certs\CA.
- **CRL** – path to the directory that contains certificate revocation lists. Value by default: certs\CRL.
- **SecurityPackage** – authentication protocol within a domain. Possible values:
 - _ Kerberos (default);
 - _ NTLM.
- **ServerSPN** – Service Provider Name (SPN) identifier of the QUIK server in the Active Directory account. Used when **SecurityPackage** is set to **Kerberos**.

To find out the SPN, run the "setspn -l <name of the domain account under which the server is running>" command from the command line or contact your administrator.

5.4 MP encryption configuration file format

The MP crypto provider is used only for Windows OS.

The file specifies the encryption settings using the MP cryptography library:

- on the QUIK server side – in the MP_Pr_server.ini file;
- on the client side – in the MP_Pr_client.ini file.

5.4.1 [Crypto Provider Settings] section

The section contains description of the cryptographic provider settings required for check of a digital signature.

To use the digital signature, specify a key container using any of the following methods:

1. Specify parameters of the container:

- _ **CSPName** – name of a Cryptographic Service Provider (CSP) used by the certificate.
- _ **CSPContainerName** – name of the key container.
- _ **CSPTYPE** – CSP provider type.

If CSPName parameter is not specified, the system provides a CSP provider, assigned by default for a specified type of cryptographic provider.



2. Specify a file of the certificate linked to the key container:

- _ **CSPCertFileName** – path to the certificate file.

3. Specify a hash of the certificate linked to the key container:

- _ **CSPCertHash** – hash of the installed certificate.

To get a certificate hash:

- _ Open the certificate file;
- _ Switch to the **Details** tab;
- _ Copy the value of the **Thumbprint** field and remove the spaces.

Additional parameters to set up the digital signature (optional):

- **CSPUseInService** – indicates that cryptographic provider works in 'silent' mode. Possible values:

- _ 0 – cryptographic provider works online;
- _ 1 – cryptographic provider works in 'silent' mode.

The cryptographic provider doesn't interact with the user in 'silent' mode. An error appears upon the occurrence of an event that requires interactive communication (for example, entry of the password to access the container).

- **CachKeyAndCertificates** – indicates that keys and certificates are cached for quick access to them. Possible values:

- _ 0 – do not cache;
- _ 1 – cache.

- **CSPUseLocalMachineStore** – private key location. Possible values:

- _ 0 (by default) – the key is located in the current user's storage;
- _ 1 – the key is located in the local computer store.

- **CheckRevocation** – indicates that the certificate is checked for revocation. Possible values:

- _ 0 – the check is not performed;
- _ 1 – the check is performed.

- **UseLocalCA** – attribute for using the local certificate storage. Possible values:

- _ 0 – local storage is not used.
- _ 1 (default) – local storage is used.

- **CA** – path to the local CA certificate storage. Default value: Certs\CA.

- **CRL** – path to the local certificate revocation list (CRL) storage. Default value: Certs\CRL.

5.4.2 [Certificate Stores] section

The section contains description of the certificates store, where the certificates are searched to set up the encryption modes and digital signature:



At least one certificate storage should be specified to ensure correct work of the cryptographic provider.

- **MY** – attribute of standard personal storage of the user certificates. Possible values:
 - _ 0 – not used;
 - _ 1 – used.
- **CA** – attribute of standard storage of the Certification Authorities' certificates. Possible values:
 - _ 0 – not used;
 - _ 1 – used.
- **ROOT** – attribute of standard root storage. Possible values:
 - _ 0 – not used;
 - _ 1 – used.
- **SPC** – attribute of standard storage of the software vendors' certificates. Possible values:
 - _ 0 – not used;
 - _ 1 – used.
- **CUSTOM_STORE** – attribute of third-party external storage. Possible values:
 - _ 0 – not used;
 - _ 1 – used.
- **CUSTOM_DIR** – attribute of directory for storing certificates. Possible values:
 - _ 0 – not used;
 - _ 1 – used.

Each store is described by the separate section of the [<storage>] view (see [5.4.3](#) sub-section).

5.4.3 [<storage>] section

- **HOST** – host where the store is located. Value: "." (full stop).
- **LOCATION** – physical store location.
- **TYPE** – store type. Possible values:
 - _ 1 – system store;
 - _ 2 – external storage located in a specific file;
 - _ 3 – directory with certificate files.
- **UPDATABLE** – feature of an updatable store. Possible values:
 - _ 0 – not allowed to be updated;
 - _ 1 – allowed to be updated.
- **PRIORITY** – priority. Value: 1.

For example:



```
[CUSTOM_STORE]
HOST=.
LOCATION=D:\QUIK\Server\CPStorage\usercerts.sto
TYPE=2
UPDATABLE=1
PRIORITY=1
```

5.4.4 [TLS Connection Settings] section

The section is used to configure a TLS connection.

- **SSLProtocols** – list of the protocols used (separated by commas). Possible values:

- _ TLSv1;
- _ TLSv1.1;
- _ TLSv1.2 (by default).

To work in encryption mode, specify a private key. To do this, specify a certificate linked to a private key by using one of the following methods:

1. Specify parameters of the container:

- _ **CSPName** – name of a Cryptographic Service Provider (CSP) used by the certificate;
- _ **CSPContainerName** – name of the key container;
- _ **CSPTYPE** – CSP provider type.

If CSPName parameter is not specified, the system provides a CSP provider, assigned by default for a specified type of cryptographic provider.

2. Specify a file of the certificate:

- _ **CSPCertFileName** – path and name of the certificate file.

3. Specify a hash of the certificate linked to the key container:

- _ **CSPCertHash** – hash of the certificate.

Authentication parameters:

- **UsedHandShakeScheme** – authentication scheme used. Possible values:

- _ 1 * – authentication by certificate, i.e. the client is checked by certificate that he provided establishing an SSL/TLS connection;
- _ 2 * – authentication by domain login and password through the Domain Controller;

The scheme 2 cannot be used, if in the QUIK Administrator's server configuration file in the [QuikOptions] section the EnableMultiple2FaLogins setting contains '1'.



- _ 4 * – authentication scheme by login and password;
- _ 7 (by default) – all schemes are supported, except for 8 (password recovery);
- _ 8 – password recovery.
- _ 32 * – authentication within the domain (by domain name of the user).

When multiple authentication schemes are supported, the setting value is set as a sum of values corresponding to the supported schemes.

(*) Server's and client's authentication schemes compatibility:

- 1. If the authentication scheme number 1 is configured on the server, the client can use the first and fourth schemes.**
- 2. If the authentication scheme number 2 is configured on the server, the client can use the second and fourth schemes.**
- 3. If the authentication scheme number 4 is configured on the server, the client can use only the fourth scheme.**
- 4. If the authentication scheme number 32 is configured on the server, the client can use only the scheme number 32.**

- **DomainController** – address of the Domain Controller being referenced when authenticating a client. This parameter is used if the authentication scheme number 2 is applied.

If the DomainController setting is not specified, the current Domain Controller is used.

- **DomainLogin *** – user's domain login for the access server authentication. Value by default: "" (empty).
- **DomainPasswd *** – user's domain password for the access server authentication. Value by default: "" (empty).

(*): These parameters are used only when connected via the access server where the authentication scheme number 2 is configured.

Additional parameters to set up the encryption mode (optional):

- **CSPUseInService** – indicates that cryptographic provider works in 'silent' mode. Possible values:
 - _ 0 – cryptographic provider works online;
 - _ 1 – cryptographic provider works in 'silent' mode.

The cryptographic provider doesn't interact with the user in 'silent' mode. An error appears upon the occurrence of an event that requires interactive communication (for example, entry of the password to access the container).



- **CSPUseLocalMachineStore** – private key location. Possible values:
 - _ 0 (by default) – the key is located in the current user store;
 - _ 1 – the key is located in the local computer store.
- **CheckRevocation** – indicates that the certificate is checked on the revoked certificates list by the cryptographic provider. Possible values:
 - _ 0 – the check is not performed;
 - _ 1 – the check is performed.
- **IOBufferMaxSize** – buffer size (in bites) for input/output data in TLS connection handshake. Value by default: 65536.
- **CheckClientCert** – indicates that the client's certificate is verified using the additional tools of Windows operating system.
- **HashCalcCount** – number of iterations of the password hash function calculation on the server. Possible values: from 1 to 10037. Value by default: 4999.

When using authentication within a domain (**UsedHandShakeScheme=32**), the following settings are additionally specified in the `MP_Pr_client.ini` client file:

- **SecurityPackage** – authentication protocol within a domain. Possible values:
 - _ Kerberos (default);
 - _ NTLM.
- **ServerSPN** – Service Provider Name (SPN) identifier of the QUIK server in the Active Directory account. Used when **SecurityPackage** is set to **Kerberos**.

To find out the SPN, run the "setspn -l <name of the domain account under which the server is running>" command from the command line or contact your administrator.

5.4.5 [TLS Extra Parameters] section

The section contains the settings (optional) for the planned change of server's certificates.

The settings of the planned change of certificate should be specified before the expiration date of the current certificate. The settings are deleted after the certificate change.

- **NewServerCertHash** – possible values:
 - _ Hash of a new server's certificate – after the certificate change, this value is written to the **ServerCertHash** setting of the `MP_Pr_client.ini` client's file and then the value of the **NewServerCertHash** setting is deleted.
 - _ DELETE – cancellation of the certificate change procedure.



Value by default: "" (empty).

- NewServerCertDomainName – possible values:
 - _ Domain name of a new certificate – after the certificate change, this value is written to the **ServerCertDomainName** setting of the **MP_Pr_client.ini** client's file and then the value of the **NewServerCertDomainName** setting is deleted.
 - _ DELETE – cancellation of the certificate change procedure.

Value by default: "" (empty).

5.5 CryptoPro configuration

Make sure that CryptoPro software is installed on the computers where the QUIK server and the server part of the QUIK Administrator are located.

To validate the digital signature on the computers where the QUIK server and the server part of the QUIK Administrator are located, all necessary root certificates of CA must be installed. The files with these certificates are loaded from external sources and can be in the .p7b, .pfx, .pem, .cer formats.

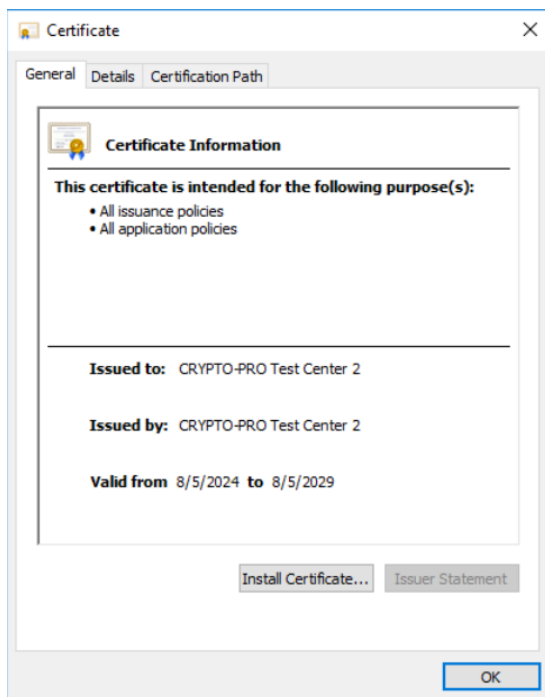
5.5.1 CA certificate installation to QUIK server when operating on Windows

When working in Windows, the CA certificates must be installed in the Trusted Root Certification Authorities such system record, under which the QUIK server software service or server part of the QUIK Administrator operates. Usually, specific user account is used, for which certificates need to be installed, working via this user account. If the software works via system account, certificates are installed in the local computer storage.

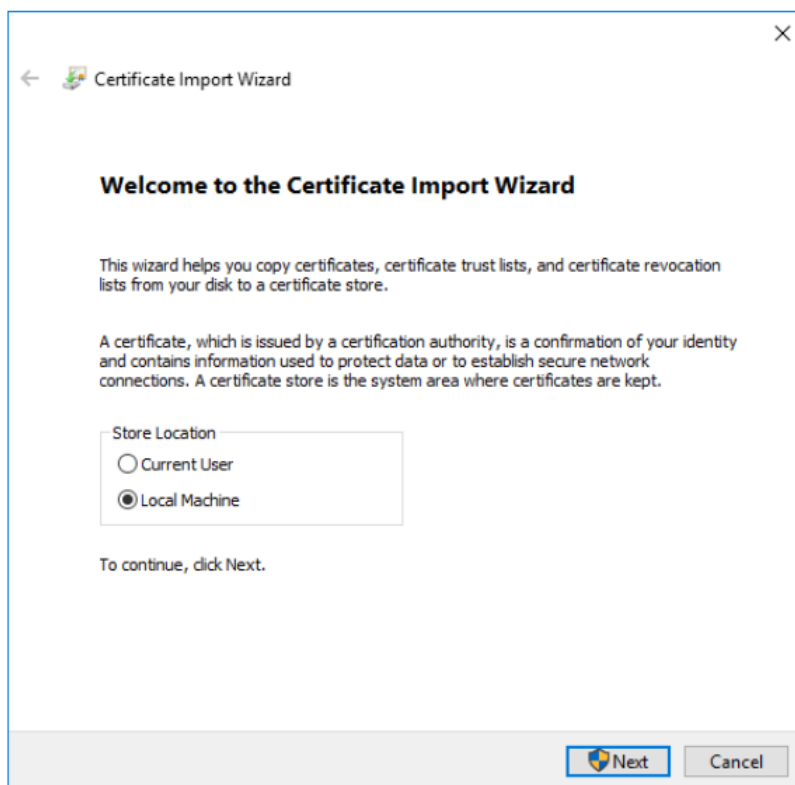
CA certificate installation to Local Machine / Trusted Root Certification Authorities storage

1. Open received file with CA certificate. Press the **Install Certificate...** button.

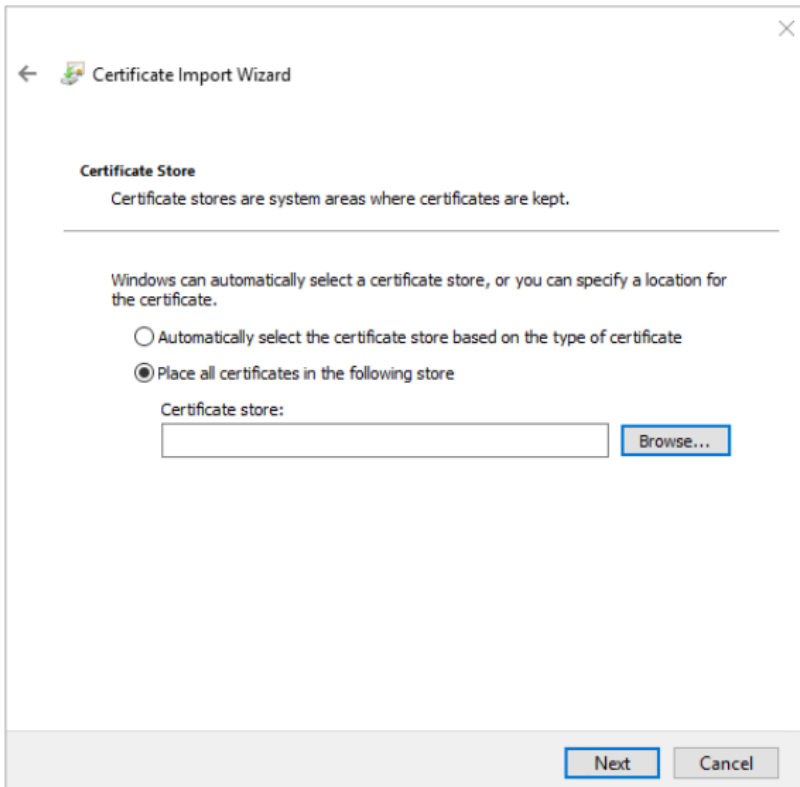




2. In the opened **Certificate Import Wizard** window, enable the **Local Machine** checkbox and click **Next**.



3. In the opened **Certificate store** window, follow the steps:



- _ Enable the **Place all certificate in the following store** checkbox;
- _ Select the **Trusted Root Certification Authorities** storage;
- _ Click **Next** button.

Verification CA certificate installation in the Local Machine / Trusted Root Certification Authorities storage

1. Click the **Start** menu.
2. Type "certlm.msc".
3. Select offered program;
4. Make sure that CA certificate is installed in the Trusted Root Certification Authorities storage in the opened Certificates window.

5.5.2 CA certificate installation to QUIK server when operating on Linux

The installation is performed if there is a file containing the root certificate received from the CA.

Certificate installation by arqasrv user

When working on Linux, the user with a specific name is used. QUIK software is installed on behalf of the arqasrv user.



1. Copy the file with the root certificate received from CA, for example, to the /opt/arqasrv/quik/quik/CryptoPro folder (where the QUIK server is installed). For example: **certnew.p7b**.
2. Run the command in the catalog with **certnew.p7b**:

```
sudo chmod 777 ./certnew.p7b
```

3. Verify the certnew.p7b certificate file owner by running the command:

```
ls -l ./certnew.p7b
```

If the certnew.p7b file owner is different from the arqasrv user, then run the following command:

```
sudo chown -R arqasrv: *
```

If the system works under the user other than arqasrv (for example, arqa), then to change the user to arqasrv, run the following commands:

- _ Configure the user permissions by running the command:

```
sudo su -  
[sudo] password for arqa: (type root password for arqa user)
```

- _ Change user to arqasrv by running the command:

```
su - arqasrv
```

4. Install the certificates:

- _ Go to the place of the certnew.p7b storage location where it was saved (/opt/arqasrv/quik/quik/CryptoPro);
- _ Install the root certificates by running the command (there can be several certificates in the PKCS7 container):

```
/opt/cprocsp/bin/amd64/certmgr -install -store root -file certnew.p7b
```

Where:

- _ /opt/cprocsp/bin/amd64/certmgr – certificate manager;
- _ -install – certificate installation command;



- _ -store root – Trusted Root Certification Authorities storage;
- _ -file certnew.p7b – name of the container file containing root certificates.

For example:

- _ As a result of running the command, the list of certificates located in the specified container will be displayed. Number 1 is the CA certificate, which must be placed in the **Trusted Root Certification Authorities**.

```
1-----
Publisher           : E=support@cryptopro.ru, C=RU, L=Moscow, O=CRYPTO-PRO
LLC, CN=CRYPTO-PRO Test Center 2
Subject            : E=support@cryptopro.ru, C=RU, L=Moscow, O=CRYPTO-PRO
LLC, CN=CRYPTO-PRO Test Center 2
Serial number      : 0x3AF80A1332834D994C4E6F0AD07C8FAD
SHA1 fingerprint   : 41dcbff2ae102339bcedd0d5100ffb8e6018353d
Key identifier     : e4c08096e4b47834123c22b975cd96204330b023
Signature algorithm : government standard P 34.11-2012/34.10-2012 256 bit
Opened key algorithm : government standard P 34.10-2012 256 bit (512 bit)
Valid from         : 04/10/2024 12:22:02 UTC
Valid to           : 04/01/2025 12:32:02 UTC
Reference to key    : No
```

- _ Press **1** and **Enter** buttons.
- _ For CA certificate containers with a password, for example, in the <PKCS12> format, the following installation command is used:

```
«/opt/cprocsp/bin/amd64/certmgr -store root -file ./test_user_cryptopro.pfx -pfx
-pin 12345»
```

Where:

- _ -pin – .pfx container password.

Verification CA certificate installation for arqasrv user

All CryptoPro certificates are stored in the specified path:

```
/var/opt/cprocsp/users
```

The arqasrv certificates are stored in the specified path:

```
/var/opt/cprocsp/users/arqasrv/stores
```



Check the certificate owner by running the following command:

```
ls -l
```

Check the installed root certificates by running the following command:

- On Windows:

```
Certutil -store root
```

- On Linux:

```
/opt/cprocsp/bin/amd64/certmgr -list -store root
```

5.5.3 **crypto.cfg** configuration file on the QUIK server

- Add the following settings to the **crypto.cfg** file on Windows:

```
[CryptoProviders]
cryptopro=1

[cryptopro]
Module=.\cpcsp_pr.dll
IniFile=.\cpcsp_pr.ini
```

- Add the following settings to the **crypto.cfg** file on Linux:

```
[CryptoProviders]
cryptopro=1

[cryptopro]
Module=./libcpcsp_pr.so
IniFile=./cpcsp_pr.ini
```

5.5.4 **qadminsrv.ini** configuration file of the QUIK Administrator server part

- Add the following settings to the **qadminsrv.ini** file of the QUIK Administrator server part on Windows:



```
[CryptoProviders]
cryptopro=1

[cryptopro]
Module=.\cpcsp_pr.dll
IniFile=.\cpcsp_pr.ini
```

- Add the following settings to the **qadminsrv.ini** file of the QUIK Administrator server part on Linux:

```
[CryptoProviders]
cryptopro=1

[cryptopro]
Module=../libcpcsp_pr.so
IniFile=../cpcsp_pr.ini
```

5.5.5 cpcsp_pr.ini configuration file of the QUIK server

Specify the following values in the [Certificate Stores] section:

```
[Certificate Stores]
CA=1
ROOT=1
CUSTOM_STORE=1
```

Where:

- CA – intermediate certificates of CA. The path to this storage is specified in the [CA] section of the cpcsp_pr.ini file:

```
[CA]
HOST=.
LOCATION=CA
TYPE=1
UPDATABLE = 1
PRIORITY = 1
```

- ROOT – root certificates of CA. The path to this storage is specified in the [ROOT] section of the cpcsp_pr.ini file:



```
[ROOT]
HOST=.
LOCATION=ROOT
TYPE=1
UPDATABLE = 1
PRIORITY = 1
```

- **CUSTOM_STORE** – local storage of the user certificates as a file. The path to this storage is specified in the [CUSTOM_STORE] section of the cpcsp_pr.ini file:

```
[CUSTOM_STORE]
HOST=.
LOCATION=./userstore.sto
TYPE=2
UPDATABLE = 1
PRIORITY = 1
```

The userstore.sto file must be created in advance, and certificates of users who signed transactions to the QUIK server are added to it using the QUIK Administrator. The owner must be the arqasrv user

5.5.6 cpcsp_pr.ini configuration file of the QUIK Administrator server part

Specify the following values in the [Certificate Stores] section:

```
[Certificate Stores]
CA=1
ROOT=1
CUSTOM_STORE=1
```

Where:

- **CA** – intermediate certificates of CA. The path to this storage is specified in the [CA] section of the cpcsp_pr.ini file:

```
[CA]
HOST=.
LOCATION=CA
TYPE=1
UPDATABLE = 1
PRIORITY = 1
```



- ROOT – root certificates of CA. The path to this storage is specified in the [ROOT] section of the cpcsp_pr.ini file:

```
[ROOT]
HOST=.
LOCATION=ROOT
TYPE=1
UPDATABLE = 1
PRIORITY = 1
```

- CUSTOM_STORE – local storage of the user certificates as a file. The path to this storage is specified in the [CUSTOM_STORE] section of the cpcsp_pr.ini file:

```
[CUSTOM_STORE]
HOST=.
LOCATION=./userstore.sto
TYPE=2
UPDATABLE = 1
PRIORITY = 1
```

userstore.sto certificate storage for the QUIK server and server part of the QUIK Administrator is general.

5.5.7 Adding the certificate to certificate storage of the QUIK server

The connection configuration file is specified in the qadminsrv.ini configuration file in the [instances] section, for example:

```
16300=SERV_03.ini
```

1. Set the values in the SERV_03.ini file:

```
[QuikCryptoProvider]
CryptoType=CryptoPro
Storage=./opt/arqasrv/quik/quik/userstore.sto
```

Where:

- userstore.sto – certificate storage. This file must be created (empty) in the location specified in the Storage parameter.



If the QUIK Administrator is installed on Linux, then file is created by the arqasrv user.

2. Open the **QUIK Server / Security / Certificates storage** menu item in the QUIK Administrator.
3. Select the **Certificates file** and specify the path to the saved User certificate and click **Open**. The information about the selected user certificate will be displayed.

[QUIK server] Certificates storage

QUIK certificates storage content

Filter

Certificate number	Certification Authority	Certificate owner	Signature algorithm	Cr
1200 65E5 9666 70C/CN=CRYPTO-PRO Test Cer/CN= Test/O=Test/Email=Te			1.2.643.7.1.1.3.2	1.2.

Certificates files

Certificates file

Certificate number	Certification Authority	Certificate owner	Signature algorithm	Cr
1200 65E5 9666 70C/CN=CRYPTO-PRO Test Cer/CN= Test/O=Test/Email=Te			1.2.643.7.1.1.3.2	1.2.

4. Click **Add certificate**.

5. The following steps should be performed on the **Edit (Users)** tab in the QUIK Administrator:

Common rights Security Trade roles Classes permissions Operations

Last login information
ATTENTION! Login failed. Login at 23.10.2024 10:11:21 from 192.168.37.167.
Error (1001): User entered incorrect two-factor authentication PIN (password) and was disconnected.

Allowed IP Addresses

Authentication by login and password
Custom... Login: 13
Access to applications:
☒ webQUIK
☒ QUIK
Temporary blocking: no
Unlock blocking

Digital signature CryptoPro
☒ Signature required

Digital signature keys

Certificate number	Assign date	Revoked
1200 65E5 9666 70C0 B3DA 456C		No

QUIK keys

Key identifier	Assign date	Revoked
85DF784F9A26C53D	8/1/2024	No

- Select the required user;
- Select the **CryptoPro** in the **Digital signature** parameter;
- Enable the **Signature required** checkbox;
- Click the **Edit...** in the **Digital signature keys**. Click **Add** in the opened window.

6. Select user certificate and click **Open**.

7. Enable the **Use to sign transactions** and **Use to encrypt data stream** in the **Electronic signature keys** window. Click **OK**.

Electronic signature keys

Serial	Certificate number	Certification Authority	Certificate owner
1	7C00 16EB AAE1 72/CN=Тестовый УЦ ООО 'К/ CN=Test/O=Test/Email=Te		

Add Edit Delete

Key properties
☒ Use to sign transactions
☒ Use to encrypt data stream
☐ Disabled

OK Cancel



8. Click **Save** to save changes.
9. Check the entry of the certificate number into the database (crypto_pro_key table).

5.5.8 Verification of the initialization of the CryptoPro crypto provider

1. Start the QUIK server.
2. Open the **events.log**.

If the service is successfully installed, the diagnostic will be displayed: "Sign provider 'CryptoPro' initialized successfully".

5.5.9 Server part migration from Windows OS to Linux OS

1. Install the CryptoPro software on the Linux server and configure the QUIK server on the CryptoPro digital signature according to the instruction of the digital signature installation.
2. Transfer the sto file with the QUIK server user certificate storage under Windows to the QUIK server under Linux. To do this, copy this file to the required storage location in Linux and specify this path in the crypto provider configuration ini file ([CUSTOM_STORE] section, LOCATION parameter).

The sto file owner must be argasrv user.

3. Configure the QUIK server on Linux to work with the same DB that is used on the QUIK server on Windows.

If the database running Microsoft SQL Server DBMS is used, it is required first migrate data from Microsoft SQL Server to PostgreSQL.

If the server installed on Windows is configured for digital signature via the MP crypto provider (the crypto.cfg file will contain a link to the mp_pr.dll file instead of cpcsp_pr.dll), then it is additionally required to modify the data in the DB, changing the cspsspi line to cryptopro in the dbo.users.crypto_type and dbo.crypto_pro_key tables. To do this, perform the following requests to DB:

```
update dbo.USERS set CRYPTO_TYPE='cryptopro' where CRYPTO_TYPE='cspsspi'
update dbo.CRYPTO_PRO_KEY set CRYPTO_TYPE='cryptopro' where CRYPTO_TYPE='cspsspi'
```

5.6 Signal-COM configuration

The **SSLAndMessagePro** crypto provider provides the opportunity to work with two cryptographic libraries: **Message-Pro** and **SSL-Pro**.



5.6.1 Encryption configuration for the main QUIK server for Windows

1. Check availability and, if it is necessary, copy files that are necessary for program operation in the work catalog of the main QUIK server:
 - _ sslpro.dll – **SSL-Pro** library, provided by company that created the Signal-COM;
 - _ signalcom_pr.dll – unified provider module of the **SSL-Pro** and **Message-Pro** libraries;
 - _ signalcom_pr.ini – file with general provider settings, connection parameters of the user and digital signature settings.
2. For encryption, the server will need set of the key files and certificates. Before starting, it is required to place the key files and certificates in the appropriate subdirectory of the directory of the crypto key device. For the description of the recommend format of the crypto key device, see [Description of the recommend format of the crypto key device for SSL-Pro and Message-Pro libraries](#).

The system operation using encryption by SSL-Pro library requires availability of a secret key for each party (QUIK server and QUIK Workstations), corresponding certificate, and also necessary set of the certificates of certification authorities. In current versions, the server key must not be password protected, otherwise if you try to request a password the server will hang. Keys and certificates are generated using software that is not included in the standard QUIK system delivery package. The question of its acquisition is resolved with company that created Signal-COM.

3. Specify the necessary encryption settings in the **signalcom_pr.ini** file of the main QUIK server corresponding with structure description of the **signalcom_pr.ini** file (see [5.6.7](#)). In this case, the **signalcom_pr.ini** file will contain common crypto provider settings and settings that will be used only for encryption.
4. Specify the following provider settings in the **qcrypto.cfg** file:
 - _ Add the **SignalComMessageAndSSLPro=1** setting in the [CryptoProviders] section:

```
[CryptoProviders]
...
SignalComMessageAndSSLPro=1
```

- _ Add the [SignalComMessageAndSSLPro] section of the following view to the file:

```
[SignalComMessageAndSSLPro]
Module=.\signalcom_pr.dll ; full or relative path to executable file
IniFile= .\signalcom_pr.ini ; full or relative path to setting file
```

5. Specify path to the previously configured **signalcom_pr.ini** file with encryption settings by SSL-Pro library that will be used by the server when the users will be connected. To do this,



create the **crypto_provider_ini_file=.\signalcom_pr.ini** setting in the [usend_module] section of the server **quik.ini** configuration file:

```
[usend_module]
...
crypto_provider_ini_file=.\signalcom_pr.ini
```

If the main QUIK server uses the only one encryption crypto provider, then to specify its settings, you can use crypto_provider_ini_file setting in the [usend_module] section.

If there are several providers, then, to specify settings, you must use the following settings:

- _ crypto_provider_ini_file_ka – key to specify the KA crypto provider;
- _ crypto_provider_ini_file_mp – key to specify the MP crypto provider;
- _ crypto_provider_ini_file_sslpro – key to specify the SSL-Pro crypto provider;
- _ crypto_provider_ini_file_openssl – key to specify the OpenSSL crypto provider.

6. Run the main QUIK server and make sure the following line is present in the **events.log** file:

```
Crypto provider 'SignalComMessageAndSSLPro' initied
```

5.6.2 Digital signature configuration for main QUIK server for Windows

1. Check availability and, if it is necessary, copy files that are necessary for program operation to the work catalog of the main QUIK server:

- _ messpro.dll – Message-Pro library, provided by company that created the Signal-COM;
- _ signalcom_pr.dll – unified provider module of the **SSL-Pro** and **Message-Pro** libraries;
- _ signalcom_pr.ini – file with general provider settings, connection parameters of the user and digital signature settings.

2. Locate files with user certificates that will send transactions with digital signature in the appropriate subdirectory of the directory of the crypto key device. For the description of the recommend format of the crypto key device, see [Description of the recommend format of the crypto key device for SSL-Pro and Message-Pro libraries](#).

The system operation using digital signature by Message-Pro library requires availability of secret key of the QUIK Workstations, corresponding certificate, and the necessary set of the certificates of certification authorities. The server side requires a directory of client certificates and the required set of certificates of certification authorities. To generate keys and certificates, software is used that is not included in the standard QUIK system delivery package. The question of its acquisition is resolved with company that created Signal-COM. Support of the server certificate dictionary in the current view



(removal of revoked and expired certificates) is also resolved by not included software.

3. Specify the necessary digital signature settings in the **signalcom_pr.ini** file of the main QUIK server corresponding to structure description of the **signalcom_pr.ini** file (see [5.6.7](#)). In this case, the **signalcom_pr.ini** file will contain common crypto provider settings and settings that will be used only for digital signature.
4. Specify the following provider settings in the **qcrypto.cfg** file:

- _ Add the **SignalComMessageAndSSLPro=1** setting in the [CryptoProviders] section:

```
[CryptoProviders]
...
SignalComMessageAndSSLPro=1
```

- _ Add the [SignalComMessageAndSSLPro] section of the following view to the file:

```
[SignalComMessageAndSSLPro]
Module=.\signalcom_pr.dll ; full or relative path to executable file
IniFile= .\signalcom_pr.ini ; full or relative path to setting file
```

- _ Specify the following settings in the [auth] section, in case of missing:

```
[auth]
...
sign=secprov.dll
```

5. Run the main QUIK server and make sure the following line is present in the **events.log** file:

```
Sign provider 'SignalComMessageAndSSLPro' initialized successfully
```

5.6.3 Encryption configuration for the main QUIK server for Astra Linux

1. Check availability and, if it is necessary, copy files that are necessary for program operation to the work catalog of the main QUIK server:
 - _ libsslprox.so.0 – **SSL-Pro** library, provided by company that created the Signal-COM;
 - _ libsignalcom_pr.so – unified provider module of the **SSL-Pro** and **Message-Pro** libraries;
 - _ signalcom_pr.ini – file with general provider settings, connection parameters of the user and digital signature settings.



2. For encryption, the server will need set of the key files and certificates. Before starting, it is required to place the key files and certificates in the appropriate subdirectory of the directory of the crypto key device. For the description of the recommend format of the crypto key device, see [Description of the recommend format of the crypto key device for SSL-Pro and Message-Pro libraries](#).

The system operation using encryption by SSL-Pro library requires availability of a secret key for each party (QUIK server and QUIK Workstations), corresponding certificate, and also necessary set of the certificates of certification authorities. In current versions, the server key must not be password protected, otherwise if you try to request a password the server will hang. Keys and certificates are generated using software that is not included in the standard QUIK system delivery package. The question of its acquisition is resolved with company that created Signal-COM.

3. Specify the necessary encryption settings in the **signalcom_pr.ini** file of the main QUIK server corresponding with structure description of the **signalcom_pr.ini** file (see [5.6.7](#)). In this case, the **signalcom_pr.ini** file will contain common crypto provider settings and settings that will be used only for encryption.
4. Specify the following provider settings in the **crypto.cfg** file:

- _ Add the **SignalComMessageAndSSLPro=1** setting in the [CryptoProviders] section:

```
[CryptoProviders]
...
SignalComMessageAndSSLPro=1
```

- _ Add the [SignalComMessageAndSSLPro] section of the following view to the file:

```
[SignalComMessageAndSSLPro]
Module=../libsignalcom_pr.so ; full or relative path to executable file
IniFile= ../signalcom_pr.ini ; full or relative path to setting file
```

5. Specify path to the previously configured **signalcom_pr.ini** file with encryption settings by SSL-Pro library that will be used by the server when the users will be connected. To do this, create the **crypto_provider_ini_file=../signalcom_pr.ini** setting in the [usend_module] section of the server **quik.ini** configuration file:

```
[usend_module]
...
crypto_provider_ini_file=../signalcom_pr.ini
```



If the main QUIK server uses the only one encryption crypto provider, then to specify its settings, you can use `crypto_provider_ini_file` setting in the `[usend_module]` section.

If there are several providers, then, to specify settings, you must use the following settings:

- _ `crypto_provider_ini_file_sslpro` – key to specify the SSL-Pro crypto provider;
- _ `crypto_provider_ini_file_openssl` – key to specify the OpenSSL crypto provider.

6. Run the main QUIK server and make sure the following line is present in the **events.log** file:

```
Crypto provider 'SignalComMessageAndSSLPro' initied
```

5.6.4 Digital signature configuration for main QUIK server for Astra Linux

1. Check availability and, if it is necessary, copy files that are necessary for program operation to the work catalog of the main QUIK server:

- _ `libmesprox.so.0` – Message-Pro library, provided by company that created the Signal-COM;
- _ `libsignalcom_pr.so` – unified provider module of the **SSL-Pro** and **Message-Pro** libraries;
- _ `signalcom_pr.ini` – file with general provider settings, connection parameters of the user and digital signature settings.

2. Locate files with user certificates that will send transactions with digital signature in the appropriate subdirectory of the directory of the crypto key device. For the description of the recommend format of the crypto key device, see [Description of the recommend format of the crypto key device for SSL-Pro and Message-Pro libraries](#).

The system operation using digital signature by Message-Pro library requires availability of a secret key of the QUIK Workstations, corresponding certificate, and the necessary set of the certificates of certification authorities. The server side requires a directory of client certificates and the required set of certificates of certification authorities. To generate keys and certificates, software is used that is not included in the standard QUIK system delivery package. The question of its acquisition is resolved with company that created Signal-COM. Support of the server certificate dictionary in the current view (removal of revoked and expired certificates) is also resolved by not included software.

3. Specify the necessary digital signature settings in the **signalcom_pr.ini** file of the main QUIK server corresponding to structure description of the **signalcom_pr.ini** file (see [5.6.7](#)). In this case, the **signalcom_pr.ini** file will contain common crypto provider settings and settings that will be used only for digital signature.

4. Specify the following provider settings in the **qcrypto.cfg** file:



- Add the **SignalComMessageAndSSLPro=1** setting in the [CryptoProviders] section:

```
[CryptoProviders]
...
SignalComMessageAndSSLPro=1
```

- Add the [SignalComMessageAndSSLPro] section of the following view to the file:

```
[SignalComMessageAndSSLPro]
Module=libsignalcom_pr.so    ; full or relative path to executable file
IniFile= ./signalcom_pr.ini  ; full or relative path to setting file
```

- Specify the following settings in the [auth] section, in case of missing:

```
[auth]
...
sign=libsecprov.so
```

5. Run the main QUIK server and make sure the following line is present in the **events.log** file:

```
Sign provider 'SignalComMessageAndSSLPro' initialized successfully
```

5.6.5 Encryption configuration of the QUIK Access server

The encryption setting for communication channel with clients is work as main QUIK server setting (see settings for [Windows](#) or [Astra Linux](#)) by 1-5 steps.

The Qrypto32 library requires for communication channel of the Access server higher-level servers or main QUIK server in the standard configuration, therefore it is not required to set up the encryption using the SSL-PRO library to protect communication channel between the Access server and servers higher in the hierarchy or main QUIK server.

1. **The SSL-Pro library requires more computing resources than Qrypto32, therefore performance level adequacy of the system must be checked before complex run in the real operation.**
2. **During the operation of the SSL-Pro library on the Access Server it is possible to use the key and certificate used by the servers higher in the hierarchy – this reduces security but simplifies the key administration procedure.**



5.6.6 Digital signature for QUIK Access server

In a typical configuration, the digital signature settings for the Access server are not required, since digital signature generation is executed by the QUIK Workstation of clients when sending transactions, and digital signature verification is executed by the main QUIK server. Setting up digital signature for the Access server is required only when clients connecting via the Access server do not use digital signature (transactions are received by the Access server without digital signature), and there is a requirement that all user transactions received by the main server must be with digital signature and must be verified by the main QUIK server. In this case, the Access server is configured so that all incoming transactions of users are signed with the Access server key. In this case:

1. If the Access server with configured digital signature generation receives a client transaction already signed with digital signature made by the same software configured for digital signature on the Access server, this transaction is sent to main QUIK server without the Access server digital signature.
2. If the crypto provider used for digital signature of the client transaction and the crypto provider configured to create the digital signature on the Access server are different, then the transaction is signed using the Access server digital signature and sent to the main QUIK server.

Setting up the digital signature on the Access server is similar to setting up main QUIK server setting (see settings for [Windows](#) or [Astra Linux](#)) by 1-5 steps. Additionally, execute the following:

1. Add [Signal_Com_Message_Pro_Sign_Settings] section in the **signalcom_pr.ini** configuration file (see [5.6.7](#)) and specify the following settings:

- _ sign_cert_file_name – path and name of the file with user certificate by which digital signature is executed;
- _ pse_path – path to directory of the root key device.

For example:

```
[Signal_Com_Message_Pro_Sign_Settings]
sign_cert_file_name=./certs/pse/cert/cert.cer
pse_path=./certs/pse
```

2. Add the **sign_provider** setting in the [dwell_module] section in the **quik.ini** configuration file of the Access server:

- _ sign_provider – crypto provider type used when digital signature is generating. Value by default: **SignalComMessageAndSSLPro** (digital signature generation by Message-Pro library).



For example:

```
[dwell_module]
...
sign_provider=SignalComMessageAndSSLPro
```

3. Run the Access server and make sure the following line is present in the **events.log** file:

```
Sign provider 'SignalComMessageAndSSLPro' initialized successfully
```

5.6.7 signalcom_pr.ini file format

Description of the encryption settings

- [SIGNAL_COM_SSL_PRO_COMMON_SETTINGS] section contains a description of the global settings of the crypto provider.
- [SIGNAL_COM_SSL_PRO_CIPHER_SETTINGS] section contains a description of the crypto provider settings required for correct creation of the encryption context:
 - PSE_PATH – path to the root directory of the key device of the Cryptographic information protection facilities (CIPF). Required parameter.
 - TRUSTED_CA_CERTS_DIR – path to the directory of the certificates of certification authorities. Required parameter, if path to file with certificate of the root certification authorities is not specified in the TRUSTED_CA_CERT_FILE_NAME setting.
 - TRUSTED_CA_CERT_FILE_NAME – path to the file with the certificates of certification authorities. Required parameter, if path to file with certificate of the root certification authorities is not specified in the TRUSTED_CA_CERTS_DIR setting.
 - UNTRUSTED_CA_CERTS_DIR – path to the directory of the certificates of the intermediate certification authority certificates. Optional parameter.
 - UNTRUSTED_CA_CERT_FILE_NAME – path to the file with certificate of the intermediate certification authority certificates. Optional parameter.
 - CRL_FILES_DIR – path to the directory with list of revoked certificates. Optional parameter.
 - CRL_FILE_NAME – path to the file with list of revoked certificates. Optional parameter.
 - CIPHER_CERT_FILE_NAME – name of the file with certificate, corresponding of the secret key, on which data is encrypted during the communication session. Required parameter.
 - CIPHER_KEY_FILE_NAME – file name with secret key on which the data is encrypted during the communication session. Optional parameter.
- [SignalComCommonSettings] section:
 - DefaultNewKeyAlgorithm – algorithm of the data encryption by the asymmetric key. Value by default: ECR3410.

For key generation in accordance with the state standard as value of the DefaultNewKeyAlgorithm setting, specify GOST3410-2012-512-A-ParamSet.



Description of digital signature verification settings

The [signal_com_message_pro_common_settings] section contains the description of the crypto provider settings required to verify digital signature:

- certificates_path – path to the directory of the used certificates. Required parameter.
- CRLs_path – path to the directory of used lists of revoked certificates. Required parameter.
- CAs_path – path to the directory of the certificate of the certification authorities. Required parameter.
- input_data_format – input data format. In the typical configuration, it is not configured by user (use value by default). Valid values:
 - _ FORMAT_ASN1;
 - _ FORMAT_PEM (by default).
- output_data_format – output data format. In the typical configuration, it is not configured by user (use value by default). Valid values:
 - _ FORMAT_ASN1;
 - _ FORMAT_PEM (by default).
- cert_storage_validation – verification downloaded certificates for relevance. When the setting is enabled, expired certificates are not downloaded. Valid values:
 - _ 0 – disabled;
 - _ 1 (by default) – enabled.
- cert_operation_approach – adding certificates to the intermediate storage for further verification for the presence of certificates in the mespro.dll library. Allows you to significantly reduce the time of searching for the required certificate and digital signature verification. Valid values:
 - _ 0 – disabled;
 - _ 1 (by default) – enabled.

Description of the recommended format of the key device for SSL-Pro and Message-Pro libraries

- SignalCom – root directory, where all necessary files (keys, certificates, lists of revoked certificates and more) are placed.
- PSE – subdirectory of the **SignalCom** directory, where keys are placed.
- CERTS – subdirectory of the **SignalCom** directory, where files of the user certificates are placed. This directory name and location can be changed by also making appropriate changes to the **Signalcom_Pr.ini** provider configuration file.
- KEYS – subdirectory of the **PSE** directory, where files of the user encryption keys are placed. The name and location of this directory relative to the root **PSE** catalog cannot be changed.
- TrustedCA – subdirectory of the **SignalCom** directory, where files of the certificates of the root certification authorities are placed. This catalog name and location can be changed by also making appropriate changes to the provider configuration file.



- CA – subdirectory of the **SignalCom** directory, where files of the certificates of the root certification authorities are placed. The location of this catalog is generally the same as the location of the **TrustedCA** catalog.
- CRLs – subdirectory of the **SignalCom** directory, where files of the lists of revoked certificates are placed. This catalog name and location can be changed by also making appropriate changes to the provider configuration file.

When using the recommended format of the key device, the **signalcom_pr.ini** crypto provider configuration file has the following format:

```
[SIGNAL_COM_SSL_PRO_COMMON_SETTINGS]

[SIGNAL_COM_SSL_PRO_CIPHER_SETTINGS]
PSE_PATH=./signalcom/pse
TRUSTED_CA_CERTS_DIR=./signalcom/trustedca
CRL_FILES_DIR=./signalcom/crls
CIPHER_CERT_FILE_NAME=./signalcom/certs/cert.cer

[SIGNAL_COM_MESSAGE_PRO_COMMON_SETTINGS]
CERTIFICATES_PATH=./signalcom/certs
CRLS_PATH=./signalcom/crl
CAS_PATH=./signalcom/ca
INPUT_DATA_FORMAT=FORMAT_ASN1
OUTPUT_DATA_FORMAT=FORMAT_ASN1

[SIGNAL_COM_MESSAGE_PRO_SIGN_SETTINGS] ; section is present when setting up digital
signature via Access server
SIGN_CERT_FILE_NAME=./signalcom/certs/cert.cer
PSE_PATH=./signalcom/pse
```

The .opq and .db3 files must match the container with which they were generated.

5.7 Stop order rollover

Stop order rollover specifics between servers united in a circuit, including between the main and backup server. The description is given for both crashed and planned rollover.

5.7.1 Scenario for stop order rollover when one of the servers crashed

- Execute the **Process all stop orders** command on the run server via QMonitor program interface. Select server label that was crashed on the identifier request (from which it is necessary to rollover the stop orders).
- A server that has crashed must be started with the **-removestops** launch parameter (a combination with the **-clean**, **-new** and **-loadlimits** launch parameters is allowed).



It is during the first start after a server crash that subsequent normal starts are performed without the **-removestops** launch parameter.

5.7.2 Scenarios of the planned rollover

It is recommended that any planned rollover be done exclusively during non-trading hours.

Rollover execution in one step

1. Ensure that the following conditions are met:

- _ Run both servers;
- _ The Replicator is connected to these servers;
- _ All trading interfaces are connected to these servers.

2. Connect to the server to which stop orders need to be rolled over using the QMonitor program interface and execute **Process all stop orders** command (in response to the identifier request, specify the label of the server from which stop orders need to be rolled over).

When executing a one-step rollover, it is required to fulfill the conditions, otherwise there will be collisions with belonging labels.

Rollover execution in two steps

1. Connect to the server to which stop orders need to be rolled over using the QMonitor program interface and execute **Process all stop orders** command (alternatively, you can start with **-removestops** launch parameter).
2. Connect to the server to which stop orders need to be rolled over using the QMonitor program interface and execute **Process all stop orders** command (at the identifier request, specify the label of the server from which the stop orders should be rolled over).

Requirements:

- If the server on which the stop orders are rolled over was run with **-clean** or **-new** launch parameters, then the situation is possible when QMonitor program interface will not display stop orders. In this case, it is required to wait for the QUIK server to process stop orders from the database. The usual order of waiting time is from 30 seconds to several minutes. The criterion of successful completion of processing of stop orders from the database is appearance of the list in the Stop Orders table (in the QMonitor program interface). For this reason, it is not recommended to start the server to which stop orders are rolled over with the **-clean** or **-new** launch parameters.
- Connection of the Replicator and trading interfaces in the **Rollover execution in two steps** scenario is not required.
- The planned rollover must be done during non-trading hours. If you need to make the planned rollover in the trading hours, then the steps described in **Rollover execution in two steps** must be done consistently and quickly (to avoid double triggering and delays in triggering) via the QMonitor program interface.

If you have any other questions, send the events.log file and a full description of the events to QUIK technical support : quiksupport@argatech.com.

5.8 Daily maintenance of QUIK server

5.8.1 Service start and stop

If you have any questions, please contact QUIK technical support: quiksupport@argatech.com.

A QUIK server and interfaces to trading systems must be restarted daily because the trading venues are restarted daily.

If there is no schedule of service restart, the current trading session date can be determined incorrectly thus resulting in connection problems of trading interfaces. Services should be stopped after trading in corresponding venues is over, and started in the morning before trading starts.

The service start schedule must be set according to the [ClearingTime](#) setting value in section [trade_date] of the quik.ini configuration file, which must correspond to the time when a service is stopped. For example, if a QUIK server stops at 2:30 AM and starts at 5:30 AM, then the value can be set as follows:

```
[trade_date]
ClearingTime=040000
```

The **ClearingTime** setting can be also set in the settings of other services and modules of the QUIK system. For more information, see Administrator's manual of these modules.

A QUIK server can be started [with](#) or [without](#) keys.

Recommended schedule

Time	Event
Before 5:30 AM	QUIK server start (QuikServer service)
5:35 AM	Start the Cross currency rates forming module (CrossRateTW)
5:40 AM	Start the Replicator (ReplTW)
5:45 AM	Start the Alert dispatch module (SMS)
5:47 AM	Start access servers and web2QUIK modules



Time	Event
5:50 AM	Start the MOEX gateways AstsBridge or DFServer (Securities, FX, clearing system)
5:53 AM	Start the SPECTRA trading system gateway of the MOEX Derivatives market (SpectraCGate)
5:55 AM	Start the Moscow Exchange FX Market trading interface (AstsCurrGate) and Moscow Exchange Clearing System of the FX Market program interface (AstsCurrRiskGate)
5:57 AM	Start the Moscow Exchange Securities Market trading interface (AstsGate) and Moscow Exchange Clearing System of the Securities Market program interface (AstsRiskGate)
6:00 AM	Start the Moscow Exchange Derivatives Market trading interface (FortsGate)
5:45 AM	Start the SPB Exchange trading interface (SpbExchangeGate)
8:55 AM	Start the MOEX gateways AstsBridge or DFServer (money market)
9:05 AM	Start the MOEX Money market trading interface (AstsGKO)
9:05 PM	Stop the MOEX Money market trading interface (AstsGKO)
9:10 PM	Stop the MOEX gateways AstsBridge, DFServer (money market)
0:00 AM	Stop the Moscow Exchange Securities Market trading interface (AstsGate), Moscow Exchange FX Market trading interface (AstsCurrGate), Moscow Exchange Clearing System of the Securities Market program interface (AstsRiskGate) and Moscow Exchange Clearing System of the FX Market program interface (AstsCurrRiskGate)
0:00 AM	Stop the Moscow Exchange Derivatives Market trading interface (FortsGate)
0:10 AM	Stop the MOEX gateways AstsBridge, DFServer (Securities, FX, clearing system)
0:10 AM	Stop the SPECTRA trading system gateway of the MOEX Derivatives market (SpectraCGate)
00:05 AM	Stop the SPB Exchange trading interface (SpbExchangeGate)
2:25 AM	Stop the Cross currency rates forming module (CrossRateTW)
2:25 AM	Stop access servers and web2QUIK modules
2:27 AM	Stop the Alert dispatch module (SMS)
2:27 AM	Stop the Replicator (ReplTW)
2:30 AM	Stop the QUIK server (QuikServer service)



Running QUIK server without keys

When running QUIK server without keys, it loads market data and orders and trades from storage files. This information is available to all users connected to a QUIK server thus allowing to load information about trades of the previous day in the morning. The previous trading session data is cleared at the moment when the first trading interface connects to a QUIK server.

5.8.2 Client position loading

Client positions must be loaded to a QUIK server daily before the trading starts because positions are not moved to the next day, and all related information is deleted when restarting a QUIK server.

In most cases, positions are loaded using the .lim file in which each string contains information about one position of a client. This file can be obtained using one of the following ways:

- From a QUIK workstation with manager permissions after the previous trading day is over – select **Dealer / Save positions to file**.
- Using the Program of loading limits to QUIK Server [FillLimits](#).
- From the accounting system of a brokerage company or a bank (recommended).

Positions can be loaded to a QUIK server using one of the following ways:

- [Starting a QUIK server](#) with the key `-loadlimits`.
- In the Program of loading limits to QUIK Server [FillLimits](#).
- In [QMonitor](#) program.
- Manually from [a QUIK workstation](#).
- Using the Position management program interface [QuikLimit](#).

Starting QUIK server using key `-loadlimits`

This method is recommended as the fastest one.

1. Copy previously prepared .lim file to the directory available to a QUIK server.
2. Start a QUIK server with parameters **`-clean -loadlimits <path to positions file>`**. For example:

```
sc start QuikServer -clean -loadlimits C:\QUIK\limits.lim
```

The key `-new` can be used instead of the key `-clean` except when incorrect cash and instrument positions were loaded to a server thus requiring loading a correct positions file.

For details, see [3.1.13](#).



Program of loading limits to QUIK server FillLimits

The program connects to a QUIK server a user but start of the program can be automated using the files .bat or .cmd that can start it by schedule. The program must be started after trading interfaces are connected and all required instruments are loaded to a QUIK server.

QMonitor program

This method is used if the positions file was not created in time and has to be loaded after starting a QUIK server.

1. Specify the path to the positions file in the server configuration file quik.ini:

```
[ctl_module]
loadlimits_file=C:\QUIK\Server\positions.lim
```

If this setting is not set, positions will be loaded from the file qmonitor.lim that must be located in the QUIK server directory.

2. In the QMonitor program, select **Actions / Load limits from file** to load positions.

QUIK workstation

A positions file can be loaded from a QUIK workstation with manager permissions by selecting **Dealer / Save positions to file**.

The disadvantages of this method are:

- this operation is performed manually by a manager,
- positions can be loaded only after trading interfaces are connected the list of available instruments is received.

Holiday Market Data interface

The interface can be used to load static information to the QUIK server on non-trading days. For example, if clients trade in Western markets via the QUIK system, such a need arises on days that are holidays in the Russian Federation but workdays in the West. The interface allows loading the following information to a QUIK server:

- description of instruments in a separate trading mode and prices to evaluate client positions value;
- information on derivatives instruments, positions, and limits by client accounts for clients who use the Unified cash position module.

5.8.3 Backup files creation

A QUIK server does not store files in system directories and neither write settings into the Windows registry. The files necessary the server functioning are stored in the directory where the QUIK server is installed.

To ensure that you do not lose important files it is recommended that:



- the QUIK server directory be backed up before updating to a newer version,
- the following files be packed up after the trading is over:

- _ quik\qadmin\server\mainbase.ini;
- _ quik\qadmin\server\qadminsrv.ini;
- _ quik\qadmin\client\qadmin.ini;
- _ quik\server*.ini.

as well as other .ini configuration files of all installed trading and market data interfaces.

- log files of a QUIK server (events.log, deallib.log), interfaces, and modules for the last 10 days be stored. It is also advisable to save .ik files of a QUIK server.

Built-in log rotation functionality of a QUIK server and other modules can also be used.

It is also recommended to schedule backups for all used databases to run automatically after hours and initiate the reindex procedure on a weekly basis on weekends.

5.8.4 Local time synchronization

Synchronization of local time with the time server on the machines where QUIK server services are installed must be made at a time when all QUIK server services are stopped and no more than once a day.

5.8.5 Recovery after system failure

If a system failure occurs during the trading, please contact QUIK technical support: quiksupport@argatech.com.

Failure of the system software or equipment where a QUIK server is installed may result in problems saving data to file storages and failure running a QUIK server.

General order of service start and stop

1. Stop of interfaces/modules connected to a QUIK server.

The following services **must** be stopped:

- _ Asts* interfaces (for example: Moscow Exchange Securities Market (Main Market Sector) trading interface (AstsGate), Moscow Exchange FX Market trading interface (AstsCurrGate), Moscow Exchange Clearing System of the Securities Market program interface (AstsRiskGate), Moscow Exchange Clearing System of the FX Market program interface (AstsCurrRiskGate).
- _ Moscow Exchange Derivatives Market trading interface (FortsGate).
- _ Broker quotation system.
- _ Multi-broker service module (Multihub) (if included in a configuration).

The need to stop these services **depends on the conditions**:

- _ SPB Exchange trading interface (Foreign securities market) (SpbExchangeGate), SPB Exchange trading interface (Russian securities market) (SpbExchangeRuGate) – must



be stopped **only in case of a failure** in the SPB Exchange trading system due to which incorrect information was loaded to a QUIK server.

No stopping is required for the following services:

- _ Client applications.
- _ Service applications (for example, an access server, the Market Data Export module, the FIX adapter program interface).
- _ Exception: Q2Q service (see [specifics](#)).

2. Stop of a QUIK server.

3. Start of a QUIK server with the key [-clean](#) (can be combined with [-loadlimits](#)) or [-new](#).

4. Start of interfaces/modules included into a configuration and used together with one of the stopped interfaces/modules).

The recommended time interval between launch of trading interfaces is 1 to 3 minutes.

Asts* interfaces, SPB Exchange trading interface (Foreign securities market) and SPB Exchange trading interface (Russian securities market) are started **only using** the key [-clean](#).

The specifics of Moscow Exchange Derivatives Market trading interface (FortsGate) start see [below](#).

Start with keys **-new** and **-clean**

A QUIK server can be started using the key **-new** or **-clean** in the following cases:

- when updating a QUIK server (**-new** or **-clean**);
- incorrect data was loaded to a server (for example, due to an exchange failure or violation of the QUIK services start procedure):
 - _ except positions or news (**-new**);
 - _ including positions in cash and instruments (**-clean**).
- a system failure (lack of disk space, a hardware problem, sudden power outages) resulted in file storage damage:
 - _ except storages of positions or news (**-new**);
 - _ including storages of positions or news (**-clean**).

In this case, a server will not run, and the type of a corrupted storage will be saved to the events.log file of a QUIK server

For the events.log file analysis, contact QUIK technical support:
quiksupport@argatech.com.



Q2Q service stop and start

The order of stopping and starting services when using the Q2Q service differs from the [General order](#) and looks as follows:

1. Stop a master server.
2. Start a master server with the key [-new](#) or [-clean](#).
3. Stop the Q2Q service and interfaces/modules (see Step 1 of the [General order](#)).
4. Stop a slave server.
5. Start a slave server with the key [-new](#) or [-clean](#).
6. Start the Q2Q service and interfaces/modules (see Step 4 of the [General order](#)).

Start of Moscow Exchange Derivatives Market trading interface

During the day trading session clients submit orders that are moved to the evening session when the main clearing ends. Other orders are cancelled by the trading system. The Moscow Exchange Derivatives Market trading interface (FortsGate) analyses a session status and cancels orders with lifetime limited to the current session once the evening clearing starts.

If a failure occurs during the main trading session, and a QUIK server is started with the key **-new** or **-clean** after the start of the evening session on the Derivatives market (after 7 PM Moscow Time), then the Interface must be started using the key combination **-new -send_esp** to load orders and trades of the main trading session as well as the incoming positions to a QUIK server for the correct operation of the Unified cash position module.

5.9 List of QUIK server error codes

Decimal error code	Error text in the interface in English
8637216	No right to trade in this instrument
8637217	Order price should be positive
8637218	Order price <Value3> is out of set interval from <Value1> to <Value2>
8637219	You have no position for this instrument
8637220	Position for the instrument is exceeded
8637221	Error in changing position for the instrument
8637222	Incorrect name of financial instrument
8637223	Market order for the client account is forbidden
8637224	Money-position for the client is missed
8637225	No right for this trading account



Decimal error code	Error text in the interface in English
8637226	Money-position is exceeded
8637227	Error while changing money-position
8637231	Securities amount exceed tolerable one
8637232	Close period order is forbidden
8637233	Order depth should be positive
8637236	Orders for this trading account and financial instrument are prohibited
8637237	Common limit is exceeded
8637238	Operations on this client code are prohibited
8637239	Maximum limit is exceeded
8637240	Security is restricted for short operation
8637242	Minimal level of client assets
8637243	Operations are not allowed for the specified counterparty
8637244	Operations are not allowed for the specified settlement code
8637245	Initial margin of contract is undefined
8637247	The order value cannot be less than <Value1>
8637248	Order volume is out of range
8637249	Total limit for collateral securities exceeded
8637250	Low discount for the selected security must not be smaller than the initial discount by more than <Value1>
8637251	High discount for the selected security must not exceed the initial discount by more than <Value1>
8637252	The order volume <Value1> exceeds the limit in <Value2> from the average daily turnover <Value3>
8637253	Operations in <Value1> class with the specified parameters are not allowed
8637254	REPO transactions on <Value1> of the specified direction and/or REPO term are prohibited
8637255	Operations with <Value1> instrument are prohibited



Decimal error code	Error text in the interface in English
8637256	Placing a market order to sell <Value1> in <Value2> class is prohibited
8637257	Placing a market short order to sell <Value1> in <Value2> class is prohibited
8637258	Maximum order value <Value1> exceeded
8637259	An order can lead to an unacceptable long position size <Value1> in <Value2> (max. position <Value3>)
8637260	An order can lead to an unacceptable short position size <Value1> in <Value2> (max. position <Value3>)
8637261	Maximum transactions quantity <Value1> on interval <Value2> exceeded
8637262	The order quantity must not exceed <Value1>
8637263	The order quantity cannot be less than <Value1>
8637264	The order price cannot be higher than <Value1>
8637265	The order price cannot be lower than <Value1>
8637266	The corrected RCS1 <Value1> is less than 0
8637267	The corrected RCS1 is less than 0. The short position in non-liquid instrument.
8637268	Security of settle currency is not found
8637269	The operation is not allowed. CFI code is not valid.
8637270	Short position <Value1> in <Value2> less liabilities
8637271	Operation is prohibited due to a short in the option <Value1>. Only position closing is allowed.
8637272	REPO trades with a term that includes the record date for coupon payment are not allowed for the class <Value1>
8637319	The order price must be lower than <Value1>
8637320	The order price must be higher than <Value1>
8637326	The maximum size of liabilities was exceeded by <Value1> <Value2>
8637336	Operations are prohibited
8637337	Operations with <Value1> instruments are prohibited

Decimal error code	Error text in the interface in English
8637338	Operations with settlement date <Value1> or greater are prohibited
8637339	Operations with settlement date <Value1> or greater are prohibited for <Value2> instruments
8637340	You cannot enter an order for <Value1>. The instrument ISIN code is not allowed
8637341	You cannot enter an order for instrument <Value1>. Operations with <Value2> are available for unqualified investors successfully tested on complex financial product type <Value3>
8637342	The limit is exceeded for the period <Value1> by <Value2> <Value3>
8637343	Insufficient funds to deposit
8637346	Buy operations are prohibited
8637347	Sell operations are prohibited
8637348	Buy operations are prohibited for <Value1> instruments
8637349	Sell operations are prohibited for <Value1> instruments
8637350	Buy operations with settlement date <Value1> or greater are prohibited
8637351	Sell operations with settlement date <Value1> or greater are prohibited
8637352	Buy operations with settlement date <Value1> or greater are prohibited for <Value2> instruments
8637353	Sell operations with settlement date <Value1> or greater are prohibited for <Value2> instruments
8637354	The available quantity of <Value1> is not sufficient to enter an order
8637355	Error while receiving a futures limit for securities
8637356	Error while receiving a futures limit for cash
8637357	Error while setting securities limit
8637358	Error while setting cash limit