



Access server

Installation manual

Version 14.2



Contents

1. General statements	3
1.1 Principle of user authorization without Access server.....	5
1.2 Principle of user authorization via Access server	5
2. Preparing to install Access server	6
3. Access server installation and configuration	7
3.1 Installation on Windows OS	7
3.2 Installation on Linux OS	10
3.3 Access server migration from Windows OS to Linux OS	16
3.4 Removing service	23
4. Using OpenSSL encryption provider	25
4.1 Certificate receipt	25
4.2 Certificate preparation.....	26
4.3 QUIK server configuration.....	29
4.4 Access server configuration.....	32
4.5 Setting up workstation for connecting via OpenSSL.....	35
4.6 Setting up Web2QUIK connection via OpenSSL	36
4.7 Adding chain of CA certificates to web2QUIK https certificate	37
4.8 Replacing expired certificate	38

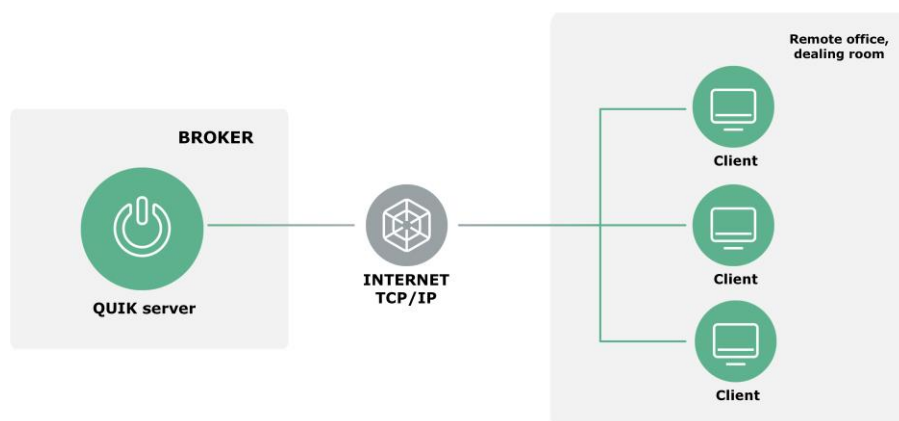
You may email your comments on this Manual to: quiksupport@argatech.com



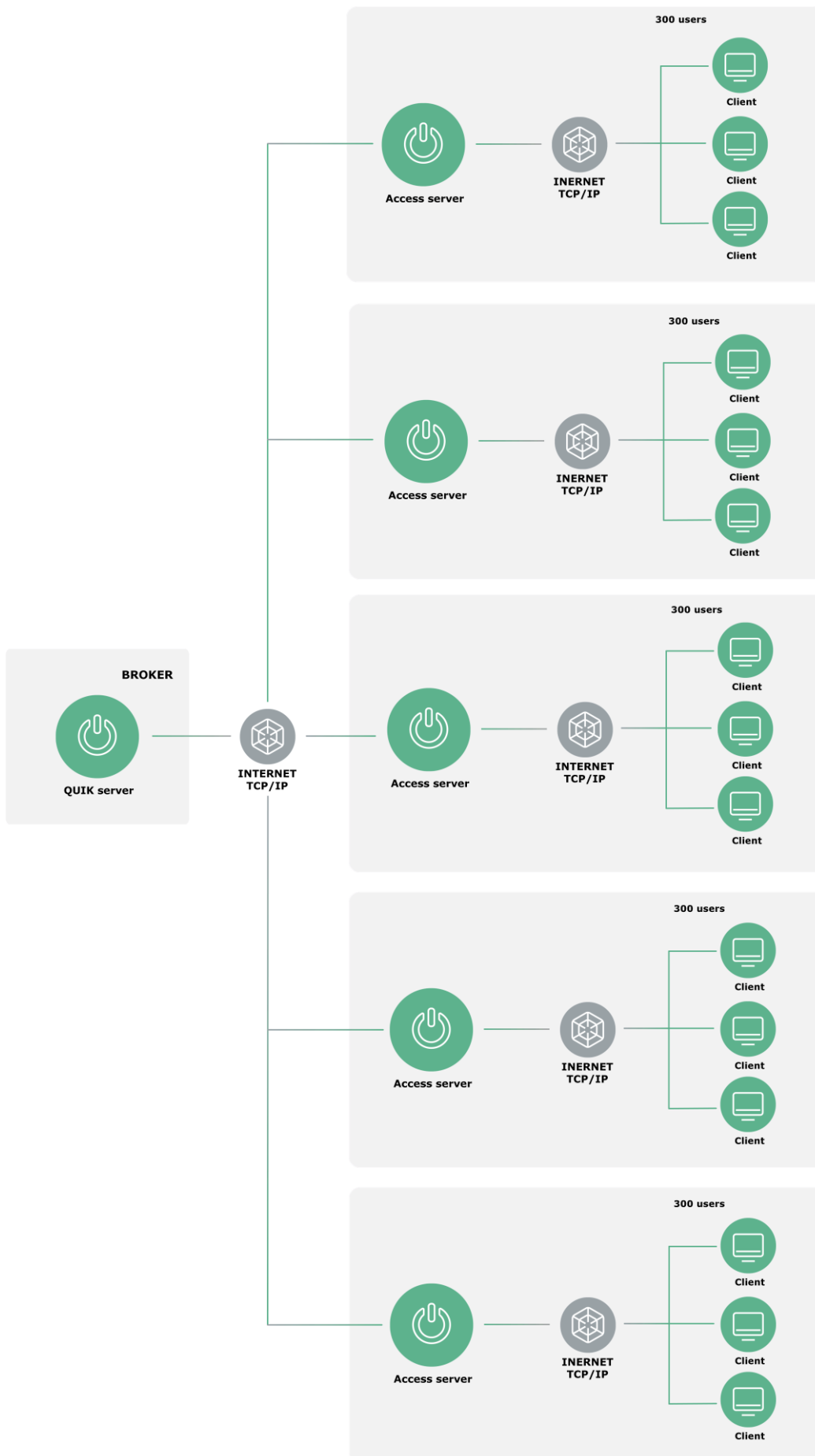
1. General statements

The access server is a separate module of the QUIK system server that serves a specific group of connected users. It is intended to reduce network traffic between the brokerage system server and a group of Internet trading users, as well as to reduce the load on the main server of the QUIK system.

The access server is recommended to use for brokers and sub-brokers have with more than 300 simultaneous client terminal connections. The main purpose of the Access server is to multiply and reduce the load on the QUIK server.



1. Without the use of an Access server traffic is proportional to the number of workstations.

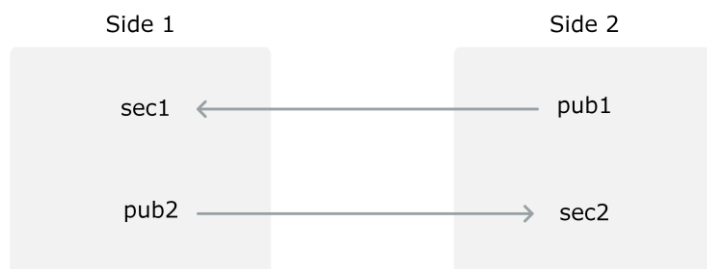


2. With an Access server the total traffic is not much higher than value for one terminal.

1.1 Principle of user authorization without Access server

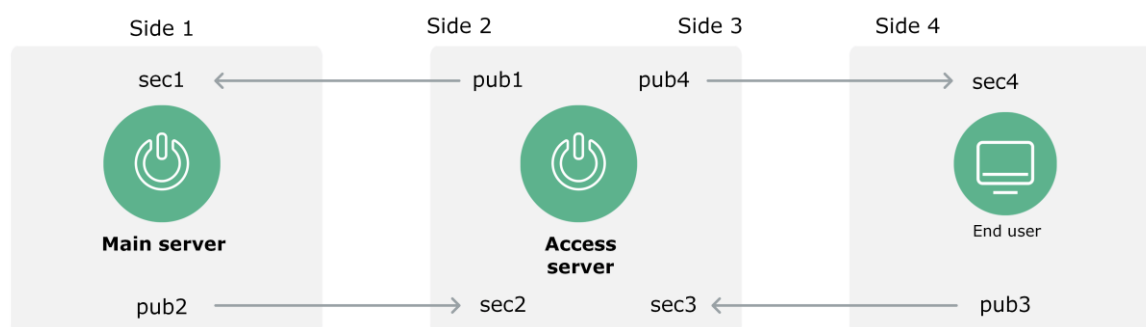
To establish a two-way secure communication using a public and secret key, two sets of keys are required (one set means the public and secret parts of one key).

Each party has a public part of the other party's key. The public part of the key is intended for data encryption.



This data can only be decrypted with the secret part of the key, which is located on the other side. Thus, in order for Side 1 to be able to accept encrypted data from Side 2, it is necessary that this data be encrypted with the public side 1 of the key, the secret part of which is held only by Side 1.

1.2 Principle of user authorization via Access server



Four sets of keys are involved in the **Main server** — **Access server** — **End user** operation:

Set of keys	Purpose
Main server key (sec1, pub1)	The main server key for connecting clients. The secret part of this key is stored on the server. The public part is distributed to all users who will access the main server, including the Access server
Access server key for main server (sec2, pub2)	Access server key for connecting to the main server. The public part of this key is registered on the main server
Access server key for clients (sec3, pub3)	Access server key for connecting clients. Instead of this key, the Main server key can be used

Client key (sec4, pub4)	The public part of the key (pub4) is stored on the main server. Clients are connecting to the Access server authorized on the main server along with the Access server
----------------------------	--

The **Access server key for main server** is the same as when connecting a regular QUIK client terminal to the QUIK server. The public part of this key must be registered in the QUIK Administrator program for the user under whose name the Access server will access the main server. In the user settings in the QUIK Administrator program, you must select the **Access server** check box.

In the QUIK Administrator program, in the **Server management** dictionary (**QUIK server/Server management** menu item), you need to add a user under whose name the Access server is connected (required to connect the QMonitor program to the access server).

2. Preparing to install Access server

When keys are generated by the KeyGen.exe program, their names must be unique.

Example of incorrect keys:

```
Key name <Access server key for main server>= "SubServer"
Key name <Access server key for clients>= "Sub"
```

The error in this example is that the names of both keys contain the "Sub" word. If clients have "Sub" written in the connection settings in the **Server key** field, the system will not be able to determine which key this connection refers to.

The scheme for connecting clients to the Access server via OpenSSL is available. For more details, see [4](#).

Access server key for client = Main server key. If you need to allow clients to connect Access server with the same public key as to the main server, then before installing the **Access server** program you must:

1. Check the presence of the following files in the root folder of the main server:

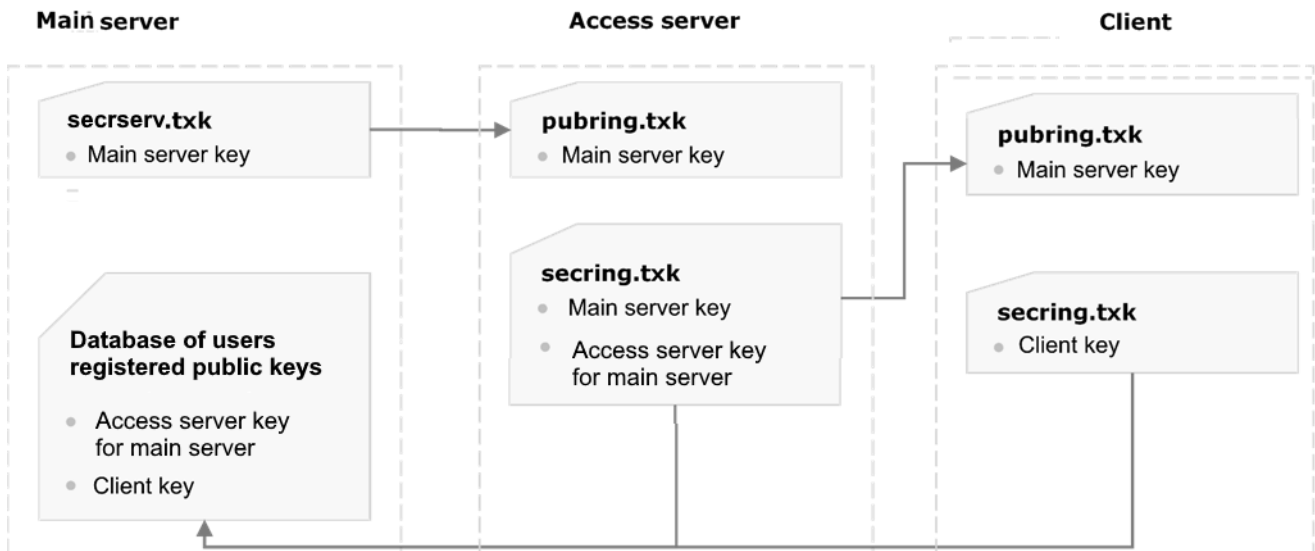
- **secrserv.txk** — file contains the secret part of the main server key and the password for this key.
- **quik.ini** — file contains the password ([usend_module] section/**password** field).

2. Create **Access server key for main server** using the key generation and management program (KeyGen).



The client's public key file must contain the same Main server key as in the Access server public key file.

Operation scheme:



3. Access server installation and configuration

3.1 Installation on Windows OS

1. Copy the secret part of the main server key (usually the `secrserv.txk` file in the main QUIK server folder) to the `secrserv.txk` file in the Access server folder (you need to copy the entire file or the key text using a text editor).
2. Generate a new key for the technological user of the Access server using the program for creating and managing keys (KeyGen). The public part of this key is used to connect clients to this Access server.
3. From the `secring.txk` file with the new key, add the secret part of the Access server technology user key to the `secrserv.txk` file. On the Access server, there should be two sections in `secrserv.txk`: "from the Access server" and "from the main server".
4. Copy `pubring.txk` of the new key to the Access server folder. Make sure that `pubring.txk` has both sections: "from the Access server" and "from the main server".
5. Check that in `crypto.cfg` file of the Access server the following parameters are specified:

```

SECRING=secrserv.txk
PUBRING=quik.dwq

[auth]
look=qrypto.cfg,db_look
look=qrypto.cfg,locke_look

[db_look]
do=_QXKey
use=dwqx.dll

[locke_look]
do=_QXKeyTX
use=dwqx.dll
locke=pubring.txk

```

6. Save the public part of the Access server key for passing to clients (the public part of the key is using for client connection to Access server).
7. Configure the quik.ini file of the Access server.

— In the [system] section, specify the service name:

```

[system]
service=QuikDwellSubServer
service-desc=QUIK dwell sub server #1

```

- In the [ctl_module] section, specify the port to connect the QMonitor program to Access server, the name of the main server and password of this key. The value of name and password parameters must match those values entered during generating the Access server key.

```

[ctl_module]
module=dwwqctl.dll
port=15121
name=broker
password=

```

- In the [usend_module] section, specify the system name, port for user connection to Access server, the name of the main server and password of this key. The value of



name and password parameters must match those values entered during generating the Access server key.

```
[usend_module]
module=dwsend.dll
sysname="broker" QUIK info trading system
name=broker
password=
port=15101
```

In the “name” field specify the name of the main server key.

- In the [dwell_module] section, specify the IP address and port for connecting the Access server to the main QUIK server, the name and password of the new key for connecting users. In the “auth_name” parameter, specify the name of the main server key.

```
[dwell_module]
module=dtwcli.dll
address=192.168.1.1
port=15100
auth_name=broker
name=brokerSD
password=
```

8. Install the Access server service by following command:

```
quik.exe -install
```

9. Configure a schedule for daily startup and shutdown of the Access server. Usually the schedule coincides with the schedule of the main server.

10. Register a technological user of the Access server on the main QUIK server (using the QUIK Administrator program). For this user, in the editing window, on the **Common rights** tab, select the **Access server** check box and on the **Classes permissions** tab, select the **Manager** rights template for all necessary classes.

11. Launch the Access server and check the connection with the QUIK Workstation.



3.2 Installation on Linux OS

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 on behalf of the arqasrv user. Installation and removing of the 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 by calling the run file.

By default, all services are installed to the **/opt/arqasrv** directory and on behalf of the **arqasrv** system user. The **<srvname>** service name for Access server matches the **quik** executable file name by default.

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

Access server is distributed for installation on computers with Astra Linux OS in the **<install_access_server-xx.xx.xx.run>** file format, where xx.xx.xx is a program version (hereinafter run file). The run file is used to install and update services in the system.

All specified parameters can be overridden by the corresponding launch parameters of the run file. All parameters are selected by default in the following information.

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_access_server-xx.xx.xx.run** file to the directory where the current user has permissions to operate in, such as the home directory (**/home/<user>** or **~/**).



2. Run the command to add execution permissions to the run file:

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

3. Install the service by running the command:

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

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

- If the **arqasrv** system user does not exist, it will be created with default permissions and included in the group with the same name. The installed service will operate on behalf of this user. The **/opt/arqasrv** directory will be set as its home directory.
- The **quik** sub-directory will be created in the **opt/arqasrv/quik** directory. This is the service installation directory into which the distribution files of the installed service will be unpacked.
- The **arqasrv** user will be assigned as the owner of the all files and sub-directories in the service directory.
- The service named **quik** will be registered in the **systemd** service administration system and run as the **arqasrv** system user.
- The file named **quik** will be created in the **/etc/arqasrv** directory to commit 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 executed by the **arqasrv** and **root** users;
 - **q_stop.sh** – stop the service. Intended to be executed by the **arqasrv** and **root** users;
 - **q_status.sh** – displaying information about service current 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@argatech.com.

The directory structure matters. Moving or renaming service files may cause installation errors.



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

```
sudo apt install libjemalloc2
```

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

5. 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 modifying to the service administration scripts for on which the user will be granted the root execution permissions, make it 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 the root by following the steps:

- Run the command:

```
sudo visudo
```

- In the opened vi-like editor, add the line:

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

This line allows the arqasrv user to execute the listed scripts on any computer in the domain as root.

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

6. Copy the secret part of the main server key (usually the secrserv.txk file in the main QUIK server folder) to the secrserv.txk file in the Access server folder (you need to copy the entire file or the key text using a text editor).



7. Generate a new key for the technological user of the Access server using the program for creating and managing keys (KeyGen). The public part of this key is used to connect clients to this Access server.
8. Add the secret part of the Access server technology user key to the secrserv.txk file from the the secring.txk file with the new key. On the Access server, there should be two sections in secrserv.txk: "from the Access server" and "from the main server".
9. Copy the pubring.txk of the new key to the Access server folder. Make sure that pubring.txk has both sections: "from the Access server" and "from the main server".
10. Check that in the qcrypto.cfg file of the Access server the following parameters are specified:

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

[auth]
look=qcrypto.cfg,db_look
look=qcrypto.cfg,locke_look

[db_look]
do=_QXKey
use=libdwqx.so

[locke_look]
do=_QXKeyTX
use=libdwqx.so
locke=pubring.txk
```

11. Save the public part of the Access server key for passing to clients (the public part of the key is used to connect clients to the Access server).
12. Configure the **quik.ini** file of the Access server.

- In the [ctl_module] section, specify the port to connect the QMonitor program to the Access server, the name of the main server and password of this key. The **name** and **password** values can be copied from the **quik.ini** file of the main server.

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



- In the [usend_module] section, specify the system name, port for user connection to the Access server, the name of the main server and password of this key. The **name** and **password** values you can copy from **quik.ini** file of the main server.

```
[usend_module]
module=libdwsend.so
sysname="broker" QUIK info trading system
name=broker
password=
port=15101
```

Name of the main server key is specified in the "name" filed.

- In the [dwell_module] section, specify the IP address and port for connecting the Access server to the main QUIK server, the name and password of the new key for connecting users. In the "auth_name" parameter, specify the name of the main server key.

```
[dwell_module]
module=libdtwcli.so
address=192.168.1.1
port=15100
auth_name=broker
name=brokerSD
password=
```

13.For the service installed by default, perform the first launch with the **-clean** launch parameter from the directory with the installed service by the following command:

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

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

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



14. Set up automatic daily start and stop of the service. To do this, use the `q_start.sh` and `q_stop.sh` command files (located in the directory of the installed service) in the automation scripts.

The Access server must be restarted daily. The start must be performed within a short period of time after the main server is started (3-5 minutes). The Access server must be stopped 3-5 minutes before the main server is stopped.

3.2.1 Launch parameters of the Linux installation run package

The run file supports the following set of launch parameters:

-i [<srvname>]

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

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 Access server – `quik`. When the `<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 only when installing the service (with the `-i`) to specify the root directory other than the default directory `/opt/arqasrv` for installing the service other than the default directory `/opt/arqasrv`. 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`) to specify the `<srvusername>` user who will be an owner of the service files and on whose behalf the installed service will run. The user name by default is **arqasrv**.

-u [<srvname>]

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

Updating 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 – `quik`.

The service location and the user-owner name of the server are not required, because they are saved in the `/etc/arqasrv/<srvname>`.

-e

Execution with this launch parameter is available for any user.



Unpacking the full archive including configuration ini files. It is unpacked into a sub-directory with a name equal to the name of the run file (without .run), which is created next to the executable file in case of absence.

-h, -?

Execution with this launch parameter is available for any user.

Display a summary of the launch parameters.

For example:

Running the run file to install the service:

```
sudo ./install_access_server-xx.xx.xx.run -i quiksub -d /opt/customdir
```

Running the run file to update the service:

```
sudo ./install_access_server-xx.xx.xx.run -u quiksub
```

Running the run file for unpacking:

```
./install_access_server-xx.xx.xx.run -e
```

3.3 Access server migration from Windows OS to Linux OS

To migrate software from Windows OS to Astra Linux OS, the following restrictions of the Linux version compared to the Windows version must be considered:

- 1.** MP crypto provider is not supported.
- 2.** 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 during strictly non-trading hours: after the end of trading and stopping the service, and before the start of trading, before the first start of the service.

Service migration is executed by two steps:



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

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

1. Copy the `install_access_server-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 `~/`).
2. Run the command to add execution permissions to the run file:

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

3. Install the service by running the command:

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

You may need to enter the current user's password to run the command. The following operations are performed during the execution of the run file:

- If the **arqasrv** system user does not exist, it will be created with default permissions and included in the group with the same name. The installed service will operate on behalf of this user. The **/opt/arqasrv** directory will be set as a home directory for this user.
- The **quik** sub-directory will be created in the **/opt/arqasrv/quik** directory. This is the service installation directory into which the distribution files of the installed software will be unpacked.
- The **arqasrv** user will be assigned as the owner of all files and sub-directories in the service directory.
- The service named **quik** will be registered in the systemd service administration system and run as the **arqasrv** system user.
- The file named **quik** will be created in the **/etc/arqasrv** directory to commit the service installation parameters.
- The following command files will be 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** – displaying information about service current 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@argatech.com.

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

```
sudo apt install libjemalloc2
```

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

5. If the administration of the installed service is expected to be performed by the arqasrv user, follow the steps:
 - _ To prevent arqasrv from making changes to the service administration scripts, that the user will be granted root permission to run, 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 service administration scripts of the service administration as the root by running the following steps:
 - _ Run the command:

```
sudo visudo
```

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

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

This line allows the **arqasrv** user to execute the listed scripts on current computer in the domain as root.



6. Copy the following files (if they exist) from the Windows service directory to the **/opt/arqasrv/quik/<srvname>** Linux service directory:

- quik.ini main configuration file;
- crypto.cfg general crypto provider configuration file;
- Other *.ini configuration files with their location saved relative to the service installation directory;
- Key files (*.txk), provider certificates catalogs:
 - The **secrserv.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.

7. 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/<srvname>** directory:

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

8. Change the settings in the migrated files:

- All references to dynamic libraries (*.dll) must be converted to the corresponding Linux name. To do this, add the "lib" prefix to the name and replace the extension with *.so, for example:

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

- List of the **quik.ini** file sections, which contain the names of dynamic libraries that must be converted:

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

[dwell_module]
module=dtwcli.dll
replace with
module=libdtwcli.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
```

- Replace the names of dynamic libraries in the `qcrypto.cfg` file. Sections that contain the names of dynamic libraries:

```
[auth]
look=qcrypto.cfg,db_look
look=qcrypto.cfg,locke_look

[locke_look]
use=dwqx.dll
replace with
use=libdwqx.so

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

- In the `cryptoprovider` settings of the `qcrypto.cfg` file, change the names of the dynamic libraries and register of the configuration file, for example:

```
[CryptoProviders]
openssl=1

[openssl]
Module=OpenSSL_Pr.dll
IniFile=OpenSSL_Pr.ini

replace with

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

- All references to file and directory paths should be converted to the Linux file system form – backslashes (“\”) should be replaced with forward slashes (“/”), for example:

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



- _ List of the **quik.ini** settings, which contain paths to files or directories with example values:

```
[ctl_module]
ip_address_file=ip.cfg
rest_users_file=restu.cfg

[usend_module]
suspect_data_file=stats/suspect.txt

[key_management]
key_info_file=key_info.ini

[dwell_module]
proxy_file_settings=quik.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=grypto.cfg
```

- _ List of the **grypto.cfg** settings, which contain paths to files or directories with example values:

```
[auth]
mail_settings_file=grypto.cfg
arw_log=stats/arw.log

[locke_look]
locke=pubbring.txk
```

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

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

- Network disks are usually mounted to local directories, so you should check with your 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
```

9. Set up automatic daily start and stop of the service. To do this, use the q_start.sh and q_stop.sh command files (located in the directory of the installed service) in the automation scripts.

The Access server must be restarted daily. The start should be performed within a short period of time after the main server is started (3-5 minutes). The Access server must be stopped 3-5 minutes before the main server is stopped.

3.4 Removing service

Removing on Windows

To remove the service, follow the steps:

1. Stop the Access server service. This can be done by the **Service** application opened as administrator or by the **Service** tab of the task manager also opened as administrator.
2. Remove the Access server service by running the command in the console running by administrator:

```
quik.exe -remove
```

3. Remove the Access server directory.



Removing on Linux

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

Below is the procedure for service removal with the default name – quik.

To remove a service, follow these steps:

1. Stop the QUIK Access server 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/quik/q_stop.sh
```

2. If the 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 by doing the following:
 - _ Run the command:

```
sudo visudo
```

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

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

- Disable the attribute of immutability of the service administration scripts 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 the following command:



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

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

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

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

- The **quik** service registration is deleted from the **systemd** service administration system.
- The **/etc/arqasrv/quik** file where the parameters of the installed service were recorded is deleted.
- The **/opt/arqasrv/quik/quik** directory where the service is installed changes its name to the **/opt/arqasrv/quik/quik.backup_YYYYY.MM.DD_hh.mm.ss**. The directory is renamed, not deleted 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@argatech.com.

4. Using OpenSSL encryption provider

4.1 Certificate receipt

To get templates of connection settings files via OpenSSL, contact the QUIK technical support.

Before Access server configuration to connect user to it by login and password you need to get the Security certificate. Certificate and a key are issued by the Security service.

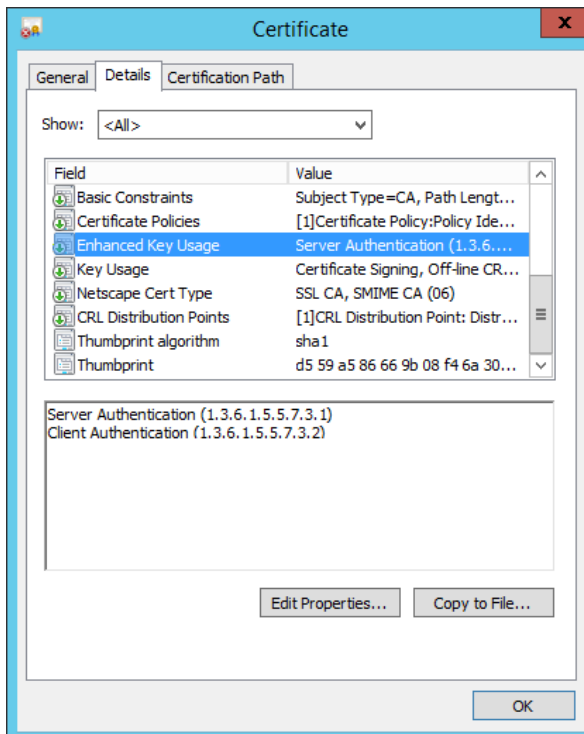
The certificate generation process consists of the following steps:

1. Key generation.
2. Generation the request for certificate.



3. Sending a request to the Certification authority and receiving a certificate.

Depending on the selected Certification authority the process may vary but the properties of the resulting certificate should always contain the value of the **Enhanced Key Usage** field — Server Authentication (the **Details** tab):

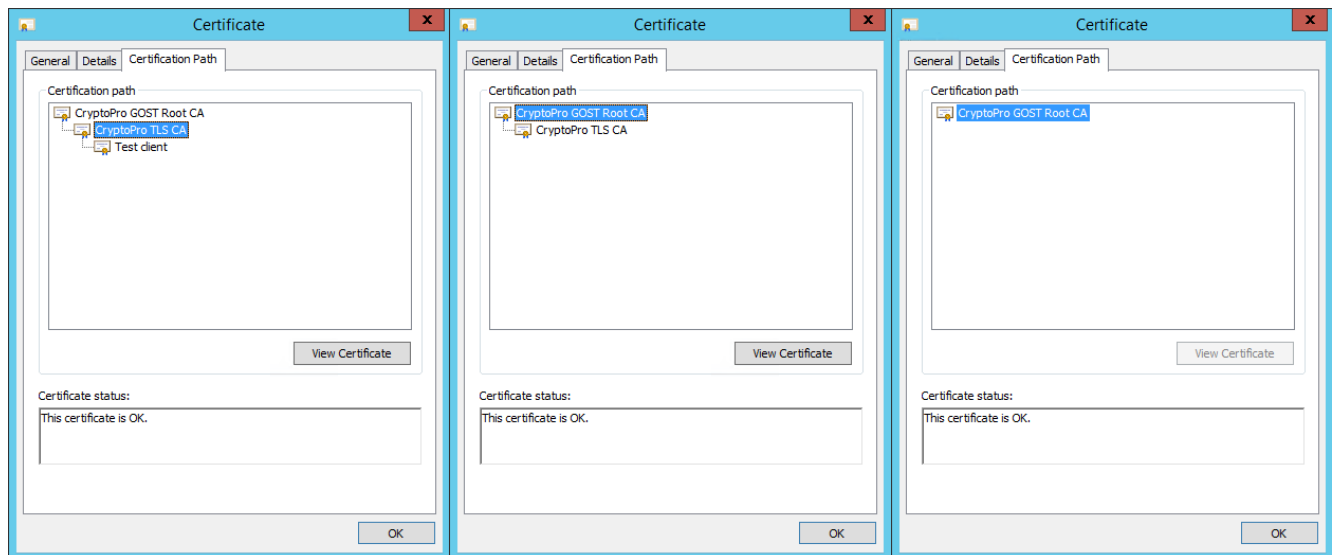


4.2 Certificate preparation

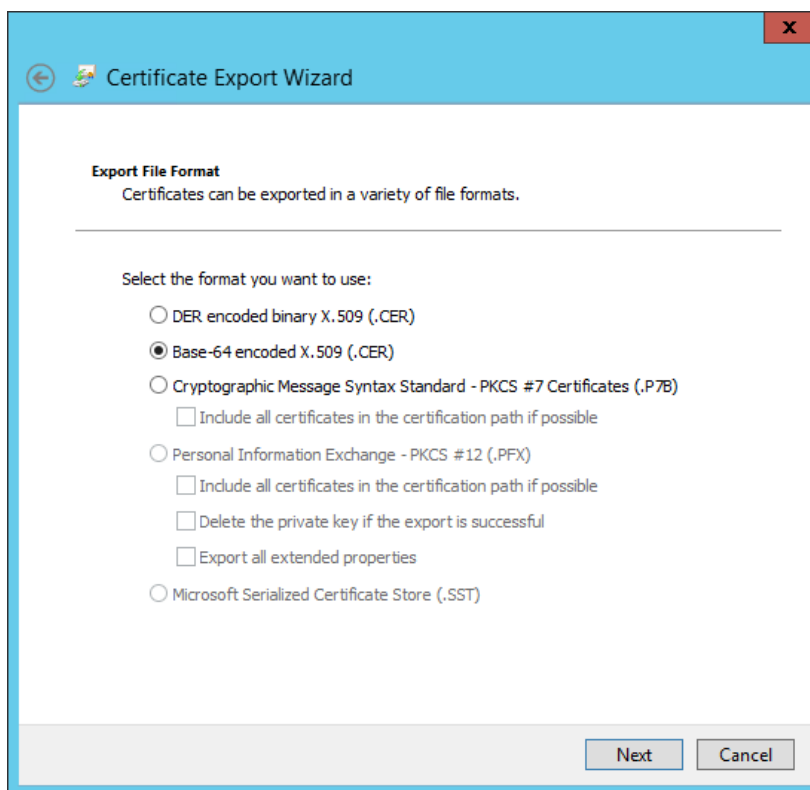
After receiving the certificate and key, open and extract the private (*.key) and public (*.cer or *.crt) keys. If the files have a *.pem extension, rename: the key to *.key and the certificate to *.cer or *.crt.

1. Open the certificate file to view. In the **Certificate** dialog that opens, go to the **Certification Path** tab and click the **View Certificate** button on each certificate (there can be two and/or more intermediate certificates) to display the entire chain.

The most recent certificate in the chain is the root one, for which the **View Certificate** button is not available.



2. Go to the **Details** tab and click the **Copy to File** button.
3. Save each previously opened certificate into a separate file, selecting the **Base-64 encoded X.509 (.CER)** file format:



It is recommended to use the following certificate naming scheme to minimize errors:

- **root.cer** – root certificate (root certificate of the Certification authority).
- **mid.cer** – intermediate Certification authority certificate.
- **end.cer** – end certificate (for example, user).

4. Place the root.cer and mid.cer files in a separate Server\certs\CA directory.

5. Move the end.cer file to the Server\certs\ directory and rename it to store.pem.

Make a copy of the finished certs directory for further use when setting up the QUIK Workplace or web2QUIK. A copy must be made before performing the next step.

6. In a text editor, open the *.key secret key file for reading and copy its contents to the clipboard.

7. In a text editor, open the store.pem file located in the Server\certs\CA directory for reading/writing. Paste the private key section from the buffer in the following order: certificate section on top and private key section below the certificate section.

1. **Changing the order of sections is unacceptable.**
2. **It is unacceptable to have additional characters or spaces before the section, between sections and after sections.**

Example:

```
-----BEGIN CERTIFICATE-----
3kZMw8o0nMIIDhTCCAm2gAwIBAgIJANX6GM1211/yMA0GCSqGSIb3DQEBCwUAMEs
BAYTA1JVMQwwCgYDVQQIEwNOU08xDTALBgNVBAoTBVbYKMF0ATk8lZXiiJa0ir2K
ZXYxEDA0BgNVBeeqiQ0+P5A+SplAMTB2FycWEucnUwHhcNMjAwMjA1MDUxOTA0Wh
q8DyFaAfVBpSJK6JvvL2ZfPAFJ3dkrJTZitmczR4GGvjT0hSaccNMjEwMjA0MDUx
VUtaZ1fwY061E3NrnQDG31ib9tzrdPUhrP9uzwmZrEFSUUEXDTALBgNVBAsTBFFE
QevKUXQQCxIUajTt1/Gtaa2yOxDSv9c3m/wk2KslgQ0zYSjutJFqWC8=
-----END CERTIFICATE-----
-----BEGIN RSA PRIVATE KEY-----
MIIEpQIBAAKCAQEAg9FF9gmLCSLbiuRZowhUtCoTkt1w6vSfAruG09KVmjSOILAKC
QAt0q+Lz1i0Xa2QgT4D0ZonX9iuTgsxnuHaMkyllR2bDJRckse3e1gT3gRetylK
JmLt5m4HiCX00Lx37weC1oJwPcu5BJhLV/vaSQ0jyMtESGPWv44qcvutuxwaIRja
OZe10/y298rg1dewtFvNl3Pq3qgxBV2ApnsSVYc2Z7Mvg41TIOK0uaexpMZzNVTZ
zwkgJb/IYgyN2rfgbIgdAoGAaZLIoeqfQK/DvSI4BhhV2o4wE2aQCetD9vKhpuWL
JyxCzAJBgNV42fwVXxdrVGRfjt7EkKbZy1EMcNJYwMDI3JUavI7IYfM=
-----END RSA PRIVATE KEY-----
```

4.3 QUIK server configuration

4.3.1 Configuration on Windows

1. From the archive received after contacting QUIK technical support, move the Server\OpenSSL_Pr.ini file to the root directory of the QUIK server. There is no need to edit the file.
2. In the directory with the QUIK server, place the certs directory, place the root.cer and mid.cer files in the certs\CA folder, place the end.cer file in the certs\ folder and rename it to store.pem.
3. Open the crypto.cfg file for editing and add the line to the [CryptoProviders] section:

```
openssl=1
```

4. Add the section below:

```
[openssl]
Module=OpenSSL_Pr.dll
IniFile=OpenSSL_pr.ini
```

Example:

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

[placebo]
Module=Placebo_Pr.dll
IniFile=

[openssl]
Module=OpenSSL_Pr.dll
IniFile=OpenSSL_pr.ini
```

5. Open the **quik.ini** file for editing and add the line to the [usend_module] section:

```
crypto_provider_ini_fileOpenSSL=OpenSSL_pr.ini.
```



Example:

```
[usend_module]
module=dwsend.dll
name=quiktest
password=qwe
port=15100
crypto_provider_ini_fileOpenSSL=OpenSSL_Pr.ini
```

6. Open for editing the OpenSSL_Pr.ini file, which should be located in the folder with the QUIK server and add the following settings:

```
[General]
CertStoreFile=certs\store.pem
ServerCertificateKey=
```

7. In the ServerCertificateKey setting, specify the password for the server key. The password may be encrypted. For encryption, the \$ character is indicated at the beginning of the password. If the server key file (in store.pem) does not have a password, then the setting can have any value, including empty.
8. Add the following setting to the OpenSSL_Pr.ini file:

```
[TLS Connection Settings]
ServerHandShakeScheme=12
```

The ServerHandShakeScheme value is 12 ensures that the password recovery mode and authentication by login and password are enabled; if recovery is not needed, then specify the ServerHandShakeScheme value is 4.

4.3.2 Configuration on Linux

1. From the archive received after contacting QUIK technical support, move the server/openssl_pr.ini file to the root directory of the QUIK server. There is no need to edit the file.
2. In the directory with the QUIK server, place the certs directory, place the root.cer and mid.cer files in the certs/ca folder, place the end.cer file in the certs/ folder and rename it to store.pem.



3. Open the `crypto.cfg` file for editing and add the line to the `[CryptoProviders]` section:

```
openssl=1
```

4. Add the section below:

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

Example:

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

[placebo]
Module=libplacebo_pr.so
IniFile=

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

5. Open the **quik.ini** file for editing and add the line to the `[usend_module]` section:

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

Example:

```
[usend_module]
module=dwsend.dll
name=quiktest
password=qwe
port=15100
crypto_provider_ini_file=OpenSSL_Pr.ini
```



6. Open for editing the openssl_pr.ini file, which should be located in the folder with the QUIK server and add the following settings:

```
[General]
CertStoreFile=certs/store.pem
ServerCertificateKey=
```

7. In the ServerCertificateKey setting, specify the password for the server key. The password may be encrypted. For encryption, the \$ character is indicated at the beginning of the password. If the server key file (in store.pem) does not have a password, then the setting can have any value, including empty.
8. Add the following setting to the openssl_pr.ini file:

```
[TLS Connection Settings]
ServerHandShakeScheme=12
```

The ServerHandShakeScheme value is 12 ensures that the password recovery mode and authentication by login and password are enabled; if recovery is not needed, then specify the ServerHandShakeScheme value is 4.

4.4 Access server configuration

4.4.1 Configuration on Windows

1. From the archive received after contacting QUIK technical support move the Server\OpenSSL_Pr.ini file to the root directory of the QUIK server. There is no need to edit the file.
2. In the directory with the Access server, place the certs directory, place the root.cer and mid.cer files in the certs\CA folder, place the end.cer file in the certs\ folder and rename it to store.pem.
3. Open the crypto.cfg file for editing and add the line to the [CryptoProviders] section:

```
openssl=1
```

4. Add the section below:



```
[openssl]
Module=OpenSSL_Pr.dll
IniFile=OpenSSL_pr.ini
```

Example:

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

[placebo]
Module=Placebo_Pr.dll
IniFile=

[openssl]
Module=OpenSSL_Pr.dll
IniFile=OpenSSL_pr.ini
```

5. Open the quik.ini file for editing and add the line to the [usend_module] section:

```
crypto_provider_ini_file=OpenSSL_pr.ini
```

Example:

```
[usend_module]
module=dwsend.dll
name=quiktest
password=qwe
port=15100
crypto_provider_ini_file=OpenSSL_Pr.ini
```

4.4.2 Configuration on Linux

1. From the archive received after contacting QUIK technical support move the server/openssl_pr.ini file to the root directory of the QUIK server. There is no need to edit the file.



2. In the directory with the Access server, place the certs directory, place the root.cer and mid.cer files in the certs/ca folder, place the end.cer file in the certs/ folder and rename it to store.pem.
3. Open the crypto.cfg file for editing and add the line to the [CryptoProviders] section:

```
openssl=1
```

4. Add the section below:

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

Example:

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

[placebo]
Module=libplacebo_pr.so
IniFile=

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

5. Open the quik.ini file for editing and add the line to the [usend_module] section:

```
crypto_provider_ini_file=openssl_pr.ini
```

Example:

```
[usend_module]
module=libdwsend.so
name=quiktest
```



```
password=qwe
port=15100
crypto_provider_ini_file=openssl_pr.ini
```

4.5 Setting up workstation for connecting via OpenSSL

In the QUIK Workstation folder, you need to create a **certs** folder, inside the **certs** folder, a **CA** folder and place the root (root.cer) and intermediate Certification authority (mid.cer) certificate files in it, for more details, see section [4.2](#).

Copy Front\OpenSSL_pr_client.ini from openssl2QUIK.zip to the QUIK Workstation folder.

Switch your terminal to OpenSSL using one of the following methods:

1. Manually:

- In the crypto.ini file, add / change the missing lines according to the example:

```
[CryptoProviders]
OpenSSL_Provider=1

[CryptoProvidersSettings]
CipherCryptoProvider=OpenSSL_Provider

[OpenSSL_Provider]
Module=.\OpenSSL_Pr.dll
IniFile=.\OpenSSL_pr_client.ini
```

- In the OpenSSL_pr_client.ini file of the [TLS Connection Settings] section, check for the presence of the **CheckPeerCertificate** setting:

```
[TLS Connection Settings]
UsedHandShakeScheme=4
CheckPeerCertificate=1
```

2. Via QUIK Workstation:

- In the **Workstation settings** window (**System / Settings / General Settings...**), select the **Encryption** tab (**Program / Encryption**) and click the  **Encrypt using CIPF button**.
- In the Crypto providers window that opens, click on the **Add new** button and fill in the parameters in the **Add crypto provider** window as follows:
 - **Name** – OpenSSL_Provider;



- **Description** – OpenSSL_Provider;
- **Executable module file** – D:\QUIK\Front\OpenSSL_Pr.dll;
- **Crypto provider settings file** – D:\QUIK\Front\OpenSSL_pr_client.ini.
- Click the **Add** button.
- In the **Crypto providers** window, select the line with OpenSSL in the **List of installed crypto providers** field and click on the **Select** button.
- In the **Workstation settings** window, on the **Encryption** tab, click on the  **Default settings** button. In the window that opens, in the **Authentication** field, select "by name and password" and click **OK**.

4.6 Setting up Web2QUIK connection via OpenSSL

To configure the connection of the web2QUIK Module to the server and the HTTPS connection of mobile applications to the web2QUIK Module, follow these steps:

1. From the archive received after contacting QUIK technical support, move the web2QUIK\OpenSSL_pr_client.ini file to the web2QUIK root directory. There is no need to edit the file.
2. Place the certs catalog in the directory with web2QUIK.
3. Place the certificates of the new Certification authority (root and intermediate - the entire chain) in the certs\CA folder. The name of the certificate must be equal to the value in subject_hash, and the extension must be 0, example:

```
645912cb.0
```

You can get Subject_hash yourself by running the command (you must have openssl.exe on your computer):

```
openssl.exe x509 -in org.crt -noout -subject_hash
```

For information about subject_hash, please contact QUIK technical support:
quiksupport@argatech.com.

4. Open the web2quik.ini file for editing:



- In the [General] section, add the line:

```
openssl=1
```

- In the [General] section, specify the path and name to the key and certificate file.

Example:

```
[General]
openssl=1
cert_file=brokerru.crt
private_key_file=brokerru.key
```

- 1. To connect the web2QUIK Module to the server, you can use the same key/certificate pair as for connecting mobile applications to the web2QUIK Module (HTTPS connection).**
- 2. If it is necessary to use two different key/certificate pairs, the second pair (web2QUIK connection to the server) can be issued by the internal security service. For this connection, certificates may be self-signed (issued by a non-globally recognized Certificate authority).**

- 5.** It is recommended to check the presence of the CheckPeerCertificate=1 setting in the [General] section of the web2quik.ini file. If the setting is not specified, the QUIK server certificate is not validated against the Certification authority certificate.

Example:

```
[General]
...
CheckPeerCertificate=1
```

4.7 Adding chain of CA certificates to web2QUIK https certificate

Some versions of the Android OS require the entire certificate chain. The procedure for creating the correct certificate for such cases is as follows:

- 1.** From the root.cer, mid.cer and end.cer files copied from sub-section [4.2](#) the end.cer file is the final certificate issued for example, for the test.webquik.ru resource.



2. Open the end.cer file for editing in a text editor and after its section add the section from the mid.cer file and after it the root.cer section.
3. Save the changes and move the file to the web2QUIK directory. Rename if necessary.

Example of the certificate with the chain of the Certification authority certificates:

```
-----BEGIN CERTIFICATE-----
pHAIV0oKHMfgosiLAInMVG8Fo4TZfkQpcSOdWigtbCmHCYuGfjjm7cFhMasp8BD9
Jxq+wKvOw24diS8m51RBcONfXHMfgosiLA6PqBd+vaCpJh21Rn9Q51bw9z130oa1
KHBdzvLKCYoEfupHd0J8h8GWLqI3dSvo8HMfgosiLAYKkeb0xKAfHG/arQvJ91w
2TpC/bBIVqY+LcLJh1feGGut4lpOSaP1NEUvJXwQIxSHMfgosiLAD/YGeXU0ljgg
PsR9le9RK92sVFnz/oetQIWs3EhwwCIvAaTNoZ6pPIaCVJc5hIwDHMfgosiLA9h2
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIGEzCCA/ugAwIBHMfgosiLAR2uhHdbBYLvFMNpzANBgkqhkiG9w0BAQwFADCB
l6lFhd2zi+WJN44pDfwGF/Y4QHMFgosiLAvzxhFoYt/jmPQT2BVPi7Fp2RBgvGQq
6jG35LWj0hSbJuMLe/0CjraZwTiXWtb2qHMfgosiLAK6s+go/lunrotEbaGmAhy
LcmsJWtyXnw00MGuf1pGg+pRyrbxmRE1a6Vqe8YAsOfHMfgosiLA8azjUeqkk+B5
yOGBQMkKW+ESPMFgKuOXwIILCypTPRpgSabuY0MLTDXJLR27lk8QyHMfgosiLAj4K
00u/I5sUKUErmgQfky3xxz1HMfgosiLA
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MAkHMfgosiLAR0IxGzAZBgNVBAGMEkdyZWFOZXIgdWVhY2hlc3RlcjEQA4GA1UE
BwwHU2FsZm9yHMfgosiLA1UECgwRQ29tb2RvIENBIExpbWl0ZWQxITAFBgNVBAMM
GEFBQSB0ZXJ0aWZpY2F0ZHMfgosiLANlczCCASIwDQYJKoZIhvcNAQEBBQADggEP
-----END CERTIFICATE-----
```

The order of certificates must be as follows:

1. **Security certificate issued to the company's domain name.**
2. **Certificate of the intermediate Certification authority.**
3. **Certificate of the root Certification authority.**

4.8 Replacing expired certificate

It is necessary to monitor the validity period of the certificate and issue a new certificate no later than 2 weeks before the expiration of the current certificate.

After receiving a new certificate and private key, you must re-configure certificates for the Access server.

If a new certificate is issued by a new Certification authority, measures must be taken to distribute new Certification authority certificates to user terminals in advance.

Distribution of Certification authority certificates to QUIK terminals is expected together with an update package located on the QUIK server or Access server, so it is recommended to place the

files in advance in the Areas\Frontend\x64\Release\certs\CA\ folder on the QUIK server or Access server so that the terminals have time to accept new certificates while the old certificate is working.

