



QUIK Transactions to XML

Руководство по администрированию

Версия 3.0



Содержание

1. Общие сведения	3
1.1 Назначение	3
1.2 Принцип работы.....	3
2. Установка и настройка Программы	4
2.1 Технические требования	4
2.2 Требования к безопасности	4
2.3 Установка	4
2.4 Запуск Программы	9
2.5 Формат файла qt2xml.ini	15
2.6 Формат файла qcrypto.ini	27
2.7 Формат XML-файла экспорта	28
3. Коды завершения приложения.....	36
4. Статусы транзакций	37
5. Типы транзакций	38
6. Дополнительные типы транзакций	38
7. Типы полей транзакций.....	38
8. Соответствия между элементами XML-файла и полями БД QUIK	39
9. Преобразование формируемого XML-файла	41
10. Шифрование паролей.....	41

Ваши пожелания и комментарии к данному Руководству направляйте по электронной почте на адрес: quiksupport@arqatech.com



1. Общие сведения

1.1 Назначение

Программа **QUIK Transactions to XML** (qt2xml) предназначена для экспорта данных по транзакциям из базы данных QUIK в файл XML-формата с возможностью фильтрации экспортируемых данных.

1.2 Принцип работы

Программа представляет собой консольное приложение, которое запускается при необходимости вывода в XML-файл данных о транзакциях, поданных пользователями за некоторый период. Возможна фильтрация данных по следующим параметрам:

- по идентификатору (ID) пользователя;
- по коду клиента;
- по коду фирмы;
- по начальным символам кода клиента;
- по номеру заявки;
- по наличию ЭЦП;
- по статусу транзакции;
- по типу транзакции.

В результате работы Программы формируется XML-файл с данными о транзакциях.

В процессе работы Программа может выводить сообщения на консоль, основные из которых:

- **«Exported %n transactions(s)»** – сообщение о количестве обработанных транзакций, где %n – количество транзакций, данные о которых выведены в файл. Сообщение выводится в случае успешного завершения работы Программы,
- **«Error: %s»** – сообщение об ошибке, где %s – текстовое описание ошибки в произвольном формате.

По окончании работы Программа передает операционной системе строку с кодом завершения приложения (errorlevel), который сигнализирует об условиях завершения работы Программы (подробнее о значениях кодов завершения приложения см. п. [3](#)).

1.2.1 Условия, необходимые для работы Программы

Для работы Программы требуется наличие доступа к базе данных QUIK.



2. Установка и настройка Программы

2.1 Технические требования

Таблицу требований к оборудованию и программному обеспечению можно посмотреть на официальном [сайте QUIK](#).

2.2 Требования к безопасности

Для работы Программы требуется указание паролей для доступа к базе данных QUIK. При этом перечисленные пароли не должны начинаться с символов «\$» и «#» (если указанные символы используются, то их следует изменить). В целях безопасности указанные пароли могут быть сохранены в файле настроек Программы в зашифрованном виде (подробнее см. п. [10](#)).

2.3 Установка

Команды, описанные ниже, приводятся для ОС Windows и ОС Linux.

1. Для ОС Windows в начале команды указывается символ «>».
2. Для ОС Linux в начале команды указывается символ «\$», если команда выполняется от текущего пользователя, или символ «#», если команда выполняется от пользователя с административными привилегиями (обычно это пользователь root, либо пользователь, включенный в группы root, wheel).
3. Символы «>», «\$», и «#» не входят в команды.

2.3.1 Установка под ОС Windows

1. Распакуйте присланный архив qt2xml_<номер версии>_upd.zip с поддиректориями и скопируйте в целевую директорию.
2. Выполните следующие настройки для ОС Windows:

Для СУБД MSSQL

- При подключении к СУБД MSSQL на компьютере, где устанавливается qt2xml, должен быть установлен ODBC-драйвер «ODBC Driver 17 for SQL Server». Для проверки наличия драйвера в меню Windows наберите в поиске текст «Источники данных ODBC» и откройте соответствующее окно. Убедитесь, что на вкладке «Драйверы» в списке присутствует драйвер с наименованием «ODBC Driver 17 for SQL Server».



Если такой драйвер отсутствует, скачайте установочный файл драйвера с официального сайта Microsoft: <https://go.microsoft.com/fwlink/?linkid=2223304> и установите его на компьютер, на котором будет работать сервис.

Для СУБД PostgresPro

- При подключении к СУБД PostgresPro на компьютере, где устанавливается qt2xml, должен быть установлен ODBC-драйвер PostgreSQL ANSI(x64) версии не ниже 13.2. Для проверки наличия драйвера и его версии в меню Windows наберите в поиске текст «Источники данных ODBC» и откройте соответствующее окно. Убедитесь, что на вкладке «Драйверы» в списке присутствует драйвер с именем «PostgreSQL ANSI(x64)» и версией 13.2.

Если такой драйвер отсутствует либо версия ниже ожидаемой, скачайте установочный файл драйвера с официального сайта postgresql: https://ftp.postgresql.org/pub/odbc/versions/msi/psqlodbc_13_02_0000-x64.zip и установите его на компьютер, на котором будет работать сервис.

3. Выполните настройки Программы в основном файле [qt2xml.ini](#).

- В секции [\[QUIK_DB\]](#) укажите сервер для подключения к БД-источнику:

```
[QUIK_DB]
Server=<Имя сервера БД>
```

- Задайте остальные параметры в секции [\[QUIK_DB\]](#).

4. Выполните запуск Программы qt2xml.

Программа qt2xml предназначена для работы в standalone-режиме в качестве утилиты, с возвратом кода завершения.

Для ОС Windows запуск утилиты выполняется вызовом бинарного файла qt2xml из директории с расположением файлов Программы.

```
> qt2xml
```

В процессе работы выполняется вывод результатов работы в терминал (при запуске в терминале) и производится запись в лог-файл (по умолчанию: qt2xml.log).

При вызове утилиты ей могут быть переданы аргументы командной строки, например:

```
> qt2xml -version
```

Аргумент «-v» или «-version» отображает версии программы.



После окончания работы, либо при возникновении ошибки, процесс завершится с соответствующим сообщением в терминале и/или лог-файле.

2.3.2 Установка под ОС Linux

1. Скопируйте присланный архив `install_qt2xml-<номер версии>.run` в целевую директорию.
2. Распакуйте архив, выполнив команды:

```
$ chmod +x install_qt2xml-<номер версии>.run
$ ./install_qt2xml-<номер версии>.run
```

3. Выполните следующие настройки для ОС Linux:

- Для работы с ODBC, необходимо установить пакеты `libodbc1` и `unixodbc` с помощью команд:

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

Далее следует настроить необходимые драйвера для СУБД.

Для СУБД MSSQL

- При подключении к СУБД MSSQL на компьютере, где устанавливается `qt2xml`, должен быть установлен ODBC-драйвер «ODBC Driver 17 for SQL Server».
- Выполните следующие команды от пользователя с административными привилегиями:

```
#curl https://packages.microsoft.com/keys/microsoft.asc | apt-key add -
#curl https://packages.microsoft.com/config/debian/9/prod.list >
/etc/apt/sources.list.d/mssql-release.list
#apt-get update
#ACCEPT_EULA=Y apt-get install -y msodbcsql17
```

- Убедитесь, что в файле `/etc/odbcinst.ini` добавилась запись драйвера:

```
[ODBC Driver 17 for SQL Server]
Description=Microsoft ODBC Driver 17 for SQL Server
Driver=/opt/microsoft/msodbcsql17/lib64/libmsodbcsql-17.10.so.4.1
```

Если запись не была добавлена, добавьте ее самостоятельно.

Убедитесь, что файл расположен по пути:



/opt/microsoft/msodbcsql17/lib64/libmsodbcsql-17.10.so.4.1 (имя файла может отличаться в зависимости от версии предоставленной Microsoft.com).

- При подключении к СУБД MSSQL Server 2012 требуется понизить дефолтный уровень шифрования до TLS1.0. Это связано с тем, что MSSQL 2012 не поддерживает TLS1.2. Для этого в файле /etc/ssl/openssl.cnf измените значения:

```
[system_default_sect]
MinProtocol = TLSv1.0
CipherString = DEFAULT@SECLEVEL=1
```

Для MSSQL 2017 и выше этого не требуется.

Для СУБД PostgresPro

- При подключении к СУБД PostgresPro на компьютере, где устанавливается qt2xml, должен быть установлен ODBC-драйвер PostgreSQL версии 13.02 или выше.
- Выполните следующие команды от пользователя с административными привилегиями:

```
#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
```

- Если установка пакетов из официального репозитория PostgreSQL недоступна (установка / обновление пакетов завершается ошибкой), скачайте пакеты:
https://apt.postgresql.org/pub/repos/apt/pool/main/p/postgresql-16/libpq5_16.1-1.pgdg100+1_amd64.deb.
https://apt.postgresql.org/pub/repos/apt/pool/main/p/psqlodbc/odbc-postgresql_16.00.0000-1.pgdg100+1_amd64.deb.
- Установите скачанные пакеты, выполнив следующие команды:

```
#apt remove libpq5
#apt remove odbc-postgresql
#dpkg -i libpq5_16.1-1.pgdg100+1_amd64.deb
#dpkg -i odbc-postgresql_16.00.0000-1.pgdg100+1_amd64.deb
```



- _ Убедитесь, что в файле `/etc/odbcinst.ini` добавилась запись драйвера:

```
[PostgreSQL Unicode]
Description=PostgreSQL ODBC driver (Unicode version)
Driver=/usr/lib/x86_64-linux-gnu/odbc/psqlodbcw.so
Setup=libodbcpsqlS.so
Debug=0
CommLog=1
UsageCount=1
```

Если запись не была добавлена, добавьте ее самостоятельно.

Убедитесь, что файл расположен по пути:

`/usr/lib/x86_64-linux-gnu/odbc/psqlodbcw.so` (имя файла может отличаться в зависимости от версии).

При работе под ОС Windows используется ANSI-драйвер, а при работе под ОС Linux используется UNICODE-драйвер.

4. Выполните настройки Программы в основном файле [qt2xml.ini](#).

- _ В секции [\[QUIK_DB\]](#) укажите сервер для подключения к БД-источнику:

```
[QUIK_DB]
Server=<Имя сервера БД>
```

- _ Задайте остальные параметры в секции [\[QUIK_DB\]](#).

5. Выполните запуск Программы qt2xml.

Запуск утилиты выполняется запуском `qt2xml` из директории с расположением файлов Программы.

```
./qt2xml
```

Где:

«../» в начале команды указывает на необходимость выполнить команду из текущей директории.

В процессе работы выполняется вывод результатов работы в терминал (при запуске в терминале) и производится запись в лог-файл (по умолчанию: `qt2xml.log`).



При вызове утилиты ей могут быть переданы аргументы командной строки, например:

```
./qt2xml -version
```

Аргумент «-v» или «-version» отображает версии программы.

После окончания работы, либо при возникновении ошибки, процесс завершится с соответствующим сообщением в терминале и/или лог-файле.

2.4 Запуск Программы

Программа представляет собой консольное приложение. При запуске Программа считывает настройки из файла настроек (подробнее о настройках, задаваемых в файле настроек см. п. [2.5](#)). Кроме файла настроек параметры могут задаваться и в командной строке. Если одна и та же настройка задана и в файле настроек, и в командной строке, то приоритет имеет настройка из командной строки.

Программа qt2xml поддерживает следующий набор ключей:

- **-period<начало периода>-<конец периода>,<начало временного промежутка>-<конец временного промежутка>** – начало и конец периода для вывода данных о транзакциях, задаются в формате <ГГГГММДД> (дата), либо в формате <ГГГГММДДЧЧММСС> (дата и время). При этом начало и конец периода могут быть заданы в разных форматах. В отчёт включаются только дата и время начала периода. Если граница периода задана только датой, то считается, что день начинается в 00:00:00 и заканчивается при наступлении следующего дня. Начало и конец временного промежутка вывода данных о транзакциях, задаются в формате <ЧЧММСС> (время). Если задан временной промежуток, то вывод данных о транзакциях выполняется каждый день в течение указанного периода в этот временной промежуток. При этом если начало и/или конец временного промежутка выходят за границы периода, то в первый и/или последний день периода вывод данных выполняется только во временных рамках начала и/или конца периода. Если ключ не задан и не используется ключ **-periodprevday**, то в качестве периода используются текущие сутки.

Пример:

```
qt2xml -period20090501-20090501183000,140000-180000
```

- **-periodprevday<начало временного промежутка>-<конец временного промежутка>** – в качестве периода выбирается указанный временной промежуток предыдущих суток. Начало и конец временного промежутка задаются в формате <ЧЧММСС>. Если временной промежуток не задан, то в качестве периода выбираются предыдущие сутки.



Пример:

```
qt2xml -periodprevday
```

- **-uid<ID пользователя>** – фильтр данных о транзакциях по идентификатору пользователя (ID). Значения в списке указываются через запятую. Если ключ не задан, то данные выводятся по всем пользователям.

Пример:

```
qt2xml -uid6,12,418
```

- **-excludeuid** – исключающий фильтр данных о транзакциях по идентификатору пользователя (ID). Значения в списке указываются через запятую. Данные выводятся по всем пользователям, кроме указанных в ключе.

Пример:

```
qt2xml -excludeuid6,12,418
```

Если заданы оба фильтра -uid<ID пользователя> и -excludeuid, то исключающий фильтр -excludeuid игнорируется.

- **-client<код клиента или маска>** – фильтр данных о транзакциях по коду клиента или маске (указываются начальные символы кода клиента и символ «*»). Если ключ не задан, то данные выводятся по всем кодам клиентов.

Пример:

```
qt2xml -clientE100 или qt2xml -clientE*
```

- **-nouserinfo** – запрет вывода элемента UserInfo.
- **-status<статус>** – фильтр данных о транзакциях по статусу транзакции (каждый статус задаётся некоторым символом, см. п. 4). Может быть указан один или несколько статусов транзакций. Статусы перечисляются в любом порядке, без разделителей. Если ключ не задан, то данные выводятся вне зависимости от статуса транзакций.

Пример:

```
qt2xml -statusof
```



- **-output<имя файла>** – имя XML-файла для вывода данных. Если ключ не задан, то данные по транзакциям выводятся в создаваемый автоматически файл **trans.xml**, расположенный в текущем каталоге.

Пример:

```
qt2xml -outputTransReport.xml
```

- **-ini<имя файла>** – абсолютный или относительный путь к файлу настроек. Если ключ не задан, то Программа ищет файл с именем **qt2xml.ini** в текущем каталоге.

Пример:

```
qt2xml -ini.\qt2xml.ini
```

- **-classes<список>** – список классов бумаг, по которым разрешена передача информации о транзакциях, через запятую.

Пример:

```
qt2xml -classesEQBR,EQNE
```

- **-excludeclasses<список>** – список классов бумаг, по которым запрещена передача информации о транзакциях, через запятую.

Пример:

```
qt2xml -excludeclassesPSEQ,PSES
```

Если списки -classes и -excludeclasses не указаны, то выводится информация по всем классам. Если указаны оба списка, то список -excludeclasses игнорируется.

- **-securities<список>** – список допустимых кодов инструментов, через запятую.

Пример:

```
qt2xml -securitiesRTKM,HYDR
```

- **-excludesecurities<список>** – список игнорируемых кодов инструментов, через запятую.



Пример:

```
qt2xml -excludesecuritiesBUSB,YKEN
```

Если списки **-securities** и **-excludesecurities** не указаны, то выводится информация по всем инструментам. Если указаны оба списка, то список **-excludesecurities** игнорируется.

- **-reply<текст>** – подстрока, которая ищется в ответе на транзакцию. В качестве <текст> может быть передана любая строка. Если она содержит пробелы, то такая строка должна быть заключена в двойные кавычки.

Пример:

```
qt2xml -reply"Вам запрещена работа по данному торговому счету"
```

- **-encodenonchar** – изменение формата вывода данных транзакции и её полей, если в транзакции присутствуют непечатные символы (кроме символа с кодом 0x20 (пробел)).

Пример:

```
qt2xml -encodenonchar
```

- **-checksign** – проверка наличия ЭЦП для каждой транзакции.

Пример:

```
qt2xml -checksign
```

1. Перед применением проверки ЭЦП на транзакциях необходимо удостовериться, что в каталоге с QT2XML присутствует файл `qcrypto.ini` с настройками крипто-провайдера (см. п. [2.6](#)), и в файле настроек `qt2xml.ini` в параметре `CryptoType` явно указан тип используемого крипто-провайдера (см. п. [2.5.3](#)).
2. Если включена проверка наличия ЭЦП, то при использовании крипто-провайдера `SignalCom` необходимо в файле `signalcom_pr.ini` в секции `[signal_com_message_pro_common_settings]` указать параметр `cert_storage_validation=0`.

- **-signoutputfile** – формирование ЭЦП для XML-файла.



Пример:

```
qt2xml -signoutputfile
```

- **-withsign** – вывод только транзакций с ЭЦП.

Пример:

```
qt2xml -withsign
```

- **-withnosign** – вывод только транзакций без ЭЦП.

Пример:

```
qt2xml -withnosign
```

Если ключи `-withsign` и `-withnosign` заданы одновременно, то ключ `-withnosign` игнорируется.

- **-transflags<набор флагов>[,<тип фильтра>]** – фильтр по полю Flags в описании транзакции (каждое значение задается определенным набором битов, см. п. [2.7.4](#)), где:
 - <набор флагов> – флаги, которые должны присутствовать в описании транзакции. Набор флагов указывается битовой маской или перечислением флагов через запятую.
 - <тип фильтра> – указывается по следующему правилу:
 - «any» – в транзакции допускается наличие любого из указанных флагов;
 - «exact» – допускается только полное соответствие перечисленных флагов;
 - «all» (по умолчанию) – допускается наличие всех указанных флагов.

При указании параметров в фильтре пробелы не допускаются.

Примеры:

Набор флагов указан битовой маской:

```
qt2xml -transflags3,any
```

Набор флагов указан перечислением:

```
qt2xml -transflags1024,2048,all
```



- **-excludetransflags<набор флагов>[,<тип фильтра>]** – фильтр по полю Flags в описании транзакции (каждое значение задается определенным набором битов, см. п. [2.7.4](#)), где:
 - <набор флагов> – флаги, которые не должны присутствовать в описании транзакции. Набор флагов указывается битовой маской или перечислением флагов через запятую.
 - <тип фильтра> – указывается по следующему правилу:
 - «any» – в транзакции не допускается наличие хотя бы одного из указанных в фильтре флагов;
 - «exact» – не допускается только полное соответствие перечисленных флагов;
 - «all» – не допускается наличие всех указанных в фильтре флагов.

При указании параметров в фильтре пробелы не допускаются.

Пример:

```
qt2xml -excludetransflags1024,any
```

- **-firms<список>** – фильтр по списку допустимых кодов фирм. Коды фирм в списке указываются через запятую.

Пример:

```
qt2xml -firmsMC0080100000,NC0058900000
```

- **-excludefirms<список>** – фильтр по списку игнорируемых кодов фирм. Коды фирм в списке указываются через запятую.

Пример:

```
qt2xml -excludefirmsNC0038900000
```

- **-ordernum<номер_1>[,...<номер_n>]** – фильтр по номерам заявок / стоп-заявок в транзакции. Номера заявок / стоп-заявок в списке указываются через запятую.

Пример:

```
qt2xml -ordernum5487,5641
```

- **-exchangenum<номер_1>[,...<номер_n>]** – фильтр по биржевым номерам заявок, неторговым поручениям в транзакции.

1. В биржевом номере не допускается использование запятых (,) и кавычек (").



2. Если хотя бы в одном номере имеется пробел, то весь список номеров должен быть заключён в кавычки (""):

```
-exchangenum"<номер_1>[,...<номер_n>]"
```

Пример:

```
qt2xml -exchangenum"ML0125,ML 0132"
```

Для некоторых ключей параметры фильтра можно не указывать. В этом случае отключается фильтр, указанный в файле настроек. Это правило применимо для следующих ключей: **-period**, **-uid**, **-excludeuid**, **-client**, **-status**, **-reply**, **-classes**, **-excludeclasses**, **-transflags**, **-excludetransflags**, **-firms**, **-excludefirms**.

- **-exportmode<режим>** – режим экспорта данных по транзакциям.

Пример:

```
qt2xml -exportmode1
```

- **-transnum<номер_1>[,...<номер_n>]** – фильтр по номерам транзакций. Номера транзакций в списке указываются через запятую.

Пример:

```
qt2xml -transnum641
```

2.5 Формат файла qt2xml.ini

2.5.1 Секция [QUIK_DB]

Секция [QUIK_DB] содержит настройки для доступа к базе данных QUIK.

- **AuthMode** – тип аутентификации. Возможные значения:
 - «login» (по умолчанию) – аутентификация через логин и пароль, заданные в параметрах «Login», «Password»;
 - «domain» – с использованием доменной аутентификации, параметры «Login» и «Password» игнорируются.
- **Type** – тип базы данных, к которой выполняется подключение. Возможные значения:



- _ «MSSQL» (по умолчанию);
 - _ «POSTGRES».
- **Driver** – имя ODBC-драйвера, используемого для подключения к базе данных.
Возможные значения:
 - _ «ODBC Driver 17 for SQL Server» (по умолчанию) – используется при работе с MS SQL Server 2014, 2016 или более новыми версиями СУБД.
 - _ «PostgreSQL ANSI(x64)» – используется при работе с PostgreSQL под Windows.
 - _ «PostgreSQL Unicode» – используется при работе с PostgreSQL под Linux.
- **SSLMode** – режим шифрования при подключении к серверу PostgreSQL. Возможные значения:
 - _ «require» – SSL-шифрование (если сервер отказал в шифровании, то выполняется отключение);
 - _ «verify-full» – SSL-шифрование с проверкой на обеих сторонах.

Если настройка отсутствует, соединение выполняется без шифрования.

- **CertCAFileName** – путь к файлу корневого сертификата.
- **CertFileName** – путь к файлу сертификата пользователя, если используется аутентификация по сертификату.
- **CertPKeyFileName** – путь к файлу приватного ключа сертификата пользователя.
- **CertPass** – пароль пользователя. Пароль не должен начинаться с символов «\$» и «#».
- Пароль может быть зашифрован (подробнее см. п. [10](#)).
- **AdditionalParameters** – дополнительные параметры соединения с базой данных.

Пример:

```
[QUIK_DB]
CertCAFileName=Certs/root.crt
CertFileName=Certs/user.crt
CertPKeyFileName=Certs/user.key
CertsPass=qwe
AdditionalParameter=MultiSubnetFailover=true <MultiSubnetFailover=true поддерживается
драйвером "ODBC Driver 17 for SQL Server" и новее>
```

Для прямого подключения к SQL-серверу в секции должны быть указаны следующие ключи:

- **Server** – имя SQL-сервера;
- **DataBase** – имя базы данных на сервере;
- **User** – логин для работы с базой данных;
- **Password** – пароль для работы с базой данных. Пароль не должен начинаться с символов «\$» и «#». Пароль может быть зашифрован (подробнее см. п. [10](#)).



Пример:

```
[QUIK_DB]
Server=QDB1
DataBase=AQ_SERV1
User=sa
Password=qwe
```

Для подключения к базе данных посредством ODBC (например, при использовании СУБД Interbase) в секции должны быть указаны следующие ключи:

- **DSN** – название ODBC-источника, указывающего на базу данных Interbase;
- **User** – логин для работы с базой данных (если требуется);
- **Password** – пароль для работы с базой данных (если требуется). Пароль не должен начинаться с символов «\$» и «#». Пароль может быть зашифрован (подробнее см. п. [10](#)).

Пример:

```
[QUIK_DB]
DSN=SERV1
User=sa
Password=qwe
```

2.5.2 Секция [General]

Секция [General] содержит ключи, значение которых аналогично значению соответствующих параметров командной строки:

- **UseXSL** – имя файла со стилем, который используется для просмотра содержимого XML-файла (подробнее о преобразовании XML-файла см. п. [9](#)).
- **Period** – формат настройки: **<начало периода>-<конец периода>, <начало временного промежутка>-<конец временного промежутка>**, где начало и конец периода для вывода данных о транзакциях, задаются в формате <ГГГГММДД> (дата), либо в формате <ГГГГММДДЧЧММСС> (дата и время). При этом начало и конец периода могут быть заданы в разных форматах. В отчёт включаются только дата и время начала периода. Если граница периода задана только датой, то считается, что день начинается в 00:00:00 и заканчивается при наступлении следующего дня.
Начало и конец временного промежутка вывода данных о транзакциях, задаются в формате <ЧЧММСС> (время). Если задан временной промежуток, то вывод данных о транзакциях выполняется каждый день в течение указанного периода в этот временной промежуток. При этом если начало и/или конец временного промежутка выходят за границы периода, то в первый и/или последний день периода вывод данных



выполняется только во временных рамках начала и/или конца периода. Если ключ не задан, то в качестве периода используются текущие сутки.

Пример:

```
Period=20090602101500-20090602183000,140000-180000
```

- **UID** – фильтр данных о транзакциях по идентификатору пользователя. Идентификаторы пользователей в списке указываются через запятую. Если фильтр не задан, то данные выводятся по всем пользователям.

Пример:

```
UID=6,12,418
```

- **ExcludeUID** – исключающий фильтр данных о транзакциях по идентификатору пользователя. Идентификаторы пользователей в списке указываются через запятую. Данные выводятся по всем пользователям, кроме указанных.

Пример:

```
UID=6,12,418
```

Если заданы оба фильтра UID и ExcludeUID, то исключающий фильтр ExcludeUID игнорируется.

- **Client** – фильтр данных о транзакциях по коду клиента или маске (указываются начальные символы кода клиента и символ «*»).

Пример:

```
Client=D123
```

- **NoUserInfo** – управление выводом элемента UserInfo. Возможные значения:
 - «0» – элемент выводится;
 - «1» – не выводится.

Пример:

```
NoUserInfo=1
```



- **Status** – фильтр данных о транзакциях по статусу транзакции (каждый статус задаётся некоторым символом, см. п. 4). Может быть указан один или несколько статусов транзакций. Статусы перечисляются в любом порядке, без разделителей. Если ключ не задан, то данные выводятся вне зависимости от статуса транзакций.

Пример:

```
Status=rg
```

- **Reply** – фильтр данных по подстроке в тексте ответа сервера QUIK на отправленную транзакцию. Если фильтр не задан, то данные выводятся вне зависимости от ответа на транзакцию.

Пример:

```
Reply=Вам запрещена работа по данному торговому счету
```

- **Output** – имя файла для вывода данных. В зависимости от режима экспорта данных, заданного в параметре [ExportMode](#), значение параметра принимает следующий вид:
 - при `ExportMode=1` задается имя и, при необходимости, путь к XML-файлу. Если ключ не задан, то данные по транзакциям выводятся в создаваемый автоматически файл **trans_info.xml**, расположенный в текущем каталоге;
 - при `ExportMode=2` и `ExportMode=3` задается путь (при необходимости) и шаблон имени файла в формате «<Путь>\<Шаблон>», где: <Шаблон> – начало имени файла. При экспорте имя файла принимает формат «<Путь>\<Шаблон><Номер транзакции>.<расширение файла>», где <Номер транзакции> – номер транзакции, дополненный слева нулями до длины в 10 символов (например, «0001234567»).

Пример:

```
Output=TransReport.xml
```

- **Classes** – список классов бумаг, по которым разрешена передача информации о транзакциях, через запятую. Параметр не является обязательным.

Пример:

```
Classes=EQBR,EQNE
```

- **ExcludeClasses** – список классов бумаг, по которым запрещена передача информации о транзакциях, через запятую. Параметр не является обязательным.

Пример:



```
ExcludeClasses=PSEQ,PSES
```

Если списки `Classes` и `ExcludeClasses` не указаны, то выводится информация по всем классам бумаг. Если указаны оба списка, то список `ExcludeClasses` игнорируется.

- **EncodeNonChar** – изменение формата вывода данных о транзакции и её полях, если в транзакции присутствуют непечатные символы (с кодом, меньшим, чем 0x20 (пробел)). Возможные значения:

- «0» (по умолчанию) – формат не изменяется;
- «1» – формат изменяется.

Пример:

```
EncodeNonChar=1
```

- **CheckSign** – проверка наличия ЭЦП для каждой транзакции. Возможные значения:
- «0» (по умолчанию) – ЭЦП не проверяется;
 - «1» – ЭЦП проверяется. Если ЭЦП имеется, то выполняется проверка корректности ЭЦП для данной транзакции и определение ключа (сертификата), с помощью которого сформирована данная ЭЦП.

Пример:

```
CheckSign=1
```

- **SignOutputFile** – формирование ЭЦП для XML-файла на основе настроек из секции [FileSignSettings] (описание см. в п. [2.5.4](#)). Возможные значения:

- «0» (по умолчанию) – файл не подписывается;
- «1» – файл подписывается.

Пример:

```
SignOutputFile=1
```

- **Withsign** – вывод только транзакций с ЭЦП. Возможные значения:

- «0» (по умолчанию) – выводятся все транзакции;
- «1» – выводятся только транзакции с ЭЦП.



Пример:

```
Withsign=1
```

- **Withnosign** – вывод только транзакций без ЭЦП. Возможные значения:

- «0» (по умолчанию) – выводятся все транзакции;
- «1» – выводятся только транзакции без ЭЦП.

Пример:

```
Withnosign=0
```

Если параметры withsign и withnosign заданы одновременно, используется только параметр withsign.

- **TransFlags** – фильтр по полю Flags в описании транзакции (каждое значение задается определенным набором битов, см. п. [2.7.4](#)) в формате:

```
TransFlags=<набор флагов>[,<тип фильтра>]
```

где:

- <набор флагов> – флаги, которые должны присутствовать в описании транзакции. Набор флагов указывается битовой маской или перечислением флагов через запятую.
- <тип фильтра> – указывается по следующему правилу:
 - «any» – требуется наличие любого из указанных в фильтре флагов;
 - «exact» – допускается только полное соответствие перечисленных флагов;
 - «all» (по умолчанию) – требуется наличие всех указанных в фильтре флагов.

Пример:

Набор флагов указан битовой маской:

```
TransFlags=3,any
```

Набор флагов указан перечислением:

```
TransFlags=8192,134217728,2147483648,any
```



- **ExcludeTransFlags** – фильтр по полю Flags в описании транзакции (каждое значение задается определенным набором битов, см. п. [2.7.4](#)) в формате:

```
ExcludeTransFlags=<набор флагов>[,<тип фильтра>]
```

где:

- <набор флагов> – флаги, которые не должны присутствовать в описании транзакции. Набор флагов указывается битовой маской или перечислением флагов через запятую.
- <тип фильтра> – указывается по следующему правилу:
 - «any» – не допускается наличие любого из указанных в фильтре флагов;
 - «exact» – не допускается только полное соответствие перечисленных флагов;
 - «all» (по умолчанию) – не допускается наличие всех указанных в фильтре флагов.

Пример:

Набор флагов указан битовой маской:

```
ExcludeTransFlags=1024,any
```

Набор флагов указан перечислением:

```
ExcludeTransFlags=512,1024,any
```

- **BaseStopOrder** – управление выводом параметров BaseStopOrder и BaseStopOrderTime для базовой стоп-заявки, породившей данную транзакцию. Возможные значения:
 - «0» (по умолчанию) – не выводить;
 - «1» – выводить.

Пример:

```
BaseStopOrder=0
```

- **Securities** – список допустимых кодов инструментов, через запятую.

Пример:

```
Securities=RTKM,HYDR
```

- **ExcludeSecurities** – список игнорируемых кодов инструментов, через запятую.



Пример:

```
ExcludeSecurities=BUSB,YKEN
```

Если списки `Securities` и `ExcludeSecurities` не указаны, то выводится информация по всем инструментам. Если указаны оба списка, то список `ExcludeSecurities` игнорируется.

- **Firms** – список допустимых кодов фирм, через запятую.

Пример:

```
Firms= MC0080100000,NC0058900000
```

- **ExcludeFirms** – список игнорируемых кодов фирм, через запятую.

Пример:

```
ExcludeFirms=NC0038900000
```

Если списки `Firms` и `ExcludeFirms` не указаны, то выводится информация по всем фирмам. Если указаны оба списка, то список `ExcludeFirms` игнорируется.

- **OrderNum** – фильтр по номерам заявок / стоп-заявок в транзакции. Номера заявок / стоп-заявок в списке указываются через запятую.

Пример:

```
OrderNum=1203,1209,123456
```

- **ExchangeNum** – фильтр по биржевым номерам заявок, неторговых поручений в транзакции, в виде произвольной строки.

Пример:

```
ExchangeNum=987654,20150725 1234567,20150725 5678901
```

- **StopOrderExtraInfo** – управление выводом дополнительной информации о стоп-заявках. Возможные значения:

– «0» (по умолчанию) – не выводить;



- «1» – выводить.

При включенном признаке дополнительно выводятся следующие атрибуты:

- ResultOrderNum – номер выставленной заявки;
- LinkedOrder – номер связанной заявки;
- GeneratedTransNum – внутренний номер транзакции на выставление заявки.

Пример:

```
StopOrderExtraInfo=1
```

- **ExportMode** – режим экспорта данных по транзакциям. Возможные значения:

- «1» (по умолчанию) – экспорт данных в формате XML;
- «2» – экспорт данных в одном файле с расширением «.sgn». Сохраняется тело транзакции и, при наличии, ЭЦП;
- «3» – экспорт данных в трех файлах. Отдельными файлами с различными расширениями сохраняются:
 - «.data» – тело транзакции;
 - «.sign» – ЭЦП;
 - «.desc» – описание полей транзакции.

Путь к файлам вывода данных задается в параметре [Output](#).

При режимах экспорта «2» и «3» параметры UseXSL, NoUserInfo, EncodeNonChar, CheckSign, SignOutputFile, BaseStopOrder, StopOrderExtraInfo (и одноимённые параметры командной строки) – игнорируются.

- **TransNum** – фильтр по номерам транзакций. Номера транзакций в списке указываются через запятую.

Пример:

```
TransNum=641,679
```

- **TransInfoType** – фильтр по типу транзакций (каждый тип транзакции задаётся некоторым кодом, см. п. [5](#)). Типы транзакций указываются через запятую.

Пример:

```
TransInfoType=67,91
```



- **ExcludeTransInfoType** – исключающий фильтр по типу транзакций (каждый тип транзакции задаётся некоторым кодом, см. п. [5](#)). Типы транзакций указываются через запятую.

Пример:

```
ExcludeTransInfoType=67,91
```

Если заданы оба фильтра TransInfoType и ExcludeTransInfoType, то исключающий фильтр ExcludeTransInfoType игнорируется.

- **TransInfoExType** – фильтр по дополнительному типу транзакций (каждый дополнительный тип транзакции задаётся некоторым кодом, см. п. [6](#)). Дополнительные типы транзакций указываются через запятую.

Пример:

```
TransInfoExType=3,4
```

- **ExcludeTransInfoExType** – исключающий фильтр по дополнительному типу транзакций (каждый дополнительный тип транзакции задаётся некоторым кодом, см. п. [6](#)). Дополнительные типы транзакций указываются через запятую.

Пример:

```
ExcludeTransInfoExType=3,4
```

Если заданы оба фильтра TransInfoExType и ExcludeTransInfoExType, то исключающий фильтр ExcludeTransInfoExType игнорируется.

- **Language** – задаёт язык, который используется при экспорте описания транзакций и описания их полей. Значение задаётся без учёта регистра. Возможные значения:

- «ru» (по умолчанию) – русский;
- «en» – английский.

Если отсутствует описание транзакции или поля транзакции на указанном языке, то используется описание на другом языке.

- **DebugLevel** – уровень логирования событий приложения. Возможные значения: от «0» до «250», где:

- «0» – события не логируются;
- «1» (по умолчанию) – логируются события с наивысшим приоритетом;



- «250» – логируются все события.

2.5.3 Секция [SignSettings]

При использовании режима проверки ЭЦП в данной секции указывается тип используемого крипто-провайдера. Настройки крипто-провайдера хранятся в файле qcrypto.ini, в текущем каталоге.

- **CryptoType** – тип используемого крипто-провайдера. Возможные значения:
 - «SignalCom» – для ОС Linux используется только этот крипто-провайдер. Для ОС Windows используется только для целей совместимости, фактически применяется «SSLAndMessagePro»;
 - «SSLAndMessagePro»;
 - «BiCrypt»;
 - «CryptoPro»;
 - «MessagePro».

Пример:

```
CryptoType=SSLAndMessagePro
```

2.5.4 Секция [FileSignSettings]

В данной секции указываются настройки крипто-провайдера, используемого для формирования ЭЦП для XML-файла.

- **CryptoType** – тип используемого крипто-провайдера. Возможные значения:
 - «SignalCom» – для ОС Linux используется только этот крипто-провайдер. Для ОС Windows используется только для целей совместимости, фактически применяется «SSLAndMessagePro»;
 - «SSLAndMessagePro»;
 - «BiCrypt»;
 - «CryptoPro»;
 - «MessagePro».

Пример:

```
CryptoType=BiCrypt
```

При одновременном использовании режимов проверки и формирования ЭЦП для XML-файла настройки крипто-провайдера в секциях [SignSettings] и [FileSignSettings] должны совпадать.



2.6 Формат файла qcrypto.ini

Настройка крипто-провайдера требуется при **CheckSign=1** и **SignOutputFile=1**.

2.6.1 Секция [CryptoProviders]

Секция содержит список настроек вида:

- **<тип крипто-провайдера>** – признак использования крипто-провайдера <тип крипто-провайдера>. Возможные значения:
 - _ «0» – не используется;
 - _ «1» – используется.

2.6.2 Секция [CryptoProvidersSettings]

- **CipherCryptoProvider** – текущий тип крипто-провайдера.
- **SignCryptoProvider** – название текущего крипто-провайдера.

2.6.3 Секции вида [<тип крипто-провайдера>]

- **module** – путь к файлу библиотеки крипто-провайдера.
- **IniFile** – путь к файлу настроек крипто-провайдера.

2.6.4 Пример файла

```
[CryptoProviders]
BiCrypt=0
SignalComMessageAndSSLPro=0
CryptoPro=1
MessagePro=0

[CryptoProvidersSettings]
SignCryptoProvider=CryptoPro
CipherCryptoProvider=CryptoPro

[MessagePro]
Module=.\signalcom_pr.dll
IniFile=.\signalcom_pr.ini

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



```
[CryptoPro]
Module=.\CPCSP_Pr.dll
IniFile=.\cryptoCSP_Pr.ini

[BiCrypt]
Module=.\bicrypt_pr.dll
IniFile=.\bicrypt_pr.ini
```

2.7 Формат XML-файла экспорта

При создании XML-файла используется версия 1.0. При работе под Windows, файл сохраняется в кодировке windows-1251, а при работе под Linux в кодировке UTF-8. В файлах экспорта все поля и типы данных принадлежат пространству имён «urn:quik:trans-info:v1.0».

Файл содержит корневой узел типа **TransactionsReport** (подробнее об атрибутах узла типа TransactionsReport см. п. [2.7.1](#)). Внутри корневого узла может присутствовать произвольное количество узлов типа **Trans**. Каждый из таких узлов описывает отдельную поданную пользователем транзакцию (подробнее об атрибутах узла типа Trans см. п. [2.7.2](#)). Узел типа **Trans** содержит следующие элементы:

- **UserInfo** – если этот элемент присутствует, то он описывает пользователя, подавшего транзакцию (подробнее об атрибутах элемента UserInfo см. п. [2.7.3](#)).
- **Data** – содержит текст транзакции в виде текстовой строки. Пробелы являются значимыми символами.
- **DataEncoded** – содержит текст транзакции в закодированном виде base64.

Если включен режим EncodeNonChar (см. п. [2.5.2](#)), то в элемент Data записывается пустая строка, а текст транзакции записывается в элемент DataEncoded. Если режим EncodeNonChar выключен, то элемент DataEncoded отсутствует.

- **PureData** – если этот элемент присутствует, то он содержит полный текст транзакции, включая ЭЦП, включая ситуации, когда ЭЦП присутствует на транзакции, но на сервере проверка ЭЦП не настроена. Содержимое транзакции закодировано с помощью алгоритма base64. Пробелы и переносы строк игнорируются.
- **Reply** – если этот элемент присутствует, то он содержит ответ торговой системы на транзакцию. Пробелы являются значимыми символами.
- **Sign** – если этот элемент присутствует, то он содержит текст ЭЦП, полученный от клиента. При этом текст ЭЦП закодирован с помощью алгоритма base64. Пробелы и переносы строк игнорируются. Отсутствие элемента означает отсутствие полученного текста ЭЦП на транзакции.



- **TransData** – узел содержит описание транзакции данного типа и значения всех её полей. Содержит в себе атрибуты, описывающие транзакцию (подробнее об атрибутах элемента типа TransData см. п. [2.7.4](#)). Внутри узла типа **TransData** содержатся элементы типа **Field**, описывающие отдельные поля транзакции (подробнее об атрибутах элемента типа Field см. п. [2.7.5](#)).
- **BaseStopOrder** – если этот элемент присутствует, то он содержит номер стоп-заявки, на основании которой была порождена данная транзакция.
- **BaseStopOrderTime** – если этот элемент присутствует, то в нем указывается дата и время регистрации стоп-заявки, на основании которой была порождена данная транзакция, с точностью до микросекунд.

2.7.1 Атрибуты узла типа TransactionsReport

Наименование

атрибута	Тип	Значение
StartDate	date	Дата начала периода для экспорта данных
EndDate	date	Дата конца периода для экспорта данных
UID	integer	Если элемент присутствует, то в файл включены только те транзакции, которые были поданы пользователем с указанным идентификатором
ExcludeUID	string	Если элемент присутствует, то в файл включены все транзакции, кроме поданных пользователем с указанным идентификатором
ClientCode	string	Если элемент присутствует, то в файл включены только те транзакции, которые были поданы по указанному коду клиента. Если значение элемента оканчивается на символ «*», то в файл включены все транзакции по кодам клиентов, начинающихся на указанные символы (* трактуется как любая подстрока)
TransStatus	string	Если элемент присутствует, то в файл включены только те транзакции, которые имеют статус транзакции, совпадающий с указанным.
TransFlags	string	Тип фильтра в формате: «<тип списка>,<флаги>,<тип фильтра>», где все параметры записываются подряд (без пробелов) и имеют следующие значения: – <тип списка>: – «1» – заданы разрешенные типы транзакций; – «0» – заданы запрещенные типы транзакций; – <флаги> – флаги, указанные в фильтре (в шестнадцатеричном формате без ведущих нулей, без префикса «0x», но с постфиксом «h»); – <тип фильтра> – одно из значений «all», «any» или «exact» (в зависимости от указанного в фильтре).
SignAvailable	integer	Признак вывода транзакций с / без ЭЦП. Возможные значения: – «0» – выводятся только транзакции без ЭЦП; – «1» – выводятся только транзакции с ЭЦП.
OrderNum	int64	Если элемент присутствует, то в файл включены только те транзакции,



Наименование**атрибута****Тип****Значение**

		которые содержат указанный номер заявки
ExchangeNum	string	Если элемент присутствует, то в файл включены только те транзакции, которые содержат указанный биржевой номер заявки, неторгового поручения
TransInfoType	string	Если элемент присутствует, то в файл включены только те транзакции, которые удовлетворяют фильтру по типу транзакций. Фильтр указывается в формате: «<тип списка>,<тип транзакции>[,<тип транзакции>]», где все параметры записываются подряд (без пробелов) и имеют следующие значения: – <тип списка>: – «1» – заданы разрешенные типы транзакций; – «0» – заданы запрещенные типы транзакций; – <тип транзакции> – значение типа транзакции
TransInfoExType	string	Если элемент присутствует, то в файл включены только те транзакции, которые удовлетворяют фильтру по дополнительному типу транзакций. Формат фильтра аналогичен формату фильтра, задаваемому в атрибуте TransInfoType

2.7.2 Атрибуты узла типа Trans**Наименование****атрибута****Тип****Значение**

TransNum	integer	Внутренний номер транзакции в БД QUIK
UID	integer	ID пользователя, подавшего транзакцию
TransID	integer	ID транзакции в данной сессии
SessionID	integer	ID сессии, в которой была подана транзакция
TradeDate	datetime	Дата торгового дня
Status	string	Статус транзакции
QuikDate	datetime	Дата регистрации транзакции на сервере QUIK
QuikTime	datetime	Время, когда сервер QUIK получил транзакцию, с точностью до микросекунд
ReplyTime	datetime	Время, когда сервер QUIK получил ответ из торговой системы на транзакцию, с точностью до микросекунд
OrderNum	int	Если присутствует, то содержит номер в торговой системе, который был присвоен заявке, выставленной по этой транзакции



Наименование

атрибута	Тип	Значение
ExchangeNum	string	Если присутствует, то содержит биржевой номер заявки, присвоенный на западном рынке, либо номер неторгового поручения
ClientCode	string	Если присутствует, то содержит код клиента, по которому была подана транзакция
SignStatus	int	Статус проверки ЭЦП. Возможные значения: – «0» – ЭЦП используется, полученный текст ЭЦП верен. Данные по ключу (сертификату) указаны в атрибутах SignerCertSerialNumber, SignerCertSubject, SignerCertIssuer, SignerCertCryptoType; – «1» – ЭЦП присутствует на транзакции, на сервере настроена проверка ЭЦП, но ЭЦП не прошла проверку. Текст сообщения об ошибке указывается в атрибуте SignErrorMsg. Данные по ключу (сертификату) указаны в атрибутах SignerCertSerialNumber, SignerCertSubject, SignerCertIssuer, SignerCertCryptoType; – «2» – ЭЦП присутствует на транзакции, но на сервере не настроена проверка ЭЦП. Текст ЭЦП, полученный от клиента, указан в элементе Sign. Если ЭЦП отсутствует на транзакции, и на сервере не настроена проверка ЭЦП, то атрибут SignStatus не отображается
SignerCertSerialNumber	string	Серийный номер ключа (сертификата), которым создана ЭЦП на транзакции. Если ЭЦП неверна или серийный номер ключа (сертификата) не получен, то атрибут не указывается. Присутствует только при наличии корректной ЭЦП и включенном режиме проверки ЭЦП. Если данные по сертификату не найдены, то выводится значение «Unavailable»
SignerCertSubject	string	Данные о владельце ключа (сертификата), которым создана ЭЦП на транзакции. Если ЭЦП неверна или данные о владельце ключа (сертификата) не получены, то атрибут не указывается. Присутствует только при наличии корректной ЭЦП и включенном режиме проверки ЭЦП. Если данные по сертификату не найдены, то выводится значение «Unavailable»
SignerCertIssuer	string	Информация об удостоверяющем центре ключа (сертификата), которым создана ЭЦП на транзакции. Если ЭЦП неверна или информация об удостоверяющем центре ключа (сертификата) не получены, то атрибут не указывается. Присутствует только при наличии корректной ЭЦП и включенном режиме проверки ЭЦП. Если данные по сертификату не найдены, то выводится значение «Unavailable»
SignerCertCryptoType	string	Тип сертификата. Присутствует, если используется ЭЦП. Если данные по сертификату не найдены, то выводится значение «Unavailable»



Наименование

атрибута	Тип	Значение
*SignErrorMsg	string	Полученное из крипто-провайдера сообщение об ошибке. Присутствует, если при проверке ЭЦП произошла ошибка (в том числе, когда не удалось найти данные по ключу, которым создана ЭЦП).
LinkedOrder	integer	Номер связанной заявки
ParentTransID	integer	ID базовой транзакции при использовании дополнительных ограничений

* – полученное из крипто-провайдера сообщение об ошибке корректируется следующим образом: удаляются непечатные символы, переводы строк заменяются на пробелы, текст сообщения обрывается до 256 символов.

2.7.3 Атрибуты элемента типа UserInfo

Наименование

атрибута	Тип	Значение
Name1	string	Имя пользователя
Name2	string	Отчество пользователя
Name3	string	Фамилия пользователя
OrgCode	integer	Код организации, к которой относится пользователь
OrgName	string	Наименование организации, к которой относится пользователь
Login	string	Логин пользователя в другой системе. Используется при авторизации средствами RSA SecurID или Active Directory

2.7.4 Атрибуты элемента типа TransData

Наименование

атрибута	Тип	Значение
ClassID	integer	ID класса инструментов, к которому относится транзакция
ClassCode	string	Код класса инструментов, к которому относится транзакция
InfoType	integer	Тип транзакции
InfoExType	integer	Дополнительный тип транзакции. Используется для более точной идентификации типа транзакции



Наименование

атрибута	Тип	Значение
Description	string	Описание (название) транзакции
FieldsNum	integer	Количество полей в транзакции
Flags	integer	Флаги транзакции (внутренние флаги QUIK). Поле имеет вид битовой маски, значения битов см. в следующей таблице
ClassType	integer	Тип класса. Возможные значения: – «1» – Ценные бумаги; – «2» – Облигации; – «3» – Фьючерсы; – «4» – Опционы; – «5» – Спреды; – «6» – Свопы; – «7» – Валюта; – «8» – Стратегии; – «9» – Индексы; – «11» – Отчёты; – «12» – Поручения; – «13» – Авточартист; – «14» – Автоследование; – «16» – Алгоритмические заявки; – «17» – SMS оповещения; – «18» – Кросс-курсы валют; – «21» – Внебиржевые сделки; – «22» – Драгоценные металлы; – «23» – Служебные данные QUIK; – «24» – Внебиржевой рынок; – «25» – Депозиты; – «26» – Паи; – «27» – Обмен сообщениями; – Иначе – неизвестный тип
ClassSubType	integer	Подтип класса. Возможные значения: – Алгоритмические заявки: – «0» – не указан; – «1» – Айсберг; – «2» – Волатильность; – «3» – TWAP; – «4» – VWAP; – «5» – Заявка со сроком действия; – «6» – Стоп-заявка; – «7» – Заявка типа Spread; – «8» – неизвестный тип. – Свопы: – «0» – не указан; – «1» – Стандартный валютный своп; – «2» – Короткий валютный своп (однодневный); – «3» – Форвардный валютный своп; – «4» – неизвестный тип. – Спреды: – «0» – не указан; – «1» – Спреды на акции; – «2» – Спреды на фьючерсы; – «3» – Спреды на опционы; – «4» – Спреды на валютные инструменты; – «5» – неизвестный тип. – Служебные данные QUIK:



Наименование

атрибута Тип Значение

		– «4» – Информация о клиенте; – «5» – Справочники. – Обмен сообщениями: – «1» – Информационный канал; – «2» – Служба поддержки; – «3» – Группа сотрудников
ClassBehavior	integer	Признак на классе. Возможные значения: – «1» – Права по параметрам; – «2» – Не показывать погашение; – «4» – SOR; – «8» – Активные операции; – «16» – Расчет доходности в котировках; – «32» – Дополнительная синхронизация; – «128» – Класс РЕПО; – «256» – Класс РПС; – «512» – Код клиента может содержать «/»; – «4096» – Класс модифицированного РЕПО; – «134217728» – OMS заявки; – «64» – РЕПО с ЦК; – «4294967296» – Внешние алго-заявки; – «8589934592» – РПС с ЦК; – «1024» – Информационный класс; – «8192» – Класс участия в размещениях
ClassSpecialBehavior	integer	Специальный признак на классе. Возможные значения: – «1» – Количество в валюте; – «2» – Дробное количество

Значения битовых флагов транзакций поля Flags:

Флаг	Значение	Флаг	Значение
0	Контроль дополнительных ограничений	1024	Удаление фьючерсного лимита
1	Снятие заявки	2048	Снятие всех активных заявок
2	Снятие внебиржевой заявки	8192, 134217728 или 2147483648	Новая стоп-заявка
4	Новая заявка	16384	Снятие стоп-заявки
8	Новая внебиржевая заявка	32768	Новая заявка РЕПО
64	Неконкурентная заявка на размещение	65536	Заявка на вывод денежных средств
128	Новый отчет на исполнение	131072	Заявка на вывод бумаг
256	Снятие отчета на исполнение	262144	Новая безадресная заявка
512	Установка фьючерсного лимита		



Флаг	Значение
524288	Новая безадресная заявка РЕПО
1048576	Снятие безадресной заявки
8388608	Установка фьючерсного лимита на фирму

Флаг	Значение
8388612	Новая алго-заявка
16777216	Сделать стоп-заявку своей
67108864	Активация стоп-заявки

Битовые маски параметра Flags могут состоять из нескольких флагов, поэтому указанные значения битов должны проверяться на вхождение в значение, а не на его равенство.

2.7.5 Атрибуты элемента типа Field

Наименование атрибута	Тип	Значение
Number	integer	Номер поля транзакции
Name	string	Название поля транзакции
Description	string	Описание поля транзакции
Size	integer	Длина поля транзакции
Type	integer	Тип поля транзакции (см. п. 2)
Flags	integer	Флаги поля транзакции (внутренние флаги QUIK)
DefValue	string	Значение поля транзакции по умолчанию.
EnumCount	integer	Количество значений перечисляемого типа в поле
EnumValues	string	Описание значений перечисляемого типа для данного поля. Содержит алгоритм преобразования значений перечисляемого типа в строки
Value	string	Значения поля транзакции для рассматриваемой транзакции. Пусто – если включен режим EncodeNonChar и поле транзакции содержит непечатные символы
PreparedValue	string	Преобразованное значение поля транзакции: учтено количество знаков после запятой для чисел с плавающей точкой, значения перечисляемого типа преобразованы в строки. Пусто – если включен режим EncodeNonChar и поле транзакции содержит непечатные символы
ValueEncoded	string	Значения поля транзакции для рассматриваемой транзакции



Наименование атрибута	Тип	Значение
		в закодированном виде base64. Если поле не содержит непечатных символов, то ValueEncoded не указывается
LotSize	integer	Размер лота инструмента
Scale	integer	Масштаб, используемый для данного поля. Параметр полей типа «0», «2», «3», «6» (см. п. 7)
DateType	integer	Тип заданной даты. Возможные значения: – «default» – дата не задана; – «today» – в поле передаётся дата текущего дня; – «GTC» – в поле передаётся значение «до отмены». – «date» – в поле передаётся дата, значение которой содержится в атрибуте PreparedValue Используется только для полей типа «5» (см. п. 7)
InfoType	integer	Тип поля транзакции
InfoExType	integer	Дополнительный тип поля транзакции. Используется для более точной идентификации типа поля

3. Коды завершения приложения

Код возврата (errorlevel)	Описание
0	Ошибки в работе Программы не выявлены, транзакции выведены в файл
1	Ошибки в работе Программы не выявлены, транзакций, удовлетворяющих условию, не обнаружено
2	Ошибка доступа к базе данных
3	Ошибка данных в таблице транзакций (в том числе ошибка в описании сессии, ошибка в описании транзакции или поля транзакции)
4	Ошибка создания/записи XML-файла
5	Неверные входные данные
6	Прочие ошибки



4. Статусы транзакций

Флаг	Значение	Статус транзакции
TRANS_STATUS_RECEIVED	r	Получено сервером QUIK
TRANS_STATUS_GATE_FAULT	g	Ошибка связи с торговой системой
TRANS_STATUS_TRADE_OK	o	Транзакция успешно выполнена торговой системой
TRANS_STATUS_TRADE_FAULT	f	Торговая система вернула ошибку
TRANS_STATUS_QUIK_CHECK_FAULT	q	Транзакция не прошла проверку на сервере QUIK
TRANS_STATUS_LIMIT_CHECK_FAULT	l	Транзакция не прошла контроль лимитов
TRANS_STATUS_ACCEPTED	a	Транзакция подтверждена менеджером
TRANS_STATUS_REJECTED	j	Транзакция отклонена менеджером
TRANS_STATUS_KILLED	k	Транзакция снята пользователем
TRANS_STATUS_NOT_SUPPORTED	u	Транзакция не поддерживается сервером
TRANS_STATUS_VERBA_FAULT	v	Ошибка электронной подписи
TRANS_STATUS_CROSS_TRADE_REJECTED	c	Транзакция отвергнута из-за кросс-сделки
TRANS_STATUS_TRANS_COLLECT_FAILED	p	Транзакция получена не полностью
TRANS_STATUS_SOFT_LIMIT_CHECK_FAULT	s	Транзакция не прошла контроль дополнительных ограничений
TRANS_STATUS_ACCEPTED_AFTER_SOFT_LIMIT_CHECK_FAULT	b	Транзакция первоначально не прошла контроль дополнительных ограничений, исполнена



Флаг	Значение	Статус транзакции
TRANS_STATUS_STOPPED_AFTER_SOFT_LIMIT_CHECK	i	Транзакция первоначально не прошла контроль дополнительных ограничений, отменена

5. Типы транзакций

Значение	Тип транзакции
1	Подача стоп-заявки
67	Снятие заявки
91	Установка денежного ограничения на срочном рынке
133	Ввод адресной заявки РЕПО

6. Дополнительные типы транзакций

Значение	Дополнительный тип транзакции
3	Обычная стоп-заявка
4	Маклерская стоп-заявка
10002	Заявка с площадки WSE
10005	Заявка с площадки LSE

7. Типы полей транзакций

Тип поля транзакции	Значение
0	Строка символов. Длина строки символов определяется размером поля транзакции
1	Целое число, записанное в виде строки, может быть записано с некоторым масштабом (масштаб определяется отдельно)



Тип поля транзакции	Значение
2, 3	Вещественное число, записанное в виде строки, может быть записано с некоторым масштабом (масштаб определяется отдельно)
4	Перечисляемый тип. Если значение поля «ENUM_COUNT» равно нулю, то в качестве значения берётся содержимое поля, то есть часть текста транзакции, соответствующая текущему полю. Иначе осуществляется поиск значения поля среди списка расшифровок, задаваемого полем «ENUM_VALUES» (представляет из себя список, состоящий из элементов типа «ЗначениеПоля _i =Расшифровка _i », разделенных символом « »)
5	Дата в формате <ГГГГММДД>. Если число представлено не в указанном формате, то считается, что поле неопределенно
6	Вещественное число, записанное в виде строки с масштабом 6

8. Соответствия между элементами XML-файла и полями БД QUIK

Элемент xml-файла	Поле в таблице базы данных
Trans.TransNum	TRANS.TRANS_ID
Trans.UID	TRANS.USER_ID
Trans.TransID	TRANS.TRANS_TYPE
Trans.SessionID	TRANS.SESSIION_ID
Trans.TradeDate	SESSIONS.TRADEDATE
Trans.Status	TRANS.TRANS_STATUS
Trans.QuikTime	TRANS.SERVER_TIME
Trans.ReplyTime	TRANS.REPLY_TIME
Trans.OrderNum	TRANS.TS_NUM
Trans.ClientCode	TRANS.CLIENT_CODE
Trans.ParentTransID	TRANS.PARENT_TRANS_ID
UserInfo.Name1	PERSON.PERSON_FIRST_NAME
UserInfo.Name2	PERSON.PERSON_MIDDLE_NAME
UserInfo.Name3	PERSON.PERSON_LAST_NAME
UserInfo.OrgCode	ORGANIZATION.ORG_CODE



Элемент xml-файла	Поле в таблице базы данных
UserInfo.OrgName	ORGANIZATION.ORG_NAME
Data	TRANS.TRANS_DATA
PureData	TRANS.PURE_TRANS_DATA
Reply	TRANS.TRANS_REPLY_DATA
Sign	Текст подписи
TransData.ClassID	TRANS_DESC.CLASS_ID
TransData.ClassCode	SECURITY_CLASS.SECURITY_CLASS_CODE
TransData.Description	TRANS_DESC.TRANS_DESCRIPTION
TransData.FieldsNum	TRANS_DESC.FIELDS_NUM
TransData.Flags	TRANS_DESC.FLAGS
TransData.InfoType	TRANS_DESC.TRANS_INFO
TransData.InfoExType	TRANS_DESC.TRANS_INFO_EX
TransData.ClassType	SECURITY_CLASS.TYPE
TransData.ClassSubType	SECURITY_CLASS.SUBTYPE
TransData.ClassBehavior	SECURITY_CLASS.BEHAVIOR
TransData.ClassSpecialBehavior	SECURITY_CLASS.SPECIAL_BEHAVIOR
Field.Number	TRANS_FIELD_DESC.FIELD_NO
Field.Name	TRANS_FIELD_DESC.FIELD_NAME
Field.Description	TRANS_FIELD_DESC.FIELD_DESC
Field.Size	TRANS_FIELD_DESC.FIELD_SIZE
Field.Type	TRANS_FIELD_DESC.FIELD_TYPE
Field.Flags	TRANS_FIELD_DESC.FLAGS
Field.DefValue	TRANS_FIELD_DESC.DEFAULT_VALUE
Field.EnumCount	TRANS_FIELD_DESC.ENUM_COUNT
Field.EnumValues	TRANS_FIELD_DESC.ENUM_VALUES
Field.Value	Значение поля транзакции
Field.PreparedValue	Преобразованное значение поля транзакции



9. Преобразование формируемого XML-файла

Для представления формируемого в результате работы Программы **qt2xml** XML-файла в более удобный формат, используйте прилагаемую схему файла **trans.xsl**. Для преобразования XML-файла добавьте в полученный файл после первой строки следующую строку:

```
<?xml-stylesheet type="text/xsl" href="trans.xsl"?>
```

В результате при повторном открытии XML-файл будет преобразован в соответствии со схемой, содержащейся в файле **trans.xsl**.

Задать имя файла, схема которого будет использована при отображении XML-файла, можно в параметре [UseXSL](#) секции [General]. При формировании файла после первой строки вставляется строка с указанным в параметре значением:

```
<?xml-stylesheet type="text/xsl" href="значение параметра UseXSL"?>
```

10. Шифрование паролей

Пароли, задаваемые в настройках Программы, могут быть зашифрованы. Пароль не должен начинаться с символов «\$» и «#». Если используется пароль, который начинается с одного из указанных символов, то такой пароль следует изменить.

Для включения шифрования паролей добавьте в файле настроек qt2xml.ini символ «\$» перед текстом пароля.

Пример:

```
Password=$secret
```

При попытке использования пароля Программа зашифрует пароль и сохранит его обратно в файл настроек qt2xml.ini в зашифрованном виде. Зашифрованный пароль маркируется символом «#» в начале строки.

При переносе Программы на другой компьютер или переустановке операционной системы, зашифрованные пароли становятся недоступными для



дешифрования. В таком случае необходимо указать соответствующие пароли заново.

