



ФИНАМ

TRANSAQ XML Connector

Версия 6.16 билд 2.21.2

Содержание

Изменения	5
1. Общее описание.....	9
1.1 Функция Initialize	10
1.2 Функция InitializeEx.....	11
1.3 Функция SetLogLevel	12
1.4 Функция SendCommand.....	13
1.5 Функция SetCallback	14
1.6 Функция SetCallbackEx	14
1.7 Функция FreeMemory	15
1.8 Функция UnInitialize	15
2. Форматы данных	16
2.1 Типы данных.....	16
2.2 Числовые значения	16
2.3 Дата и время	17
2.4 Идентификация финансовых инструментов в командах и сообщениях.....	18
2.5 Формат команд.....	18
3. Список команд принимаемых коннектором	19
3.1 connect.....	19
3.2 disconnect	23
3.3 server_status	23
3.4 get_securities.....	23
3.5 subscribe.....	24
3.6 unsubscribe.....	25
3.7 get_history_data.....	26
3.8 neworder.....	27
3.9 newcondorder	29
3.10 newstoporder.....	31
3.11 cancelorder.....	37
3.12 cancelstoporder	37
3.13 get_forts_position.....	37
3.14 get_client_limits	37
3.15 get_markets.....	37



3.16 get_servtime_difference.....	38
3.17 change_pass	38
3.18 subscribe_ticks.....	38
3.19 get_connector_version	39
3.20 get_server_id	39
3.21 get_securities_info	39
3.22 moveorder	40
3.23 get_portfolio.....	41
3.24 get_max_buy_sell_tplus	41
3.25 get_united_portfolio.....	42
3.26 get_united_equity.....	42
3.27 get_united_go	42
3.28 get_cln_sec_permissions	43
4. Список возможных сообщений, передаваемых Callback функции	44
4.1 Исторические данные.....	44
4.2 Состояние сервера	45
4.3 Клиентские счета.....	47
4.4 Доступные рынки.....	48
4.5 Информация о доступных периодах свечей	48
4.6 Список инструментов.....	49
4.7 Информация по инструменту	52
4.8 Обновление информации по инструменту.....	54
4.9 Котировки по инструменту(ам)	55
4.10 Сделки по инструменту(ам)	57
4.11 Глубина рынка по инструменту(ам)	58
4.12 Заявка(и) клиента	60
4.13 Сделка(и) клиента	64
4.14 Позиции клиента	65
4.15 Лимиты клиента на срочном рынке.....	71
4.16 Клиентский портфель Т+	73
4.17 Возможность рыночных заявок.....	76
4.18 Тиковые данные	76
4.19 Справочник режимов торгов.....	77
4.20 Параметры инструмента в режиме торгов.....	77



4.21 Максимальная покупка/продажа для T+	77
4.22 Текстовые сообщения	78
4.23 Возможные ошибки	78
4.24 Клиентский единый портфель	79
4.25 Актуальная оценка ликвидационной стоимости Единого портфеля.....	81
4.26 Размер средств, заблокированных биржей (FORTS) под срочные позиции клиентов юниона	81
4.27 Юнионы, находящиеся в управлении клиента	81
4.28 Свойства инструмента, действующие в отношении конкретного клиента	82
5. Получение новостей в TXmlConnector	83
5.1 Спецификация команд	83
5.2 Список возможных сообщений, передаваемых callback функции.....	84
6. Приложения	86

Изменения

05.09.2019 | Версия 6.16 билд 2.21.2

- В структуру trades::trade добавлены поля bypass и venue

04.06.2019 | Версия 6.14 билд 2.21.1

- В структуре trade изменен тип поля quantity на integer64

14.09.2018 | Версия 6.12 билд 2.20.34

- Автоматическое создание директории для хранения файлов отчетов

04.07.2018 | Версия 6.09 билд 2.20.32

- Исправлена ошибка расчета MaxBuy

28.04.2018 | Версия 6.09 билд 2.20.31

- Максимальная длина кода MIC увеличена до 5 символов

09.04.2018 | Версия 6.09 билд 2.20.30

- Добавлен запрос get_cln_sec_permission
- Добавлены поля currency, InstrClass, MIC и ticker в структуру security

29.03.2018 | Версия 6.06 билд 2.20.29

- В структуру server_status добавлен билд сервера

08.02.2018 | Версия 6.06 билд 2.20.27, 2.20.26

- Исправлены найденные ошибки
- Исправлено отсутствие атрибута union в ответе для united_equity и united_go

24.08.17 | Версия 6.06 билд 2.20.25

- Добавлена функция InitializeEx
- В команде connect минимальное допустимое значение rqdelay изменено на 10 мс

23.06.17 | Версия 6.04 билд 2.20.23

- В описание инструмента <security> добавлен элемент <instrclass>

24.05.17 | Версия 6.03 билд 2.20.22

- Исправлена ошибка отсутствия закрывающего тэга asset в united_portfolio

03.05.17 | Версия 6.03 билд 2.20.21

- В структуру connect добавлен параметр language

14.03.17 | Версия 5.45 билд 2.20.19

- Исправлены некоторые ошибки

31.01.17 | Версия 5.45 билд 2.20.18

- В структуру united_portfolio добавлено поле vm_mma

12.01.17 | Версия 5.45 билд 2.20.17

- В структуры portfolio_tplus, united_portfolio добавлены поля балансовая цена (balance_prc), нереализованные прибыли (unrealized_pnl)

07.10.16 | Версия 5.43 билд 2.20.14

- Исправления в структуре sec_info:
 - экранирование невалидных в xml символов в элементе secname
 - put_call и opt_type теперь приходят только для опционов
 - исправлена ошибка форматирования элемента facevalue
 - исправлена ошибка форматирования bgo_deposit, sell_deposit, bgo_c, bgo_nc, bgo_buy

- Исправления в структуре sec_info_upd:
 - исправлена ошибка форматирования bgo_deposit, sell_deposit, bgo_c, bgo_nc, bgo_buy
- Исправлены ошибки в механизме автоматической отдачи текущих показателей Единого портфеля (connect::push_u_limits) и стоимости позиций (connect::push_pos_equity)
- Исправлена ошибка форматирования цены в структурах tick и trade

01.07.16 | Версия 5.43 билд 2.20.12

- Исправлена ошибка автоматической отдачи forts_position и forts_money в режиме autopos=true при изменениях вариационной маржи, которые не сопровождаются изменениями позиций

23.06.16 | Версия 5.41 билд 2.20.11

- Исправлен формат логирования команды connect в случае ошибок парсинга xml

21.06.16 | Версия 5.41 билд 2.20.10

- Исправлена ошибка: остановка работы библиотеки в случае вызова команды get_securities_info и остановки сервера

06.06.16 | Версия 5.41 билд 2.20.9

- Исправлена ошибка: в справочнике <markets> не отдавался рынок залогов

18.03.16 | Версия 5.41 билд 2.20.8

- MaxBuy и MaxSell в United Portfolio

29.02.16 | Версия 5.41 билд 2.20.7

- Багфиксинг команды get_history_data

20.02.16 | Версия 5.41 билд 2.20.6

- Багфиксинг и рефакторинг обработчика команды change_pass

09.02.16 | Версия 5.40 билд 2.20.5

- Исправлена ошибка обязательного элемента quantity при moveflag=0 в команде moveorder

04.02.16 | Версия 5.40 билд 2.20.4

- Исправлена ошибка отдачи структур quotations без подписки
- Исправлена ошибка формирования элемента accruedintvalue в структуре quotation

29.01.16 | Версия 5.40 билд 2.20.3

- Исправлено падение библиотеки в некоторых случаях отмены заявок

27.01.16 | Версия 5.40 билд 2.20.2

- Стоимость позиции, к которой применяются ставки рисков/резервов, теперь содержит в себе НКД

1. Общее описание

TXmlConnector.dll – загружаемая библиотека функций, предназначенных для организации взаимодействия пользовательских приложений с сервером TRANSAQ от имени и с правами доступа зарегистрированного торгового клиента (далее по тексту - «Коннектор»).

Коннектор экспортирует следующие функции:

```
BYTE* Initialize(const BYTE* logPath, int logLevel);  
BYTE* SetLogLevel(int logLevel).  
BYTE* SendCommand(BYTE* pData);  
bool SetCallback(tcallback pCallback);  
bool SetCallbackEx(tcallbackEx pCallbackEx, void* userData);  
bool FreeMemory(BYTE* pData);  
BYTE* UnInitialize();
```

Тип tcallback объявлен следующим образом:

```
typedef bool (*tcallback)(BYTE* pData);  
typedef bool (*tcallbackEx)(BYTE* pData, void*);
```

Обмен данными осуществляется посредством текстовых сообщений, представленных в виде XML структур, описываемых ниже.

Сообщения формируются в кодировке UTF-8.

1.1 Функция Initialize

BYTE* Initialize(const BYTE* logPath, int logLevel);

Выполняет инициализацию библиотеки: запускает поток обработки очереди обратных вызовов, инициализирует систему логирования библиотеки.

Параметры

<i>logPath</i>	Путь к директории, в которую будут сохраняться файлы отчетов
<i>logLevel</i>	Глубина логирования Предусмотрено три уровня логирования, в соответствии с детализацией и размером лог-файла: 1 – минимальный 2 – стандартный (рекомендуемый); 3 – максимальный

Данная функция в качестве аргументов принимает путь к папке (const BYTE* logPath), в которой будут созданы лог-файлы (XDF*.log, DSP*.txt, TS*.log), и уровень логирования (int logLevel).

logPath должен включать в себя завершающий символ "\" и заканчиваться на терминальный символ «\0». Пример:

```
logPath = "D:\\Logs\\0";
```

Функция Initialize может быть вызвана в процессе работы с Коннектором повторно для изменения директории и уровня логирования, но только в случае, когда библиотека остановлена, то есть была выполнена команда disconnect или соединение еще не было установлено.

Возвращаемое значение

В случае успешного выполнения функция возвращает 0. В случае ошибки возвращается указатель на сообщение об ошибке. Память, выделенная библиотекой под сообщение об ошибке, подлежит очистке с помощью функции FreeMemory.



Функция должна быть выполнена перед началом работы с библиотекой, то есть перед первой отправкой команды. Каждый успешный вызов функции Initialize должен сопровождаться вызовом функции UnInitialize.

1.2 Функция InitializeEx

BYTE* InitializeEx(const BYTE* data);

Аналогично функции Initialize выполняет инициализацию библиотеки. Отличие InitializeEx заключается в формате входных данных – параметры инициализации передаются в виде xml документа вида:

```
<init log_path = "" log_level="" logfile_lifetime=""/>
```

Параметры

<i>logPath</i>	Путь к директории, в которую будут сохраняться файлы отчетов
<i>logLevel</i>	Глубина логирования Предусмотрено три уровня логирования, в соответствии с детализацией и размером лог-файла: 1 – минимальный 2 – стандартный (рекомендуемый); 3 – максимальный
<i>logfile_lifetime</i>	Необязательный параметр. Время хранения файлов отчетов в днях (например: logfile_lifetime="7"). По умолчанию 7 дней.

Возвращаемое значение

В случае успешного выполнения функция возвращает 0. В случае ошибки возвращается указатель на сообщение об ошибке. Память, выделенная библиотекой под сообщение об ошибке, подлежит очистке с помощью функции FreeMemory.



Требования аналогичны функции Initialize. Вызовы функций InitializeEx и Initialize являются взаимоисключающими.

1.3 Функция SetLogLevel

BYTE* SetLogLevel(int logLevel)

Изменяет уровень логирования без остановки библиотеки.

Параметры

<i>logLevel</i>	Глубина логирования Предусмотрено три уровня логирования, в соответствии с детализацией и размером лог-файла: 1 – минимальный 2 – стандартный (рекомендуемый); 3 – максимальный
-----------------	---

Возвращаемое значение

В случае успешного выполнения функция возвращает 0. В случае ошибки возвращается указатель на сообщение об ошибке. Память, выделенная библиотекой под сообщение об ошибке, подлежит очистке с помощью функции FreeMemory.

1.4 Функция SendCommand

BYTE* SendCommand(BYTE* pData);

Служит для передачи команд Коннектору.

Параметры

pData Указатель на строку, содержащую xml команду для библиотеки TXmlConnector

Возвращаемое значение

Указатель на XML-сообщение, содержащее данные об успешности или неуспешности выполнения команды. В случае успеха сообщение имеет следующий вид:

```
<result success="true"/>
```

В случае неуспешного выполнения команды сообщение будет иметь вид:

```
<result success="false">  
    <message>error message</message>  
</result>
```

В случае исключительной ситуации может быть выдано сообщение:

```
<error> Текст сообщения об ошибке</error>
```

Память, выделенная библиотекой под XML- сообщение, подлежит очистке с помощью функции FreeMemory.



Функция может выполняться только в период между вызовами функций Initialize и UnInitialize.

1.5 Функция SetCallback

```
typedef bool (*tcallback)(BYTE* pData);  
typedef bool (*tcallbackEx)(BYTE* pData, void*);  
bool SetCallback(tcallback pCallback);  
bool SetCallbackEx(tcallbackEx pCallbackEx, void* userData);
```

Устанавливает указатель на функцию обратного вызова клиента, которая будет принимать асинхронные информационные сообщения от Коннектора. С помощью функции SetCallbackEx можно передать Коннектору указатель на данные пользователя (например, строку-идентификатор Коннектора, какой-либо пользовательский объект), который будет передаваться в callback функцию при каждом ее вызове. Ответственность за высвобождение памяти, аллоцированной Коннектором для размещения данных, лежит на пользователе Коннектора. Для этих целей следует использовать функцию FreeMemory.

Параметры

<i>pCallback</i>	Указатель на функцию обратного вызова клиента, имеющую сигнатуру <code>bool Callback(BYTE* data)</code>
<i>pCallbackEx</i>	Указатель на функцию обратного вызова клиента, имеющую сигнатуру <code>bool Callback(BYTE* data, void* user_data)</code>
<i>userData</i>	Указатель на данные пользователя, который передается в функцию обратного вызова клиента

Возвращаемое значение

В случае успешного выполнения функция возвращает true, в противном случае false.



В callback функции не должны выполняться операции, которые могут заблокировать поток на длительное время. Также из callback функции не следует вызывать управляющие функции библиотеки: Initialize, UnInitialize, SetCallback, SetCallbackEx, SendCommand, так как это может привести к взаимной блокировке потоков библиотеки.

1.6 Функция SetCallbackEx

С помощью функции SetCallbackEx можно передать Коннектору указатель на данные пользователя (например, строку-идентификатор Коннектора, какой-либо пользовательский объект), который будет передаваться в callback функцию при каждом ее вызове.

1.7 Функция FreeMemory

bool FreeMemory(BYTE* pData);

Выполняет очистку памяти, выделенной библиотекой

Параметры

pData Указатель на выделенную библиотекой память

Возвращаемое значение

В случае успешного выполнения функция возвращает true, в противном случае false.

1.8 Функция UnInitialize

BYTE* UnInitialize()

Выполняет остановку внутренних потоков библиотеки, в том числе завершает поток обработки очереди обратных вызовов. Останавливает систему логирования библиотеки и закрывает файлы отчетов.

Параметры

(нет)

Возвращаемое значение

В случае успешного выполнения функция возвращает 0. В случае ошибки возвращается указатель на сообщение об ошибке. Память, выделенная библиотекой под сообщение об ошибке, подлежит очистке с помощью функции FreeMemory.



Функция должна быть выполнена перед выгрузкой библиотеки из памяти.



В случае успешного выполнения функций *Initialize*, *UnInitialize*, *SetLogLevel* возвращается 0, иначе – указатель на XML-сообщение об ошибке (память, на которую указывает данный указатель, подлежит очистке с помощью функции *FreeMemory*).

Параметр *logLevel* в функциях *Initialize* и *SetLogLevel* определяет глубину расширенного логирования (запросов клиента и ответов коннектора).

2. Форматы данных

2.1 Типы данных

В настоящем руководстве для описания элементов структур используются следующие типы данных:

integer	– 32-х битное целое число;
integer64	– 64-х битное целое число;
double	– число двойной точности с плавающей точкой;
date	– дата и/или время (см. пункт 2.3).
string	– текстовое значение

Большинство параметров с типом string имеет длину, не превышающую 50 символов.

Исключения:

<sec_info><secname>	= 75 символов
<news_header><title>	= 128 символов
<news_body><text>	= Максимально возможное количество

Данные типа orderno и tradeno в биржевых системах ММВБ и РТС имеют тип integer64.

Значения поля <point_cost> иногда могут быть представлены в экспоненциальной записи – "2.5e-002"

2.2 Числовые значения

Разделителем между целой и дробной частями числовых значений является точка («.»).

2.3 Дата и время

Дата и время представляются в следующем формате:

"DD.MM.YYYY hh:mm:ss", где

DD	– день месяца начиная с 1 до 31
MM	– месяц в году, начиная с 1 до 12
YYYY	– год
hh	– час в сутках с 00 до 23
mm	– минуты в часу с 00 до 59
ss	– секунды в минуте 00 до 59
mmm	– миллисекунды (для некоторых элементов)

В некоторых случаях часть DD.MM.YYYY может отсутствовать, это означает что время указано на текущую дату.

Например, при работе с коннектором 27 Января.2011 получение в какой-либо структуре элемента

```
<Time>12:15:31</Time>
```

эквивалентно получению элемента:

```
<Time>27.01.2010 12:15:31</Time>
```

2.4 Идентификация финансовых инструментов в командах и сообщениях

Постоянными уникальными идентификаторами (ключами) финансовых инструментов являются:

- пара `seccode+board` для объектов, связанных с торгами (котировки, сделки, заявки и т.п.)
- пара `seccode+market` для клиентских позиций

Для использования данного типа идентификации добавлены новые форматы команды с использованием элемента `security`.

Также финансовые инструменты могут идентифицироваться целочисленными идентификаторами `secid` (не рекомендуется), неизменность которых гарантируется только в течение времени текущего подключения к серверу TRANSAQ. При использовании `secid` в командах подразумевается, что данная команда должна быть выполнена в так называемом "режиме торгов по умолчанию" (см. комментарий к элементу `<board>` структуры `<securities>`).

Идентификация бумаг по паре `seccode+board` позволяет избежать проблем, связанных с тем, что `secid` для бумаги может измениться при переподключении к серверу.

Кроме того, использование пары `seccode+board` позволяет явным образом задавать режим торгов в командах и работать в дополнительных режимах торгов ("Неполные лоты", "Крупные лоты" и т.д.)

2.5 Формат команд

При формировании команд важно учитывать, что все элементы и атрибуты команд чувствительны к регистру.

3. Список команд принимаемых коннектором

3.1 connect

Установить соединение с сервером:

```
<command id="connect">
  <login>user_id</login>
  <password>user_password</password>
  <host>server_ip</host>
  <port>server_port</port>
  <language>ru/en</language>
  <autopos>true/false</autopos>
  <micex_registers>true/false</micex_registers>
  <milliseconds>true/false</milliseconds>
  <utc_time>true/false</utc_time>
  <proxy type="тип" addr="адрес" port="порт" login="логин"
  password="пароль"/>
  <rqdelay>Частота запросов к серверу</rqdelay>
  <session_timeout>Таймаут на сессию в секундах</session_timeout>
  <request_timeout>Таймаут на запрос в секундах
  </request_timeout>
  <push_u_limits>Период в секундах</push_u_limits>
  <push_pos_equity>Период в секундах</push_pos_equity>
</command>
```

Немедленным результатом на команду connect будет результат синтаксического разбора запроса. После этого в асинхронном режиме будет либо передано сообщение об ошибке подключения, либо будет предоставлена информация о доступных периодах исторической ценовой информации – свечей/баров (candlekinds) и рынках (markets). Когда все данные будут загружены, будет получена структура <server_status connected="true"/>.

Результатом является структура server_status.

В случае успешного выполнения команды connect повторная посылка команды возможна только после отключения от сервера с помощью команды disconnect.

Информация о финансовых инструментах (securities) может быть получена после появления соответствующей информации на сервере (после 9:50 мск).

<i>language</i>	Параметр language задает язык системных сообщений.						
<i>autopos</i>	Параметр autopos указывает на необходимость автоматического запроса клиентских позиций на срочном рынке после каждой клиентской сделки. Если autopos не указан, по умолчанию он принимается равным true. Использование <autopos>>false</autopos> при активной торговле ускоряет взаимодействие с сервером.						
<i>proxy</i>	<p>Тэг proxy следует включать только тогда, когда есть необходимость соединяться через прокси-сервер. Аналогично атрибуты login и password используются только тогда, когда имеется необходимость авторизации на сервере. В противном случае не нужно указывать их вообще.</p> <p>Типы прокси-сервера доступны следующие: SOCKS4, SOCKS5, HTTP-CONNECT (задаются именно в таком виде, с учетом регистра и дефиса).</p>						
<i>rqdelay</i>	<p>Rqdelay задает частоту обращений Коннектора к серверу Transaq в миллисекундах. Минимальные значения:</p> <table><tr><td>Торговый сервер Transaq</td><td>10</td><td>мс</td></tr><tr><td>Торговый сервер HFT Transaq</td><td>10</td><td>мс</td></tr></table>	Торговый сервер Transaq	10	мс	Торговый сервер HFT Transaq	10	мс
Торговый сервер Transaq	10	мс					
Торговый сервер HFT Transaq	10	мс					
<i>session_timeout</i>	Session_timeout – интервал времени, в течении которого коннектор в случае ошибок связи будет выполнять попытки переподключения к серверу без повторной загрузки информации о доступных периодах исторической ценовой информации – свечах/барах (candlekinds), финансовых инструментах (securities) и рынках (markets). Если данный параметр не будет указан, используется значение по умолчанию равное 120 секундам.						
<i>request_timeout</i>	Request_timeout - таймаут на выполнение запроса. Если данный параметр не будет указан, используется значение по умолчанию равное 20 секундам. Значение параметра session_timeout должно быть больше значения параметра request timeout.						

Параметры session_timeout и request_timeout не регулируют timeout в случае недоступности сервера. Минутный таймаут в данном случае не зависит от коннектора и является результатом работы внутренних функций операционной системы, отвечающих за работу с сетью.

micex_registers Значение элемента *micex_registers* определяет набор данных, передаваемый в структурах *<money_position>* и *<sec_position>* (см. раздел "Позиции клиента").

milliseconds Значение элемента *milliseconds* определяет формат элементов типа "Дата и время" в некоторых структурах. Если задано *<milliseconds>true</milliseconds>*, то следующие данные отдаются в формате "DD.MM.YYYY hh:mm:ss.mmm":

- элемент *time* структуры *alltrade*
- элемент *time* структуры *ticks*
- элемент *time* структуры *trade*
- элементы *time* и *withdrawtime* структуры *order*

utc_time Значение элемента *utc_time* определяет таймзону некоторых элементов "Дата и время". Если задано *<utc_time>true</utc_time>*, то следующие элементы передаются в UTC:

- элемент *time* структуры *alltrade*
- элемент *tradetime* структуры *tick*
- элемент *time* структуры *trade*
- элемент *time* структуры *quotation*
- элемент *date* структуры *candle*
- элементы *time* и *withdrawtime* структуры *order*
- элемент *conditionvalue* (если *condition="Time"* и в нем задано дата-время)
- элементы *accepttime*, *validafter* и *validbefore* структуры *order*
- элементы *validbefore*, *withdrawtime* и *accepttime* структуры *stoporder*
- элемент *date* структуры *message*
- элемент *time_stamp* структуры *news_header*

Значение *utc_time* НЕ влияет на следующие структуры

- элементы *mat_date*, *coupon_date* структуры *sec_info*
- *expdate* структуры *order*

Если задано *<utc_time>true</ utc_time>*, то следующие элементы необходимо указывать в UTC:

- параметры *validafter*, *validbefore* и *cond_value* команды *newcondorder* (если они заданы и не заданы спец.значения *validafter=0* или *validbefore=0*)
- параметр *validfor* команды *newstoporder* (если не задано спец.значения *validfor=0*)

Независимо от значения *<utc_time>*:

- в структуру *securities* добавляется элемент *sec_tz*,

содержащий имя таймзоны инструмента (типа "Russian Standard Time", "USA=Eastern Standard Time")

- в структуре `server_status` добавляется элемент `server_tz`, содержащий имя таймзоны сервера

push_u_limits Наличие элемента `push_u_limits` является инструкцией обеспечивать для каждого юниона информирование пользователя о текущих показателях Единого портфеля - всякий раз при возникновении существенных событий (изменение состояния заявок или сделок клиентов юниона), но не реже чем один раз в N секунд.

Допустимыми значениями элемента `push_u_limits` являются натуральные десятичные числа.

Если значение элемента равно нулю `<push_u_limits>0</push_u_limits>`, это эквивалентно отсутствию элемента `<push_u_limits>`.

push_pos_equity Наличие элемента `push_pos_equity` является инструкцией раз в N секунд информировать пользователя о текущей стоимости позиций, передавая ему массив структур `<positions><sec_position>`, соответствующих множеству удерживаемых клиентами позиций (за исключением позиций FORTS).

Допустимыми значениями элемента `push_pos_equity` являются натуральные десятичные числа.

Если значение элемента равно нулю `<push_pos_equity>0</push_pos_equity>`, это эквивалентно отсутствию элемента `<push_pos_equity>`.

3.2 disconnect

Разорвать соединение.

```
<command id="disconnect"/>
```

Результатом является структура `server_status`.

Данная команда является синхронной, т.е. при ее выполнении `SendCommand` вернет результат только после того, как коннектор отключится от сервера. При этом через `callback` функцию будет отдана структура `<server_status>` со значением `connected=false`.



Если в процессе работы коннектора подключение к серверу будет потеряно (при этом приходит структура `<server_status connected=false/>`), то вызов команды `disconnect` перед новым подключением с помощью команды `connect` - не требуется.

3.3 server_status

Получить текущее состояние соединения с сервером.

```
<command id="server_status"/>
```

Результатом является структура `"server_status"`.

3.4 get_securities

Получить список доступных инструментов.

```
<command id="get_securities"/>
```

Результатом является структура `"securities"`.

3.5 subscribe

Подписаться на получение котировок, сделок и глубины рынка (стакана) по одному или нескольким инструментам.

Формат команды с использованием элемента security для идентификации бумаг:

```
<command id="subscribe">
  <alltrades> - подписка на сделки рынка
    <security>
      <board> идентификатор режима торгов</board>
      <seccode>код инструмента</seccode>
    </security>
    ...
  </alltrades>
  <quotations> - подписка на изменения показателей торгов
    <security>
      <board> идентификатор режима торгов</board>
      <seccode>код инструмента</seccode>
    </security>
    ...
  </quotations>
  <quotes> - подписка на изменения «стакана»
    <security>
      <board> идентификатор режима торгов</board>
      <seccode>код инструмента</seccode>
    </security>
    ...
  </quotes>
</command>
```

Результатом работы является начало передач всех изменений по заданным бумагам путем передачи сообщений quotations, alltrades, quotes.

3.6 unsubscribe

Прекратить получение котировок, сделок и глубины рынка (стакана) по одному или нескольким инструментам.

Формат команды с использованием элемента security для идентификации бумаг:

```
<command id="unsubscribe">
  <alltrades> - подписка на сделки рынка
    <security>
      <board> идентификатор режима торгов </board>
      <seccode>код инструмента</seccode>
    </security>
    ...
  </alltrades>
  <quotations> - подписка на изменения показателей торгов
    <security>
      <board> идентификатор режима торгов </board>
      <seccode>код инструмента</seccode>
    </security>
    ...
  </quotations>
  <quotes> - подписка на изменения «стакана»
    <security>
      <board> идентификатор режима торгов </board>
      <seccode>код инструмента</seccode>
    </security>
    ...
  </quotes>
</command>
```

Результатом работы является прекращение передач всех изменений по заданным бумагам путем передачи сообщений quotations, alltrades, quotes.

3.7 get_history_data

Выдать последние N свечей заданного периода, по заданному инструменту.

Формат команды с использованием элемента security для идентификации бумаг:

```
<command id="gethistorydata">
  <security>
    <board> идентификатор режима торгов </board>
    <seccode> код инструмента </seccode>
  </security>
  <period>идентификатор периода</period>
  <count>количество свечей</count>
  <reset>true/false</reset>
</command>
```

Параметр reset="true" говорит, что нужно выдавать самые свежие данные, в противном случае будут выданы свечи в продолжение предыдущего запроса.

Результатом работы является структура вида candles.

Возможные значения для period присылаются при установке соединения с сервером в полях candlekinds (поле id).



В period необходимо указать нужное значение поля <id> из структуры <candlekinds>.

3.8 neworder

Подать новую заявку на биржу

```
<command id="neworder">
  <security>
    <board> идентификатор режима торгов </board>
    <seccode> код инструмента </seccode>
  </security>
  <client>клиент</client>
  <union>union code</union>
  <price>цена</price>
  <hidden>скрытое количество в лотах</hidden>
  <quantity>количество в лотах</quantity>
  <buysell>B</buysell>
  ("B" - покупка, или "S" - продажа)
  <bymarket/>
  <brokerref>примечание</brokerref>
  (будет возвращено в составе структур order и trade)
  <unfilled>PutInQueue</unfilled>
  (другие возможные значения: FOK, IOC)
  <usecredit/>
  <nosplit/>
  <expdate>дата экспирации (только для ФОРТС)</expdate>
  (задается в формате 23.07.2012 00:00:00 (не обязательно))
</command>
```

Для идентификации инструмента необходимо задать элемент <security>.

Usecredit, nosplit и bymarket должны быть заданы пустым тегом, либо отсутствовать вообще. При наличии тега bymarket, тег price игнорируется и может отсутствовать. Поле hidden является необязательным.

Для инструментов MMA Usecredit недоступен. Его нельзя указывать.

Так как в TC FORTS не предусмотрены заявки без цены, то рыночные заявки для фьючерсов эмулируются с помощью лимитированных следующим образом: заявки на покупку подаются по максимально возможной цене сессии, а заявки на продажу - по минимально возможной. Для таких заявок также автоматически устанавливается признак "Снять остаток".

Для опционов рыночные заявки не принимаются.

Значения unfilled:

PutInQueue: неисполненная часть заявки помещается в очередь заявок Биржи.

FOK: сделки совершаются только в том случае, если заявка может быть удовлетворена полностью.

IOC: неисполненная часть заявки снимается с торгов

Немедленным результатом запроса neworder будет структура <result>, сообщающая о том, была ли принята заявка сервером TRANSAQ (не Биржей!). В случае успеха результатом будет следующий тег:

```
<result success="true" transactionid="id"/>
```

Transactionid - это целочисленный номер, который однозначно идентифицирует транзакцию внутри текущего сеанса связи с сервером.

После того, как в результате транзакции на Бирже появится соответствующий объект (заявка), основным идентификатором этого объекта становится регистрационный номер Биржи.

Transactionid является также основным идентификатором условной заявки, являющейся результатом транзакции, до тех пор, пока заявка не будет выставлена на Биржу и приобретёт регистрационный номер Биржи.

После того, как заявка будет принята Биржей, в асинхронном режиме будет получена структура <order>, имеющая в составе данных ссылку на номер транзакции.

Иногда бывает так, что ордер приходит от Биржи только через какое-то время после того как Биржа примет транзакцию по выставлению заявки. В этом случае Коннектор сначала пришлет активный ордер с нулевым <orderno>, а после получения ордера от Биржи пришлет ордер с биржевым номером (и возможно с новым статусом, например, "исполнен"). Из этого следует, что внешнее приложение должно интерпретировать активный ордер с нулевым <orderno> как "заявка выставляется на Биржу" (по аналогии с тем, что при выполнении cancelorder значение <status> cancelled и нулевое значение элемента <withdrawtime> означает, что заявка находится в процессе снятия).



В условных заявках <orderno> будет нулевым до тех пор, пока не исполнится условие (и заявка не будет отправлена на Биржу)!



Brokerref позже будут возвращены в структурах order и trade.

Максимальная длина в символах Brokerref:

КЦБ ММВБ max = 12 минус clientcode

например код клиента 44567/44567 макс. размер примечания 1 символ

ГЦБ ММВБ max = 7

RTS CFK max = 12 минус clientcode
FOB MMB max = 12 минус clientcode
FORTS max = 13
XETRA max = 0



Только для FORTS. На рынке FORTS не доступны параметры usecredit и nosplit. Также для параметра unfilled не доступно значение FOK. Для опционов также не доступны рыночные заявки.

3.9 newcondorder

Выставить новую условную заявку на Сервер TRANSAQ

```

<command id="newcondorder">
  <security>
    <board> идентификатор режима торгов </board>
    <seccode> код инструмента </seccode>
  </security>
  <client>клиент</client>
  <union>union code</union>
  <price>цена</price>
  <hidden>скрытое количество в лотах</hidden>
  <quantity>количество в лотах</quantity>
  <buysell>"B" - покупка, "S" - продажа</buysell>
  <bymarket/>
  <brokerref>примечание</brokerref>
  <cond_type>тип условия (см. ниже)</cond_type>
  <cond_value>значение</cond_value>
  <validafter></validafter>
  <validbefore></validbefore>
  <nosplit/>
  <usecredit/>
  <expdate>дата экспирации (только для ФОРТС)</expdate>
  (задается в формате 23.07.2012 00:00:00 (не обязательно))
</command>

```

Для идентификации инструмента необходимо задать элемент <security>.

Validafter и validbefore задаются в форме даты, описанном выше. Для validafter можно передать значение "0", если заявка начинает действовать немедленно. Для validbefore значение "0" означает, что заявка будет действительна до конца сессии. Так же validbefore может принимать текстовое значение "till_canceled", которое говорит о том, что заявка будет актуальна пока ее не снимут.

Допустимые типы условия:

Bid	= лучшая цена покупки
BidOrLast	= лучшая цена покупки или сделка по заданной цене и выше
Ask	= лучшая цена продажи
AskOrLast	= лучшая цена продажи или сделка по заданной цене и ниже
Time	= время выставления заявки на Биржу
CovDown	= обеспеченность ниже заданной
CovUp	= обеспеченность выше заданной
LastUp	= сделка на рынке по заданной цене или выше
LastDown	= сделка на рынке по заданной цене или ниже

Принцип работы запроса newcondorder тот же, что и запроса neworder. После запроса возвращается либо сообщение об ошибке, либо структура <result> с номером транзакции. Заявка при этом не попадает сразу на биржу, а дожидается наступления условия. В отдаваемой асинхронно структуре order заявка будет иметь статус watching. Как только условие будет выполнено, заявка будет передана на биржу, в результате чего callback будет вызван еще раз со структурой order, которая отразит биржевой номер заявки и ее характеристики. Элемент <orderno> будет нулевым до тех пор, пока не исполнится условие (и заявка не будет отправлена на Биржу).

Поле hidden является необязательным.

3.10 newstoporder

Выставить стоп-заявку:

```
<command id="newstoporder">
  <security>
    <board> идентификатор режима торгов </board>
    <seccode> код инструмента </seccode>
  </security>
  <client>идентификатор клиента</client>
  <union>union code</union>
  <buysell>B/S</buysell>
  <linkedorderno>номер связной заявки</linkedorderno>
  (при отсутствии тэга без привязки)
  <validfor>заявка действительно до</validfor>
  (не обязательно)
  <expdate>дата экспирации (только для ФОРТС)</expdate>
  (не обязательно)
  <stoploss>
    <activationprice>Цена активации</activationprice>
    <orderprice>Цена заявки</orderprice>
    <bymarket/> - Выставить заявку по рынку
    (в этом случае orderprice игнорируется)
    <quantity>Объем</quantity>
    <usecredit/> - использование кредита
    <guardtime>Защитное время</guardtime>
    (не обязательно)
    <brokerref>Примечание брокера</brokerref>
    (не обязательно)
  </stoploss>
  <takeprofit>
    <activationprice>Цена активации</activationprice>
    <quantity>Количество</quantity>
    <usecredit/> - использование кредита
    <guardtime>Защитное время</guardtime>
    (не обязательно)
    <brokerref>Примечание брокера</brokerref>
    (не обязательно)
    <correction>Коррекция</correction>
    <spread>Защитный спрэд</spread>
    <bymarket/>
  </takeprofit>
</command>
```

Защитный спрэд, объем quantity для stop loss и коррекцию можно задавать как в абсолютной величине, так и в процентах (от цены либо от позиции клиента по смыслу). Для задания процентов, достаточно поставить после числа символ '%', например:

```
<quantity>10%</quantity>
```

Существенно, для стоп-заявок brokerref не сохраняется локально в notes_file, и таким образом для стопов допустимы только короткие brokerref.

Expdate – параметр типа :date, который в составе заявки передается в ТС FORTS. Биржа будет перевыставлять эту заявку в своей системе до наступления указанной даты.

Параметр validfor может принимать значения:

0	–	до конца торговой сессии
till_canceled	–	до отмены
ДД.ММ.ГГГГ ЧЧ:ММ:СС	–	до указанной даты и времени

3.10.1 Работа со стоп-заявками

Стоп-заявка в системе TRANSAQ (или стоп) — это распоряжение, предписывающее серверу TRANSAQ выполнить в автоматизированном режиме транзакционные действия, реализующие логику стоп-лосс или тейк-профит в зависимости от ценовых условий на рынке.

Основное предназначение стоп-заявок — автоматизировать управление существующей или потенциальной позицией клиента, которое заключается в своевременном выполнении одного из следующих действий:

- либо при неблагоприятном изменении цены своевременно закрыть позицию, ограничив убытки заранее определённой величиной (исполнить стоп-лосс);
- либо зафиксировать прибыль после достижения ценой достаточно прибыльного уровня при возникновении признаков окончания благоприятного тренда (исполнить тейк-профит).

Стоп-заявки могут также с успехом применяться для автоматизированного открытия новой позиции при определённых условиях, например, при возникновении признаков смены тренда, или наоборот, для открытия позиции «по тренду».

Стоп-заявка состоит из двух частей:

- Стоп-лосс (далее SL)
- Тейк-профит (далее TP)

При выполнении условия для одной части стоп-заявки, вторая ее часть снимается.



Вторая часть стоп-заявки снимается даже в том случае, если попытка выставить заявку на Биржу при исполнении оказалась unsuccessful. Допускаются стоп-заявки, содержащие только одну часть (SL или TP).



Для закрытия коротких позиций следует выставлять стопы на покупку, для закрытия длинных позиций - стопы на продажу.

Обе части стоп-заявки допускают использование признака «по рынку».

3.10.2 Стоп-лосс

SL предназначен для закрытия позиций с целью ограничения убытков от удержания позиции при неблагоприятном движении цены на рынке. Следовательно,

- для стопа на продажу часть SL активируется, когда цена на рынке станет меньше либо равна цене активации стоп-лосса.
- для стопа на покупку часть SL активируется, когда цена на рынке станет больше либо равна цене активации стоп-лосса.

При выставлении SL необходимо задать Цену активации и Цену заявки, которая будет отправлена на Биржу при достижении рынком цены активации.

Использование необязательного параметра Защитное время позволяет предотвратить исполнение стопа при «проколах» на рынке, т.е. в таких ситуациях, когда цены на рынке лишь кратковременно достигают уровня цены активации, и вскоре возвращаются обратно. В частности, такие явления наблюдаются при интервенциях или ошибках крупных участников, когда на рынке внезапно регистрируется значительное отклонение цен произошедших сделок, но реальной возможности совершить сделки по таким ценам на рынке не возникает.

Защитное время задаётся в целых секундах и отсчитывается от времени получения сервером первой сделки, удовлетворяющей цене активации. Если в течение защитного времени на рынке будут зафиксированы сделки, не подтверждающие наступление уровня цены активации, (т.е. сделки по цене выше цены активации для SL на продажу, либо по цене ниже цены активации для SL на покупку) то стоп-лосс возвращается в состояние «Ожидает активации».

Если защитное время не задано, то даже одна сделка на рынке по цене, удовлетворяющей цене активации, приводит к исполнению стоп-лосса.

3.10.3 Тейк-профит

TP предназначены для закрытия позиций с фиксацией прибыли. Следовательно,

- для стопа на продажу часть TP активируется, когда цена на рынке станет больше либо равна цене активации.
- для стопа на покупку часть TP активируется, когда цена на рынке станет меньше либо равна цене активации.

3.10.3.1 Простой TP

Для ввода простого TP достаточно задать Цену активации и Количество. В этом случае на Биржу будет отправлена заявка с ценой, равной цене первой же сделки на рынке, которая удовлетворяет цене активации.

Можно увеличить вероятность совершения сделки при исполнении стопа, задав Защитный спрэд. Для определения цены заявки, исполняющей TP на покупку, защитный спрэд прибавляется к цене рынка. Для определения цены заявки, исполняющей TP на продажу, защитный спрэд вычитается из цены рынка.

Для простого TP также можно установить защитное время (см. выше описание SL).



При использовании защитного времени, в качестве цены рынка (от которой определяется цена заявки исполняющей TP) принимается цена последней по времени сделки, полученной сервером в течение защитного времени.

3.10.3.2 Следящий (trailing) TP

Позволяет выставить на Биржу заявку, закрывающую позицию в момент появления признаков окончания благоприятного тренда на рынке.

Для того, чтобы включить механизм отслеживания тренда, необходимо при выставлении TP задать значение поля Коррекция.

Это значение используется сервером TRANSAQ для определения момента окончания благоприятного тренда следующим образом:

- для TP на продажу считается, что растущий тренд заканчивается в тот момент, когда после того, как рынок вырос до уровня цены активации или выше, он снизится на величину коррекции от максимальной цены.
- для TP на покупку считается, что нисходящий тренд заканчивается в тот момент, когда после того, как рынок снизился до уровня цены активации или ниже, он вырастет на величину коррекции от минимальной цены.

Величина коррекции может быть задана как в виде абсолютного изменения цены, так и в виде процента от цены. Для следящего TP также могут быть заданы Защитный спрэд и/или Защитное время.

Если для следящего TP задано защитное время, то оно действует как при определении факта достижения цены активации, так и при определении факта окончания тренда (достижения заданного уровня коррекции).

3.10.4 Связанные заявки

Любая стоп-заявка может иметь «связь по исполнению» с активной заявкой на бирже. Такая связь означает, что действие стоп-заявки начнётся не с момента её ввода в TRANSAQ, а в тот момент, когда соответствующая активная заявка исполнится (или частично исполнится) на Бирже. «Связь по исполнению» удобно применять в тех случаях, когда необходимо обеспечить стоп-процессинг для позиции, которая ещё не открыта, но может быть открыта при исполнении активной заявки.

Несколько стоп-заявок могут иметь связь по исполнению с одной и той же активной заявкой. Это позволяет организовать ступенчатое закрытие позиции частями на разных ценовых уровнях в заданных пропорциях.

При снятии активной заявки до её (полного или частичного) исполнения на Бирже все стоп-заявки, имеющие с ней связь по исполнению, также автоматически будут сняты.

При вводе связанного стопа его направление (покупка/продажа) устанавливается противоположным направлению активной заявки, поля

Инструмент, Режим и Клиент также копируются из активной заявки, но при необходимости могут быть изменены.

Если в момент ввода связанного стопа заявка уже исполнена (в т.ч. частично), то он сразу переходит в состояние ожидания активации.

Связь по исполнению может быть использована как способ автоматизации торговых операций в одном финансовом инструменте по условию цены в другом инструменте. Выставляется «триггерная» заявка на минимальный объём в индикативном инструменте, и к ней привязывается стоп по исполнению, открывающий или закрывающий позицию в торговом инструменте. К этой же «триггерной» заявке можно привязать стоп, который автоматически закроет позицию, возникшую при ее исполнении.

3.11 cancelorder

Отменить заявку

```
<command id="cancelorder">  
    <transactionid>номер из структуры orders</transactionid>  
</command>
```

Результатом работы является отмена (снятие) заявки. Если произойдут какие-либо ошибки (заявка уже исполнена, уже снята, некорректный номер и т. д.), об этом будет сообщено в возвращаемом теге <result>.

3.12 cancelstoporder

Отменить стоп-заявку

```
<command id="cancelstoporder">  
    <transactionid>номер из структуры orders</transactionid>  
</command>
```

3.13 get_forts_position

Запрос позиций клиента по FORTS:

```
<command id="get_forts_positions" client="клиент"/>
```

Запрашивает у сервера позиции клиента на FORTS. Если не указать клиента, запрос будет выполнен по всем доступным клиентам.

3.14 get_client_limits

Запрос лимитов клиента на срочном рынке.

```
<command id="get_client_limits" client="клиент" />
```

3.15 get_markets

Запросить список доступных рынков:

```
<command id="get_markets"/>
```

3.16 get_servtime_difference

```
<command id="get_servtime_difference"/>
```

Получить разницу между серверным временем и временем на компьютере пользователя. Формат ответа:

```
<result success="true" diff="кол-во секунд"/>
```

3.17 change_pass

```
<command id="change_pass"  
  oldpass="текущий пароль"  
  newpass="новый пароль"  
>
```

Запрос на смену пароля. Выполняется синхронно.

Указываются действующий и новый желаемый пароль в качестве параметров. Результатом является сообщение об успешной смене пароля, либо сообщение об ошибке с указанием ошибки. На демо-сервере смена пароля запрещена, выдается сообщение: «Вам не разрешено изменять пароль».

Смена пароля необходима в случае получения сообщения от сервера Transaq в ответ на выставленную заявку: «Срок действия пароля истек. Вы сможете выполнять транзакции только после изменения своего пароля».

Максимальная длина пароля = 19 символов.

3.18 subscribe_ticks

Формат команды с использованием элемента security для идентификации бумаг:

```
<command id="subscribe_ticks">  
  <security>  
    <board>идентификатор режима торгов</board>  
    <seccode>код инструмента</seccode>  
    <traden>номер сделки</traden>  
  </security>  
  <filter>true/false</filter>  
</command>
```

Подписка на тиковые данные. С атрибутом filter="true" будут отдаваться только сделки нормального периода торгов. По умолчанию отдаются все сделки.

В tradeno указывается номер сделки, после которой надо начать отдавать тики по данному инструменту. Если 0, то с момента выполнения подписки. Чтобы получить все известные серверу тики, то можно задать заведомо малое значение tradeno, например, единицу.

Тики выдаются в хронологическом порядке вперемешку по инструментам. Каждый последующим запрос отменяет существующую подписку. Для того, чтобы отменить подписку вообще, необходимо сделать запрос с пустым списком инструментов.

Обращаем внимание, что для подписки на большое количество инструментов необходимо использовать `<command id="subscribe"><alltrades>`.

3.19 get_connector_version

```
<command id = "get_connector_version"/>
```

Запрос на получение версии модуля XmlConnector. В ответ на данную команду внешнему приложению возвращается структура

```
<connector_version>Номер_версии_коннектора</connector_version>
```

3.20 get_server_id

```
<command id = "get_server_id"/>
```

Запрос на получение идентификатора сервера. Выполняется синхронно. В ответ на данную команду внешнему приложению будет передано сообщение "current_server".

3.21 get_securities_info

Формат команды с использованием элемента security для идентификации бумаг:

```
<command id = "get_securities_info">  
  <security>  
    <market> внутренний код рынка </market>  
    <seccode> код инструмента </seccode>  
  </security>  
</command>
```

Запрос на получение информации по инструменту.

В ответ на данную команду внешнему приложению возвращается структура `<sec_info>`.

3.22 moveorder

```
<command id="moveorder">  
  <transactionid>идентификатор заменяемой заявки FORTS  
  </transactionid>  
  <price>цена</price>  
  <moveflag>способ замены</moveflag>  
  <quantity>количество, лот</quantity>  
</command>
```

Запрос на изменение заявки. Команда `moveorder` применима только для заявок с фиксированной ценой по инструментам Срочного рынка.

Элемент `moveflag` может принимать следующие значения:

- 0: не менять количество;
- 1: изменить количество;
- 2: при несовпадении количества с текущим – снять заявку.

Элемент `quantity` учитывается только в том случае, если элемент `moveflag` имеет значение 1 или 2. После того, как заявка будет принята Биржей, в асинхронном режиме будет получена структура `<orders>`, имеющая в составе данных ссылку на номер транзакции.

При `moveflag=0` в новой заявке остается текущий фактический объем заменяемой заявки на момент замены (баланс).

При `moveflag=1` кол-во лотов в новой заявке принимается равным значению `quantity`, заданному в команде `moveorder`.

При `moveflag=2` заявка снимается, если баланс заменяемой заявки не равен значению `quantity`, заданному в команде `moveorder`.

3.23 get_portfolio

```
<command id="get_portfolio" client="код клиента"/>
```

Команда имеет один обязательный параметр – клиент, для которого запрашивается портфель Т+.

Запрос по данной команде исполняется однократно, то есть на одну команду приходит ровно один ответ. При необходимости получить обновление данных команду нужно выполнить вновь. Ответная структура всегда приходит целиком, инкрементального обновления не предусматривается.

Результатом команды является структура <portfolio_tplus>.

3.24 get_max_buy_sell_tplus

Команда для получения информации о максимально возможных объемах заявок на покупку и на продажу по перечисленным бумагам сектора Т+ фондового рынка для заданного клиента. В данной команде можно задавать только инструменты фондового рынка и валютной секции МБ.

Формат команды с использованием элемента security для идентификации бумаг:

```
<command id="get_max_buy_sell_tplus" client="код клиента">
  <security>
    <market>Внутренний код рынка</market>
    <seccode>Код инструмента</seccode>
  </security>
  .....
  <security>
    <market>Внутренний код рынка</market>
    <seccode>Код инструмента</seccode>
  </security>
</command>
```

Расчет производится для режима торгов по умолчанию (элемент <board> структуры <securities>).

Если заданный в параметрах запроса инструмент не является инструментом фондового рынка или валютной секции МБ, то для него возвращаются нулевые maxbuy и maxsell. Результатом команды является структура <max_buy_sell_tplus>.

3.25 get_united_portfolio

```
<command id="get_united_portfolio" client="код клиента" union="код юниона" />
```

В команде необходимо задать только один из параметров (client или union). Если задан client, то возвращается состояние того единого портфеля (ЕП), в который включен данный клиент. (Код ЕП клиента отдается в элементе <union> структуры <client>). Запрос по данной команде выполняется однократно, то есть на одну команду приходит ровно один ответ. При необходимости получить обновление данных команду нужно выполнить вновь. Ответная структура всегда приходит целиком, инкрементального обновления не предусматривается. Результатом команды является структура <united_portfolio>.

3.26 get_united_equity

```
<command id="get_united_equity" union="код юниона" />
```

Получить актуальную оценку ликвидационной стоимости Единого портфеля, соответствующего юниону.

Результатом команды является структура <united_equity>.

3.27 get_united_go

```
<command id="get_united_go" union="код юниона" />
```

Получить размер средств, заблокированных биржей (FORTS) под срочные позиции клиентов юниона

Результатом команды является структура <united_go>.

3.28 get_cln_sec_permissions

Получить свойства конкретного инструмента, действующие в отношении конкретного клиента.

```
<command id=" get_cln_sec_permissions">  
  <security>  
    <market>Внутренний код рынка</market>  
    <seccode>Код инструмента</seccode>  
  </security>  
  <client>клиент</client>  
</command>
```

Результатом команды является структура <cln_sec_permissions>

4. Список возможных сообщений, передаваемых CallBack функции

4.1 Исторические данные

```
<candles secid=".." period="идентификатор периода" status=".."
board="идентификатор режима торгов" seccode="код инструмента">
  <candle date="дата" open="..." high="..." low="..." close="..."
    volume="..." oi="open_interest"/>
</candles>
```

secid	<i>:integer</i>	open	<i>:double</i>
period	<i>:integer</i>	high	<i>:double</i>
status	<i>:integer</i>	low	<i>:double</i>
board	<i>:string</i>	close	<i>:double</i>
seccode	<i>:string</i>	volume	<i>:integer</i>
date	<i>:date</i>	oi	<i>:integer</i>

Параметр "status" показывает, осталась ли еще история. Возможные значения:

- 0 - данных больше нет (дочерпали до дна)
- 1 - заказанное количество выдано (если надо еще - делать еще запрос)
- 2 - продолжение следует (будет еще порция)
- 3 - требуемые данные недоступны (есть смысл попробовать запросить позже).

Open interest передается только для фьючерсов и опционов.

Для идентификации инструмента, к которому относятся свечи, рекомендуем пользоваться парой board+seccode. Secid оставлен для совместимости.

4.2 Состояние сервера

```
<server_status id="ID сервера" connected="true/false/error"  
recover="true/атрибут отсутствует" server_tz="имя таймзоны сервера"  
sys_ver="версия системы" build="билд сервера"/>
```

id	<i>:integer</i>
connected	<i>:string</i>
recover	<i>:string</i>
server_tz	<i>:string</i>
sys_ver	<i>:integer</i>
build	<i>:integer</i>

Данное сообщение передается после выполнения команды "connect", при изменении состояния соединения с сервером в процессе работы, либо по команде "server_status".

Значением атрибута id является идентификационный номер сервера, с которым в данный момент работает коннектор.

Атрибут recover – необязательный параметр. Его наличие означает, что коннектор пытается восстановить потерянное соединение с сервером (находится в состоянии рекавера).

Если потеря соединения происходит на этапе подключения к серверу (то есть когда загрузка начальных данных еще не выполнена), клиенту будет передана структура вида:

```
<server_status id="ID сервера" connected="false" recover="true"/>
```

При этом на этапе подключения может быть передано несколько сообщений о рекавере. Например, при подключении могут быть переданы следующие структуры:

```
<markets>...</markets>  
...  
<securities>...</securities>  
<server_status id="ID сервера" connected="false" recover="true"/>  
<securities>...</securities>  
<server_status id="ID сервера" connected="false" recover="true"/>  
<securities>...</securities>  
...  
<server_status id="ID сервера" connected="true"/>
```

При потере соединения с сервером в процессе работы (после того, как связь с сервером была успешно установлена, т.е. значение connected="true" в составе <server_status> уже было получено), клиенту будет передана

структура вида:

```
<server_status id="ID сервера" connected="true" recover="true"/>
```

Далее, в случае успешного рекавера, будет передана структура:

```
<server_status id="ID сервера" connected="true"/>
```

Если же рекавер закончится неудачей и соединение с сервером будет потеряно, то структура "server_status" будет иметь вид:

```
<server_status id="ID сервера" connected="false"/>
```

При значении connected="error", тег будет содержать сообщение об ошибке. Атрибуты id и recover будут отсутствовать. Например:

```
<server_status connected="error">Сервер недоступен</server_status>
```

Данная структура может быть получена в случае если библиотека не может подключиться к серверу по следующим причинам:

- неверно указаны логин/пароль;
- сервер недоступен;
- нет прав на подключение к серверу;
- не удается открыть соединение;
- внутренних ошибках коннектора при подключении



<server_status id="N" connected="true/false"> отправляется для информирования пользователя о состоянии библиотеки в случае установки/потери связи с сервером, в случае начала/завершения процесса recover.

Появление структуры <server_status connected="error"> означает, что произошла ошибка при подключении к серверу.

При получении структуры <server_status connected="error"> необходимо выполнять повторное соединение, так как соединение не установлено. Необходимо устранить ошибку (если это возможно, например, в случае ошибки в параметрах учетной записи) и повторить попытку подключения.

4.3 Клиентские счета

```

<client id="CLIENT_ID" remove="true/false">
  <type>тип клиента</type>
  <currency>валюта фондового портфеля клиента</currency>
  <market>id рынка </market>
  <union>код юниона </union>
  <forts_acc>счет FORTS </forts_acc>
</client>

```

id	<i>:string</i>	market	<i>:integer</i>
remove	<i>:string</i>	union	<i>:string</i>
type	<i>:string</i>	forts_acc	<i>:string</i>
currency	<i>:string</i>		

Данные сообщения для каждого из клиентских счетов передаются сразу после коннекта. Кроме того, это сообщение передается в случае добавления/удаления администратором TRANSAQ доступа к клиентскому счету во время текущей сессии.

Возможные типы клиента: spot (кассовый), leverage (плечевой), margin_level (маржинальный), mct (клиент ММА).

Параметр remove говорит о том, добавился ли клиент или удалился. В случае удаления клиента никаких свойств клиента передано не будет.

Валюта фондового портфеля currency может принимать следующие значения: NA (если клиент не имеет фондового портфеля), RUB, EUR, USD.

Union – код Единого Портфеля, в который включен данный клиент. Если клиент не включён в юнион, то элемент <union> в структуре client отсутствует.

market – идентификатор рынка, на котором разрешено работать данному клиенту (значение id из структуры markets).

forts_acc - счет FORTS клиента. Если клиент не имеет счета FORTS, то элемент не передается.

4.4 Доступные рынки

```
<markets>  
  <market id="внутренний код рынка">название рынка</market>  
</markets>
```

id	<i>:integer</i>
название рынка	<i>:string</i>

4.5 Информация о доступных периодах свечей

```
<andlekinds>  
  <kind>  
    <id>идентификатор периода</id>  
    <period>количество секунд в периоде</period>  
    <name>наименование периода</name>  
  </kind>  
</andlekinds>
```

id	<i>:integer</i>
period	<i>:string</i>
name	<i>:string</i>

4.6 Список инструментов

```

<securities>
  <security secid="внутренний код" active="true/false">
    <seccode> Код инструмента </seccode>
    <instrclass>Символ категории (класса) инструмента
    </instrclass>
    <board>Идентификатор режима торгов по умолчанию
    </board>
    <market>Идентификатор рынка</market>
    <shortname>Наименование бумаги</shortname>
    <decimals>Количество десятичных знаков в
    цене</decimals>
    <minstep>Шаг цены</minstep>
    <lotsize>Размер лота</lotsize>
    <point_cost>Стоимость пункта цены</point_cost>
    <opmask usecredit="yes/no" bymarket="yes/no"
    nosplit="yes/no" fok="yes/no" ioc="yes/no "/>
    <sectype>Тип бумаги</sectype>
    <sec_tz>имя таймзоны инструмента (типа "Russian Standard
    Time", "USA=Eastern Standard Time"), содержит секцию
    CDATA</sec_tz>
    <quotestype>0 - без стакана; 1 - стакан типа OrderBook; 2 -
    стакан типа Level2</quotestype>
    <MIC>код биржи листинга по стандарту ISO</MIC>
    <ticker>тикер на бирже листинга</ticker>
    <currency>валюта цены</currency>
  </security>
</securities>

```

secid	<i>:integer</i>
active	<i>:string</i>
seccode	<i>:string</i>
instrclass	<i>:string</i>
board	<i>:string</i>
market	<i>:integer</i> или <i>:string</i> (в случае ошибки)
shortname	<i>:string</i>
decimals	<i>:integer</i>
minstep	<i>:double</i>
lotsize	<i>:integer</i>
point_cost	<i>:double</i>
usecredit	<i>:string</i>
bymarket	<i>:string</i>
nosplit	<i>:string</i>
fok	<i>:string</i>
ioc	<i>:string</i>
sectype	<i>:string</i>
sec_tz	<i>:string</i>

quotestype	<i>:integer</i>
MIC	<i>:string</i>
ticker	<i>:string</i>
currency	<i>:string</i>

Массив инструментов выдается автоматически после успешного подключения к серверу.

Может быть также получен впоследствии по запросу `get_securities`. Следует иметь в виду, что список инструментов может приходить не единым блоком, а несколькими, а также в ходе сессии по мере подключения рынков, и динамического получения доступа к отдельным инструментам.

Secid действителен в течение сессии, постоянным уникальным ключом инструмента между сессиями является `Seccode+Market`.

Атрибут `active` указывает, является ли бумага торгуемой в обычном режиме. Для неторгуемых бумаг не будут возвращаться поля `minstep`, `lotsize` и `ormask`.

В качестве `sectype` могут встречаться следующие поля:

Торгуемые инструменты:

- SHARE - акции
- BOND - облигации корпоративные
- FUT - фьючерсы FORTS
- OPT - опционы
- GKO - гос. бумаги
- FOB - фьючерсы ММББ

Неторгуемые (все кроме `IDX` приходят только с зарубежных площадок):

- IDX - индексы
- QUOTES - котировки (прочие)
- CURRENCY - валютные пары
- ADR - АДР
- NYSE - данные с NYSE
- METAL - металлы
- OIL - нефтяной сектор

Также:

ERROR - в случае внутренней ошибки (не должно появляться)

`Ormask` возвращается только для торгуемых инструментов (для которых `active="true"`), его параметрами являются параметры заявок (см. `neworder`).

`Ormask usecredit="yes"` указывает на возможность использования свойства

"Использовать кредит" в диалоге ввода заявки. Для фьючерсов, например, usecredit="no" (всегда).

Стоимость шага цены вычисляется как:

$$\text{Стоимость_шага_цены} = \text{point_cost} * \text{minstep} * 10^{\text{decimals}}$$

Полученное значение выражено в копейках. Для перевода значения в рубли необходимо его разделить на 100.



Передаваемый в этой структуре режим торгов по умолчанию (элемент <board>) используется системой в том случае, когда внешняя программа в командах идентифицирует инструмент с помощью secid (а не пары seccode+board или seccode+market).

Значения decimals, minstep, lotsize и point_cost в этом списке инструментов даны для режима торгов по умолчанию. Данные параметры инструмента для других режимов торгов доступны в структуре <pit>.

Для инструментов ММВБ, торгуемых одновременно и в секторе T0 и в секторе T+ "режим торгов по умолчанию" определяется Брокером. Это может быть как режим сектора T0, так и режим сектора T+.

4.7 Информация по инструменту

```

<sec_info secid="идентификатор бумаги">
  <secname> полное наименование инструмента</secname>
  <seccode> Код инструмента</seccode>
  <market>Внутренний код рынка </market>
  <pname> единицы измерения цены</pname>
  <mat_date> дата погашения</mat_date>
  <clearing_price>цена последнего клиринга (только FORTS)
</clearing_price>
  <minprice>минимальная цена (только FORTS)</minprice>
  <maxprice>максимальная цена (только FORTS)</maxprice>
  <buy_deposit>ГО покупателя (фьючерсы FORTS,
руб.)</buy_deposit>
  <sell_deposit>ГО продавца (фьючерсы FORTS,
руб.)</sell_deposit>
  <bgo_c>ГО покрытой позиции (опционы FORTS, руб.)</bgo_c>
  <bgo_nc>ГО непокрытой позиции (опционы FORTS,
руб.)</bgo_nc>
  <accruedint>текущий НКД, руб</accruedint>
  <coupon_value>размер купона, руб</coupon_value>
  <coupon_date>дата погашения купона </coupon_date>
  <coupon_period>период выплаты купона, дни</coupon_period>
  <facevalue>номинал облигации или акции, руб</facevalue>
  <put_call>тип опциона Call(C)/Put(P)</put_call>
  <point_cost>Стоимость пункта цены</point_cost>
  <bgo_buy>Базовое ГО под покупку маржируемого опциона
</bgo_buy>
  <opt_type>маржинальный(M)/премия(P)</opt_type>
  <lot_volume>количество базового актива (FORTS)</lot_volume>
</sec_info>

```

sec_id	<i>:integer</i>	bgo_nc	<i>:double</i>
secname	<i>:string</i>	accruedint	<i>:double</i>
seccode	<i>:string</i>	coupon_value	<i>:double</i>
market	<i>:integer</i>	coupon_date	<i>:date</i>
pname	<i>:string</i>	coupon_period	<i>:integer</i>
mat_date	<i>:date</i>	facevalue	<i>:double</i>
clearing_price	<i>:double</i>	put_call	<i>:string</i>
minprice	<i>:double</i>	opt_type	<i>:string</i>
maxprice	<i>:double</i>	point_cost	<i>:double</i>
buy_deposit	<i>:double</i>	bgo_buy	<i>:double</i>
sell_deposit	<i>:double</i>	lot_volume	<i>:integer</i>
bgo_c	<i>:double</i>		

В структуру включаются только те элементы, значение которых отлично от нуля.

Элементы:

- buy_deposit
- sell_deposit
- bgo_c
- bgo_nc
- bgo_buy

форматируются следующим образом:

- элемент не добавляется в структуру, если значение соответствующего поля равно 0
- элемент форматируется как [0-9]+.[0-9]{2}, если значение поля не равно 0

4.8 Обновление информации по инструменту

```

<sec_info_upd>
  <secid>идентификатор бумаги</secid>
  <market>Внутренний код рынка</market>
  <seccode> Код инструмента</seccode>
  <minprice>минимальная цена (только FORTS)</minprice >
  <maxprice>максимальная цена (только FORTS)</maxprice >
  <buy_deposit>ГО покупателя (фьючерсы FORTS, руб.)
</buy_deposit>
  <sell_deposit>ГО продавца (фьючерсы FORTS, руб.)
</sell_deposit>
  <bgo_c>ГО покрытой позиции (опционы FORTS, руб.)</bgo_c>
  <bgo_nc>ГО непокрытой позиции (опционы FORTS, руб.)
</bgo_nc>
  <bgo_buy>Базовое ГО под покупку маржируемого опциона
</bgo_buy>
  <point_cost>Стоимость пункта цены</point_cost>
</sec_info_upd>

```

secid	<i>:integer</i>
market	<i>:integer</i>
seccode	<i>:string</i>
minprice	<i>:double</i>
maxprice	<i>:double</i>
buy_deposit	<i>:double</i>
sell_deposit	<i>:double</i>
bgo_c	<i>:double</i>
bgo_nc	<i>:double</i>
bgo_buy	<i>:double</i>
point_cost	<i>:double</i>

Элементы:

- buy_deposit
- sell_deposit
- bgo_c
- bgo_nc
- bgo_buy

форматируются следующим образом:

- элемент не добавляется в структуру, если соответствующее поле не пришло от сервера
- элемент форматируется как 0.00, если поле пришло от сервера и его значение равно 0
- элемент форматируется как [0-9]+.[0-9]{2}, если поле пришло от сервера и его значение не равно 0

4.9 Котировки по инструменту(ам)

```
<quotations>
  <quotation secid ="внутренний код">
    <board>Идентификатор режима торгов по умолчанию
    </board>
    <seccode> Код инструмента </seccode>
    <accruedintvalue>НКД на дату торгов в расчете на одну
    бумагу, руб.</accruedintvalue>
    <open>Цена первой сделки</open>
    <waprice>Средневзвешенная цена</waprice>
    <biddepth> Кол-во лотов на покупку по лучшей цене
    </biddepth>
    <biddeptht>Совокупный спрос</biddeptht>
    <numbids>Заявок на покупку</numbids>
    <offerdepth>Кол-во лотов на продажу по лучшей
    цене</offerdepth>
    <offerdeptht>Совокупное предложение</offerdeptht>
    <bid>Лучшая котировка на покупку</bid>
    <offer>Лучшая котировка на продажу</offer>
    <numoffers>Заявок на продажу</numoffers>
    <numtrades>Сделок</numtrades>
    <voltoday>Объем совершенных сделок в лотах</voltoday>
    <openpositions>Общее количество открытых
    позиций(FORTS)</openpositions>
    <deltapositions>Изм.открытых
    позиций(FORTS)</deltapositions>
    <last>Цена последней сделки</last>
    <quantity>Объем последней сделки, в лотах.</quantity>
    <time>Время заключения последней сделки</time>
    <change>Изменение цены последней сделки по отношению
    к цене последней сделки предыдущего торгового
    дня</change>
    <priceminusprevwaprice>Цена последней сделки к оценке
    предыдущего дня</priceminusprevwaprice>
    <valtoday>Объем совершенных сделок, млн. руб</valtoday>
    <yield>Доходность, по цене последней сделки</yield>
    <yieldatwaprice>Доходность по средневзвешенной
    цене</yieldatwaprice>
    <marketpricetoday>Рыночная цена по результатам торгов
    сегодняшнего дня</marketpricetoday>
    <highbid>Наибольшая цена спроса в течение торговой
    сессии</highbid>
    <lowoffer>Наименьшая цена предложения в течение
    торговой сессии</lowoffer>
    <high>Максимальная цена сделки</high>
    <low>Минимальная цена сделки</low>
```

```

    <closeprice>Цена закрытия</closeprice>
    <closeyield>Доходность по цене закрытия</closeyield>
    <status>Статус «торговые операции
    разрешены/запрещены»</status>
    <tradingstatus>Состояние торговой сессии по
    инструменту</tradingstatus>
    <buydeposit>ГО покупок/покр</buydeposit>
    <selldeposit>ГО продаж/непокр</selldeposit>
    <volatility>Волатильность </volatility >
    <theoreticalprice>Теоретическая цена</theoreticalprice>
    <bgo_buy>Базовое ГО под покупку маржируемого опциона
    </bgo_buy>
    <point_cost>Стоимость пункта цены</point_cost>
    <lcurrentprice> Официальная текущая цена Биржи
    </lcurrentprice>
  </quotation>
  <quotation secid ="внутренний код">
    ....
  </quotation>
</quotations>

```

secid	<i>:integer</i>
board	<i>:string</i>
seccode	<i>:string</i>
accruedintvalue	<i>:double</i>
open	<i>:double</i>
waprice	<i>:double</i>
biddepth	<i>:integer</i>
biddeptht	<i>:integer</i>
numbids	<i>:integer</i>
offerdepth	<i>:integer</i>
offerdeptht	<i>:integer</i>
bid	<i>:double</i>
offer	<i>:double</i>
numoffers	<i>:integer</i>
numtrades	<i>:integer</i>
voltoday	<i>:integer</i>
openpositions	<i>:integer</i>
deltapositions	<i>:integer</i>
last	<i>:double</i>
quantity	<i>:integer</i>
time	<i>:date</i>
change	<i>:double</i>
priceminusprevwaprice	<i>:double</i>
valtoday	<i>:double</i>
yield	<i>:double</i>
yieldatwaprice	<i>:double</i>
marketpricetoday	<i>:double</i>

highbid	<i>:double</i>
lowoffer	<i>:double</i>
high	<i>:double</i>
low	<i>:double</i>
closeprice	<i>:double</i>
closeyield	<i>:double</i>
status	<i>:string</i>
tradingstatus	<i>:string</i>
buydeposit	<i>:double</i>
selldeposit	<i>:double</i>
volatility	<i>:double</i>
theoreticalprice	<i>:double</i>
bgo_buy	<i>:double</i>
point_cost	<i>:double</i>
lcurrentprice	<i>:double</i>

Передается после подписки путем команды subscribe. В сообщении в первый раз передаются все релевантные инструменту поля, для которых имеются актуальные данные, впоследствии передаются только измененные поля.

4.10 Сделки по инструменту(ам)

```

<alltrades>
  <trade secid ="внутренний код">
    <seccode> Код инструмента </seccode>
    <tradenno>биржевой номер сделки</tradenno>
    <time>время сделки</time>
    <board> наименование борда </board>
    <price>цена сделки</price>
    <quantity>объем сделки</quantity>
    <buysell>покупка (B) / продажа (S)</buysell>
    <openinterest>кол-во открытых позиций на срочном
рынке</openinterest>
    <period>Период торгов (O - открытие, N - торги, C -
закрытие)</period>
  </trade>
  <trade secid ="внутренний код">
    ....
  </trade>
</alltrades>

```

secid	<i>:integer</i>	price	<i>:double</i>
seccode	<i>:string</i>	quantity	<i>:integer</i>
tradenno	<i>:integer64</i>	buysell	<i>:string</i>
time	<i>:date</i>	openinterest	<i>:integer</i>
board	<i>:string</i>	period	<i>:string</i>

Передается после подписки путем команды subscribe. В сообщении могут быть переданы не все поля, а только те, у которых есть значения.

Информация по следкам РПС, РЕПО и сделкам по неполным лотам не передается.

Параметр open interest передается только для фьючерсов и опционов.



Переговорные сделки передаются в alltrades пустым тэгом:

```
<alltrades>
</alltrades>
```

4.11 Глубина рынка по инструменту(ам)

```
<quotes>
  <quote secid ="внутренний код">
    <board>Идентификатор режима торгов по умолчанию
    </board>
    <seccode> Код инструмента </seccode>
    <price>цена</price>
    <source> Источник котировки (маркетмейкер)</source>
    <yield>доходность (актуально только для
    облигаций)</yield>
    <buy>количество бумаг к покупке</buy>
    <sell>количество бумаг к продаже</sell>
  </quote>
  <quote secid ="внутренний код">
    ....
  </quote>
</quotes>
```

secid	<i>:integer</i>
board	<i>:string</i>
seccode	<i>:string</i>
price	<i>:double</i>
source	<i>:string</i>
yield	<i>:integer</i>
buy	<i>:integer</i>
sell	<i>:integer</i>

Передается после подписки путем команды subscribe. В сообщении передаются не все поля, а только те, у которых изменились значения.

Ключом, идентифицирующим строку в «стакане» для большинства инструментов, является значение элемента price.

Для некоторых инструментов в составе структуры quote передается также элемент source. В этом случае в качестве ключа, идентифицирующего строку в «стакане», нужно принимать пару price + source.

- Значение «-1» в поле buy означает, что в данной строке «стакана» больше нет заявок на покупку.
- Значение «-1» в поле sell означает, что в данной строке «стакана» больше нет заявок на продажу.
- Значение «-1» одновременно и в поле sell и в поле buy означает, что строка с данной ценой (или с данным значением пары price + source) удалена из «стакана».

4.12 Заявка(и) клиента

```
<orders>
  <order transactionid ="идентификатор транзакции сервера
Transaq">
    <orderno>биржевой номер заявки</orderno>
    <secid>идентификатор бумаги</secid>
    <board>идентификатор борда</board>
    <seccode>код инструмента</seccode>
    <client>идентификатор клиента</client>
    <union>Код юниона </union>
    <status>статус заявки (см. ниже в таблице 5)</status>
    <buysell>покупка (B) / продажа (S)</buysell>
    <time>время регистрации заявки биржей</time>
    <expdate>дата экспирации (только для ФОРТС)</expdate>
    (задается в формате 23.07.2012 00:00:00 (не обязательно))
    <origin_orderno>биржевой номер родительской
заявки</origin_orderno>
    <accepttime>время регистрации заявки сервером Transaq
(только для условных заявок)</accepttime>
    <brokerref>примечание</brokerref>
    <value>объем заявки в валюте инструмента</value>
    <accruedint>НКД</accruedint>
    <settlecode>Код поставки (значение биржи, определяющее
правила расчетов - смотрите подробнее в документах
биржи)</settlecode>
    <balance>Неудовлетворенный остаток объема заявки в
лотях (контрактах)</balance>
    <price>Цена</price>
    <quantity>Количество</quantity>
    <hidden>Скрытое количество в лотах</hidden>
    <yield>Доходность</yield>
    <withdrawtime>Время снятия заявки, 0 для
активных</withdrawtime>
    <condition>Условие, см. newcondorder</condition>
    <conditionvalue>Цена для условной заявки, либо
обеспеченность в процентах</conditionvalue>
    <validafter>с какого момента времени действительна (см.
newcondorder)</validafter>
    <validbefore>до какого момента действительно (см.
newcondorder)</validbefore>
    <maxcomission>максимальная комиссия по сделкам
заявки</maxcomission>
    <result>сообщение биржи в случае отказа выставить
заявку</result>
  </order>
  <stoporder transactionid="идентификатор стопа">
    <activeorderno>номер заявки биржевой части
```

```

стопа</activeorderno>
<secid></secid>
<board></board>
<seccode>код инструмента</seccode>
<client></client>
<union>Код юниона </union>
<buysell></buysell>
<canceller></canceller>
<alltradeno></alltradeno>
<validbefore></validbefore>
<author></author>
<accepttime></accepttime>
<linkedorderno></linkedorderno>
<expdate></expdate>
<status></status>
<stoploss usecredit="yes/no">
    <activationprice></activationprice>
    <guardtime></guardtime>
    <brokerref></brokerref>
    <quantity></quantity>
    <bymarket/>
    <orderprice></orderprice>
</stoploss>
<takeprofit>
    <activationprice></activationprice>
    <guardtime></guardtime>
    <brokerref></brokerref>
    <quantity></quantity>
    <extremum></extremum>
    <level></level>
    <correction></correction>
    <guardspread></guardspread>
</takeprofit>
<withdrawtime>Время снятия заявки, 0 для
активных</withdrawtime>
<result>сообщение биржи в случае отказа выставить заявку
</result>
</stoporder>
</orders>

```

<order>

transactionid	<i>:integer</i>
orderno	<i>:integer64</i>
secid	<i>:integer</i>
board	<i>:string</i>
seccode	<i>:string</i>
client	<i>:string</i>
union	<i>:string</i>

status	<i>:string</i>
buysell	<i>:string</i>
time	<i>:date</i>
expdate	<i>:date</i>
origin_orderno	<i>:integer64</i>
accepttime	<i>:date</i>
brokerref	<i>:string</i>
value	<i>:double</i>
accruedint	<i>:double</i>
settlecode	<i>:string</i>
balance	<i>:integer</i>
price	<i>:double</i>
quantity	<i>:integer</i>
hidden	<i>:integer</i>
yield	<i>:double</i>
withdrawtime	<i>:date</i>
condition	<i>:string</i>
conditionvalue	<i>:double</i>
validafter	<i>:date</i>
validbefore	<i>:date</i>
maxcomission	<i>:double</i>
result	<i>:string</i>
<stoporder>	
transactionid	<i>:integer</i>
activeorderno	<i>:integer64</i>
secid	<i>:integer</i>
board	<i>:string</i>
seccode	<i>:string</i>
client	<i>:string</i>
union	<i>:string</i>
buysell	<i>:string</i>
canceller	<i>:string</i>
alltradeno	<i>:integer64</i>
validbefore	<i>:date</i>
author	<i>:string</i>
accepttime	<i>:date</i>
linkedorderno	<i>:integer64</i>
expdate	<i>:date</i>
status	<i>:string</i>
<stoploss>	
usecredit	<i>:string</i>
activationprice	<i>:double</i>
guardtime	<i>:date</i>
brokerref	<i>:string</i>
quantity	<i>:integer</i> или <i>:double</i> (в случае %)
orderprice	<i>:double</i>
<takeprofit>	
activationprice	<i>:double</i>
guardtime	<i>:date</i>

brokerref	<i>:string</i>
quantity	<i>:integer</i> или <i>:double</i> (в случае %)
extremum	<i>:double</i>
level	<i>:double</i>
correction	<i>:double</i>
guardspread	<i>:double</i>
withdrawtime	<i>:date</i>
result	<i>:string</i>

Передается автоматически после установки соединения, а также после изменения какого-либо из полей. В последнем случае передаются только изменившиеся поля. В таких сообщениях для условных заявок идентификатором является transactionid, для всех остальных – orderno.

Transactionid - это идентификатор транзакции сервера Transaq, неизменность которого гарантируется внутри текущего сеанса связи с сервером. Он служит для идентификации условных заявок, а также заявок, отклоненных Биржей. После выставления заявки на биржу, ей присваивается параметр orderno - это постоянный идентификатор заявки, который не меняется вплоть до окончания биржевой торговой сессии.

Для стоп-заявок поля соответствуют аналогам при выставлении заявок. Статусы стоп-заявок и обычных заявок различаются - смотрите их список в конце документации. Любое из полей может отсутствовать, если оно неактуально или не было задано при выставлении заявки.

Значение <status> cancelled и нулевое значение элемента <withdrawtime> означает, что заявка находится в процессе снятия.

Ненулевое значение <withdrawtime> при значении <status> cancelled означает, что заявка снята.

Alltradeno - биржевой регистрационный номер сделки, явившейся основанием для перехода стопа в текущее состояние.

Canceller - идентификатор трейдера, который отменил стоп.

Activeorderno - биржевой регистрационный номер заявки, выставленной на рынок в результате исполнения стопа.

Maxcomission - это максимальная комиссия, которая может быть взята по сделкам по этой заявке. Фактически показывает, сколько денег заблокировано под комиссию для данной заявки. Maxcomission будет ненулевым пока заявка активна и равен нулю для заявок ФОРТС и для ВСЕХ условных заявок.

4.13 Сделка(и) клиента

```

<trades>
  <trade>
    <secid>Идентификатор бумаги</secid>
    <tradeno>Номер сделки на бирже</tradeno>
    <orderno>Номер заявки на бирже</orderno>
    <board>Идентификатор борда</board>
    <seccode> Код инструмента </seccode>
    <client>Идентификатор клиента</client>
    <union>Код юниона</union>
    <buysell>В - покупка, S - продажа</buysell>
    <time>время сделки</time>
    <brokerref>примечание</brokerref>
    <value>объем сделки</value>
    <comission>комиссия</comission>
    <price>цена</price>
    <items>кол-во инструмента в сделках в штуках </items>
    <quantity>количество лотов</quantity>
    <yield>доходность</yield>
    <accruedint>НКД</accruedint>
    <tradetype>тип сделки: 'Т' – обычная 'N' – РПС 'R' – РЕПО
    'P' – размещение</tradetype>
    <settlecode>код поставки</settlecode>
    <currentpos>Текущая позиция по инструменту</currentpos>
    <bypass>Признак внебиржевой сделки, 1 или поле
    отсутствует </bypass>
    <venue>Площадка (execution place)</venue>
  </trade>
</trades>

```

secid	<i>:integer</i>	commission	<i>:double</i>
tradeno	<i>:integer64</i>	price	<i>:double</i>
orderno	<i>:integer64</i>	items	<i>:integer64</i>
board	<i>:string</i>	quantity	<i>:integer64</i>
seccode	<i>:string</i>	yield	<i>:double</i>
client	<i>:string</i>	accruedint	<i>:double</i>
union	<i>:string</i>	tradetype	<i>:string</i>
buysell	<i>:string</i>	settlecode	<i>:string</i>
time	<i>:date</i>	currentpos	<i>:integer64</i>
brokerref	<i>:string</i>	bypass	<i>:integer</i>
value	<i>:double</i>	venue	<i>:string</i>

Передается автоматически после установки соединения (для уже совершенных сделок), а также по мере появления новых сделок.

4.14 Позиции клиента

```

<positions>
  <money_position>
    <markets>
      <market>Внутренний код рынка</market>
      ...
      <market>Внутренний код рынка</market>
    </markets>
    <register>Регистр учета</register>
    <asset>Код вида средств</asset>
    <client>Идентификатор клиента </client>
    <union>Код юниона </union>
    <shortname>Наименование вида средств</shortname>
    <saldoin>Входящий остаток</saldoin>
    <bought>Куплено</bought>
    <sold>Продано</sold>
    <saldo>Текущее сальдо</saldo>
    <ordbuy>В заявках на покупку + комиссия</ordbuy>
    <ordbuycond>В условных заявках на покупку</ordbuycond>
    <comission>Сумма списанной комиссии</comission>
  </money_position>
  <sec_position>
    <secid>Код инструмента</secid>
    <market>Внутренний код рынка</market>
    <seccode>Код инструмента</seccode>
    <register>Регистр учета</register>
    <client>Идентификатор клиента</client>
    <union>Код юниона </union>
    <shortname>Наименование инструмента</shortname>
    <saldoin>Входящий остаток</saldoin>
    <saldomin>Неснижаемый остаток</saldomin>
    <bought>Куплено</bought>
    <sold>Продано</sold>
    <saldo>Текущее сальдо</saldo>
    <ordbuy>В заявках на покупку</ordbuy>
    <ordsell>В заявках на продажу</ordsell>
    <amount>Текущая оценка стоимости позиции, в валюте
инструмента</amount>
    <equity>Текущая оценка стоимости позиции, в рублях
    </equity>
  </sec_position>
  <forts_position>
    <secid>Код инструмента</secid>
    <markets>
      <market>Внутренний код рынка</market>
      ...
      <market>Внутренний код рынка</market>

```

```

</markets>
<seccode>Код инструмента</seccode>
<client>Идентификатор клиента</client>
<union>Код юниона </union>
<startnet>Входящая позиция по инструменту</startnet>
<openbuys>В заявках на покупку</openbuys>
<opensells>В заявках на продажу</opensells>
<totalnet>Текущая позиция по инструменту</totalnet>
<todaybuy>Куплено</todaybuy>
<todaysell>Продано</todaysell>
<optmargin>Маржа для маржируемых
опционов</optmargin>
<varmargin>Вариационная маржа</varmargin>
<expirationpos>Опционов в заявках на
исполнение</expirationpos>
<usedsellspotlimit>Объем использованного спот-лимита на
продажу</usedsellspotlimit>
<sellspotlimit>текущий спот-лимит на продажу,
установленный Брокером</sellspotlimit>
<netto>нетто-позиция по всем инструментам данного
спота</netto>
<kgo>коэффициент ГО для спота</kgo>
</forts_position>
<forts_money> - деньги ФОРТС
  <client>Идентификатор клиента</client>
  <union>Код юниона </union>
  <markets>
    <market>Внутренний код рынка</market>
    ...
    <market>Внутренний код рынка</market>
  </markets>
  <shortname>Наименование вида средств</shortname>
  <current>Текущие</current>
  <blocked>Заблокировано</blocked>
  <free>Свободные</free>
  <varmargin>Опер. маржа</varmargin>
</forts_money>
<forts_collaterals> - залоги ФОРТС
  <client>Идентификатор клиента</client>
  <union>Код юниона </union>
  <markets>
    <market>Внутренний код рынка</market>
    ...
    <market>Внутренний код рынка</market>
  </markets>
  <shortname>Наименование вида средств</shortname>
  <current>Текущие</current>
  <blocked>Заблокировано</blocked>
  <free>Свободно</free>

```

```

    </forts_collaterals>
    <spot_limit> - лимиты ФОРТС
      <client>Идентификатор клиента</client>
      <union>Код юниона </union>
      <markets>
        <market>Внутренний код рынка</market>
        ...
        <market>Внутренний код рынка</market>
      </markets>
      <shortname>Наименование вида средств</shortname>
      <buylimit>Текущий лимит</buylimit>
      <buylimitused>Заблокировано лимита</buylimitused>
    </spot_limit>
  </positions>

```

<money_position>

client	<i>:string</i>
union	<i>:string</i>
market	<i>:integer</i>
register	<i>:string</i>
asset	<i>:string</i>
shortname	<i>:string</i>
saldoin	<i>:double</i>
bought	<i>:double</i>
sold	<i>:double</i>
saldo	<i>:double</i>
ordbuy	<i>:double</i>
ordbuycond	<i>:double</i>
commission	<i>:double</i>

<sec_position>

secid	<i>:integer</i>
market	<i>:integer</i>
seccode	<i>:string</i>
register	<i>:string</i>
client	<i>:string</i>
union	<i>:string</i>
shortname	<i>:string</i>
saldoin	<i>:integer64</i>
saldomin	<i>:integer64</i>
bought	<i>:integer64</i>
sold	<i>:integer64</i>
saldo	<i>:integer64</i>
ordbuy	<i>:integer64</i>
ordsell	<i>:integer64</i>
amount	<i>:double</i>
equity	<i>:double</i>

<forts_position>

secid	<i>:integer</i>
market	<i>:integer</i>

seccode	<i>:string</i>
client	<i>:string</i>
union	<i>:string</i>
startnet	<i>:integer</i>
openbuys	<i>:integer</i>
opensells	<i>:integer</i>
totalnet	<i>:integer</i>
todaybuy	<i>:integer</i>
todaysell	<i>:integer</i>
optmargin	<i>:double</i>
varmargin	<i>:double</i>
expirationpos	<i>:integer64</i>
usedsellspotlimit	<i>:double</i>
sellspotlimit	<i>:double</i>
netto	<i>:double</i>
kgo	<i>:double</i>
<forts_money>	
client	<i>:string</i>
union	<i>:string</i>
market	<i>:integer</i>
shortname	<i>:string</i>
current	<i>:double</i>
blocked	<i>:double</i>
free	<i>:double</i>
varmargin	<i>:double</i>
<forts_collaterals>	
client	<i>:string</i>
union	<i>:string</i>
market	<i>:integer</i>
shortname	<i>:string</i>
current	<i>:double</i>
blocked	<i>:double</i>
free	<i>:double</i>
<spot_limit>	
client	<i>:string</i>
union	<i>:string</i>
market	<i>:integer</i>
shortname	<i>:string</i>
buylimit	<i>:double</i>
buylimitused	<i>:double</i>

Передается автоматически после установки соединения, а также по мере появления новых позиций и изменении какого-либо из полей. В последнем случае передаются только изменившиеся поля.

Элемент `markets` в структуре `money_position` устанавливает зависимость между денежными позициями клиента и рынками, на которых их можно использовать.

Forts_position возвращается для фьючерсов, опционов, позиций по инструментам РТС Стандарт.

Теги kgo, netto, sellspotlimit и usedsellspotlimit отдаются только для позиций по инструментам, являющимся «главными спотами» РТС Стандарт.

В forts_position возвращаются всегда все поля для любых инструментов. Поля, которые не имеют смысла для инструмента (например, varmargin для опционов), возвращаются с нулевым значением.

Набор данных в структурах <money_position> и <sec_position> зависит от значения элемента micex_registers в команде connect. Если значение элемента micex_registers задано как false, то в <money_position> и <sec_position>. для фондового рынка ММББ будут отдаваться только данные с регистром "T0". Если micex_registers=true, то в <money_position> и <sec_position> для фондового рынка ММББ будут отдаваться данные по всем регистрам, а также будет присутствовать элемент <register>, определяющий учетный регистр позиции.

В случае наличия в команде connect элемента

<push_u_limits>Период в секундах</push_u_limits>

всякий раз при возникновении существенных событий (изменение состояния заявок или сделок клиентов юниона), но не реже чем один раз в N секунд будет передана структура следующего вида:

```
<positions>
  <united_limits union="код юниона">
    <open_equity>Входящая оценка стоимости единого
    портфеля</open_equity>
    <equity>Текущая оценка стоимости единого портфеля
    </equity>
    <requirements>Начальные требования </requirements>
    <free>Свободные средства </free>
    <vm>Вариационная маржа FORTS </vm>
    <finres>Финансовый результат последнего клиринга FORTS
    </finres>
    <go>Размер требуемого ГО, посчитанный биржей FORTS
    </go>
  </united_limits>
</positions>
```

union	<i>:string</i>
open_equity	<i>:double</i>
equity	<i>:double</i>
requirements	<i>:double</i>
free	<i>:double</i>
vm	<i>:double</i>
finres	<i>:double</i>
go	<i>:double</i>

Также для каждого клиента ФОРТС, входящего в юнион, будет передана структура:

```
<positions>
  <forts_money>
    <client>Клиент </client>
    <union>Юнион </union>
    <current>Текущие в ЕП </current>
    <free>Свободные в ЕП </free>
  </forts_money>
</positions>
```

client	<i>:string</i>
union	<i>:string</i>
current	<i>:double</i>
free	<i>:double</i>

В случае наличия в команде connect элемента

```
<push_pos_equity>Период в секундах</push_pos_equity>
```

раз в N секунд выполняется информирование пользователя о текущей стоимости позиций - передается массив структур <positions><sec_position>, соответствующих множеству удерживаемых клиентами позиций (за исключением позиций FORTS).

Поля amount и equity передаются только в случае наличия в команде connect элемента <push_pos_equity>.

4.15 Лимиты клиента на срочном рынке

```
<clientlimits client="клиент">
  <cbplimit>стоимостной лимит открытых позиций (СЛОП срочн.
рынок ММББ) </cbplimit>
  <cbplused>стоимостная оценка текущих чистых позиций (СОЧП
срочн. рынок ММББ) </cbplused>
  <cbplplanned>СОЧП с учетом активных заявок (срочный рынок
ММББ) </cbplplanned>
  <fob_varmargin>Вар. маржа срочного рынка
ММББ</fob_varmargin>
  <coverage>Обеспеченность срочного портфеля
(FORTS)</coverage>
  <liquidity_c>Коэффициент ликвидности(FORTS)</liquidity_c>
  <profit>Доход(FORTS)</profit>
  <money_current>Деньги текущие. То же, что current в
forts_money</money_current>
  <money_reserve>Деньги заблокированные. То же, что blocked в
forts_money </money_reserve>
  <money_free>Деньги свободные. То же, что free в forts_money
</money_free>
  <options_premium>Премии по
опционам(FORTS)</options_premium>
  <exchange_fee>Биржевой сбор(FORTS)</exchange_fee>
  <forts_varmargin>Вар. маржа текущая (FORTS)</forts_varmargin>
  <varmargin>Операционная маржа. То же, что varmargin в
forts_money </varmargin>
  <pclmargin> Перечисленная в пром.клиринге вариационная
маржа(FORTS)</pclmargin>
  <options_vm> Вар. маржа по опционам(FORTS) </ options_vm >
  <spot_buy_limit>Лимит на покупку спот </spot_buy_limit>
  <used_spot_buy_limit> Лимит на покупку спот
использованный</used_spot_buy_limit>
  <collat_current>Залоги текущие. То же, что current в
forts_collaterals </collat_current>
  <collat_blocked>Залоги заблокированные. То же, что blocked в
forts_collaterals </collat_blocked>
  <collat_free> Залоги свободные. То же, free в forts_collaterals
</collat_free>
</clientlimits>
```

client	<i>:string</i>	options_premium	<i>:double</i>
cbplimit	<i>:double</i>	exchange_fee	<i>:double</i>
cbplused	<i>:double</i>	forts_varmargin	<i>:double</i>
cbplplanned	<i>:double</i>	varmargin	<i>:double</i>
fob_varmargin	<i>:double</i>	pclmargin	<i>:double</i>
coverage	<i>:double</i>	options_vm	<i>:double</i>
liquidity_c	<i>:double</i>	spot_buy_limit	<i>:double</i>
profit	<i>:double</i>	used_stop_buy_limit	<i>:double</i>
money_current	<i>:double</i>	collat_current	<i>:double</i>
money_reserve	<i>:double</i>	collat_blocked	<i>:double</i>
money_free	<i>:double</i>	collat_free	<i>:double</i>

4.16 Клиентский портфель Т+

```

<portfolio_tplus client="код клиента">
  <coverage_fact>"фактическая обеспеченность"</coverage_fact>
  <coverage_plan>"плановая обеспеченность"</coverage_plan>
  <coverage_crit>"критическая
обеспеченность, %"</coverage_crit>
  <open_equity>"входящая оценка портфеля без
дисконта"</open_equity>
  <equity>"текущая оценка портфеля без дисконта"</equity>
  <cover>"плановое обеспечение (оценка ликвидационной
стоимости портфеля)"</cover>
  <init_margin>"плановая начальная маржа (оценка портфельного
риска)"</init_margin>
  <pnl_income>"прибыль/убыток по входящим
позициям"</pnl_income>
  <pnl_intraday>"прибыль/убыток по сделкам"</pnl_intraday>
  <leverage>"фактическое плечо портфеля Т+"</leverage>
  <margin_level>"фактический уровень маржи портфеля
Т+"</margin_level>
  <money>
    <open_balance>"входящая денежная
позиция"</open_balance>
    <bought>"Затрачено на покупки"</bought>
    <sold>"выручено от продаж"</sold>
    <settled>"исполнено"</settled>
    <balance>"текущая денежная позиция"</balance>
    <tax>"уплачено комиссии"></tax>
    <value_part register="регистр учёта">
      <open_balance>"входящая денежная
позиция"</open_balance>
      <bought>"Затрачено на покупки"</bought>
      <sold>"выручено от продаж"</sold>
      <settled>"исполнено"</settled>
      <balance>"текущая денежная позиция"</balance>
    </value_part>
    ... (<value_part>)
    ... (<value_part>)
  </money>
  <security secid="id инструмента">
    <market>"id рынка"</market>
    <seccode>"обозначение инструмента"</seccode>
    <price>"текущая цена"</price>
    <open_balance>"входящая позиция, штук"</open_balance>
    <bought>"куплено, штук"</bought>
    <sold>"продано, штук"</sold>
    <balance>"текущая позиция, штук"</balance>
    <balance_prc>Балансовая цена</balance_prc>
  </security>
</portfolio_tplus>

```

```

    <unrealized_pnl>Нереализованные
    прибыли/убытки</unrealized_pnl>
    <buying>"заявлено купить, штук"</buying>
    <selling>"заявлено продать, штук"</selling>
    <cover>"вклад бумаги в плановое обеспечение"</cover>
    <init_margin>"плановая начальная маржа
    (риск)"</init_margin>
    <riskrate_long>"ставка риска для лонгов,
    %"</riskrate_long>
    <riskrate_short>"ставка риска для шортов,
    %"</riskrate_short>
    <pnl_income>"прибыль/убыток по входящим позициям
    "</pnl_income>
    <pnl_intraday>"прибыль/убыток по сделкам"</pnl_intraday>
    <maxbuy>"максимальная покупка, в лотах"</maxbuy>
    <maxsell>"максимальная продажа, в лотах"</maxsell>
    <value_part register="регистр учёта">
        <open_balance>"входящая позиция,
        штук"</open_balance>
        <bought>"куплено, штук"</bought>
        <sold>"продано, штук"</sold>
        <settled>"исполнено, штук"</settled>
        <balance>"текущая позиция, штук"</balance>
        <buying>"заявлено купить, штук"</buying>
        <selling>"заявлено продать, штук"</selling>
    </value_part>
    ... (<value_part>)
    ... (<value_part>)
</security>
... (<security>)
... (<security>)
</portfolio_tplus>

```

<portfolio_tplus>

client	<i>:string</i>
coverage_fact	<i>:double</i>
coverage_plan	<i>:double</i>
coverage_crit	<i>:double</i>
open_equity	<i>:double</i>
equity	<i>:double</i>
cover	<i>:double</i>
init_margin	<i>:double</i>
pnl_income	<i>:double</i>
pnl_intraday	<i>:double</i>
leverage	<i>:double</i>
margin_level	<i>:double</i>
<money>	
open_balance	<i>:double</i>
bought	<i>:double</i>

sold	<i>:double</i>
settled	<i>:double</i>
balance	<i>:double</i>
tax	<i>:double</i>
<value_part>	
register	<i>:string</i>
open_balance	<i>:double</i>
bought	<i>:double</i>
sold	<i>:double</i>
settled	<i>:double</i>
balance	<i>:double</i>
<security>	
secid	<i>:integer</i>
market	<i>:integer</i>
seccode	<i>:string</i>
price	<i>:double</i>
open_balance	<i>:integer</i>
bought	<i>:integer</i>
sold	<i>:integer</i>
balance	<i>:integer</i>
balance_prc	<i>:double</i>
unrealized_pnl	<i>:double</i>
buying	<i>:integer</i>
selling	<i>:integer</i>
cover	<i>:double</i>
init_margin	<i>:double</i>
riskrate_long	<i>:double</i>
riskrate_short	<i>:double</i>
pnl_income	<i>:double</i>
pnl_intraday	<i>:double</i>
maxbuy	<i>:integer</i>
maxsell	<i>:integer</i>
<value_part>	
register	<i>:string</i>
open_balance	<i>:integer</i>
bought	<i>:integer</i>
sold	<i>:integer</i>
settled	<i>:integer</i>
balance	<i>:integer</i>
buying	<i>:integer</i>
selling	<i>:integer</i>

4.17 Возможность рыночных заявок

```
<marketord secid="id" seccode="Код инструмента" permit="yes/no" />
```

secid	<i>:integer</i>
seccode	<i>:string</i>
permit	<i>:string</i>

Приходит в случае, если во время работы Коннектора изменится возможность выставления или снятия рыночной заявки по инструменту с идентификатором id.

4.18 Тиковые данные

```
<ticks>
  <tick>
    <secid>идентификатор бумаги</secid>
    <tradeno>номер сделки</tradeno>
    <tradetime>время сделки</tradetime>
    <price>цена</price>
    <quantity>количество лотов (контрактов)</quantity>
    <period>торговый период (О - открытие, N - торги, С -
    закрытие; передается только для ММВБ)</period>
    <buysell>В - покупка, S - продажа (с точки зрения того, кто
    инициировал сделку, приняв условия выставленной ранее
    заявки - передается только когда есть такая
    информация)</buysell>
    <openinterest> кол-во открытых позиций на срочном рынке
    </openinterest>
    <board>Идентификатор режима торгов по умолчанию
    </board>
    <seccode> Код инструмента </seccode>
  </tick>
</ticks>
```

secid	<i>:integer</i>	period	<i>:string</i>
tradeno	<i>:integer64</i>	buysell	<i>:string</i>
tradetime	<i>:date</i>	openinterest	<i>:integer</i>
price	<i>:double</i>	board	<i>:string</i>
quantity	<i>:integer</i>	seccode	<i>:string</i>

4.19 Справочник режимов торгов

```

<boards>
  <board id = "Идентификатор режима торгов">
    <name>Наименование режима торгов</name>
    <market>Внутренний код рынка </market>
    <type> тип режима торгов 0=FORTS, 1=T+, 2=T0</type>
  </board>
</boards>

```

board	<i>:string</i>	market	<i>:integer</i>
name	<i>:string</i>	type	<i>:integer</i>

4.20 Параметры инструмента в режиме торгов

```

<pit seccode="Код инструмента" board="Идентификатор режима
торгов">
  <market>Идентификатор рынка</market>
  <decimals>Количество десятичных знаков в цене</decimals>
  <minstep>Шаг цены</minstep>
  <lotsize>Размер лота</lotsize>
  <point_cost>Стоимость пункта цены</point_cost>
</pit>

```

seccode	<i>:string</i>	minstep	<i>:double</i>
board	<i>:string</i>	lotsize	<i>:integer</i>
market	<i>:integer</i>	point_cost	<i>:double</i>
decimals	<i>:integer</i>		

4.21 Максимальная покупка/продажа для T+

```

<max_buy_sell_tplus client="код клиента ">
  <security secid="код бумаги">
    <market>Внутренний код рынка</market>
    <seccode>Код инструмента</seccode>
    <maxbuy>максимум купить (лот.)</maxbuy>
    <maxsell>максимум продать (лот.)</maxsell>
  </security>
</max_buy_sell_tplus>

```

secid	<i>:integer</i>	seccode	<i>:string</i>
client	<i>:integer</i>	maxbuy	<i>:integer64</i>
market	<i>:integer</i>	maxsell	<i>:integer64</i>

4.22 Текстовые сообщения

```
<messages>
  <message>
    <date>Дата и время</date>
    <urgent>Срочное: Y/N</urgent>
    <from>Отправитель</from>
    <text>Текст сообщения</text>
  </message>
</messages>
```

date	<i>:date</i>	from	<i>:string</i>
urgent	<i>:string</i>	text	<i>:string</i>

Поле text содержит секцию CDATA.

4.23 Возможные ошибки

В случае возникновения внутренних ошибок dll, они будут возвращены в callback-функцию в следующем виде:

```
<error>Описание ошибки</error>
```

error	<i>:string</i>
-------	-----------------------

4.24 Клиентский единый портфель

```

<united_portfolio union="код юниона" client="код клиента">
  <open_equity>Входящая оценка стоимости единого портфеля
    :double </open_equity>
  <equity>Текущая оценка стоимости единого портфеля :double
    </equity>
  <chrgoff_ir>Корреляционный вычет планового риска :double
    </chrgoff_ir>
  <init_req>Плановый риск (размер начальных требований)
    :double </init_req>
  <chrgoff_mr>Корреляционный вычет минимальных требований
    :double </chrgoff_mr>
  <maint_req>Размер минимальных требований :double
    </maint_req>
  <reg_equity>Стоимость портфеля нормативная :double
    </reg_equity>
  <reg_ir>Плановая начальная маржа нормативная :double
    </reg_ir>
  <reg_mr>Минимальная маржа нормативная :double </reg_mr>
  <vm>Вариационная маржа FORTS :double </vm>
  <finres>Финансовый результат последнего клиринга FORTS
    :double </finres>
  <go>Размер требуемого ГО, посчитанный биржей FORTS :double
    </go>
  <vm_mma>Вариационная маржа MMA, рубли:double
</vm_mma>
  <money name="Наименование денежного раздела :string">
    <open_balance>Входящая денежная позиция :double
      </open_balance>
    <bought>Затрачено на покупки :double </bought>
    <sold>Выручено от продаж :double </sold>
    <settled>Исполнено :double </settled>
    <balance>Текущая денежная позиция :double </balance>
    <tax>Уплачено комиссии :double </tax>
    <value_part register="регистр учёта :string">
      <open_balance>Входящая денежная позиция :double
        </open_balance>
      <bought>Затрачено на покупки :double </bought>
      <sold>Выручено от продаж :double </sold>
      <settled>Исполнено :double </settled>
      <balance>Текущая денежная позиция
        :double</balance>
    </value_part>
    ... (<value_part>)
    ... (<value_part>)
  </money>
  ... (<money>)

```

```

... (<money>)
<asset code="Код базового актива :string" name="Наименование
базового актива :string">
  <setoff_rate>Ставка перекрытия :double</setoff_rate>
  <init_req>Плановый риск :double</init_req>
  <maint_req>Минимальная маржа :double</maint_req>
  <security secid="id инструмента :integer">
    <market>id рынка :integer</market>
    <seccode>Код инструмента :string</seccode>
    <price>Текущая цена :double</price>
    <open_balance>Входящая нетто-позиция, штук
    :integer</open_balance>
    <bought>Куплено, штук :integer</bought>
    <sold>Продано, штук :integer</sold>
    <balance>Текущая нетто-позиция, штук
    :integer</balance>
    <balance_prc>Балансовая цена :double
    </balance_prc>
    <unrealized_pnl>Нереализованные прибыли/убытки
    :double <unrealized_pnl>
    <buying>Заявлено купить, штук :integer</buying>
    <selling>Заявлено продать, штук :integer</selling>
    <equity>Оценка (в составе обеспечения ЕП)
    :double</equity>
    <reg_equity>Стоимость в обеспечении портфеля
    нормативная :double</reg_equity>
    <riskrate_long>Ставка риска для лонгов, %
    :double</riskrate_long>
    <riskrate_short>Ставка риска для шортов, %
    :double</riskrate_short>
    <reerate_long>Ставка резерва для лонгов, %
    :double</reerate_long>
    <reerate_short>Ставка резерва для шортов, %
    :double</reerate_short>
    <pl>Прибыль/убыток общий:double</pl>
    <pnl_income>Прибыль/убыток по входящим позициям
    :double</pnl_income>
    <pnl_intraday>Прибыль/убыток по сделкам
    :double</pnl_intraday>
    <maxbuy>Максимальная покупка, в лотах :integer
    </maxbuy>
    <maxsell>Максимальная продажа, в лотах :integer
    </maxsell>
    <value_part register="регистр учёта :string">
    <open_balance>Входящая позиция, штук :integer
    </open_balance>
    <bought>Куплено, штук :integer </bought>
    <sold>Продано, штук :integer </sold>
    <settled>Исполнено, штук :integer </settled>

```



```

        <balance>Текущая позиция, штук :integer </balance>
        <buying>Заявлено купить, штук :integer </buying>
        <selling>Заявлено продать, штук :integer </selling>
    </value_part>
    ... (<value_part>)
    ... (<value_part>)
</security>
... (<security>)
... (<security>)
</asset>
... (<asset>)
... (<asset>)
</united_portfolio>

```

4.25 Актуальная оценка ликвидационной стоимости Единого портфеля

```

<united_equity union="код юниона">
    <equity>Текущая оценка стоимости единого портфеля</equity>
</united_equity>

```

equity **:double**

4.26 Размер средств, заблокированных биржей (FORTS) под срочные позиции клиентов юниона

```

<united_go union="код юниона">
    <go>Размер требуемого ГО, посчитанный биржей FORTS</go>
</united_go>

```

go **:double**

4.27 Юнионы, находящиеся в управлении клиента

```

<union id="код юниона :string" remove="true/false :string

```

Список юнионов передаётся пользователю при подключении. Также структура <union> используется для информирования пользователя об изменении состава доступных клиенту юнионов.

4.28 Свойства инструмента, действующие в отношении конкретного клиента

```
<cln_sec_permissions client="код клиента" secid="">
  <security>
    <seccode> Код инструмента :string </seccode>
    <market> Внутренний код рынка :integer </market>
    <board> Идентификатор режима торгов :string </board>
  </security>
  <tradeable> Торговые операции разрешены (true/false)
  </tradeable>
  <marketable> Рыночные заявки разрешены (true/false)
  </marketable>
  <longable> Открытие длинных позиций разрешено (true/false)
  </longable>
  <shortable> Открытие коротких позиций разрешено (true/false)
  </shortable>
  <regratelong> Ставка начальной маржи лонг :double
  </regratelong>
  <regrateshort> Ставка начальной маржи шорт :double
  </regrateshort>
  <regratelongx> Ставка минимальной маржи лонг :double
  </regratelongx>
  <regrateshortx> Ставка минимальной маржи шорт :double
  </regrateshortx>
  <riskrate_long> Единая ставка риска лонг :double </riskrate_long>
  <reerate_long> Единая ставка резерва лонг :double
  </reerate_long>
  <riskrate_short> Единая ставка риска шорт :double
  </riskrate_short>
  <reerate_short> Единая ставка резерва шорт :double
  </reerate_short>
  <riskrate_longx> Единая ставка мин. риска лонг :double
  </riskrate_longx>
  <reerate_longx> Единая ставка мин. резерва лонг :double
  </reerate_longx>
  <riskrate_shortx> Единая ставка мин. риска шорт :double
  </riskrate_shortx>
  <reerate_shortx> Единая ставка мин. резерва шорт
  :double </reerate_shortx>
</cln_sec_permissions>
```


5. Получение новостей в TXmlConnector


После подключения к серверу внешнее приложение будет автоматически получать заголовки свежих новостей в виде структур `news_header` (см. пункт 5.2.2).

Для получения заголовков более старых новостей необходимо выполнять команду `"get_old_news"` (см. пункт 5.1.2).

Каждая новость имеет уникальный целочисленный идентификатор. Более свежие новости имеют большие значения идентификаторов.

Чтобы получить тело новости для известного заголовка необходимо выполнить команду `"get_news_body"` (см. пункт 5.1.3).

 Команду `"get_news_body"` можно вызывать для получения только тех новостей, заголовки которых были получены внешним приложением в текущем сеансе подключения к серверу Transaq.

 При реализации хранения новостей во внешнем приложении следует учитывать, что на каждом сервере Transaq ведется своя нумерация новостей. Для получения идентификатора сервера используйте команду `"get_server_id"`.

5.1 Спецификация команд

5.1.1 Получение идентификатора сервера

```
<command id = "get_server_id">
```

Запрос на получение идентификатора сервера. Выполняется синхронно. В ответ на данную команду внешнему приложению будет передано сообщение `"current_server"` (см. пункт 5.2.1).

5.1.2 Получение заголовков более старых новостей

```
<command id = "get_old_news" count = "OLD_NEWS_COUNT"/>
```

Значение атрибута `"count"` определяет максимальное количество новостей, которое необходимо получить, и не может превышать 100.

В ответ на данную команду внешнему приложению будет отдано запрошенное количество заголовков новостей (структур `"news_header"`), если они существуют на сервере (см. пункт 5.2.2).

5.1.3 Получение тела новости

```
<command id = "get_news_body" news_id = "NEWS_ID"/>
```

Значение атрибута " news_id " задает номер новости из полученного ранее заголовка, для которой необходимо получить тело. В ответ на данную команду внешнему приложению будет передано сообщение "news_body" (см. пункт 5.2.3).

5.2 Список возможных сообщений, передаваемых callback функции

5.2.1 Идентификатор сервера

```
<current_server id = "SERVER_ID"/>
```

id ***:integer***

Здесь: id = "SERVER_ID" – идентификатор сервера, с которым установлено текущее соединение.

5.2.2 Заголовок новости

```
<news_header>  
  <id> порядковый номер новости </id>  
  <timestamp> дата-время новости (от источника)</timestamp>  
  <source><![CDATA[ источник новости ]]></source>  
  <title><![CDATA[ заголовок новости ]]></title>  
</news_header >
```

id ***:integer***
timestamp ***:date***
source ***:string***
title ***:string***

5.2.3 Тело новости

```
<news_body>
  <id> порядковый номер новости </id>
  <text><![CDATA[ текст новости в виде обычного текста
]]></text>
</news_body >
```

id	<i>:integer</i>
text	<i>:string</i>

6. Приложения

Таблица 1. Возможные значения для поля status в теге quotation

Символ	Значение
A	Операции разрешены
N	Заблокировано для торгов, разрешено исполнение сделок
S	Операции запрещены

Таблица 2. ММВБ/РТС. Возможные значения для tradingstatus в теге quotation

Символ	Значение
N	Недоступно для торгов (ММВБ)
O	Период открытия (ММВБ)
C	Торги закрыты (ММВБ)
F	Период закрытия (ММВБ)
B	Перерыв (ММВБ)
T	Торговая сессия (ММВБ)
L	Период послеторгового аукциона (ММВБ)
E	Период торгов по цене Аукциона Закрытия
D	Аукцион крупными пакетами
I	Дискретный аукцион
0	Сессия назначена. Нельзя ставить заявки, но можно удалять (FORTS)
1	Сессия идет. Можно ставить и удалять заявки (FORTS)
2	Приостановка торгов по всем инструментам. Нельзя ставить заявки, но можно удалять (FORTS)
3	Сессия принудительно завершена. Нельзя ставить и удалять заявки (FORTS)
4	Сессия завершена по времени. Нельзя ставить и удалять заявки (FORTS)
5	Приостановка торгов по этому инструменту. Нельзя ставить заявки, но можно удалять по этому инструменту.

Таблица 3. Атрибуты заявки

Наименование	Описание
Bymarket / По рынку	Заявка по рынку. При наличии тега bymarket, тег price игнорируется и может отсутствовать.
Usecredit / Использовать кредит	Чтобы воспользоваться кредитом при заключении сделки, установите параметр «Использовать кредит» в поручении на выставление заявки.
Nosplit / По одной цене	При наличии встречных заявок с пересекающимися ценами, сделка совершается по цене и в объеме лучшей встречной заявки. Неисполненная часть подаваемой заявки помещается в очередь заявок с ценой, равной цене совершенной сделки.
PutInQueue / Поставить в очередь	При поступлении заявки в торговую систему Биржи производится проверка наличия в очереди встречных заявок, цены которых совпадают или пересекаются с ценой подаваемой заявки. При наличии таких заявок в системе производятся сделки по цене ранее поданной заявки. Затем количество лотов инструмента, указанное в заявке, уменьшается на количество лотов в совершенной сделке и производится аналогичная процедура сопоставления новой заявки с оставшимися встречными заявками. Неисполненная часть заявки помещается в очередь заявок.
FOK / Немедленно или отклонить	Аналогично Поставить в очередь, но сделки совершаются только в том случае, если заявка может быть удовлетворена полностью. В противном случае заявка не выставляется
IOC / Снять остаток	Аналогично Поставить в очередь, но неисполненная часть заявки снимается с торгов

Таблица 4. Статусы ордеров

Идентификатор	Значение
none	
active	Активная
cancelled	Снята трейдером (заявка уже попала на рынок и была отменена)
denied	Отклонена Брокером
disabled	Прекращена трейдером (условная заявка, которую сняли до наступления условия)
expired	Время действия истекло
failed	Не удалось выставить на биржу
forwarding	Выставляется на биржу
inactive	Статус не известен из-за проблем со связью с биржей
matched	Исполнена
refused	Отклонена контрагентом
rejected	Отклонена биржей
removed	Аннулирована биржей
wait	Не наступило время активации
watching	Ожидает наступления условия

Таблица 5. Статусы стоп-заявок

Идентификатор	Значение
cancelled	Снята трейдером (заявка уже попала на рынок и была отменена)
denied	Отклонена Брокером
disabled	Прекращена трейдером (стоп-заявка, которую сняли до наступления условия)
expired	Время действия истекло
failed	Не удалось выставить на биржу
linkwait	Ожидает исполнения связанной заявки
rejected	Отклонена биржей
sl_executed	Выполнена (Stop Loss)
sl_forwarding	Стоп выставляется на биржу (Stop Loss)
sl_guardtime	Стоп ожидает исполнения в защитном периоде
tp_correction	Ожидает исполнения в режиме коррекции (Take Profit)
tp_correction_guardtime	Стоп ожидает исполнения в защитном режиме после коррекции (Take Profit)
tp_executed	Выполнен (Take Profit)
tp_forwarding	Стоп выставляется на биржу (Take Profit)
tp_guardtime	Стоп ожидает исполнения в защитном периоде (Take Profit)
watching	Ожидает наступления условия

Таблица 6. Возможные значения tradetype в структуре trades

Константа	Описание
Валютный рынок Московской Биржи	
T	Обычная
N	Внебиржевая
P	Первичное размещение
S	Расчетная по операции своп
W	Расчетная по внебиржевой операции своп
F	Перевод денег/бумаг
E	Расчетная сделка бивалютной корзины
K	Расчетная внебиржевая сделка бивалютной корзины
Фондовый рынок Московской Биржи	
T	Обычная
N	Адресная
P	Первичное размещение
F	Перевод денег/бумаг
R	Адресная сделка первой части РЕПО
G	Сделка по операции РЕПО с ЦК
H	Первая часть сделки по операции РЕПО с ЦК
h	Вторая часть сделки по операции РЕПО с ЦК
I	Адресная сделка по операции РЕПО с ЦК
J	Первая часть адресной сделки по операции РЕПО с ЦК
j	Вторая часть адресной сделки по операции РЕПО с ЦК
L	Техническая сделка по возврату активов РЕПО с ЦК
M	Первая часть адресной сделки РЕПО с корзиной
m	Вторая часть адресной сделки РЕПО с корзиной

Таблица 7. Параметры торговых серверов Transaq

IP-адрес	Порт
Тестовый сервер	
tr1-demo5.finam.ru	3939
Торговые сервера АО "ФИНАМ"	
tr1.finam.ru	3900
tr2.finam.ru	3900
Торговые сервера АО "Банк Финам"	
tr1.finambank.ru	3324
Высокоскоростной Transaq (HFT) АО "ФИНАМ"	
hft.finam.ru	13900