

Just2Trade

Руководство пользователя
TRANSAQ XML Connector

Версия 6.09 билд 2.20.32

Содержание

1 ОБЩЕЕ ОПИСАНИЕ	6
2.1 ТИПЫ ДАННЫХ	12
2.2 Числовые значения	12
2.3 Дата и время.....	12
2.4 Идентификация финансовых инструментов в командах и сообщениях	12
3. СПЕЦИФИКАЦИЯ КОМАНД	14
3.1 Команда connect	14
3.2 Команда disconnect	19
3.3 Команда server_status	19
3.4 Команда get_securities	19
3.5 Команда subscribe	19
3.6 Команда unsubscribe	20
3.7 Команда get_history_data	22
3.8 Команда neworder	22
3.9 Команда newcondorder	24
3.10 Команда newstoporder	26
3.11 Команда cancelorder	27
3.12 Команда cancelstoporder	27
3.13 Команда get_forts_positions	27
3.14 Команда get_client_limits	27
3.15 Команда get_markets	27
3.16 Команда get_servtime_difference	28
3.17 Команда get_leverage_control.....	28
3.18 Команда change_pass.....	28
3.19 Команда subscribe_ticks.....	28
3.20 Команда get_server_id.....	29

3.21 Команда get_news_body	29
3.22 Команда get_old_news	29
3.23 Команда get_connector_version.....	30
3.24 Команда get_securities_info	30
3.25 Команда moveorder.....	31
3.26 Команда get_portfolio	31
3.27 Команда get_max_buy_sell_tplus	31
3.28 Команда get_portfolio_mct	32
3.29 Команда get_united_portfolio.....	33
3.30 Команда get_united_equity	33
3.31 Команда get_united_go	33
3.32 Команда get_cln_sec_permissions	34
4. СПИСОК ВОЗМОЖНЫХ СООБЩЕНИЙ, ПЕРЕДАВАЕМЫХ CALLBACK ФУНКЦИИ	35
4.1 Исторические данные	35
4.2 Состояние сервера	35
4.3 Клиентские счета.....	36
4.4 Доступные рынки	37
4.5 Информация о доступных периодах свечей	37
4.6 Список инструментов	37
4.7 Показатели торгов по инструментам	39
4.8 Сделки рынка по инструментам	41
4.9 Глубина рынка по инструментам (“стакан”).....	41
4.10 Заявки клиентов	43
4.11 Сделки клиентов.....	46
4.12 Позиции клиентов	47
4.13 Лимиты клиента на срочном рынке	51
4.14 Режим кредитования	52
4.15 Признак допустимости рыночных заявок для инструмента	52

4.16 Максимальная покупка/продажа и плечо для режимов T0	52
4.17 Тиковые данные	53
4.18 Идентификатор сервера	53
4.19 Тело новости	53
4.20 Заголовок новости	54
4.21 Информация по инструменту	54
4.22 Обновление информации по инструменту	55
4.23 Справочник режимов торгов	56
4.24 Параметры инструмента в режиме торгов	56
4.25 Клиентский портфель T+	56
4.28 Клиентский портфель МСТ	58
4.29 Клиентский единый портфель	60
4.30 Актуальная оценка ликвидационной стоимости Единого портфеля	62
4.31 Размер средств, заблокированных биржей (FORTS) под срочные позиции клиентов юниона	62
4.32 Максимальная покупка/продажа для режимов T+	62
4.33 Текстовые сообщения	63
4.34 Юнионы, находящиеся в управлении клиента	63
4.35 Свойства инструмента, действующие в отношении конкретного клиента	63
4.36 Возможные ошибки	64
5 ДОПОЛНИТЕЛЬНЫЕ КОМАНДЫ И СООБЩЕНИЯ ДЛЯ РАБОТЫ С АДРЕСНЫМИ СДЕЛКАМИ ММВБ	65
5.1 Команда newtrpsorder	65
5.2 Команда newmrepoorder	65
5.3 Команда newrepoorder	66
5.4 Команда cancelnegdeal	67
5.5 Команда newreport4	67
5.6 Команда cancelreport4	67
5.7 Сделки для исполнения	68

5.8 Отчеты на исполнение	69
6 ПРИЛОЖЕНИЯ.....	71
6 СПИСОК ИЗМЕНЕНИЙ	74

1 Общее описание

TXmlConnector.dll – загружаемая библиотека функций, предназначенная для организации взаимодействия пользовательских приложений с сервером TRANSAQ. (Далее по тексту – «Коннектор».)

Коннектор экспортирует следующие функции (соглашение вызова __stdcall):

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

Тип `tcallback` объявлен следующим образом (соглашение вызова __stdcall):

```
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.

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`.

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

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

Возвращаемое значение

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

Требования

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.

Initialize

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

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

Параметры

logPath Путь к директории, в которую будут сохраняться файлы отчетов

logLevel Глубина логирования

Предусмотрено три уровня логирования, в соответствии с детализацией и размером лог-файла:

- **Минимальный** (logLevel = 1) – логируются команды и результаты их выполнения, ошибки, сообщения типа «warning», команды и результат их выполнения.
- **Стандартный** (logLevel = 2, рекомендуемый) – логируются команды и результаты их выполнения, структуры server_status, trades, orders, positions, client, ошибки, сообщения типа «warning», команды и результат их выполнения
- **Максимальный** (logLevel = 3) – логируются все сообщения, приходящие в колбэк, команды и результат их выполнения, а также ошибки и сообщения типа «warning»

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

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.

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

- | | |
|-------------------------|---|
| log_path | Путь к директории, в которую будут сохраняться файлы отчетов |
| log_level | <p>Глубина логирования</p> <p>Предусмотрено три уровня логирования, в соответствии с детализацией и размером лог-файла:</p> <ul style="list-style-type: none"> • Минимальный (log_level="1") – логируются команды и результаты их выполнения, ошибки, сообщения типа «warning», команды и результат их выполнения. • Стандартный (log_level="2", рекомендуемый) – логируются команды и результаты их выполнения, структуры server_status, trades, orders, positions, client, ошибки, сообщения типа «warning», команды и результат их выполнения • Максимальный (log_level="3") – логируются все сообщения, приходящие в колбэк, команды и результат их выполнения, а также ошибки и сообщения типа «warning» |
| logfile_lifetime | 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.

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.

Требования

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

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.

Требования

Функция должна быть выполнена перед выгрузкой библиотеки из памяти.

SetLogLevel

BYTE* SetLogLevel(int logLevel)

Изменяет уровень логирования без остановки библиотеки.

Параметры

logLevel Глубина логирования

Предусмотрено три уровня логирования, в соответствии с детализацией и размером лог-файла:

- Минимальный (logLevel = 1);
- Стандартный (logLevel = 2, рекомендуемый);
- Максимальный (logLevel = 3).

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.

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).

Тип данных указывается в виде “:тип_данных”, например:

- <secid>идентификатор бумаги :integer</secid>

2.2 Числовые значения

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

2.3 Дата и время

Время представляется в следующем формате:

“DD.MM.YYYY hh:mm:ss”

Здесь:

- DD – день месяца от 1 до 31
- MM – месяц в году от 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

(См. также комментарий к элементу milliseconds команды connect)

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

Например, при работе с коннектором 27.01.2011 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 в Коннектор добавлены новые форматы команды с использованием элемента security.

Также финансовые инструменты могут идентифицироваться целочисленными идентификаторами secid, 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 isec для бумаги может измениться при пере-подключении 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.)

3. Спецификация команд

3.1 Команда 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>
  <logsdire>logs_directory</logsdire> - устаревший
  <loglevel>log level</loglevel> - устаревший
  <autopos>true/false</autopos>
  <micex_registers>true/false</micex_registers>
  <milliseconds>true/false</milliseconds>
  <utc_time>true/false</utc_time>
  <notes_file>notes filename</notes_file>
  <proxy type="тип" addr="адрес" port="порт" login="логин"
  password="пароль"/>
  <rqdelay>Период агрегирования данных</rqdelay>
  <session_timeout>Таймаут на сессию в секундах</session_timeout>
  <request_timeout>Таймаут на запрос в секундах </request_timeout>
  <csp>опционально, см. описание ниже</csp>
  <push_u_limits>Период в секундах :integer </push_u_limits>
  <push_pos_equity>Период в секундах :integer </push_pos_equity>
</command>
```

При исполнении команды “connect” функция SendCommand вернет результат разбора переданной команды.

Далее, если все параметры команды заданы верно, начнется процедура подключения к серверу и получения необходимых начальных данных. В процессе получения данных они сразу же отдаются приложению через callback функцию.

После того, как все начальные данные будут загружены, приложению будет отдана структура <server_status> со значением connected="true". Это означает, что коннектор готов к работе.

При возникновении ошибок в процессе подключения приложению будет отдана структура <server_status> со значением connected="false" или connected="error" (подробнее см. описание структуры <server_status>).

Параметры logsdire и loglevel являются устаревшими и при разборе структуры команды не анализируются. Управление параметрами логирования осуществляется с помощью функций

Initialize и SetLogLevel.

Параметр autopos указывает на необходимость автоматического запроса клиентских позиций на срочном рынке FORTS после каждой клиентской сделки. Если autopos не указан, по умолчанию он принимается равным true. Задание `<autopos>>false</autopos>` при активной торговле ускоряет взаимодействие с сервером.

Параметр notes_file задает имя файла, в котором будут храниться примечания к заявкам. Если не указать имя файла, по умолчанию будет использоваться файл notes.xml. Примечания хранятся в xml-формате.

Тэг проху следует включать только тогда, когда подключение осуществляется через прокси-сервер. При необходимости можно задать атрибуты login и password для авторизации на прокси-сервере.

Доступны следующие типы прокси-сервера: SOCKS4, SOCKS5, HTTP-CONNECT (задаются именно в таком виде, с учетом регистра и дефиса).

rqdelay задает период агрегирования данных на сервере Transaq в миллисекундах. Минимальное допустимое значение – 10 мсек.

Параметр language задает язык системных сообщений.

session_timeout – интервал времени, в течении которого коннектор в случае ошибок связи будет выполнять попытки переподключения к серверу без повторного получения списка финансовых инструментов ([securities](#)) и других справочников. Если данный параметр не задан, используется значение по-умолчанию равное 120 секундам.

request_timeout - таймаут на выполнение запроса. Если данный параметр не задан, используется значение по-умолчанию равное 20 секундам.

Значение параметра session_timeout должно быть больше значения параметра request_timeout.

Значение элемента micex_registers определяет набор данных, передаваемый в структурах `<money_position>` и `<sec_position>` (см. раздел [4.12 Позиции клиента](#)).

Значение элемента milliseconds определяет формат элементов типа "Дата и время". Если задано `<milliseconds>true</milliseconds>`, то поля, содержащие время, форматируются в виде "DD.MM.YYYY hh:mm:ss.mmm".

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

- negdeal (time, activationtime, withdrawtime)
- ustrade (match_time, settle_date, tradedate)
- report (time, withdrawtime)
- 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 структуру server_status добавляется элемент server_tz, содержащий имя таймзоны сервера

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>.

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>`.

Коннектор поддерживает несколько типов криптозащиты:

1. TRANSAQ. Встроенные средства криптозащиты TRANSAQ. Для аутентификации трейдера требуется его идентификатор и пароль (задаются элементами `login` и `password`).
2. RSA. Встроенные средства криптозащиты Microsoft Windows, аутентификация с использованием сертификатов X.509 (ключи RSA).

Для использования данного типа криптозащиты необходимо в команду "connect" добавить элемент:

```
<csp type="MS">  
    <key_path>Полный путь к каталогу с ключами пользователя</key_path>  
    <key_id>Идентификатор ключа пользователя</key_id>  
    <password>Пароль доступа к личному ключу</password>  
    <server_cert>Полный путь к файлу сертификата сервера</server_cert>  
</csp>
```

3. Крипто-Про. Криптопровайдер компании »Крипто-Про», аутентификация с использованием сертификатов X.509 (ключи ГОСТ 34.10-2001).

Для использования данного типа криптозащиты необходимо в команду "connect" добавить элемент:

```
<csp type="CP">  
    <container_name>Имя контейнера личного ключа пользователя (без префикса  
    "TRANSAQ_") </container_name>  
    <server_cert>Полный путь к файлу сертификата сервера</server_cert>  
</csp>
```

При подключении к серверу может быть использован только один тип криптозащиты. Вариант криптозащиты, который должен использоваться при подключении к серверу, определяется Брокером, которому принадлежит данный сервер.

4. Крипто-Про (УЦ). Криптопровайдер компании »Крипто-Про», аутентификация с использованием сертификатов X.509, выпущенных Удостоверяющим центром (ключи ГОСТ 34.10-2001).

Для использования данного типа криптозащиты необходимо в команду "connect" добавить элемент:

```
<csp type="CPBB">  
    <storage>Имя хранилища личного сертификата пользователя  
    </storage>  
    <serial>Серийный номер личного сертификата пользователя</serial>  
    <server_cert>Полный путь к файлу сертификата сервера</server_cert>  
</csp>
```

При подключении к серверу может быть использован только один тип криптозащиты. Вариант криптозащиты, который должен использоваться при подключении к серверу, определяется Брокером, которому принадлежит данный сервер.

Пример подключения к серверу TRANSAQ с использованием криптозащиты RSA.

Пусть ключи клиента (файлы exchange.*) находятся в каталоге c:\transaq\keys\rsa\1771, а сертификат сервера (файл server.ms.cer) - в директории c:\transaq\keys\rsa\, тогда команда на подключение к серверу может иметь вид:

```
<command id='connect'>  
    <login>USERLOGIN</login>  
    <host>127.0.0.1</host>  
    <port>3900</port>  
    <logsdir>.\LOGS</logsdir>  
    <loglevel>0</loglevel>  
    <csp type="MS">  
        <key_path>c:\transaq\keys\rsa</key_path>  
        <key_id>1771</key_id>  
        <password>KEYPASSWORD</password>  
        <server_cert> c:\transaq\keys\rsa\server.ms.cer</server_cert>  
    </csp> </command>
```

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

3.2 Команда disconnect

Завершить подключение к Серверу.

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

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.

ПРИМЕЧАНИЕ. 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 Команда server_status

Получить информацию о текущем состоянии соединения с Сервером.

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

Результатом является структура "[server_status](#)".

3.4 Команда get_securities

Получить список доступных инструментов.

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

Результатом является структура "[securities](#)".

3.5 Команда subscribe

Запросить подписку на получение показателей торгов, сделок рынка и глубины рынка (стакана) по одному или нескольким инструментам.

1) Формат команды с использованием secid для идентификации бумаг

```
<command id="subscribe">
  <alltrades> - подписка на все сделки
    <secid>...</secid>
    ...
    <secid>...</secid>
  </alltrades>
  <quotations> - подписка на показатели торгов
    <secid>...</secid>
    ...
```

```

    <secid>...</secid>
</quotations>
<quotes> - подписка на изменения «стакана»
    <secid>...</secid>
    ...
    <secid>...</secid>
</quotes>
</command>

```

2) 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>

```

В результате выполнения команды приложение начинает получать изменения данных о торгах по заданным бумагам в виде сообщений [quotations](#), [alltrades](#), [quotes](#).

3.6 Команда unsubscribe

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

1) Формат команды с использованием secid для идентификации бумаг

```

<command id="unsubscribe">
  <alltrades> - отказ от подписки на все сделки
    <secid>...</secid>
    ...
    <secid>...</secid>
  </alltrades>
  <quotations> - отказ от подписки на показатели торгов
    <secid>...</secid>
    ...
    <secid>...</secid>
  </quotations>
  <quotes> - отказ от подписки на изменения «стакана»
    <secid>...</secid>
    ...
    <secid>...</secid>
  </quotes>
</command>

```

2) Формат команды с использованием элемента security для идентификации бумаг

```

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

```

```
...
</quotes>
</command>
```

Результатом выполнения команды является прекращение передачи изменений по заданным бумагам в виде сообщений [quotations](#), [alltrades](#), [quotes](#).

3.7 Команда `get_history_data`

Выдать последние N свечей заданного периода, по заданному инструменту.

1) Формат команды с использованием `secid` для идентификации бумаги

```
<command id="gethistorydata"  secid="..." period="идентификатор"
count="количество свечей" reset="true/false"/>
```

2) Формат команды с использованием элемента `security` для идентификации бумаги

```
<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`).

3.8 Команда `neworder`

Выставить новую заявку на Биржу

```
<command id="neworder">
  <secid>идентификатор бумаги</secid>
  <security>
    <board> идентификатор режима торгов </board>
    <seccode>Код инструмента</seccode>
  </security>
  <client>клиент</client>
  <union>union code :string</union>
```

```

<price>цена</price>
<hidden>Скрытое количество :integer</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: IOC,
                                FOK)
<usecredit/>
<nosplit/>
<expdate>дата экспирации (только для ФОРТС), задается в формате
23.07.2012 00:00:00</expdate> (не обязательно)
</command>

```

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

Для идентификации инструмента необходимо задать либо secid либо элемент <security>. Если заданы оба, то используется <security>.

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 FORTS 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 команды 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"/>
```

Transaction_id - 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.

Transaction_id 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).

ВНИМАНИЕ!

В условных заявках <orderno> будет нулевым до тех пор, пока заявка не будет принята Биржей (после того, как исполнится условие активации заявки)

Примечания brokerref позже будут возвращены в структурах order и trade. Сохранаются примечания локально на диске в файле notes.xml. При утрате этого файла, или при запуске dll из разных мест, примечания не сохраняются.

Примечание 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 Команда newcondorder

Выставить новую условную заявку на сервере TRANSAQ

```
<command id="newcondorder">
  <secid>идентификатор бумаги</secid>
  <security>
    <board> идентификатор режима торгов </board>
    <seccode>Код инструмента</seccode>
  </security>
  <client>клиент</client>
  <union>union code :string </union>
  <price>цена</price>
```

```

<hidden>Скрытое количество :integer</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), задается в формате
23.07.2012 00:00:00</expdate> (не обязательно)
</command>

```

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. Также validbefore может принимать текстовое значение "till_canceled", которое означает, что заявка должна быть действительна до ее отмены.

Для идентификации инструмента необходимо задать либо secid либо элемент <security>. Если заданы оба, то используется <security>.

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

После выполнения команды newcondorder возвращается либо сообщение об ошибке, либо структура <result> с номером транзакции.

До тех пор, пока не будут выполнены заданные пользователем условия, структура order условной заявки будет иметь статус watching и нулевой элемент <orderno>. Если после выполнения условий, заявка будет передана на биржу, то callback будет вызван еще раз со структурой order,

где элемент `<orderno>` будет содержать биржевой номер заявки.

3.10 Команда `newstoporder`

Выставить стоп-заявку:

```
<command id="newstoporder">
  <secid>идентификатор бумаги</secid>
  <security>
    <board> идентификатор режима торгов </board>
    <seccode>Код инструмента</seccode>
  </security>
  <client>идентификатор клиента</client>
  <union>union code :string </union>
  <buysell>B/S</buysell>
  <linkedorderno>номер связанной активной заявки</linkedorderno>
  (опционально)
  <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/> - «по рынку», в этом случае orderprice игнорируется
    <quantity>Количество</quantity>
    <usecredit/> - «использовать кредит»
    <guardtime>Защитное время</guardtime> (не обязательно)
    <brokerref>Примечание пользователя</brokerref> (не обязательно)
  </stoploss>
  <takeprofit>
    <activationprice>Цена активации</activationprice>
    <quantity>Количество</quantity>
    <usecredit/> - «использовать кредит»
    <guardtime>Защитное время</guardtime> (не обязательно)
    <brokerref>Примечание пользователя</brokerref> (не обязательно)
    <correction>Коррекция</correction> (не обязательно)
    <spread>Защитный спред</spread> (не обязательно)
    <bymarket/> «по рынку», в этом случае spread игнорируется
  </takeprofit>
</command>
```

`validfor` задаётся в формате даты, описаном выше. Для `validfor` значение "0" означает, что заявка будет действительна до конца сессии. Также `validfor` может принимать текстовое значение

"till_canceled", которое означает, что данная СТОП-заявка будет актуальна до тех пор, пока она не исполнится, либо не будет снята.

Для идентификации инструмента необходимо задать либо secid либо элемент <security>. Если заданы оба, то используется <security>.

Защитный спрэд и коррекцию можно задавать как в виде значения цены, так и в процентах от цены. Например: <correction>3.25</correction> или <correction>0.5%</correction> Количество инструмента также может быть задано либо в лотах, либо в процентах от текущей позиции. Например: <quantity>176</quantity> или <quantity>50%</quantity>

Примечание. Для стоп-заявок brokerref не сохраняется в файле notes_file, следовательно, для СТОП-заявок допустимы только короткие примечания.

3.11 Команда 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 Команда cancelstoporder

Отменить стоп-заявку

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

3.13 Команда get_forts_positions

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 Команда get_client_limits

Запрос лимитов клиента по FORTS:.

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

3.15 Команда get_markets

Запросить список доступных рынков:

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

3.16 Команда `get_servtime_difference`

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

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

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

3.17 Команда `get_leverage_control`

Начиная с версии 5.30.2.16, данная команда не поддерживается.

3.18 Команда `change_pass`

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

Запрос на смену пароля. Выполняется синхронно.

ВНИМАНИЕ! Значение атрибута `newpass` не должно содержать символа ';'.

3.19 Команда `subscribe_ticks`

1) Формат команды с использованием `secid` для идентификации бумаги

```
<command id="subscribe_ticks" filter="true/false">
  <security secid="..." tradeno="..." />
  ...
  <security secid="..." tradeno="..." />
</command>
```

2) Формат команды с использованием элемента `<security>` для идентификации бумаги

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

Подписка на тиковые данные. Если задан атрибут `filter="true"`, то будут отдаваться сделки только нормального периода торгов. По умолчанию отдаются все сделки.

В `tradenom` указывается номер сделки, после которой надо начать отдавать тики по данному

инструменту. Если задано tradeno=0, то отдаются сделки с момента выполнения подписки. Чтобы получить все известные серверу тики, нужно задать заведомо малое значение tradeno, например, tradeno=1.

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

3.20 Команда `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 Команда `get_news_body`

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

Запрос на получение тела новости для известного заголовка.

ВНИМАНИЕ! Команду “get_news_body” можно вызывать для получения только тех новостей, заголовки которых были получены внешним приложением в текущем сеансе подключения к серверу Транзак.

ВНИМАНИЕ! При реализации хранения новостей во внешнем приложении следует учитывать, что на каждом сервере Транзак ведется своя нумерация новостей. Для получения идентификатора сервера используйте команду “[get_server_id](#)”.

Значение атрибута “news_id” задает номер новости из полученного ранее заголовка, для которой необходимо получить тело.

После подключения к серверу внешнее приложение будет автоматически получать заголовки свежих новостей в виде структур [news_header](#).

Для получения заголовков более старых новостей необходимо выполнять команду “[get_old_news](#)”.

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

В ответ на данную команду внешнему приложению будет передано сообщение “[news_body](#)”.

3.22 Команда `get_old_news`

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

Запрос на получение заголовков более старых новостей.

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.

3.23 Команда **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 (например: 1.47)</connector_version>`

3.24 Команда **get_securities_info**

1) Формат команды с использованием secid для идентификации бумаги

```
<command id = "get_securities_info">
  <secid>идентификатор бумаги</secid>
</command>
```

2) Формат команды с использованием элемента <security> для идентификации бумаги

```
<command id = "get_securities_info">
  <security>
    <market>Внутренний код рынка :integer </market>
    <seccode>Код инструмента </seccode>
  </security>
</command>
```

Запрос на получение информации по инструменту.

В ответ на данную команду внешнему приложению возвращается структура [<sec_info>](#).

3.25 Команда 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.

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.

Немедленным результатом команды moveorder будет структура <result>, сообщающая о том, была ли принята заявка сервером TRANSAQ. В случае успеха результатом будет следующий тег:

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

После того, как заявка будет принята Биржей, в асинхронном режиме будет получена структура [<orders>](#), имеющая в составе данных ссылку на номер транзакции.

3.26 Команда 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>](#).

3.27 Команда 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 фондовых рынков и валютной секции МБ.

1) Формат команды с использованием secid для идентификации бумаг

```
<command id="get_max_buy_sell_tplus" client="код клиента">
    <secid>идентификатор бумаги :integer</secid>
    .....
    <secid>идентификатор бумаги :integer</secid>
</command>
```

2) Формат команды с использованием элемента security для идентификации бумаг

```
<command id="get_max_buy_sell_tplus" client="код клиента">
    <security>
        <market>Внутренний код рынка :integer</market>
        <seccode>Код инструмента :string</seccode>
    </security>
    .....
    <security>
        <market>Внутренний код рынка :integer </market>
        <seccode>Код инструмента :string</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>](#).

3.28 Команда get_portfolio_mct

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

Команда имеет один обязательный параметр - клиент, для которого запрашивается портфель МСТ.

Запрос по данной команде выполняется однократно, то есть на одну команду приходит ровно

один

При необходимости получить обновление данных команду нужно выполнить вновь. Ответная структура всегда приходит целиком, инкрементального обновления не предусматривается.

Результатом команды является структура [<portfolio_mct>](#)

3.29 Команда **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 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 [<united_portfolio>](#)

3.30 Команда **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>](#)

3.31 Команда **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>](#)

3.32 Команда get_cln_sec_permissions

Получить свойства конкретного инструмента, действующие в отношении конкретного клиента.

Возможны два варианта команды:

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

или

```
<command id="get_cln_sec_permissions">  
  <secid>идентификатор бумаги</secid>  
  <client>клиент</client>  
</command>
```

Результатом команды является структура [<cln_sec_permissions>](#)

4. Список возможных сообщений, передаваемых callback функции

4.1 Исторические данные

```
<candles secid=".. :integer" period="идентификатор периода :integer"
    status=".. :integer" board=" Идентификатор режима торгов
    :string " seccode="Код инструмента:string">

    <candle date="дата :date" open=".. :double" high=".. :double"
    low=".. :double" close=".. :double" volume=".. :integer"
    oi="open_interest :integer"/>

<candles/>
```

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

- 0 - данных больше нет (дочерпали до дна)
- 1 - заказанное количество выдано, если нужны еще данные – можно выполнять еще команду
- 2 - продолжение следует, будет еще порция данных по этой команде
- 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 :integer" connected="true/false/error
:string" recover="true/атрибут отсутствует :string " server_tz=" имя
таймзоны сервера :string" sys_ver="версия системы : integer"
build="билд сервера : integer">
```

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>
```

4.3 Клиентские счета

```

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

```

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.

Значение remove="true" означает, что клиент доступен. Значение remove="false" означает, что клиент не доступен. При удалении клиента структура не содержит никаких параметров клиента.

Возможные типы клиента: spot (кассовый), leverage (маржинальный), mct (клиент MMA)

Валюта фондового портфеля currency может принимать следующие значения: NA (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 :integer">название рынка :string
</market>
</markets>
```

4.5 Информация о доступных периодах свечей

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

4.6 Список инструментов

```
<securities>
  <security secid="внутренний код :integer" active="true/false
:string" >
    <seccode> Код инструмента:string </seccode>
    <instrclass>Символ категории (класса)
инструмента:string</instrclass>
    <board>Идентификатор режима торгов по умолчанию:string</board>
    <market>Идентификатор рынка :integer(:string в случае ошибки)
</market>
    <shortname>Наименование бумаги :string</shortname>
```

```

<decimals>Количество десятичных знаков в цене :integer
</decimals>

<minstep>Шаг цены :double</minstep>

<lotsize>Размер лота :integer</lotsize>

<point_cost>Стоимость пункта цены :double</point_cost>

<opmask usecredit="yes/no :string" bymarket="yes/no :string"
nosplit="yes/no :string" fok="yes/no :string" ioc="yes/no
:string"/>

<sectype>Тип бумаги :string</sectype>

<sec_tz> имя таймзоны инструмента (типа "Russian Standard Time",
"USA=Eastern Standard Time"), содержит секцию CDATA
:string</sec_tz>

<quotestype>
    0 - без стакана
    1 - стакан типа OrderBook
    2 - стакан типа Level2
</quotestype>

<MIC>код биржи листинга по стандарту ISO :string </MIC>

<ticker>тикер на бирже листинга :string </ticker>

</security>
</securities>

```

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.

Инструменты рынка МСТ всегда обязательно имеют категорию (instrclass). Для инструментов других рынков элемент instrclass может быть пустым.

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

sectype может принимать следующие значения:

- Торгуемые инструменты:
 - SHARE - акции
 - BOND - облигации корпоративные
 - FUT - фьючерсы FORTS
 - OPT - опционы
 - GKO - гос. Бумаги (не используется)
 - FOB - фьючерсы ММББ (не используется)

- 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 - нефтянка
- Так же:
 - 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](#)).

Чтобы получить стоимость шага цены необходимо значение элемента <point_cost> умножить на значение элемента <minstep>.

ПРИМЕЧАНИЕ

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 Показатели торгов по инструментам

```
<quotations>
  <quotation secid ="внутренний код">
    <board> Идентификатор режима торгов :string <board>
    <seccode>Код инструмента:string</seccode>
    <accruedintvalue>НКД на дату торгов в расчете на одну бумагу,
    руб. :double </accruedintvalue>
    <open>Цена первой сделки. :double</open>
    <waprice>Средневзвешенная цена :double</waprice>
    <biddepth> Кол-во лотов на покупку по лучшей цене :integer
    </biddepth>
    <biddeptht>Совокупный спрос :integer</biddeptht>
```

```

<numbids>Заявок на покупку :integer</numbids>
<offerdepth>Кол-во лотов на продажу по лучшей цене :integer
</offerdepth>
<offerdeptht>Совокупное предложение :integer</offerdeptht>
<bid>Лучшая цена на покупку :double</bid>
<offer>Лучшая цена на продажу :double</offer>
<numoffers>Заявок на продажу :integer</numoffers>
<numtrades>Сделок :integer</numtrades>
<voltoday>Объем совершенных сделок в лотах :integer</voltoday>
<openpositions>Общее количество открытых позиций (FORTS) :integer
</openpositions>
<deltapositions>Изм.открытых позиций (FORTS) :integer
</deltapositions>
<last>Цена последней сделки :double</last>
<quantity>Объем последней сделки, в лотах. :integer</quantity>
<time>Время заключения последней сделки :date</time>
<change>Абсолютное изменение цены последней сделки по отношению
к цене последней сделки предыдущего торгового дня
:double</change>
<priceminusprevwaprice>Цена последней сделки к оценке
предыдущего дня :double</priceminusprevwaprice>
<valtoday>Объем совершенных сделок, млн. руб. :double</valtoday>
<yield>Доходность, по цене последней сделки :double</yield>
<yieldatwaprice>Доходность по средневзвешенной цене :double
</yieldatwaprice>
<marketpricetoday>Рыночная цена по результатам торгов
сегодняшнего дня :double</marketpricetoday>
<highbid>Наибольшая цена спроса в течение торговой сессии
:double </highbid>
<lowoffer>Наименьшая цена предложения в течение торговой сессии
:double</lowoffer>
<high>Максимальная цена сделки :double</high>
<low>Минимальная цена сделки :double</low>
<closeprice>Цена закрытия :double</closeprice>
<closeyield>Доходность по цене закрытия :double</closeyield>
<status>Статус «торговые операции разрешены/запрещены» :string
</status>
<tradingstatus>Состояние торговой сессии по инструменту :string
</tradingstatus>
<buydeposit>ГО покупок/покр :double</buydeposit>
<selldeposit>ГО продаж/непокр :double</selldeposit>
<volatility>Волатильность :double</volatility >
<theoreticalprice>Теоретическая цена :double</theoreticalprice>
<bgo_buy>Базовое ГО под покупку маржируемого опциона :double

```

```

</bgo_buy>
<point_cost>Стоимость пункта цены :double</point_cost>

<lcurrentprice> Официальная текущая цена Биржи :double
</lcurrentprice>
</quotation>
<quotation secid ="внутренний код :integer">
    ...
</quotation>
</quotations>

```

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.

4.8 Сделки рынка по инструментам

```

<alltrades>
  <trade secid ="внутренний код :integer">
    <seccode>Код инструмента:string</seccode>
    <tradenom>биржевой номер сделки :integer64</tradenom>
    <time>время сделки :date</time>
    <board> Идентификатор режима торгов :string </board>
    <price>цена сделки :double</price>
    <quantity>объем сделки :integer</quantity>
    <buysell>покупка (B) / продажа (S) :string</buysell>
    <openinterest>... :integer</openinterest>
    <period>Период торгов (O - открытие, N - торги, C - закрытие)
    :string </period>
  </trade>
  <trade secid ="внутренний код">
    ...
  </trade>
</alltrades>

```

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.

4.9 Глубина рынка по инструментам (“стакан”)

```

<quotes>
  <quote secid ="внутренний код :integer">

```

```

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

```

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».

4.10 Заявки клиентов

```

<orders>

  <order transactionid ="идентификатор транзакции сервера Transaq
  :integer">

    <orderno>биржевой номер заявки :integer64</orderno>

    <secid>идентификатор бумаги :integer</secid>

    <board> Идентификатор режима торгов :string </board>

    <seccode>Код инструмента:string</seccode>

    <client>идентификатор клиента :string</client>

    <union>Код юниона :string</union>

    <status>статус заявки (см. ниже в таблице 3) :string</status>

    <buysell>покупка (B) / продажа (S) :string</buysell>

    <time>время регистрации заявки биржей :date</time>

    <expdate>дата экспирации (только для ФОРТС) :date </expdate>

    <origin_orderno>первоначальный биржевой номер заявки ФОРТС, в
    которой задана дата экспирации :integer64</origin_orderno>

    <accepttime>время регистрации заявки сервером Transaq (только
    для условных заявок) :date</accepttime>

    <brokerref>примечание :string</brokerref>

    <value>объем заявки в копейках :double</value>

    <accruedint>НКД :double</accruedint>

    <settlecode>Код поставки (значение биржи, определяющее правила
    расчетов - смотрите подробнее в документах биржи) :string
    </settlecode>

    <balance>Неудовлетворенный остаток объема заявки в лотах
    (контрактах) :integer</balance>

    <price>Цена :double</price>

    <quantity>Количество :integer</quantity>

    <hidden>Скрытое количество :integer</hidden>

    <yield>Доходность :double</yield>

    <withdrawtime>Время снятия заявки, 0 для активных :date
    </withdrawtime>

    <condition>Условие, см. Newcondorder :string</condition>

    <conditionvalue>Цена для условной заявки, либо обеспеченность в
    процентах :double</conditionvalue>

    <validafter>с какого момента времени действительна (см.
    newcondorder) :date </validafter>

    <validbefore>до какого момента действительно (см. newcondorder)
    :date </validbefore>

    <maxcomission>максимальная комиссия по сделкам заявки :double
    </maxcomission>

    <result>сообщение биржи в случае отказа выставить заявку :string
    </result>

  </order>

```

```

<stoporder transactionid="идентификатор стопа">
  <activeorderno>:integer64</activeorderno>
  <secid>:integer</secid>
  <board>:string</board>
  <seccode>Код инструмента:string</seccode>
  <client>:string</client>
  <union>Код юниона :string</union>
  <buysell>:string</buysell>
  <canceller>:string</canceller>
  <alltradeno>:integer64</alltradeno>
  <validbefore>:data</validbefore>
  <author>:string</author>
  <accepttime>:data</accepttime>
  <linkedorderno>:integer64</linkedorderno>
  <expdate>:data</expdate>
  <status>:string</status>
  <stoploss usecredit="yes/no :string">
    <activationprice>:double</activationprice>
    <guardtime>:data</guardtime>
    <brokerref>:string</brokerref>
    <quantity>:integer(:double в случае %)</quantity>
    <bymarket/>
    <orderprice>:double </orderprice>
  </stoploss>
  <takeprofit>
    <activationprice>:double</activationprice>
    <guardtime>:data</guardtime>
    <brokerref>:string</brokerref>
    <quantity>:integer(:double в случае %)</quantity>
    <extremum>:double</extremum>
    <level>:double</level>
    <correction>:double</correction>
    <guardspread>:double</guardspread>
  </takeprofit>
</stoporder>

<negdeal transactionid = "идентификатор транзакции
  сервера Transaq :integer">
  <secid>идентификатор бумаги :integer</secid>
  <inout> 1 - отправленная, 2 - полученная, 3 - внутренняя
  :integer </inout>

```

```

<orderno> биржевой номер заявки :integer64</orderno>
<buysell> направление заявки: 'B' - покупка 'S' - продажа) :string
</buysell>
<board> идентификатор режима торгов :string </board>
<seccode>Код инструмента:string</seccode>
<price2> цена выкупа РЕПО :double </price2>
<repovaluel> сумма РЕПО :double </repovaluel>
<repo2value> стоимость выкупа РЕПО :double </repo2value>
<matchref> ссылка:string </matchref>
<client>код клиента в СВО TRANSAQ :string </client>
<union>Код юниона :string</union>
<price>цена заявки :double </price>
<items>кол-во инструмента в сделке (шт.) :integer64 </items>
<volume>объем сделки без учета НКД (руб) :double </volume>
<accruedint>НКД на единицу инструмента (руб) :double
</accruedint>
<maxcomission>максимальная комиссия по сделкам
заявки (руб) :double </maxcomission>
<time>дата и время регистрации заявки на Бирже :data </time>
<author>идентификатор трейдера :string </author>
<brokerref>примечание трейдера, подавшего заявку :string
</brokerref>
<settlecode>код поставки :string </settlecode>
<cpfirmid>Идентификатор фирмы-контрагента :string </cpfirmid>
<repoterm>срок РЕПО, дней :integer </repoterm>
<reporate>ставка РЕПО :double </reporate>
<startdiscount>начальный дисконт :double </startdiscount>
<lowerdiscount>нижний предел дисконта :double </lowerdiscount>
<upperdiscount>верхний предел дисконта :double </upperdiscount>
<activationtime>дата и время активации заявки :data
</activationtime>
<blocksecurities>признак блокировки бумаг на время РЕПО
'Y'/'N' :string </blocksecurities>
<withdrawtime>дата и время снятия заявки :data </withdrawtime>
<ordertype> тип адресной заявки:
    • RPS - РПС
    • EXTCREPO
    • EXTREPO - модифицированное РЕПО
    • REPO - классическое РЕПО

```

```

    </ordertype>
    <status> статус заявки (см. ниже в таблице 3) :string
  </status>
</negdeal>
</orders>

```

Передается автоматически после установки соединения, а так же 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.

Элемент negdeal содержит информацию об адресных заявках.

Для условных заявок: поле validafter принимает значение "0", если заявка начинает действовать немедленно; поле validbefore принимает значение "0", если заявка действительна до конца сессии; поле validbefore пустое (<validbefore></validbefore>), если заявка действительна до отмены.

Для стоп-заявок: поле validbefore отсутствует, если заявка действительна до конца сессии; поле validbefore пустое (<validbefore></validbefore>), если заявка действительна до отмены.

4.11 Сделки клиентов

```

<trades>
  <trade>
    <secid>Идентификатор бумаги :integer</secid>
    <tradenno>Номер сделки на бирже :integer64</tradenno>
    <orderno>Номер заявки на бирже :integer64</orderno>
    <board>Идентификатор режима торгов :string</board>
  </trade>
</trades>

```

```

<seccode>Код инструмента:string</seccode>
<client>Идентификатор клиента :string</client>
<union>Код юниона :string</union>
<buysell>B - покупка, S - продажа :string</buysell>
<time>время сделки :date</time>
<brokerref>примечание :string</brokerref>
<value>объем сделки :double</value>
<comission>комиссия :double</comission>
<price>цена :double</price>
<items>кол-во инструмента в сделках в штуках:integer64</items>
<quantity>количество лотов :integer</quantity>
<yield>доходность :double</yield>
<accruedint>НКД :double</accruedint>
<tradetype>тип сделки: 'T' - обычная 'N' - РПС 'R' - РЕПО
'P' - размещение :string</tradetype>
<settlecode>код поставки :string</settlecode>
<currentpos>Текущая позиция :integer64</currentpos>
</trade>
</trades>

```

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

4.12 Позиции клиентов

```

<positions>
  <money_position>
    <markets>
      <market>Внутренний код рынка :integer </market>
      ...
      <market>Внутренний код рынка :integer </market>
    </markets>
    <register>Регистр учета :string</register>
    <asset>Код вида средств :string</asset>
    <client>Идентификатор клиента :string</client>
    <union>Код юниона :string</union>
    <shortname>Наименование вида средств :string</shortname>
    <saldoin>Входящий остаток :double</saldoin>
    <bought>Куплено :double</bought>
    <sold>Продано :double</sold>
    <saldo>Текущее сальдо :double</saldo>
    <ordbuy>В заявках на покупку + комиссия :double</ordbuy>
  </money_position>
</positions>

```

```

<ordbuycond>В условных заявках на покупку :double</ordbuycond>
<comission>Сумма списанной комиссии :double</comission>
</money_position>
<sec_position>
  <secid>Код инструмента :integer</secid>
  <market>Внутренний код рынка :integer </market>
  <seccode>Код инструмента:string</seccode>
  <register>Регистр учета:string</register>
  <client>Идентификатор клиента :string</client>
  <union>Код юниона :string</union>
  <shortname>Наименование инструмента :string</shortname>
  <saldoin>Входящий остаток :integer64</saldoin>
  <saldomin>Неснижаемый остаток :integer64</saldomin>
  <bought>Куплено :integer64</bought>
  <sold>Продано :integer64</sold>
  <saldo>Текущее сальдо :integer64</saldo>
  <ordbuy>В заявках на покупку :integer64</ordbuy>
  <ordsell>В заявках на продажу :integer64</ordsell>
  <amount>Текущая оценка стоимости позиции, в валюте инструмента
  :double. Примечание 1</amount>
  <equity>Текущая оценка стоимости позиции, в рублях :double
  Примечание 1</equity>
</sec_position>
<forts_position>
  <secid>Код инструмента :integer</secid>
  <markets>
    <market>Внутренний код рынка :integer </market>
    ...
    <market>Внутренний код рынка :integer </market>
  </markets>
  <seccode> Код инструмента:string </seccode>
  <client>Идентификатор клиента :string</client>
  <union>Код юниона :string</union>
  <startnet>Входящая позиция по инструменту :integer</startnet>
  <openbuys>В заявках на покупку :integer</openbuys>
  <opensells>В заявках на продажу :integer</opensells>
  <totalnet>Текущая позиция по инструменту :integer</totalnet>
  <todaybuy>Куплено :integer</todaybuy>
  <todaysell>Продано :integer</todaysell>
  <optmargin>Маржа для маржируемых опционов :double</optmargin>
  <varmargin>Вариационная маржа :double</varmargin>

```

```

<expirationpos>Опционов в заявках на исполнение
:integer64</expirationpos>

<usedsellspotlimit>Объем использованного спот-лимита на продажу
:double</usedsellspotlimit>

<sellspotlimit>текущий спот-лимит на продажу, установленный
Брокером :double</sellspotlimit>

<netto>нетто-позиция по всем инструментам данного спота :double
</netto>

<kgo>коэффициент ГО c</kgo>
</forts_position>
<forts_money> - деньги ФОРТС
  <client>Идентификатор клиента :string</client>
  <union>Код юниона :string</union>
  <markets>
    <market>Внутренний код рынка :integer </market>
    ...
    <market>Внутренний код рынка :integer </market>
  </markets>
  <shortname>Наименование вида средств :string</shortname>
  <current>Текущие :double</current>
  <blocked>Заблокировано :double</blocked>
  <free>Свободные :double</free>
  <varmargin>Опер. Маржа :double</varmargin>
</forts_money>
<forts_collaterals> - залоги ФОРТС
  <client>Идентификатор клиента :integer</client>
  <union>Код юниона :string</union>
  <markets>
    <market>Внутренний код рынка :integer </market>
    ...
    <market>Внутренний код рынка :integer </market>
  </markets>
  <shortname>Наименование вида средств :string</shortname>
  <current>Текущие :double</current>
  <blocked>Заблокировано :double</blocked>
  <free>Свободно :double</free>
</forts_collaterals>
<spot_limit> - лимиты ФОРТС
  <client>Идентификатор клиента :integer</client>
  <markets>
    <market>Внутренний код рынка :integer </market>

```

```

...
    <market>Внутренний код рынка :integer </market>
  </markets>
  <shortname>Наименование вида средств :string</shortname>
  <buylimit>Текущий лимит :double</buylimit>
  <buylimitused>Заблокировано лимита :double</buylimitused>
</spot_limit>
</positions>

```

Передается автоматически после установки соединения, а так же 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. (см. [4.4 Доступные рынки](#)).

`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 ММВБ будут отдаваться только данные с регистром "T0".

Если `micex_registers=true`, то в `<money_position>` и `<sec_position>` для фондового рынка ММВБ 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 :string">
    <open_equity>Входящая оценка стоимости единого портфеля

```

```

:double</open_equity>
<equity>Текущая оценка стоимости единого портфеля
:double</equity>
<requirements>Начальные требования :double</requirements>
<free>Свободные средства :double</free>
<vm>Вариационная маржа FORTS :double</vm>
<finres>Финансовый результат последнего клиринга FORTS
:double</finres>
<go>Размер требуемого ГО, посчитанный биржей FORTS :double</go>
</united_limits>
<positions>

```

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

```

<positions>
  <forts_money>
  <client>Клиент :string</client>
  <union>Юнион :string</union>
  <current>Текущие в ЕП :double</current>
  <free>Свободные в ЕП :double</free>
  </forts_money>
</positions>

```

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).

Примечание 1. Поля amount и equity передаются только в случае наличия в команде connect элемента <push_pos_equity>

4.13 Лимиты клиента на срочном рынке

```

<clientlimits client="клиент :string">
  <cbplimit>стоимостной лимит открытых позиций (СЛОП срочн. рынок
ММВБ) :double</cbplimit>
  <cbplused>стоимостная оценка текущих чистых позиций (СОЧП срочн.
рынок ММВБ) :double</cbplused>
  <cbplplanned>СОЧП с учетом активных заявок (срочный рынок ММВБ)
:double </cbplplanned>
  <fob_varmargin>Вар. маржа срочного рынка ММВБ :double
</fob_varmargin>

```

```

<coverage>Обеспеченность срочного портфеля (FORTS) :double
</coverage>

<liquidity_c>Коэффициент ликвидности(FORTS) :double</liquidity_c>

<profit>Доход(FORTS) :double</profit>

<money_current>Деньги текущие. То же, что current в forts_money
:double</money_current >

<money_reserve>Деньги заблокированные. То же, что blocked в
forts_money :double</money_reserve>

<money_free>Деньги свободные. То же, что free в forts_money
:double</money_free >

<options_premium>Премии по опционам(FORTS) :double
</options_premium>

<exchange_fee>Биржевой сбор(FORTS) :double</exchange_fee>

<forts_varmargin>Вар. маржа текущая (FORTS) :double
</forts_varmargin>

<varmargin>Операционная маржа. То же, что varmargin в forts_money
:double</varmargin>

<pclmargin> Перечисленная в пром.клиринге вариационная
маржа(FORTS) :double</pclmargin>

<options_vm> Вар. маржа по опционам(FORTS) :double</ options_vm >

<spot_buy_limit>Лимит на покупку спот :double</spot_buy_limit>

<used_stop_buy_limit> Лимит на покупку спот использованный :double
</used_spot_buy_limit>

<collat_current>Залоги текущие. То же, что current в
forts_collaterals :double</collat_current>

<collat_blocked>Залоги заблокированные. То же, что blocked в
forts_collaterals :double</collat_blocked>

<collat_free> Залоги свободные. То же, free в forts_collaterals
:double<collat_free>

</clientlimits>

```

4.14 Режим кредитования

```
<overnight status="true/false :string"/>
```

Ночной или дневной режим кредитования (больше не используется в системе).

4.15 Признак допустимости рыночных заявок для инструмента

```

<marketord secid="id :integer" seccode=" Код инструмента:string"
permit="yes/no :string"/>

```

Передается в случае, если во время работы Коннектора изменится возможность выставления рыночных заявок по данному инструменту.

4.16 Максимальная покупка/продажа и плечо для режимов T0

```

<leverage_control client="код клиента :string"
leverage_plan="плановое плечо :double" leverage_fact="фактическое
плечо :double">

```

```

<security secid="код бумаги :integer" board=" Идентификатор режима
торгов :string " seccode=" Код инструмента:string "
maxbuy="макс.покупка :integer64" maxsell="макс.продажа :integer64"
/>

<security secid="код бумаги" board=" Идентификатор режима торгов
:string " seccode=" Код инструмента:string " maxbuy="макс.покупка"
maxsell="макс.продажа" />

.....

<security secid="код бумаги" board=" Идентификатор режима торгов
:string " seccode=" Код инструмента:string " maxbuy="макс.покупка"
maxsell="макс.продажа" />

</leverage_control>

```

Начиная с версии 5.30.2.16, leverage_control не передается.

4.17 Тиковые данные

```

<ticks>
  <tick>
    <secid>идентификатор бумаги :integer</secid>
    <tradeno>номер сделки :integer64</tradeno>
    <tradetime>время сделки :date</tradetime>
    <price>цена :double</price>
    <quantity>количество лотов (контрактов) :integer</quantity>
    <period>торговый период (O - открытие, N - основные торги, C -
заккрытие; передается только для ММВБ) :string</period>
    <buysell>В - покупка, S - продажа (с точки зрения того, кто
инициировал сделку, приняв условия выставленной ранее заявки.
Передается только когда есть такая информация) :string
    </buysell>
    <openinterest>кол-во открытых позиций на срочном рынке
    :integer</openinterest>
    <board>Идентификатор режима торгов :string</board>
    <seccode>Код инструмента:string</seccode>
  </tick>
</ticks>

```

4.18 Идентификатор сервера

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

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

4.19 Тело новости

```

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

```

```

</text>
</news_ body >

```

4.20 Заголовок новости

```

<news_header>
  <id> порядковый номер новости :integer</id>
  <time_stamp> дата-время новости (от источника) :date</time_stamp>
  <source><![CDATA[ источник новости ]]> :string</source>
  <title><![CDATA[ заголовок новости ]]> :string</title>
</news_header >

```

4.21 Информация по инструменту

```

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

```

```

    <lot_volume>количество базового актива (FORTS) :integer
  </lot_volume>

</sec_info>

```

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

4.22 Обновление информации по инструменту

```

<sec_info_upd>
  <secid>идентификатор бумаги :integer</secid>

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

</sec_info_upd>

```

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

4.23 Справочник режимов торгов

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

4.24 Параметры инструмента в режиме торгов

```
<pit seccode="Код инструмента:string">
  board="Идентификатор режима торгов :string">
    <market>Идентификатор рынка :integer </market>

    <decimals>Количество десятичных знаков в цене :integer </decimals>

    <minstep>Шаг цены:double</minstep>

    <lotsize>Размер лота:integer</lotsize>

    <point_cost>Стоимость пункта цены :double</point_cost>
  </pit>
```

4.25 Клиентский портфель T+

```
<portfolio_tplus client="код клиента :string">
  <coverage_fact>Фактическая обеспеченность, %
    :double</coverage_fact>

  <coverage_plan>Плановая обеспеченность, % :double</coverage_plan>

  <coverage_crit>критическая
    обеспеченность, % :double</coverage_crit>
```

```
<open_equity>Входящая оценка портфеля без дисконта
:double</open_equity>

<equity>Текущая оценка портфеля без дисконта :double</equity>

<cover>Плановое обеспечение (оценка стоимости портфеля)
:double</cover>

<init_margin>Плановая начальная маржа (оценка портфельного риска)
:double</init_margin>

<pnl_income>Прибыль/убыток по входящим позициям
:double</pnl_income>

<pnl_intraday>Прибыль/убыток по сделкам :double</pnl_intraday>

<leverage>Фактическое плечо портфеля T+ :double</leverage>

<margin_level>Фактический уровень маржи портфеля T+ :double
</margin_level>

<money>

  <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>

<security secid="id инструмента :integer">

  <market>id рынка :integer</market>
  <seccode>Код инструмента :string</seccode>
  <price>Текущая цена :double</price>
  <open_balance>Входящая позиция, штук :integer</open_balance>
  <bought>Куплено, штук :integer</bought>
```

```

<sold>Продано, штук :integer</sold>
<balance>Текущая 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>
<cover>Вклад бумаги в плановое обеспечение :double</cover>
<init_margin>Плановая начальная маржа (риск)
:double</init_margin>
<riskrate_long>Ставка риска для лонгов,
% :double</riskrate_long>
<riskrate_short>Ставка риска для шортов,
% :double</riskrate_short>
<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>)
</portfolio_tplus>

```

4.28 Клиентский портфель МСТ

```

<portfolio_mct client="код клиента :string">
  <portfolio_currency>"валюта портфеля
клиента:string"</portfolio_currency>
  <capital>"величина капитала :double"</capital>
  <utilization_fact>"использование капитала факт :double"
</utilization_fact>

```

```

<utilization_plan>"использование капитала план :double"
</utilization_plan>

<coverage_fact>"фактическая обеспеченность
:double"</coverage_fact>

<coverage_plan>"плановая обеспеченность :double"</coverage_plan>

<open_balance>"входящее сальдо :double"</open_balance>

<tax>"суммарная комиссия :double"</tax>

<pnl_income>"суммарная прибыль/убыток по входящим позициям :
double"</pnl_income>

<pnl_intraday>"суммарная прибыль/убыток по сделкам
:double"</pnl_intraday>

<security secid="id инструмента:integer">
  <market>Внутренний код рынка :integer</market>
  <seccode>"код инструмента:string"</seccode>
  <security_currency>"валюта цены
инструмента:string"</security_currency>
  <go_rate>"ставка ГО (либо long, либо short, в зависимости от
позиции клиента) по инструменту для клиента :double"</go_rate>
  <go_rate_long>"ставка ГО long по инструменту для клиента
:double"</go_rate_long >
  <go_rate_short>"ставка ГО short по инструменту для клиента
:double"</go_rate_short>
  <price>"текущая цена инструмента в валюте инструмента :double
"</price>
  <init_rate>"входящая цена позиции (цена последнего клиринга
:double)"</init_rate>
  <cross_rate>"кросс-курс валюты портфеля к валюте контракта
:double"</cross_rate>
  <init_cross_rate>"входящий кросс-курс валюты портфеля к валюте
контракта :double"</init_cross_rate>
  <open_balance>"входящая позиция, штук :integer"</open_balance>
  <bought>"куплено, штук :integer"</bought>
  <sold>"продано, штук :integer"</sold>
  <balance>"текущая 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>
  <pos_cost>"текущая стоимость позиции :double"</pos_cost>
  <go_pos_fact>"ГО позиции факт :double"</go_pos_fact>
  <go_pos_plan>"ГО позиции план :double"</go_pos_plan>
  <tax>"комиссия по сделкам в инструменте :double"</tax>
  <pnl_income>"прибыль/убыток по входящим позициям в инструменте

```

```

:double"</pnl_income>
<pnl_intraday>"прибыль/убыток по сделкам в инструменте
:double"</pnl_intraday>
<maxbuy>"макс. возможная покупка, в лотах :integer"</maxbuy>
<maxsell>"макс. возможная продажа, в лотах :integer"</maxsell>
<bought_average>"средняя цена покупки :double"</bought_average>
<sold_average>"средняя цена продажи :double"</sold_average>
</security>
... (<security>)
... (<security>)
</portfolio_mct>

```

4.29 Клиентский единый портфель

```

<united_portfolio
    union="код юниона :string" client="код клиента :string">
    <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 ММА, 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>
        <reserate_long>Ставка резерва для лонгов, %
        :double</reserate_long>
        <reserate_short>Ставка резерва для шортов, %
        :double</reserate_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>

```

4.30 Актуальная оценка ликвидационной стоимости Единого портфеля

```

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

```

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

```

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

```

4.32 Максимальная покупка/продажа для режимов T+

```

<max_buy_sell_tplus client="код клиента: integer">
  <security secid="код бумаги :integer">
    <market>Внутренний код рынка :integer</market>
    <seccode>Код инструмента :string</seccode>

```

```

    <maxbuy>максимум купить (лот.):integer64</maxbuy>

    <maxsell>максимум продать (лот.):integer64</maxsell>

</security>

</max_buy_sell_tplus>

```

4.33 Текстовые сообщения

```

<messages>
  <message>

    <date>Дата и время :date </date>

    <urgent>Срочное: Y/N :string </urgent>

    <from>Отправитель :string </from>

    <text>Текст сообщения :string </text>

  </message>

</messages>

```

Поле text содержит секцию CDATA.

4.34 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.

4.35 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>

```

4.36 Возможные ошибки

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

```

<error>Описание ошибки :string</error>

```

5 Дополнительные команды и сообщения для работы с адресными сделками ММВБ

5.1 Команда newrpsorder

Выставить новую заявку РПС на ММВБ

```
<command id="newrpsorder">
  <secid> идентификатор бумаги </secid>

  <security>
    <board> идентификатор режима торгов </board>
    <seccode> Код инструмента </seccode>
  </security>
  <client> идентификатор клиента </client>

  <buysell> B/S </buysell>

  <cpfirmid> Идентификатор фирмы-контрагента </cpfirmid>

  <matchref> ссылка (необязательный параметр) </matchref>

  <brokerref> примечание (необязательный параметр) </brokerref>

  <price> Цена </price>

  <quantity> Количество </quantity>

  <settlecode> Код поставки </settlecode>

  <refundrate> Ставка возмещения, сотые процента
    </refundrate>

</command>
```

Для идентификации инструмента необходимо задать либо secid либо элемент <security>. Если заданы оба, то используется <security>.

5.2 Команда newmrepoorder

Выставить новую заявку РЕПО-М на ММВБ

```
<command id="newmrepoorder">
  <secid> идентификатор бумаги </secid>

  <security>
    <board> идентификатор режима торгов </board>
    <seccode> Код инструмента </seccode>
  </security>
  <client> идентификатор клиента </client>
```

```

<buysell> B/S </buysell>

<cpfirmid> Идентификатор фирмы-контрагента </cpfirmid>

<matchref> ссылка (необязательный параметр)</matchref>

<brokerref> примечание (необязательный параметр)</brokerref>

<price> Цена </price>

<quantity> Количество </quantity>

<settlecode> Код поставки </settlecode>

<refundrate> Ставка возмещения, сотые процента (необязательный
параметр)</refundrate>

<value> Объем репо </value>

<repoterm> срок РЕПО, дней (необязательный параметр)</repoterm>

<reporate> ставка РЕПО (необязательный параметр)</reporate>

<startdiscount> начальный дисконт (необязательный
параметр)</startdiscount>

<lowerdiscount> нижний предел дисконта (необязательный
параметр)</lowerdiscount>

<upperdiscount> верхний предел дисконта (необязательный
параметр)</upperdiscount>

  <blocksecurities> признак блокировки бумаг на время РЕПО
  'Y'/'N' (необязательный параметр)</blocksecurities>

</command>

```

Для идентификации инструмента необходимо задать либо `secid` либо элемент `<security>`. Если заданы оба, то используется `<security>`.

5.3 Команда `newrepoorder`

Выставить новую заявку РЕПО на ММВБ

```

<command id="newrepoorder">
  <secid> идентификатор бумаги </secid>

  <security>
    <board> идентификатор режима торгов </board>
    <seccode> Код инструмента </seccode>
  </security>

  <client> идентификатор клиента </client>

```

```

<buysell> B/S </buysell>

<cpfirmid> Идентификатор фирмы-контрагента </cpfirmid>

<matchref> ссылка (необязательный параметр)</matchref>

<brokerref> примечание (необязательный параметр)</brokerref>

<price> Цена </price>

<quantity> Количество </quantity>

<settlecode> Код поставки </settlecode>

<refundrate> Ставка возмещения, сотые процента (необязательный
параметр)</refundrate>

<reporate> ставка РЕПО (необязательный параметр)</reporate>

</command>

```

Для идентификации инструмента необходимо задать либо `secid` либо элемент `<security>`. Если заданы оба, то используется `<security>`.

5.4 Команда `cancelnegdeal`

Снять заявку РПС, РЕПО, РЕПО-М

```

<command id="cancelnegdeal">
  <transactionid> идентификатор транзакции сервера Transaq
</transactionid>

</command>

```

5.5 Команда `newreport4`

Выставление нового отчета на исполнение

```

<command id="newreport4">
  <ustrade_leg>
    <ustradeno> номер сделки для исполнения, включаемой в отчет
    :integer64 </ustradeno>
    <buysell>> направление этой сделки для исполнения : 'B'-покупка
    'S'- продажа) :string </buysell>
  </ustrade_leg>
  ...
</command>

```

Допустимо наличие до четырех элементов `<ustrade_leg>` внутри `newreport4`

Результатом команды является структура `<report>`.

5.6 Команда `cancelreport4`

```

<command id="cancelreport4">

    <reportid> transactionid снимаемого отчета на
    исполнение :integer</reportid>

</command>

```

5.7 Сделки для исполнения

<ustrades> – сделки для исполнения

```

<ustrade>
    <secid> Идентификатор бумаги :integer </secid>
    <tradeno>номер сделки для исполнения :integer64</tradeno>
    <orderno>Номер заявки, на основании которой была заключена
    данная сделка :integer64 </orderno>
    <board>идентификатор режима торгов :string</board>
    <seccode>Код инструмента:string</seccode>
    <client> код клиента в СВО TRANSAQ :string </client>
    <buysell>> направление заявки: 'B' -покупка 'S' - продажа) :string
</buysell>
    <items>текущее кол-во инструмента в заявке (шт.) :integer64
    </items>
    <volume>объем заявки без учета НКД (руб) :double </volume>
    <settle_date>требуемая дата исполнения :date </settle_date>
    <tradedate>дата сделки:date </tradedate>
    <match_time>дата и время, когда сделка была исполнена :date
</match_time>
    <repotradeno>номер сделки РЕПО :integer64</repotradeno>
    <brokerref>примечание трейдера, подавшего заявку :string
</brokerref>
    <comission>комиссия Биржи (руб) :double </comission>
    <price>цена заявки (руб) :double </price>
    <price2> цена выкупа РЕПО :double </price2>
    <repovaluel> сумма РЕПО :double </repovaluel>
    <repo2value> стоимость выкупа РЕПО :double </repo2value>
    <cpfirmid>Идентификатор фирмы-контрагента :string </cpfirmid>
    <accruedint>НКД на единицу инструмента(руб) :double
</accruedint>
    <settlecode> код поставки :string </settlecode>
    <settled> Состояние расчетов по сделке :string
        • 'S'=Рассчитана
        • 'U'=Не рассчитана
        • 'P'=В расчетах

```

```

</settled>
<clearing_type>Тип клиринга :string
    • Элемент отсутствует = Не установлен
    • 'S'=Простой
    • 'M'=Многосторонний
</clearing_type>
<type> Тип сделки для исполнения :string
    • 'N'=Адресная сделка
    • 'R'=Первая часть сделки РЕПО
    • 'r'=Вторая часть сделки РЕПО
    • 'D'=Компенсационный взнос
    • 'P'=Первичное размещение
    • 'C'=Отложенные обязательства и требования дефолтера
    • 'c'=Отложенные обязательства и требования пострадавшего
</type>
<status> Текущий статус сделки для исполнения :string
    • 'U' - Не исполнена
    • 'P' - Включена в отчет
    • 'M' - Исполнена
    • 'G' - Требуется взнос
    • 'N' - Включена в отказ
    • 'C' - Отменена биржей
    • 'W' - Отменена
</status>
</ustrade>
</ ustrades >

```

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

5.8 Отчеты на исполнение

```

<report transactionid = "идентификатор транзакции сервера Transaq
:integer">
    <secid>идентификатор бумаги :integer</secid>
    <board>идентификатор режима торгов :string </board>
    <seccode> Код инструмента:string </seccode>
    <reportno> биржевой номер отчета на
    исполнение:integer64</reportno>

```

```
<time>дата и время регистрации отчета на Бирже :data </time>

<inout> 1 - отправленный, 2 - полученный, 3 - внутренний :integer
</inout>

<author>идентификатор трейдера, выставившего отчет :string
</author>

<cpfirmid>идентификатор фирмы-контрагента :string </cpfirmid>

<quantity>кол-во инструмента в отчете (лот.) :integer64 </quantity>

<volume>объем (руб) :double </volume>

<comission>комиссия Биржи (руб) :double </maxcomission>

<status>статус отчета, 'O'=Ждет исполнения, 'W'= Снят,
'M'=Исполнен, 'C'=Снят торговой системой :string </status>

<withdrawtime>дата и время снятия отчета на исполнение :data
</withdrawtime>

<reporttype> тип отчета, 'S'=Отчет на исполнение, 'C'=Отчет на
отказ от исполнения :string </reporttype>

<reportkind> вид отчета, '1'=Обычный отчет, 'N'=Спец. отчет, 'S'=
Срочный отчет :string </reportkind>

<result>сообщение биржи в случае отказа выставить заявку :string
</result>

</report>
```

Структура присылается автоматически в случае изменения одного или нескольких полей.

6 Приложения

Таблица 1 - Возможные значения для поля status в tere quotation

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

Таблица 2 - Возможные значения для tradingstatus в tere 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	Аукцион закрытия (ММВБ)
D	Аукцион крупными пакетами(ММВБ)
I	Дискретный аукцион (ММВБ)
E	Торги по цене Аукциона Закрытия (ММВБ)
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 (FORTS)

Таблица 3 - Статусы заявок

Идентификатор	Значение
active	Активная
cancelled	Снята трейдером (заявка уже попала на рынок и была отменена)
matched	Исполнена
forwarding	Выставляется на биржу
watching	Ожидает наступления условия
disabled	Прекращена трейдером (условная заявка, которую сняли до наступления условия)
expired	Время действия истекло
denied	Отклонена Брокером
rejected	Отклонена биржей
failed	Не удалось выставить на биржу
removed	Аннулирована биржей
refused	Отклонена контрагентом
wait	Не наступило время активации
inactive	Статус не известен из-за проблем со связью с биржей

Таблица 4 - Статусы стоп-заявок

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

tp_correction_guardtime	Стоп ожидает исполнения в защитном режиме после коррекции (Take Profit)
-------------------------	---

6 Список изменений

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

- Багфиксинг
- Исправлено отсутствие атрибута union в ответе для united_equity и united_go

23.06.2017 Версия 6.05 билд 2.20.25

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

23.06.2017 Версия 6.04 билд 2.20.23

- В данном документе ImmOrCancel заменен на FOK, cancelbalance – на IOC. В приложениях можно использовать как новые так и старые значения
- В описание инструмента (<security>) добавлен элемент <instrclass>

24.05.2017 Версия 6.03 билд 2.20.22

- Исправлена ошибка отсутствия закрывающего тэга asset в united_portfolio.

03.05.2017 Версия 6.03 билд 2.20.21

- В структуру connect добавлен параметр language.

14.03.2017 Версия 5.45 билд 2.20.19

- Bug fixing

31.01.2017 Версия 5.45 билд 2.20.18

- В структуру united_portfolio добавлено поле vm_mma.

12.01.2017 Версия 5.45 билд 2.20.17

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

07.10.2016 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

01.07.2016 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.

23.06.2016 Версия 5.43 билд 2.20.11

- Исправлен формат логирования команды connect в случае ошибок парсинга xml

21.06.2016 Версия 5.41 билд 2.20.10

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

06.06.2016 Версия 5.41 билд 2.20.9

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

18.03.2016 Версия 5.41 билд 2.20.8

- MaxBuy и MaxSell в United Portfolio

29.02.2016 Версия 5.41 билд 2.20.7

- Багфиксинг команды get_history_data

20.02.2016 Версия 5.41 билд 2.20.6

- Багфиксинг и рефакторинг обработчика команды change_pass

09.02.2016 Версия 5.40 билд 2.20.5

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

04.02.2016 Версия 5.40 билд 2.20.4

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

29.01.2016 Версия 5.40 билд 2.20.3

- исправлено падение библиотеки в некоторых случаях отмены заявок

27.01.2016 Версия 5.40 билд 2.20.2

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

21.01.2016 Версия 5.40 билд 2.20.1

- Исправили ошибку экранирования кавычек в атрибутах xml-ответа на команды выставления заявок

11.01.2016 Версия 5.40 билд 2.20.0

- Добавлено информирование пользователя о [юнионах](#), находящихся в его управлении
- Добавлено автоинформирование пользователя о [текущих показателях Единого портфеля и текущей стоимости позиций](#).
- Добавлены параметры push_u_limits и push_pos_equity в команду connect

26.11.2015 Версия 5.32 билд 2.19.1

- Добавлен элемент lcurrentprice в структуру quotations

20.11.2015 Версия 5.32 билд 2.19.0

- Добавлен атрибут union в структуру united_portfolio
- Добавлены команды get_united_equity и get_united_go
- В команды "neworder", "newcondorder" и "newstoporder" добавлен элемент <union>
- В состав передаваемых пользователю информационных объектов, ассоциированных с клиентом, добавлен элемент <union>. Такими информационными объектами являются:

```
<orders><order>
<orders><stoporder>
<orders><negdeal>
<trades><trade>
<positions><money_position>
<positions><sec_position>
<positions><forts_position>
<positions><forts_money>
```

<positions><forts_collaterals>

16.10.2015 Версия 5.30 билд 2.18.0

- Добавлен элемент <quotestype> в структуру [security](#)

05.10.2015 Версия 5.30 билд 2.17.0

- Изменен состав логируемой информации на уровне 2

08.09.2015 Версия 5.30 билд 2.16.1

- В структуру [client](#) добавлен новый параметр:
<forts_acc>счет FORTS: string</forts_acc>
- В команде "get_united_portfolio" вместо параметра client можно задавать union

07.09.2015 Версия 5.30 билд 2.16.0

- Команда get_leverage_control больше не поддерживается

14.08.2015 Версия 5.20 билд 2.20.12

- Добавлена команда для получения единого портфеля get_united_portfolio
- В структуру [client](#) добавлены новые параметры:
 <market>код рынка клиента :integer</market>
 <union>код юниона :string</union>
 Также из этой структуры удалены следующие параметры:
 <ml_intraday>дневной кредит :integer</ml_intraday>
 <ml_overnight>овернайт кредит :integer</ml_overnight>
 <ml_restrict>у.м. ограничительный :double</ml_restrict>
 <ml_call>у.м. требования :double</ml_call>
 <ml_close>у.м. закрытия :double</ml_close>
 Типы клиентов "margin_level" и "risk" больше не используются

10.07.2015 Версия 5.13 билд 2.13.11

- Исправлена ошибка в запросе get_client_limits: в случае частых запросов с разными клиентами ответ приходил по последнему клиенту

23.03.15 Версия 5.10 билд 2.10.9

- Исправлена ошибка выделения памяти для alltrades
- Изменен формат лог-файлов
- Параметры logsdир и loglevel исключены из команды «connect».

14.01.15 Версия 5.10 билд 2.10.8

- Исправлена ошибка автоматической отписки от тиковых данных при отправке команды «unsubscribe»

15.11.14 Версия 5.10 билд 2.10.7

- Дополнительное логирование тиковых данных

17.10.14 Версия 5.10 билд 2.10.6

- Исправлена ошибка пропадающих строк в quotes
- Исправлена ошибка появления openinterest в alltrades для инструментов, не являющихся фьючерсами, опционами или инструментами MCT.

29.09.14 Версия 5.10 билд 2.10.5

- Исправлена ошибка форматирования структуры server_status при отсутствии таймзоны сервера

08.09.14 Версия 5.10 билд 2.10.3

- Исправлена ошибка в tick::tradetime – не отдавались микросекунды
- Добавлены MoneyMct в positions::money_position::asset и Money MCT в positions::money_position::shortname

11.07.14 Версия 5.10 билд 2.10.2

- Добавлен элемент utc_time в команду connect
- Добавлен элемент sec_tz в структуру security
- Добавлен элемент server_tz в структуру server_status
- Добавлен элемент go_rate_long, go_rate_short в структуру portfolio_mct::security

27.05.14 Версия 5.10 билд 2.10.1

- Добавлен элемент source в структуру quotes
- Добавлен элемент milliseconds в команду connect

13.05.14 Версия 5.09.2.9.2

- Изменен порядок вызова функции Initialize. На данный момент каждый успешный вызов функции Initialize должен сочетаться с вызовом функции UnInitialize.

17.04.14 Версия 5.09.2.9.1

- Исправлена ошибка, приводящая к зависанию библиотеки в активном состоянии в случае сетевых проблем при подключении к серверу.

16.04.14 Версия 5.09.2.9

- Исправлена ошибка, приводящая к падению библиотеки при подключении к недоступному или несовместимому серверу.

05.03.14 Версия 5.09.2.8

- В структуру "portfolio_tplus" добавлены элементы coverage_crit, riskrate_long, riskrate_short.

27.11.13 Версия 5.08.2.4

- В структуре "sec_info" идентификатор sec_id заменен на secid

14.10.13 Версия 5.08.2.1

- Добавлена команда get_max_buy_sell_tplus.

07.08.13 Версия 5.08.1.88

- В структуру pit добавлен элемент market.

23.07.13 Версия 5.08.1.85

1. Добавлены элементы в команды:

- connect - micex_registers
- subscribe, unsubscribe - security
- get_leverage_control - security
- get_securities_info – security

2. Добавлены элементы в структуры:

- sec_info - market
- positions::sec_position - market, register
- positions::forts_position - markets
- positions::money_position – register

3. Добавлены seccode и board

- в команды
 - neworder (элемент)
 - newcondorder (элемент)
 - newstoporder (элемент)
 - gethistorydata (атрибут)
 - subscribe_ticks (атрибут)
 - newrpsorder (элемент, только seccode)
 - newmrepoorder (элемент, только seccode)
 - newrepoorder (элемент, только seccode)
- в структуры
 - quotation (элемент)
 - quote (элемент)
 - candles (атрибут)
 - leverage_control::security (атрибут)
 - securities (элемент, только board)
 - alltrades (элемент, только seccode)
 - orders::order (элемент, только seccode)
 - orders::stoporder (элемент, только seccode)
 - orders::negdeal (элемент, только seccode)
 - trades (элемент, только seccode)
 - ustrades (элемент, только seccode)
 - positions::sec_position (элемент, только seccode)

- positions::forts_position (элемент, только seccode)
- marketord (аттрибут, только seccode)
- sec_info (элемент, только seccode)
- sec_info_upd (элемент, только seccode)
- report (элемент, только seccode)

4. Добавлены структуры

- boards
- pit
- portfolio_tplus

5. Добавлена команда get_portfolio