

# **Moscow Exchange Market Data Multicast FIX/FAST Platform**

*Руководство пользователя*

---

Московская биржа  
Version 3.3. September 29, 2014

## Содержание

1.	О платформе MOEX Market Data Multicast FIX/FAST .....	5
1.1.	История изменений .....	5
1.2.	Потоковая передача данных .....	6
1.3.	Инкрементальные сообщения .....	6
1.4.	FIX формат .....	6
1.5.	Кодирование в FAST формат .....	6
1.6.	Получение данных с помощью Multicast .....	7
1.7.	Восстановление данных .....	7
2.	Работа с платформой MOEX Market Data FIX/FAST Multicast .....	8
2.1.	Подключение до старта Торговой системы .....	8
2.2.	Подключение после старта Торговой системы .....	8
2.3.	Обработка дублирующихся данных в потоках А и В .....	9
3.	Функциональность системы .....	11
3.1.	Архитектура системы .....	11
3.2.	FAST формат .....	14
3.2.1	Общее описание .....	14
3.2.2	Кодирование стоп-бита .....	14
3.2.3	Неявное тэгирование .....	14
3.2.4	Возможности кодирования полей .....	15
3.2.5	FAST-шаблон .....	15
3.2.6	Процесс декодирования .....	17
3.2.7	Пример FAST-шаблона .....	18
3.3.	Основные потоки UDP .....	20

3.3.1	Потоки Instrument Definitions .....	21
3.3.2	Потоки OrderBook, Market Statistics, Orders, и Trades .....	21
3.3.3	Потоки Recovery .....	23
3.3.4	Сессии для запроса пропущенных сообщений по TCP .....	23
3.4.	Восстановление пропущенных данных .....	23
3.4.1	Восстановление пропущенных данных из потоков Recovery (UDP) .....	24
3.4.2	Процесс восстановления данных .....	25
3.4.3	Восстановление пропущенных данных по TCP-соединению .....	26
4.	Публичный FIX интерфейс .....	27
4.1.	Группы полей .....	27
4.1.1	Заголовок .....	27
4.1.2	Группа Instrument .....	28
4.1.3	Группа Instrument Extension .....	30
4.1.4	Группа Market Segment .....	30
4.2.	Сообщения сессионного уровня .....	32
4.2.1	Logon (A) .....	32
4.2.2	Logout (5) .....	32
4.2.3	Heartbeat (0) .....	33
4.3.	Сообщения бизнес уровня .....	33
4.3.1	Security Definition (d) .....	33
4.3.2	Security Status (f) .....	34
4.3.3	Trading Session Status (h) .....	36
4.3.4	Market Data Request (V) .....	37
4.3.5	Market Data - Snapshot/Full Refresh (W) .....	37

4.3.6	Market Data - Incremental Refresh (X) .....	44
5.	Настройка сетевого соединения .....	51
5.1.	Настройка VPN соединения с MOEX на базе Windows XP .....	51
5.2.	Настройка VPN соединения с MOEX на базе Windows 7 .....	64
5.3.	Настройка VPN соединения с MOEX на базе OpenSUSE .....	74
5.4.	Часто возникающие вопросы и методы их решения .....	78
6.	Сертифицированные средства работы .....	83
6.1.	Библиотека FIX Antenna <sup>TM</sup> от EPAM – B2Bits® .....	83
6.1.1	Quick Start – примеры кода .....	84
6.1.2	Обзор API.....	88

## 1. О платформе MOEX Market Data Multicast FIX/FAST

Система MOEX Market Data Multicast FIX/FAST Platform представляет собой новый, высокоэффективный механизм для передачи рыночных данных о торгах на Московской Бирже (далее используется сокращение MOEX). Данный механизм сочетает в себе структуру и синтаксис сообщений FIX протокола, хорошие возможности для оптимизации потоков данных FAST протокола, и возможности быстрой и эффективной передачи данных большому количеству пользователей UDP протокола.

Система MOEX Market Data Multicast FIX/FAST Platform включает следующие аспекты: потоковые данные, инкрементальные сообщения, FIX формат сообщений, кодирование сообщений в формат FAST, получение данных большим количеством пользователей, возможность восстановления пропущенных данных.

### 1.1. История изменений

Issue	Date	Description
1.0	25 мая 2011	Исходная версия документа
2.0	12 декабря 2012	Добавления для улучшения понимания спецификаций и устранения ошибок в документе
3.3	08 апреля 2013	Добавлены поля, специфичные для сделок переговорных режимов и РЕПО Изменение форматов сообщений для разделения полей Режима, торгового статуса инструмента и периода торгов по разным полям в FIX сообщениях. Дополнительные поля для полной поддержки публикации данных режимов РЕПО с ЦК, аукциона закрытия, дискретных аукционов, аукционов крупных пакетов, и данных рынка T+2. Новый шаблон компрессии FAST Редакторские правки документации и устранение неточностей
3.3.1	24 мая 2013	Устранение ошибок и добавление пояснений по результатам обратной связи с клиентами. Удаление неиспользуемых полей из документа. Исправлен шаблон компрессии В документе сохранены правки в режиме рецензирования для выделения измененных частей .
3.3.2	04 сентября 2013	Уточнены единицы (лоты или количества ценных бумаг), используемые при публикации объемов аукционов в полях 271.
3.3.3	26 марта 2014	В блок Market Segment добавлено поле OrderNote (9680), указывающее уровень

		листинга финансового инструмента на фондовом рынке. <b>Изменен шаблон FAST компрессии</b> Исправлены неточности в документации.
--	--	---

## 1.2. Поточковая передача данных

Использование потоковой передачи данных позволяет передавать информацию от источника к получателю, не разбивая ее на отдельные сообщения для каждого события. Несколько таких событий могут быть включены в одно сообщение. Это позволяет существенно снизить задержки и увеличить скорость передачи данных.

## 1.3. Инкрементальные сообщения

Использование инкрементальных сообщений позволяет значительно снизить объемы отправляемых данных. Используются только данные, изменившиеся под воздействием рыночных событий. Минимальное количество команд используется для их обновления: добавление новой записи, изменение записи, удаление записи.

## 1.4. FIX формат

Система MOEX Market Data Multicast FIX/FAST Platform использует формат и синтаксис FIX сообщений. Сообщение состоит из заголовка, тела сообщения и трейлера. Поля в сообщении разделены между собой с помощью ASCII символа - <SOH>.

Для более подробного ознакомления с составом сообщений см. 4. Публичный FIX .

## 1.5. Кодирование в FAST формат

FAST (FIX Adapted for Streaming) представляет собой алгоритм сжатия, который позволяет в значительной степени оптимизировать FIX сообщения. FAST уменьшает размер данных без внесения задержек, что позволяет увеличить количество отправляемых данных и уменьшить время их передачи.

FAST Protocol для сжатия сообщений использует следующее:

- Неявное тэгирование;
- Возможности кодирования полей;
- Использование Pmap;
- Кодирование стоп-бита;
- Использование бинарного кодирования.

В большинстве случаев правила кодирования в FAST формат согласовываются между контрагентами путем предоставления XML шаблонов.

Для более подробного ознакомления с использованием FAST кодирования см. 3.2. FAST .

## 1.6. Получение данных с помощью Multicast

Для распространения сообщений используется UDP протокол, который позволяет передавать пакеты сразу нескольким получателям.

В один UDP пакет могут быть включены сразу несколько FIX сообщений, закодированных в FAST. Но в настоящее время система MOEX Market Data Multicast FIX/FAST Platform обеспечивает отправку только одного закодированного в FAST сообщения. FAST сообщение специально формируется таким образом, чтобы размер UDP пакета не превышал типичного для сети Ethernet значения параметра MTU в 1500 байт. .

Во избежание путаницы MOEX Market Data Multicast FIX/FAST Platform посылает данные из разных таблиц на бирже разным multicast группам.

## 1.7. Восстановление данных

Для клиентов очень важно постоянное «присутствие» на рынке. Если случится так, что какие-то данные будут потеряны в процессе работы, то просто необходимо их быстрое восстановление.

MOEX Market Data Multicast FIX/FAST Platform обеспечивает восстановление данных 2 способами:

- Восстановление большого объема данных с помощью отправки клиенту снимков (к примеру, для клиентов присоединившихся после начала торгов);
- Восстановление небольшого объема данных по TCP – соединению (к примеру, когда отдельные сообщения были утеряны при трансфере).

## 2. Работа с платформой MOEX Market Data FIX/FAST Multicast

### 2.1. Подключение до старта Торговой системы

Клиентам рекомендуется подключиться к системе MOEX Market Data Multicast FIX/FAST Platform еще до открытия торгов. Это гарантирует, что клиент начнет получать актуальные данные без необходимости обращения к каким-либо способам восстановления пропущенных данных.

Данный сценарий является основным. Клиенту следует выполнить следующую последовательность действий:

1. Скачать файл конфигурации Каналов и Поточков с ftp-сервера. Конфигурационный файл в формате .xml описывает параметры подключения (IP адреса multicast, номера портов и т.д.). Скачать файл FAST-шаблона с ftp-сервера. Для получения дополнительной информации см. пункт 3.2.5 с описанием шаблона.
2. Начать слушать Поток Instruments Definitions, OrderBook и/или Orders, Statistics, Trades (клиент может слушать только интересующие его потоки) и применять получаемые данные в обычном порядке.

### 2.2. Подключение после старта Торговой системы

При подключении к Системе позже начала Торгов для получения полной рыночной информации следует придерживаться следующей процедуры:

1. Скачать файл конфигурации Каналов и Поточков с ftp-сервера. Конфигурационный файл в формате .xml описывает параметры подключения (IP адреса multicast, номера портов и т.д.). Скачать файл FAST-шаблона с ftp-сервера. Для получения дополнительной информации см. пункт 3.2.5 с описанием шаблона.
2. Начать слушать Поток Instruments Definitions.
3. Начать слушать Поток OrderBook и/или Orders, Statistics, Trades (клиент может слушать только интересующие его потоки) и накапливать получаемые сообщения.
4. Начать слушать Поток OrderBook Recovery и/или Orders Recovery, Statistics Recovery, Trades Recovery. Получить по этим Поточкам актуальный снимок, применить полученный снимок. Процесс можно проводить как последовательно (сначала получить снимки по всем инструментам, а потом обрабатывать накопленные обновления), так и параллельно (по мере получения снимков по инструментам обрабатывать накопленные обновления по полученному инструменту).
5. Перестать слушать Поток Recovery.
6. Продолжить обычную обработку потоков инкрементальных обновлений.



## 2.3. Обработка дублирующихся данных в потоках А и В

Данные во всех UDP-потоках распространяются в двух экземплярах (А и В) на двух разных multicast-адресах. Клиенту рекомендуется обрабатывать оба потока в виду негарантированности доставки UDP-пакетов. Обработка двух идентичных потоков позволяет снизить вероятность потерь по меньшей мере в 2 раза.

В каком именно из потоков (А или В) сообщение появится первым, не оговаривается. Для обработки потоков следует использовать порядковый номер сообщения из преамбулы или тэга 34-MsgSeqNum. Использование преамбулы позволяет определить порядковый номер не прибегая к декодированию FAST-сообщения.

Обработку потоков А и В следует производить по следующему алгоритму:

1. Слушать потоки А и В.
2. Обрабатывать сообщения по порядковому номеру.
3. Отбрасывать полученное сообщение, если сообщение с таким порядковым номером уже получалось ранее.
4. Если обнаруживается пропуск в порядковых номерах в обоих каналах, то это, скорее всего, свидетельствует о потере пакетов как в потоке А, так и в потоке В. Клиенту следует инициировать одну из процедур восстановления пропущенных данных. Впрочем, клиент может подождать некоторое (разумное) время, возможно пропущенный пакет придёт несколько позже, так как протокол UDP не гарантирует последовательность доставки пакетов.

Пример:

Поток А
34-MsgSeqNum = 59
34-MsgSeqNum = 60
34-MsgSeqNum = 62
34-MsgSeqNum = 63
34-MsgSeqNum = 65

Поток В
34-MsgSeqNum = 59
34-MsgSeqNum = 60
34-MsgSeqNum = 61
34-MsgSeqNum = 62
34-MsgSeqNum = 65

Сообщения получают из потоков А и В.

1. Получили 59-е сообщение из А, обработали его.
2. Получили 59-е сообщение из В, отбросили его, так как обработали его ранее.
3. Получили 60-е сообщение из А, обработали его.
4. Получили 60-е сообщение из В, отбросили его, так как обработали его ранее.
5. Получили 62-е сообщение из А, отбросили его, так как ожидается 61-е.
6. Получили 61-е сообщение из В, обработали его.

7. Получили 62-е сообщение из В, обработали его.
8. Получили 62-е сообщение из А, отбросили его, так как обработали его ранее.
9. Получили 63-е сообщение из А, обработали его.
10. Получили 65-е сообщение из А, отбросили его, так как ожидается 64-е.
11. Получили 65-е сообщение из В, отбросили его, так как ожидается 64-е.
12. Перешли к процедуре восстановления пропущенных данных, так как обнаружен пропуск сообщения.

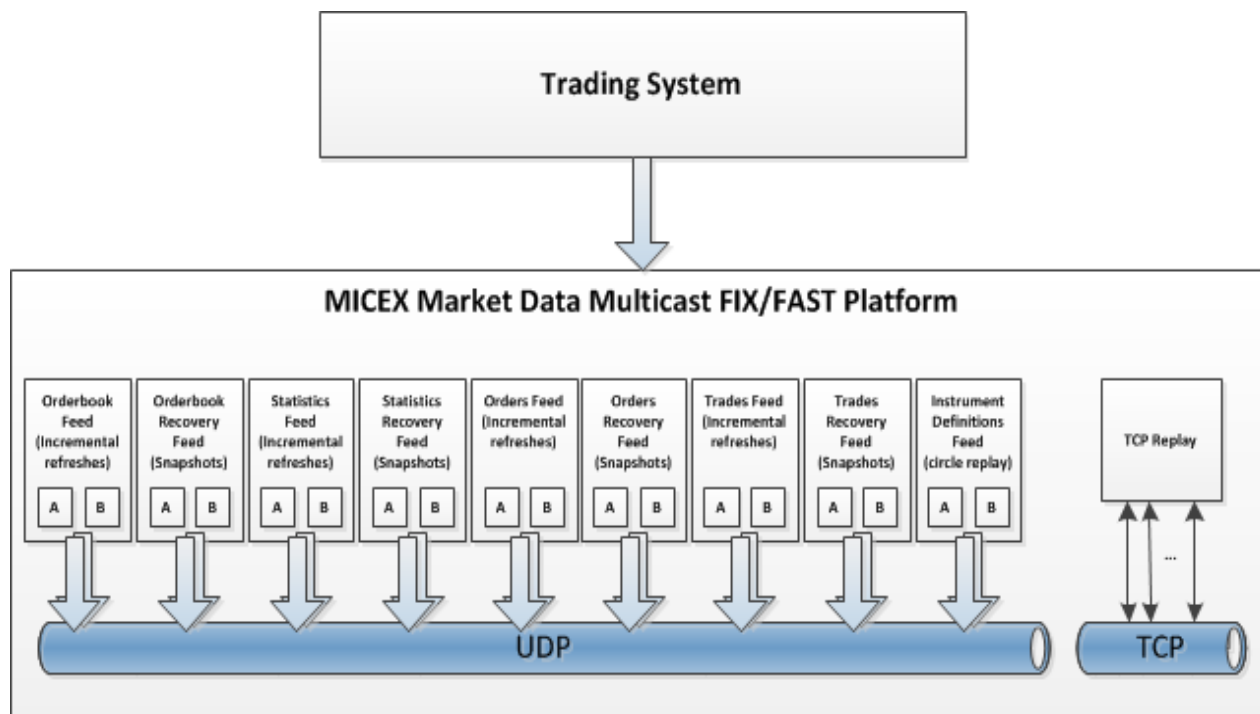
## 3. Функциональность системы

### 3.1. Архитектура системы

Для распространения рыночных данных используется транспортный протокол UDP, а для запроса пропущенных данных реализуются механизмы восстановления по протоколу UDP и повторного получения данных по протоколу TCP.

В системе используются следующие виды информационных потоков:

1. Основные потоки.
  - 1.1. Потоки распространения инкрементальных обновлений рыночных данных.
  - 1.2. Потоки распространения описаний финансовых инструментов.
2. Потоки восстановления
  - 2.1. Потоки распространения снимков рыночных данных.
  - 2.2. Сессии для запроса пропущенных данных.



MOEX Market Data Multicast FIX/FAST Platform обеспечивает вещание по следующим Потокam:

- Основные потоки:
  - Aggregated OrderBook Feeds(OBR), 20 лучших цен на покупку и продажу:
    - OrderBook Feed A;
    - OrderBook Feed B.
  - Market Statistics Feeds(MSR):
    - Statistics Feed A;
    - Statistics Feed B.
  - Active Orders List Feeds(OLR):
    - Orders Feed A;
    - Orders Feed B.
  - Trades List Feeds(TLR):
    - Trades Feed A;

- Trades Feed B.
- Потoki Recovery:
  - Aggregated OrderBook Recovery Snapshot Feeds(OBS):
    - OrderBook Recovery Feed A;
    - OrderBook Recovery Feed B.
  - Market Statistics Recovery Snapshots Feeds(MSS):
    - Statistics Recovery Feed A;
    - Statistics Recovery Feed B.
  - Active Orders List Recovery Snapshots Feeds(OLS):
    - Orders Recovery Feed A;
    - Orders Recovery Feed B.
  - Trades List Recovery Snapshot Feeds(TLS):
    - Trades Recovery Feed A;
    - Trades Recovery Feed B.
- Instruments Definitions Feeds(IDF):
  - Instruments Definitions Feed A;
  - Instruments Definitions Feed B.

Помимо трансляции данных в UDP-потоках, MOEX Market Data Multicast FIX/FAST Platform может принимать входящие TCP-соединения, по которым клиенты могут запросить пропущенные данные. По TCP-соединению могут быть запрошены пропущенные сообщения в одном из следующих UDP-потоков:

- OrderBook Feed (OBR)
- Statistics Feed (MSR)
- Orders Feed (OLR)
- Trades Feed (TLR)

Существуют некоторые ограничения при запросе данных по TCP-соединению:

1. данные доступны за ограниченный период времени (не более чем с начала дня);
2. количество отсылаемых за один раз сообщений ограничено (около 500 сообщений в запросе);
3. общее количество запрашиваемых в день сообщений ограничено.

## 3.2. FAST формат

### 3.2.1 Общее описание

Все сообщения, отправляемые MOEX Market Data Multicast, представляют собой сообщения в FIX-формате, закодированные по протоколу FAST (FIX Adapted for Streaming). Протокол FAST был разработан FIX Market Data Optimization Working Group для оптимизации электронного обмена финансовой информацией, в частности, для распространения большого объема данных с минимальной задержкой.

Особенностью распространения данных в информационных потоках от MOEX Market Data Multicast является то, что перед каждым FAST-сообщением добавляется 4-байтовая преамбула, в которой содержится значение 34-го тэга (SeqNum) следующего за преамбулой FAST-сообщения (рис. 1). Это позволяет получить порядковый номер сообщения (как при обработке сообщений из потоков А и В, так и при обнаружении пропусков), не прибегая к декодированию самого FAST-сообщения – это значительно экономит время при обработке потока.

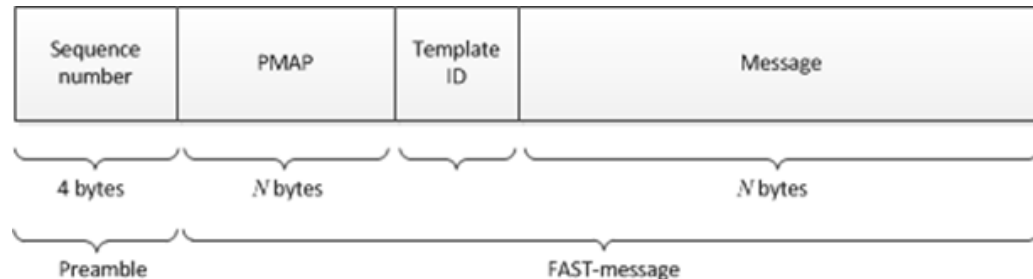


Figure 1

### 3.2.2 Кодирование стоп-бита

Кодирование стоп-бита является одним из составляющих процессов FAST, который позволяет исключить избыточность на уровне передачи полей с данными используя стоп-бит вместо привычного байтового разделителя. В FAST стоп-бит используется вместо стандартного FIX разделителя – байта <SOH>; таким образом 7 битов каждого байта используются для передачи данных, а 8й бит служит обозначением окончания поля.

### 3.2.3 Неявное тэгирование

По стандарту FIX протокола каждое сообщение имеет вид «Тег = Значение <SOH>», где:

Тег – номер поля, которое в данный момент передается;

Значение – фактическое содержание данных этого поля;

<SOH> – ASCII символ, который используется в качестве байтового разделителя поля.

Например:

35=x|268=3 (заголовок сообщения)

279=0|269=2|270=9462.50|271=5|48=800123|22=8 (сделка)

279=0|269=0|270=9462.00|271=175|1023=1|48=800123|22=8|346=15 (новое предложение 1)

279=0|269=0|270=9461.50|271=133|1023=2|48=800123|22=8|346=12 (новое предложение 2)

FAST устраняет избыточность используя шаблон, который описывает структуру всего сообщения. Такой механизм называется «неявным тегированием», т.к. FIX теги становятся неявной частью передаваемых данных. FAST-шаблон заменяет синтаксис «Тег = Значение» на «неявное тегирование» по таким правилам:

- номера тэгов не передаются в сообщении, но заданы в шаблоне;
- последовательность полей в сообщении такая же, как и тэгов в шаблоне;
- шаблон определяет упорядоченный набор полей с операторами.

### 3.2.4 Возможности кодирования полей

FAST действует как машина состояний, которая в каждый момент должна знать, какие значения необходимо содержать в памяти. FAST сравнивает текущее значение поля с его предыдущим значением, и определяет, какое действие требуется предпринять:

- использовать в качестве нового значения константу (заданную в шаблоне),
- значение по умолчанию (применять, если новое значение поля отсутствует),
- сделать копию (продублировать предыдущее значение этого тэга),
- вычислить дельту (для целочисленных – арифметическая разность между текущим и предыдущим значением, также используется со строковыми значениями),
- проинкрементировать предыдущее значение (только для целочисленных). Словарем называется кэш, в котором хранятся предыдущие значения полученные системой. Содержимое словаря сбрасывается в начале каждого UDP пакета. Так как в одном UDP-пакете отправляется только одно FAST-сообщение, то дельта в такой реализации использоваться не будет.

### 3.2.5 FAST-шаблон

FAST-шаблон соответствует типу FIX сообщения, и однозначно определяет порядок полей в нем. Шаблон также содержит синтаксис, указывающий тип поля, и какой метод декодирования применять при передаче. Шаблон задается в XML виде. Каждое FAST сообщение в свою очередь содержит идентификатор шаблона, по которому будет происходить декодирование.

Пример шаблона сообщения Market Data – Incremental Refresh (MsgType=X):

```

103 <!-- Market Data - Incremental Refresh -->
104 <template name="X" id="6" xmlns="http://www.fixprotocol.org/ns/fast/td/1.1">
105   <string name="MessageType" id="35">
106     <constant value="X"/>
107   </string>
108   <string name="ApplVerID" id="1128"><copy/></string>
109   <string name="SenderCompID" id="49"><copy/></string>
110   <uInt32 name="MsgSeqNum" id="34"><increment/></uInt32>
111   <uInt64 name="SendingTime" id="52"><copy/></uInt64>
112   <byteVector name="MessageEncoding" id="347" presence="optional"><default/></byteVector>
113   <sequence name="GroupMDEntries">
114     <length name="NoMDEntries" id="268"/>
115     <uInt32 name="MDUpdateAction" id="279"><copy/></uInt32>
116     <string name="MDEntryType" id="269" presence="optional"><copy/></string>
117     <byteVector name="MDEntryID" id="278" presence="optional"><copy/></byteVector>
118     <byteVector name="Symbol" id="55" presence="optional"><copy/></byteVector>
119     <int32 name="RptSeq" id="83" presence="optional"><copy/></int32>
120     <decimal name="MDEntryPx" id="270" presence="optional"><copy/></decimal>
121     <decimal name="MDEntrySize" id="271" presence="optional"><copy/></decimal>
122     <uInt32 name="MDEntryDate" id="272" presence="optional"><copy/></uInt32>
123     <uInt32 name="MDEntryTime" id="273" presence="optional"><copy/></uInt32>
124     <byteVector name="TradingSessionID" id="336" presence="optional"><copy/></byteVector>
125     <byteVector name="QuoteCondition" id="276" presence="optional"><copy/></byteVector>
126     <byteVector name="TradeCondition" id="277" presence="optional"><copy/></byteVector>
127     <uInt32 name="OpenCloseSettleFlag" id="286" presence="optional"><default/></uInt32>
128     <decimal name="NetChgPrevDay" id="451" presence="optional"><copy/></decimal>
129     <decimal name="Yield" id="236" presence="optional"><copy/></decimal>
130     <decimal name="AccruedInterestAmt" id="5384" presence="optional"><copy/></decimal>
131     <decimal name="ChgFromWAPrice" id="5510" presence="optional"><copy/></decimal>
132     <decimal name="ChgOpenInterest" id="5511" presence="optional"><copy/></decimal>
133     <int32 name="TotalNumOfTrades" id="6139" presence="optional"><copy/></int32>
134     <decimal name="TradeValue" id="6143" presence="optional"><copy/></decimal>
135     <int32 name="OfferNbOr" id="9168" presence="optional"><copy/></int32>
136     <int32 name="BidNbOr" id="9169" presence="optional"><copy/></int32>
137     <decimal name="ChgFromSettlmnt" id="9750" presence="optional"><copy/></decimal>
138     <int32 name="SumQtyOfBest" id="10503" presence="optional"><copy/></int32>
139     <string name="OrderSide" id="10504" presence="optional"><copy/></string>
140     <string name="OrdStatus" id="10505" presence="optional"><copy/></string>
141     <decimal name="OrdBalance" id="10506" presence="optional"><copy/></decimal>
142     <decimal name="OrdValue" id="10507" presence="optional"><copy/></decimal>
143     <decimal name="MinCurrPx" id="10509" presence="optional"><copy/></decimal>
144     <uInt32 name="MinCurrPxChgTime" id="10510" presence="optional"><copy/></uInt32>
145   </sequence>
146 </template>

```

Figure 2



### 3.2.6 Процесс декодирования

Процесс декодирования происходит в следующей последовательности:

Шаг 1. Транспорт: Клиент системы получает закодированное FAST сообщение.

Шаг 2. Декодирование пакета:

- Определение шаблона;
- Извлечение бинарных закодированных бит;
- Построение соответствия между полученными битами и полями в шаблоне.

Шаг 3. Декодирование полей: применение операторов для определения значения на основании шаблона.

Шаг 4. Построение FIX сообщения

Шаг 5. Обработка FIX сообщения.

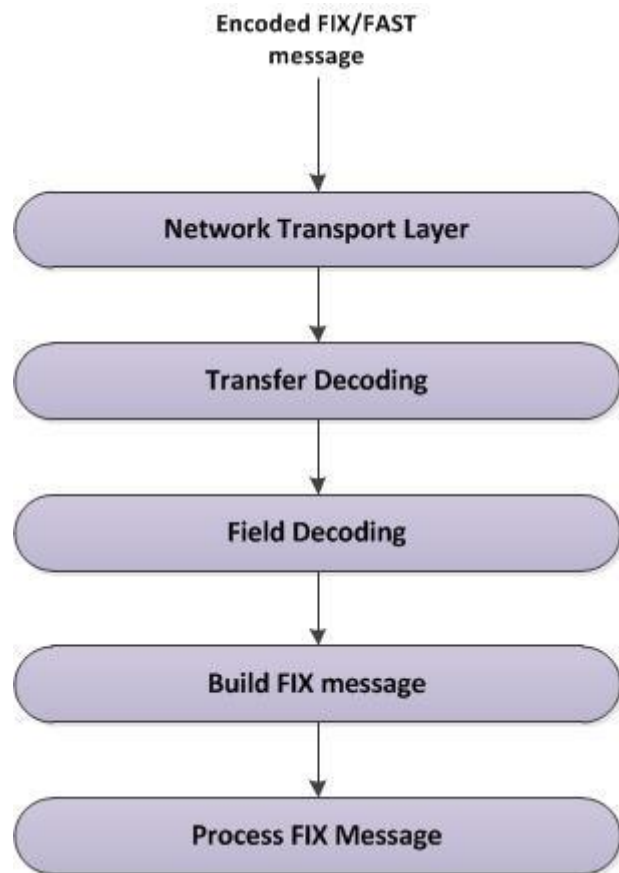


Figure 3

### 3.2.7 Пример FAST-шаблона

Table 1

Line #	Template Syntax	Use and Description
1	<template name="X" id="6">	Идентификатор и название шаблона.

	<code>xmlns="http://www.fixprotocol.org/ns/fast/td/1.1"&gt;</code>	
2	<code>&lt;string name="MessageType" id="35"&gt;     &lt;constant value="X" /&gt; &lt;/string&gt;</code>	MessageType определен как тип данных string, идентификатор = 35.
3	<code>&lt;string name="ApplVerID" id="1128"&gt;&lt;copy/&gt;&lt;/string&gt;</code>	ApplVerID определен как тип данных string, идентификатор = 1128
4	<code>&lt;string name="SenderCompID" id="49"&gt;&lt;copy/&gt;&lt;/string&gt;</code>	SenderCompID определен как тип данных string, идентификатор = 49
5	<code>&lt;uInt32 name="MsgSeqNum" id="34"&gt;&lt;increment/&gt;&lt;/uInt32&gt;</code>	MsgSeqNum определен как тип данных unsigned integer, идентификатор = 34
6	<code>&lt;uInt64 name="SendingTime" id="52"&gt;&lt;copy/&gt;&lt;/uInt64&gt;</code>	SendingTime определен как тип данных unsigned integer, идентификатор = 52
7	<code>&lt;byteVector name="MessageEncoding" id="347" presence="optional"&gt;&lt;default/&gt;&lt;/byteVector&gt;</code>	MessageEncoding определен как тип данных byte vector, идентификатор = 347
8	<code>&lt;sequence name="GroupMDEntries"&gt;     &lt;length name="NoMDEntries" id="268"/&gt; &lt;/sequence&gt;</code>	Определение репитинг группы MDEntries. 'NoMDEntries' показывает количество повторяющихся элементов.
9	<code>&lt;uInt32 name="MDUpdateAction" id="279" presence="optional"&gt;&lt;copy/&gt;&lt;/uInt32&gt;</code>	MDUpdateAction определен как тип данных unsigned integer, идентификатор = 279
10	<code>&lt;string name="MDEntryType" id="269" presence="optional"&gt;&lt;copy/&gt;&lt;/string&gt;</code>	MDEntryType определен как тип данных string, идентификатор = 269
11	<code>&lt;byteVector name="MDEntryID" id="278" presence="optional"&gt;&lt;copy/&gt;&lt;/byteVector&gt;</code>	MDEntryID определен как тип данных byte vector, идентификатор = 278
12	<code>&lt;byteVector name="Symbol" id="55" presence="optional"&gt;&lt;copy/&gt;&lt;/byteVector&gt;</code>	Symbol определен как тип данных byte vector, идентификатор = 55.
13	<code>&lt;int32 name="RptSeq" id="83" presence="optional"&gt;&lt;copy/&gt;&lt;/int32&gt;</code>	RptSeq определен как тип данных signed integer, идентификатор = 83
14	<code>&lt;decimal name="MDEntryPx" id="270" presence="optional"&gt;&lt;copy/&gt;&lt;/decimal&gt;</code>	MDEntryPx определен как тип данных decimal, идентификатор = 270
15	<code>&lt;decimal name="MDEntrySize" id="271" presence="optional"&gt;&lt;copy/&gt;&lt;/decimal&gt;</code>	MDEntrySize определен как тип данных decimal, идентификатор = 271
16	<code>&lt;uInt32 name="MDEntryDate" id="272" presence="optional"&gt;&lt;copy/&gt;&lt;/uInt32&gt;</code>	MDEntryDate определен как тип данных unsigned integer, идентификатор = 272.
17	<code>&lt;uInt32 name="MDEntryTime" id="273" presence="optional"&gt;&lt;copy/&gt;&lt;/uInt32&gt;</code>	MDEntryTime определен как тип данных unsigned integer, идентификатор = 273
18	<code>&lt;byteVector name="TradingSessionID" id="336" presence="optional"&gt;&lt;copy/&gt;&lt;/byteVector&gt;</code>	TradingSessionID определен как тип данных byte vector, идентификатор = 336
19	<code>&lt;byteVector name="QuoteCondition" id="276" presence="optional"&gt;&lt;copy/&gt;&lt;/byteVector&gt;</code>	QuoteCondition определен как тип данных byte vector, идентификатор = 276
20	<code>&lt;byteVector name="TradeCondition" id="277" presence="optional"&gt;&lt;copy/&gt;&lt;/byteVector&gt;</code>	TradeCondition определен как тип данных byte vector, идентификатор = 277
21	<code>&lt;byteVector name="OpenCloseSettlFlag" id="286" presence="optional"&gt;&lt;copy/&gt;&lt;/byteVector&gt;</code>	OpenCloseSettlFlag определен как тип данных byte vector, идентификатор = 286
22	<code>decimal name="NetChgPrevDay" id="451" presence="optional"&gt;&lt;copy/&gt;&lt;/decimal&gt;</code>	NetChgPrevDay определен как тип данных decimal, идентификатор = 451.

23	<decimal name="AccruedInterestAmt" id="5384" presence="optional"><copy/></decimal>	AccruedInterestAmt определен как тип данных decimal, идентификатор = 5384
24	<decimal name="ChgFromWAPrice" id="5510" presence="optional"><copy/></decimal>	ChgFromWAPrice определен как тип данных decimal, идентификатор = 5510
25	<int32 name="TotalNumOfTrades" id="6139" presence="optional"><copy/></int32>	TotalNumOfTrades определен как тип данных signed integer, идентификатор = 6139
26	<decimal name="TradeValue" id="6143" presence="optional"><copy/></decimal>	TradeValue определен как тип данных decimal, идентификатор = 6143
27	<decimal name="Yield" id="236" presence="optional"><copy/></decimal>	Yield определен как тип данных decimal, идентификатор = 236
28	<int32 name="OfferNbOr" id="9168" presence="optional"><copy/></int32>	OfferNbOr определен как тип данных signed integer, идентификатор = 9168
29	<int32 name="BidNbOr" id="9169" presence="optional"><copy/></int32>	BidNbOr определен как тип данных signed integer with identifier = 9169
30	<string name="OrderSide" id="10504" presence="optional"><copy/></string>	OrderSide определен как тип данных string, идентификатор = 10504.
31	<string name="OrderStatus" id="10505" presence="optional"><copy/></string>	OrderStatus определен как тип данных string, идентификатор = 10505
32	<decimal name="MinCurrPx" id="10509" presence="optional"><copy/></decimal>	MinCurrPx определен как тип данных decimal, идентификатор = 10509.
33	<uint32 name="MinCurrPxChgTime" id="10510" presence="optional"><copy/></uint32>	MinCurrPxChgTime определен как тип данных unsigned integer, идентификатор = 10510.

### 3.3. Основные потоки UDP

В основных потоках (OrderBook, Statistics, Orders, Trades – Feed A и Feed B) в режиме multicast по протоколу UDP распространяются следующие рыночные данные:

- В потоке OrderBook – обновления таблицы котировок.
- В потоке Statistics – статистика рынка.
- В потоке Orders – обновления таблицы заявок.
- В потоке Trades – обновления таблицы сделок.

Все перечисленные Потоки транслируются по протоколу UDP multicast. Каждый Поток транслируется на отдельном multicast-адресе. В соответствующих потоках А и В транслируются идентичные сообщения. Дублирование обеспечивает статистическое снижение вероятности потерь UDP-пакетов.

### 3.3.1 Поток Instrument Definitions

В потоках Instrument Definitions (Feed A и Feed B) с фиксированной периодичностью рассылаются описания финансовых инструментов в виде FIX-сообщений Security Definition (d), закодированных в формат FAST. Одно сообщение содержит описание одного финансового инструмента.

Пример сообщения:

```
8=FIXT.1.1|9=400|35=d|1128=9|34=1551|460=5|423=2|911=1572|49=MOEX|55=VRSBP|48=RU000A0DPG75|22=4|461=EPXXXX|167=PS|
107=Voronezh
EnergoSbyt.Comp(pref)|15=RUB|120=RUB|5217=2-01-55029-
E|5385=FOND|969=0.001|5508=0.4|7595=18716678|350=54|351=»Воронеж.энергосб.комп» ОАО ап|5382=20|5383=ВоронЭнСбп|52=20110503-
08:29:32.968|870=2|871=27|872=3|871=8|872=0|1310=1|561=1|1309=1|336=SMAL|10=000|
```

Примечание: каждая ценная бумага (тег 55 'Symbol') может торговаться в различных режимах, отличающихся правилами. Тег 336 содержит код режима торгов. Для каждой ценной бумаги могут быть доступны несколько режимов < Board> . Каждую комбинацию тегов 55 и 336 в Security Definition следует рассматривать как отдельный объект с отдельным потоком обновлений рыночных данных.

### 3.3.2 Поток OrderBook, Market Statistics, Orders, и Trades

Следующие рыночные данные распространяются в отдельных потоках:

- Поток OrderBook (A и B) – передают обновления из таблицы ORDERBOOK.

Виды обновлений:

1. *Add* – создает/добавляет новую запись, записи отсортированы по цене ( MDUpdateAction(279) =0 );
2. *Change* – изменяет параметры записи (MDUpdateAction (279) = 1 );
3. *Delete* – удаляет запись (MDUpdateAction (279) = 2).

Обновления применимы к элементам рыночных данных – MDEntryType (269) = '0' (Котировки на покупку), '1' (Котировки на продажу), 'J' – нет данных.

- Поток Statistics (A и B) – передают рыночную статистику, обновления из таблицы SECURITIES.

Виды обновлений: Add, Change, и Delete. Элементы рыночных данных:

- '0' (Котировки на покупку);
- '1' (Котировки на продажу);
- '2' (Информация по последней сделке);
- '3' (Список индексов);
- '4' (Цена открытия/цена первой сделки);
- '5' (Цена закрытия/цена последней сделки предыдущего дня);
- '7' (Максимальная цена сделки);

'8' (Минимальная цена сделки);

'9' (Средневзвешенные цены);

'A' (Дисбаланс);

'B' (Объемы сделок);

'J' (Пустой снэпшот)

'N' (Максимальная цена спроса в течение сессии);

'O' (Минимальная цена предложения в течение сессии);

'Q' (Расчетная цена аукциона)

'W' (Цена аукциона закрытия);

'c' (Объем аукциона закрытия);

'f' (Объем спроса рыночной заявки в аукцион закрытия);

'g' (Объем предложения рыночной заявки в аукцион закрытия)

'i' (Спрос сессии);

'j' (Предложение сессии);

'h' (Цена предторгового периода);

'k' (Цена послеторгового периода);

'l' (Рыночная цена 2); Для валютного рынка – цена валютного фиксинга, рассчитанная за период 11:29-12:00 московского времени

'm' (Рыночная цена); Для валютного рынка – цена валютного фиксинга

'o' (Официальная цена открытия);

'p' (Официальная текущая цена);

'q' (Признаваемая котировка); Для валютного рынка – международная цена валютного фиксинга

'r' (Официальная цена закрытия);

'v' (Совокупный спрос);

'w' (Совокупное предложение);

's' (Цена аукциона крупными пакетами);

'x' (Объем аукциона крупными пакетами);

'y' (Накопленный купонный доход на дату расчетов, в рублях, в пересчете на единицу финансового инструмента)

'u' (Дюрация);

- Потоки Orders (A и B) – передают обновления из таблицы ORDERS.

Виды обновлений: Add, Change, и Delete. Элементы рыночных данных: '0' (заявка на покупку), '1' (заявка на продажу), 'J' – нет данных.

- Потоки Trades (A и B) – передают обновления из таблицы TRADES.

Виды обновлений: только Add (MDUpdateAction(279) =0). Элементы рыночных данных: MDEntryType (269) = 'z' (Все сделки/список обезличенных сделок), 'J' – нет данных..

Данные распространяются в виде FIX-сообщений Market Data – Incremental Refresh (X), закодированных в формат FAST. Каждое сообщение может содержать обновления по нескольким финансовым инструментам.

При изменении состояния соединения с Торговой системой в соответствующие UDP-потoki инкрементальных обновлений отправляется сообщение Trading Session Status (h).

При изменении торгового статуса инструмента во все UDP-потoki инкрементальных обновлений отправляется сообщение Security Status (f).

### 3.3.3 Потoki Recovery

В потоках Recovery (OrderBook, Statistics, Orders, Trades) в режиме multicast по протоколу UDP с фиксированной периодичностью распространяются текущие снэпшоты соответствующих данных в виде FIX-сообщений Market Data – Snapshot/Full Refresh (W), закодированных в формат FAST. Каждое сообщение содержит информацию по одному инструменту. Информация включает текущий торговый статус инструмента и текущее состояние соединения с Торговой системой.

Клиенты не должны слушать эти потoki постоянно. К ним необходимо подключаться только в случае необходимости восстановить пропущенную в основных потоках информацию. После восстановления клиенту рекомендуется прекратить слушать данные потoki.

### 3.3.4 Сессии для запроса пропущенных сообщений по TCP

Данный сервис позволяет клиенту запросить пересылку набора сообщений в заданном диапазоне номеров, уже опубликованных в одном из UDP-потокoв.

В запросе клиент указывает диапазон порядковых номеров для пересылки, а так же идентификатор UDP-потoka, из которого запрашивается информация. Максимальное количество сообщений, которое может запросить клиент, ограничено (около 500 сообщений в запросе). Запрос отправляется в виде FIX-сообщения Market Data Request (V). Запрос отправляется по TCP-соединению, инициируемому клиентом. Ответные сообщения отправляются клиенту по этому же TCP-соединению в виде FIX-сообщений. По завершению отправки ответных сообщений MOEX Market Data Multicast закрывает это TCP-соединение. Все ответные сообщения закодированы в FAST-формат. Данный сервис клиент должен использовать лишь в крайнем случае, если другие методы восстановления невозможны. Сервис не обеспечивает высокую производительность

## 3.4. Восстановление пропущенных данных

Данные во всех UDP-потоках распространяются в двух экземплярах (A и B) на двух разных multicast-адресах. Клиенту рекомендуется обрабатывать оба потока в виду негарантированности доставки UDP-пакетoв.

Может случиться так, что будут утеряны пакеты из обоих потокoв, в этом случае нужно воспользоваться процедурой восстановления данных.

Понять, что сообщение утеряно можно по пропускам в порядковых номерах сообщений 34-MsgSeqNum или по пропускам в номерах инкрементальных обновлений 83-RptSeq. Это означает, что рыночные данные больше не достоверны и клиент получает их не в полном объеме. Необходимо воспользоваться процедурой восстановления.

MOEX Market Data Multicast предоставляет несколько механизмов для восстановления данных. Рекомендуется в первую очередь использовать потоки Recovery. Восстановления при помощи TCP-соединения более медленный процесс, при котором разрешено запрашивать ограниченное количество сообщений, им рекомендуется пользоваться в исключительных случаях, когда другие средства по каким-либо причинам недоступны.

### 3.4.1 Восстановление пропущенных данных из потоков Recovery (UDP)

Восстановление пропущенных данных из Потоков Recovery может быть использовано для получения большого объема потерянных данных и для подключения после старта Торгов. В потоках Recovery через фиксированный интервал времени распространяются снэпшоты рыночных данных. В каждом сообщении Market Data – Snapshot/Full Refresh (W) тэг 369-LastMsgSeqNumProcessed соответствует тэгу 34-MsgSeqNum последнего сообщения Market Data – Incremental Refresh (X) в соответствующем потоке, включенного в данный снэпшот, а номер обновления каждого инструмента, содержащийся в тэге 83-RptSeq сообщения Market Data – Snapshot/Full Refresh (W), соответствует номеру инкрементального обновления, содержащегося в тэге 83-RptSeq соответствующего MDEntry последнего сообщения Market Data – Incremental Refresh (X), включенного в данный снэпшот. Таким образом, по пропуску в последовательности 34-MsgSeqNum можно определить произошедшую потерю данных, а по пропуску в последовательностях 83-RptSeq определить, по каким именно инструментам данные пропущены.

Данные по инструменту в канале инкрементальных обновлений следует считать актуальными с того момента, как номер обновления этого инструмента в тэге 83-RptSeq сообщения Market Data – Incremental Refresh (X) станет больше этого номера в аналогичном тэге сообщения Market Data – Snapshot/Full Refresh (W) для этого инструмента.

Также данные по инструменту в канале инкрементальных обновлений можно считать актуальными с того момента, как порядковый номер сообщения Market Data – Incremental Refresh (X) станет больше значения тэга 369-LastMsgSeqNumProcessed сообщения Market Data – Snapshot/Full Refresh (W) по этому инструменту.

Нумерация сообщений в каждом цикле отправки снэпшотов начинается с 1. Поэтому все снэпшоты следует считать полученными, когда приходит сообщение с порядковым номером 1, которое относится к следующему циклу.

Тэг 893-LastFragment значением ‘Y’ отмечает последнее сообщение в снэпшоте по данному инструменту. Поэтому снэпшот по инструменту следует считать полученным, когда получено сообщение с 893-LastFragment = ‘Y’.

Пока идёт получение снэпшота, клиент должен накапливать сообщения из канала инкрементальных обновлений, чтобы применить их после получения снэпшота.

Последовательность шагов при восстановлении соответствует шагам 3–6, приведенным в разделе 2.2.

После восстановления пропущенных сообщений клиенту следует прекратить слушать поток Recovery, чтобы не перегружать свою сетевую инфраструктуру.



### 3.4.2 Процесс восстановления данных

Процесс восстановления затрагивает только потоки с пропущенными сообщениями. Остальные потоки могут быть обработаны двумя способами: они могут быть перемещены в очередь, до тех пор, пока не будут получены все пакеты из потока Recovery, либо они могут быть обработаны параллельно с потоками Recovery.

#### 3.4.2.1.1. Перемещение пакетов в очередь

Данный процесс применяется к сообщениям из потоков обновлений во время обработки пакетов из потока Recovery. Во избежание накопления слишком большого количества пакетов в очереди рекомендуется обрабатывать обновления сразу же, как только будет получен соответствующий им снэпшот.

1. Определить поток, в котором пропущено сообщение.
2. Получить и положить в очередь сообщения из потоков обновлений.
3. Получить снэпшоты из потока Recovery, который соответствует потоку обновлений с пропущенным сообщением.
4. Проверить что все нужные снэпшоты были получены:
  - a. Порядковый номер сообщений в цикле снэпшотов начинается с 1. Чтобы определить конец цикла, нужно дождаться следующего сообщения с 34-MsgSeqNum = 1.
  - b. Снэпшоты в потоках Recovery отправляются в таком же порядке, как и описания инструментов в потоках Instrument Definitions. По значению 893-LastFragment в снэпшоте можно понять, что это последнее сообщение для данного инструмента. Получение снэпшота с 893-LastFragment для последнего инструмента означает получение последнего сообщения в цикле.
5. Забрать из очереди все сообщения, в которых:
  - a. Значение 34-MsgSeqNum больше самого минимального значения 369-LastMsgSeqNumProcessed сообщения Market Data – Snapshot/Full Refresh (W).

Или

- b. Значение 83-RptSeq из сообщения Market Data Incremental – Refresh (X) больше, чем самое минимальное значение 83-RptSeq из снэпшота.
6. Продолжить получение обновлений.

#### 3.4.2.1.2. Параллельная обработка

Данный процесс позволяет осуществлять получение обновлений по инструментам и одновременно восстановление пропущенных данных.

1. Определить поток, в котором пропущено сообщение.
2. Получать обновления, и возможно пропущенные данные обновятся и потеряют уже актуальность.
3. Получить снэпшоты из потока Recovery, который соответствует потоку обновлений с пропущенным сообщением.
4. Для каждого инструмента:

- a. Сравнить значение 369-LastMsgSeqNumProcessed из снэпшота со значением 34-MsgSeqNum из обновления, и убедиться, что 34-MsgSeqNum не меньше.

Или

- b. Сравнить значение 83-RptSeq из снэпшота со значением 83-RptSeq из обновления, и убедиться что значение 83-RptSeq из обновления не меньше.
5. Продолжить получение обновлений.

#### **3.4.2.1.3. Инкрементальные обновления инструмента**

Сообщения из потоков с обновлениями содержат номера обновлений для каждого инструмента (tag 83-RptSeq). В каждой репитинг группе элемента рыночных данных содержится номер инкрементального обновления инструмента (tag 83-RptSeq).

Клиенты могут отслеживать порядок номеров инкрементальных обновлений для быстрого обнаружения пропуска сообщений.

- Если порядок номеров 83-RptSeq нарушен, это говорит о том, что часть рыночных данных по инструменту была пропущена.
- Если порядок номеров 83-RptSeq не нарушен, это говорит о том, что данные по инструменту верны и актуальны.

#### **3.4.2.1.4. Восстановление по инкрементальным обновлениям**

Как правило, клиенты должны отслеживать состояние данных по котировкам. Но возможно при потерях данных инкрементальные обновления лучше позволят отобразить актуальное состояние котировок, даже без необходимости обращаться к процедурам восстановления. Этот процесс называется восстановлением по инкрементальным обновлениям. Для ликвидных инструментов большая вероятность быстрого обновления данных и как следствие быстрая потеря актуальности пропущенных данных.

### **3.4.3 Восстановление пропущенных данных по TCP-соединению**

Восстановление данных, пропущенных в потоках OrderBook, Statistics, Orders, Trades, можно выполнить, запросив их по TCP-соединению. Данный способ восстановления не является высокопроизводительным, и его следует использовать только в крайнем случае и только для запроса небольшого количества пропущенных сообщений. Количество сообщений, которое может быть запрошено клиентом за одно подключение, задаётся в конфигурационном файле сервера MOEX Market Data Multicast.

Для запроса пропущенных данных клиент должен выполнить следующие действия:

1. Установить TCP-соединение с сервером MOEX Market Data Multicast.
2. Отправить серверу FIX-сообщение Logon(A). В случае успешной авторизации, сервер ответит FAST-сообщением Logon(A).
3. Отправить серверу FIX-сообщение Market Data Request (V), в котором необходимо указать:
  - a. Идентификатор UDP-потока, из которого запрашиваются сообщения – в тэге 1180-ApplFeedID.
  - b. Диапазон порядковых номеров запрашиваемых сообщений – в тэгах 1182-ApplBeginSeqNo и 1183-ApplEndSeqNo.

Если запрос может быть обработан, сервер отправляет клиенту запрошенные FAST сообщения с порядковыми номерами, под которыми эти сообщения изначально были опубликованы в соответствующем Потоке.

Если запрос не может быть обработан, клиенту отправляется FAST-сообщение Logout (5) с описанием причины отказа.

После отправки ответа сервер закрывает соединение.

Сервер обрабатывает только первый запрос от клиента. Если клиент посылает более одного запроса, второй и последующие игнорируются.

Если сервер не получит Market Data Request (V) в течение определённого интервала времени после аутентификации, подключение закрывается.

## 4. Публичный FIX интерфейс

Описание интерфейса базируется на спецификации протокола FIX (Financial Information Exchange, <http://fixprotocol.org/>) версии 5.0 SP2; предполагается, что читатель уже знаком с основами этого протокола.

Системой используются только те сообщения (группы) и их поля, которые описаны в данном публичном интерфейсе. *Следует обратить внимание, что поля, присутствующие в стандарте 5.0 SP2 (обязательные и не обязательные), но не перечислены в данном публичном интерфейсе, считаются необязательными и **игнорируются биржей**. Значения полей, присутствующие в списке допустимых значений в стандарте 5.0 SP2, но не описанные в этом документе, считаются некорректными – и поступающие сообщения с такими данными будут отклонены.*

### 4.1. Группы полей

#### 4.1.1 Заголовок

Table 2

Tag	Поле	Наличие	Тип	Допустимые значения	Примечание
1128	AppVerID	O	String (1)	'9' (FIX50SP2)	Определяет версию протокола для application messages. Всегда содержит незашифрованные данные, должно быть первым полем в сообщении.
35	MsgType	O	String (10)		Определяет тип сообщения. Всегда содержит незашифрованные данные, должно быть третьим полем в сообщении.
49	SenderCompID	O	String (12)		Идентификатор компьютера – отправителя сообщения. Всегда содержит незашифрованные данные. Если сообщение отправляется на сервер TCP Replay биржи, то это поле может содержать произвольное значение.
34	MsgSeqNum	O	SeqNum		Порядковый номер сообщения.
52	SendingTime	O	UTCTimestamp		Время передачи сообщения (выражено в формате UTC).

347	MessageEncoding	H	String(11)	'UTF-8' (Unicode)	Тип кодирования (не ASCII символы). Обязательное, если хотя бы одно из полей в сообщении имеет кодировку, отличающуюся от ASCII.
-----	-----------------	---	------------	-------------------	---

#### 4.1.2 Группа Instrument

Table 3

Tag	Поле	Наличие	Тип	Допустимые значения	Примечание
55	Symbol	O	String(12)		Код/аббревиатура ценной бумаги. В его качестве используется внутренний идентификатор финансового инструмента на MOEX (SECCODE). Примечание: финансовый инструмент с кодом SecCode может быть доступен для торгов в разных режимах. Вы должны рассматривать комбинацию Symbol (55) + TradingsessionId (336) как отдельный инструмент с отдельными котировками и таблицами сделок и заявок.
48	SecurityID	H	String		Идентификатор финансового инструмента (например, CUSIP, SEDOL, ISIN, и т.п.).
22	SecurityIDSource	H	String	'4' (ISIN)	Тип идентификатора финансового инструмента. Поле обязательное, если определено значение поля SecurityID (48).
460	Product	H	int	'3' (CORPORATE); '4' (CURRENCY); '5' (EQUITY); '6' (GOVERNMENT); '7' (INDEX); '10' (MORTGAGE) '11' (MUNICIPAL); '12' (OTHER); '13' (FINANCING).	Тип продукта, с которым связана ценная бумага.
461	CFICode	H	String		Тип ценной бумаги по стандарту ISO 10962. CFI код (Classification of Financial Instruments).
167	SecurityType	H	String	'CORP' (Корпоративные облигации) 'FOR' (Валютный контракт) 'CS' (Акции обыкновенные)	Тип ценной бумаги.

				'PS' (Акции привилегированные) 'EUSOV' (Еврооблигация) BN' (Ценные бумаги, выпущены банком) 'MF' (Паи инвестиционных фондов) 'MLEG' (Multi-leg инструмент) 'MUNI' (Муниципальные облигации) RDR – Российские депозитарные расписки ETF – Бумаги иностранных инвестиционных фондов (ETF) 'COFP' (Ипотечные сертификаты участия) 'XCN' (Корзина бумаг) 'STRUCT' (Дополнительный идентификатор списка) 'WAR' (Инструмент «заявки-списки»)	
224	CouponPaymentDate	H	LocalMktDate		Дата выплаты накопленного купонного дохода.
223	CouponRate	H	Percentage		Процентная ставка, по которой рассчитывается размер купонной выплаты по облигации (купонная ставка).
107	SecurityDesc	H	String		Описание ценной бумаги. На MOEX это поле содержит наименование финансового инструмента на английском языке.
350	EncodedSecurityDescLen	H	Length		Длина поля EncodedSecurityDesc (351) в байтах.
351	EncodedSecurityDesc	H	data		Название ценной бумаги на русском языке (не ASCII символы). Тип кодировки указан в поле MessageEncoding (347) в заголовке сообщения.
5217	StateSecurityID	H	String		Номер государственной регистрации.
5382	EncodedShortSecurityDescLen	H	Length		Длина поля EncodedShortSecurityDesc (5383) в байтах.
5383	EncodedShortSecurityDesc	H	data		Краткое (не ASCII символы) наименование ценной бумаги на русском языке. Тип кодировки указан в поле MessageEncoding (347) в заголовке сообщения.
5556	BaseSwapPx	H	Price		Базовый курс при торговле СВОП инструментам.
5558	BuyBackPx	H	Price		Цена досрочного выкупа облигации, заполняется если бумага предусматривает досрочный выкуп (до погашения). Если

					указана, то расчет доходности производится по этой цене (в качестве цены погашения). При заполнении BUYBACKPRICE обязательно заполняется поле BUYBACKDATE
5559	BuyBackDate	H	LocalMktDate		Дата досрочного выкупа облигации, заполняется если бумага предусматривает досрочный выкуп (до погашения). Если указана, то расчет доходности производится с использованием этой даты, в качестве даты погашения.

### 4.1.3 Группа Instrument Extension

Table 4

Tag	Поле	Наличие	Тип	Допустимые значения	Примечание
870	NoInstrAttrib	H	NumInGroup		Количество элементов в группе InstrAttribs.
=> 871	InstrAttribType	H	int	'8' (Купонный период) '27' (Кол-во десятичных знаков (DECIMALS) в ценах финансового инструмента)	Тип атрибута ценной бумаги. Поле обязательное, если NoInstrAttrib (870) > 0.
=> 872	InstrAttribValue	H	String		Значение атрибута ценной бумаги (если применимо).

### 4.1.4 Группа Market Segment

Table 5

Tag	Поле	Наличие	Тип	Допустимые значения	Примечание
1310	NoMarketSegments	H	NumInGroup		Количество элементов в группе MarketSegmentGrp.
=> 561	RoundLot	H	Qty		Количество ценных бумаг в одном стандартном лоте.
=> 1309	NoTradingSessionRules	H	NumInGroup		Количество элементов в группе TradingSessionRulesGrp.
=> => 336	TradingSessionID	H	String		Идентификатор торговой сессии, на котором торгуется финансовый инструмент. Используется для указания режима торгов SECBOARD. Примечание: финансовый инструмент может быть доступен для торгов в разных режимах. Вы должны

					рассматривать комбинацию Symbol (55) + TradingSessionId (336) как отдельный инструмент с отдельными котировками и таблицами сделок и заявок.
=>=>625	TradingSessionSubID	H	String	<p>NA – нет торгов  O – период открытия  C – период закрытия  F – финальный период закрытия  N – нормальный период торгов  L – аукцион закрытия  I – дискретный аукцион  D – аукцион крупных пакетов  E – период торгов по цене аукциона закрытия</p>	<p>Указывает период торгов</p> <p>Примечания:</p> <ul style="list-style-type: none"> <li>• Код периода не определен перед началом торгов и после их окончания</li> <li>• Переключение между периодами обычно происходит с кратковременным переключением торгового статуса в состояние Нет Торгов, в котором код периода не определен (625=NA)</li> <li>• Порядок смены периодов торгов и расписание определяется Правилами торгов и рыночными условиями</li> <li>• Код периода в этой группе полей указывает на период, который продолжался в момент начала цикла вещания Security Definitions. Обновления статуса инструмента, которые могут приходиться во время вещания цикла должны заменять статус на статус из сообщений 35=f.</li> </ul>
=>=>326	SecurityTradingStatus	H	int	<p>18 – нет торгов  118 – период открытия  18 – торги закрыты  103 – период закрытия  2 – перерыв в торгах  17 – нормальный период торгов  102 – аукцион закрытия  106 – аукцион крупных пакетов  107 – дискретный аукцион  120 – период торгов по цене аукциона закрытия</p>	<p>Указывает торговый статус инструмента</p> <p>Примечание: перерыв в торгах обозначается как 326=2 и идентификатор периода в поле 625.</p> <ul style="list-style-type: none"> <li>• Статусы Нет Торгов и Торги Закрыты означают разные технологические состояния торгов по инструменту. Однако, оба эти статуса означают запрет торговых операций, поэтому им присваиваются одинаковые значения поля 326.</li> <li>• Код статуса в этой группе полей указывает на статус, который существовал в момент начала цикла вещания Security Definitions. Обновления статуса инструмента, которые могут приходиться во время вещания цикла должны заменять статус на</li> </ul>

				статус из сообщений 35=f.
=>=>9680	OrderNote	H	Char	Уровень листинга.

## 4.2. Сообщения сессионного уровня

### 4.2.1 Logon (A)

Logon сообщение от пользователя к MOEX:

Table 6

Tag	Поле	Наличие	Тип	Допустимые значения	Примечание
<Standard Message Header>		O			Тип сообщения = 'A'
553	Username	Y	String		Имя пользователя (логин).
554	Password	Y	String		Пароль пользователя.
1137	DefaultApplVerID	O	String	'9' (FIX50SP2)	Определяет версию протокола на сессионном уровне.

Logon сообщение от MOEX к пользователю:

Table 7

Tag	Поле	Наличие	Тип	Допустимые значения	Примечание
<Standard Message Header>		O			MsgType = 'A'
108	HeartBtInt	O	int		Интервал ожидания торговых сообщений или сообщений HeartBeat.
1137	DefaultApplVerID	O	String	'9' (FIX50SP2)	Определяет версию протокола на сессионном уровне.

### 4.2.2 Logout (5)

Table 8



Tag	Поле	Наличие	Тип	Допустимые значения	Примечание
<Standard Message Header>		O			Тип сообщения = '5'
58	Text	H	String		Причина завершения сессии.

### 4.2.3 Heartbeat (0)

Table 9

Tag	Поле	Наличие	Тип	Допустимые значения	Примечание
<Standard Message Header>		O			Тип сообщения = '0'

## 4.3. Сообщения бизнес уровня

### 4.3.1 Security Definition (d)

Table 10

Tag	Поле	Наличие	Тип	Допустимые значения	Примечание
<Standard Message Header>		O			Тип сообщения = 'd'
911	TotNumReports	O	int		Количество сообщений по всем инструментам.
Component block <Instrument>		O			Данные по финансовому инструменту.
Component block <Instrument Extension>		H			Дополнительная информация по финансовому инструменту.
15	Currency	H	Currency		Код валюты, в которой выражен номинал ценной бумаги.
Component block <Market Segment>		H			Дополнительная информация по финансовому инструменту, его листингу, включая торговый статус и период на момент начала цикла вещания. Эти данные позволяют определить статус инструмента и период торгов

					для поздних подключений, когда сообщения типа 35=f были пропущены.
120	SettlCurrency	H	Currency		Код валюты, в которой производятся расчеты по данному финансовому инструменту.
423	PriceType	H	int	'1' (В процентах от номинала) '2' (За единицу (например, за акцию или за контракт))	Тип цены. Для РЕПО с ЦК принимает значение '1', с указанием ставки РЕПО, а не номинала акции или облигации
5385	MarketCode	H	String		Код рынка, на котором торгуется финансовый инструмент. Примечание: MarketCode указывает на группу режимов торгов с похожими правилами торгов. Значение MarketCode может совпадать со значением <Market> 336 тэга, но иметь иное назначение.
64	SettlDate	H	LocalMktDate		Дата расчетов, выраженная в ГТТГММДД формате. Для системных режимов фондового и валютного рынка – указывает дату расчетов Для переговорных режимов – указывает дату расчетов для используемого по умолчанию кода расчетов. Фактическая дата расчетов по сделке может отличаться от этой даты и публикуется в поле SettlDate Потока обезличенных сделок. Для валютных своп-инструментов: дата расчетов для обратной сделки.
969	MinPriceIncrement	H	float		Минимальный шаг изменения цены.
5508	FaceValue	H	Amt		Номинальная стоимость одной ценной бумаги, в валюте инструмента.
5850	OrigIssueAmt	H	Int		Объем в обращении
7595	NoSharesIssued	H	Qty		Объем выпуска.

#### 4.3.2 Security Status (f)

Сообщения Security status уведомляют об изменениях торгового статуса и периода торгов по инструменту. Пожалуйста, имейте в виду, что публикация большого числа сообщений в ограниченных по трафику каналах вещания занимает некоторое время, и происходит параллельно с публикацией обновлений торговых данных. Параллельная публикация может приводить к получению торговых обновлений

из последующего периода торгов немного раньше получения изменения периода торгов, или к получению торговых обновлений из предыдущего периода торгов после получения сообщения о смене статуса.

Table 11

Tag	Поле	Наличие	Тип	Допустимые значения	Примечание
<Standard Message Header>		O			Тип сообщения = 'Г'
83	RptSeq	O	int		Порядковый номер обновления информации по инструменту.
55	Symbol	O	String		Код/аббревиатура ценной бумаги. В его качестве используется внутренний идентификатор финансового инструмента на Московской бирже (SECCODE). Примечание: финансовый инструмент с кодом SecCode может быть доступен для торгов в разных режимах. Вы должны рассматривать комбинацию Symbol (55) + TradingSessionId (336) как отдельный инструмент с отдельными котировками и таблицами сделок и заявок.
336	TradingSessionID	H	String		Идентификатор торговой сессии, на котором торгуется финансовый инструмент. Используется для указания режима торгов SECBOARD. Примечание: финансовый инструмент с кодом SecCode может быть доступен для торгов в разных режимах. Вы должны рассматривать комбинацию Symbol (55) + TradingSessionId (336) как отдельный инструмент с отдельными котировками и таблицами сделок и заявок.
625	TradingSessionSubID	H	String	NA – нет торгов O – период открытия C – период закрытия F – финальный период закрытия N – нормальный период торгов L – аукцион закрытия I – дискретный аукцион D – аукцион крупных пакетов E – период торгов по цене аукциона закрытия	Указывает период торгов <ul style="list-style-type: none"> <li>• Код периода не определен перед началом торгов и после их окончания</li> <li>• Переключение между периодами обычно происходит с кратковременным переключением торгового статуса в состояние Нет Торгов, в котором код периода не определен (625=NA)</li> <li>• Порядок смены периодов торгов и расписание определяется Правилами торгов и</li> </ul>

					рыночными условиями
326	SecurityTradingStatus	H	int	18 – нет торгов 118 – период открытия 18 – торги закрыты 103 – период закрытия 2 – перерыв в торгах 17 – нормальный период торгов 102 – аукцион закрытия 106 – аукцион крупных пакетов 107 – дискретный аукцион 120 – период торгов по цене аукциона закрытия	Указывает торговый статус инструмента Примечания: <ul style="list-style-type: none"> <li>перерыв в торгах обозначается как 326=2 и идентификатор периода в поле 625</li> <li>Статусы Нет Торгов и Торги Закрыты означают разные технологические состояния торгов по инструменту. Однако, оба эти статуса означают запрет торговых операций, поэтому им присваиваются одинаковые значения поля 326.</li> </ul>
5509	AuctionIndicator	H	Boolean	'Y' (Да); 'N' (Нет).	Индикатор информирующий, что по данному инструменту проводится аукцион первичного размещения. В настоящее время данные аукционов первичного размещения не публикуются в потоках. Примечания: <ul style="list-style-type: none"> <li>5509=N для всех типов аукционов, кроме аукциона первичного размещения</li> <li>Boolean значения передаются в FAST сообщениях как двоичные целые: 0 для N и 1 для Y</li> </ul>

### 4.3.3 Trading Session Status (h)

Table 12

Tag	Поле	Наличие	Тип	Допустимые значения	Примечание
<Standard Message Header>		O			Тип сообщения = 'h'
336	TradingSessionID	O	String		Идентификатор торговой сессии. Поле содержит идентификатор рынка.
340	TradSesStatus	O	int	'100' (Соединение с MOEX установлено); '101' (Соединение с MOEX потеряно); '102' (Соединение восстановлено, торговая система не перезапустилась);	Статус торговой сессии. Информировать о состоянии подключения сервера MOEX Market Data Multicast FIX/FAST Platform к торговой системе биржи. Примечание: получение весьма маловероятного

				'103' (Соединение восстановлено, торговая система перезапускалась)	сообщения 340=103 означает, что торговая система была перезапущена с полной потерей предыдущего состояния. Вы должны удалить все полученные данные и начать процедуру их получения с самого начала.
58	Text	H	String		Текстовая строка в свободном формате. Может использоваться для комментариев, дополнительной информации касающейся подключения к конкретному рынку MOEX.

#### 4.3.4 Market Data Request (V)

Table 13

Tag	Поле	Наличие	Тип	Допустимые значения	Примечание
<Standard Message Header>		O			Тип сообщения = 'V'
1180	ApplID	H	String	OLR, OBR, TLR, MSR	Идентификатор UDP-потока.
1182	ApplBegSeqNum	H	SeqNum		Порядковый номер первого запрашиваемого сообщения.
1183	ApplEndSeqNum	H	SeqNum		Порядковый номер последнего запрашиваемого сообщения. Если запрос на одно сообщение, то ApplBegSeqNum(1182) = ApplEndSeqNum(1183). Если запрос на все сообщения (но не более максимального числа пересылаемых сообщений) после определенного сообщения, то ApplEndSeqNum(1183) = '0'(бесконечность).

#### 4.3.5 Market Data - Snapshot/Full Refresh (W)

Table 14

Tag	Поле	Наличие	Тип	Допустимые значения	Примечание
<Standard Message Header>		O			Тип сообщения = 'W'
83	RptSeq	O	int		Номер обновления инструмента. Соответствует RptSeq(83) в сообщении Market Data - Incremental Refresh (X).
369	LastMsgSeqNumProcessed	H	SeqNum		Значение, соответствующее MsgSeqNum(34) из последнего сообщения Market Data - Incremental Refresh (X), которое было получено и обработано.

340	TradSesStatus	H	int	'100' (Соединение с MOEX установлено); '101' (Соединение с MOEX потеряно); '102' (Соединение восстановлено, торговая система не перезапускалась); '103' (Соединение восстановлено, торговая система перезапускалась).	Статус соединения с Торговой системой. Информировать о состоянии подключения сервера MOEX Market Data Multicast FIX/FAST Platform к торговой системе биржи. Примечание: получение весьма маловероятного сообщения 340=103 означает, что торговая система была перезапущена с полной потерей предыдущего состояния. Вы должны удалить все полученные данные и начать процедуру их получения с самого начала.
55	Symbol	O	String		Код/аббревиатура ценной бумаги. В его качестве используется внутренний идентификатор финансового инструмента на MOEX (SECCODE). Примечание: финансовый инструмент с кодом SecCode может быть доступен для торгов в разных режимах. Вы должны рассматривать комбинацию Symbol (55) + TradingsessionId (336) как отдельный инструмент с отдельными котировками и таблицами сделок и заявок.
893	LastFragment	H	Boolean	'N' (Сообщение не последнее, снимок еще не сформирован); 'Y' (Последнее сообщение, снимок сформирован).	Индикатор, показывающий, является ли сообщение последним в серии, которая формирует снимок по данному инструменту. Boolean значения передаются в FAST сообщениях как двоичные целые: 0 для N и 1 для Y
1682	MDSecurityTrading Status	H	int	18 – нет торгов 118 – период открытия 18 – торги закрыты 103 – период закрытия 2 – перерыв в торгах 17 – нормальный период торгов 102 – аукцион закрытия 106 – аукцион крупных пакетов 107 – дискретный аукцион 120 – период торгов по цене аукциона закрытия	Состояние торгов по инструменту. Примечания: <ul style="list-style-type: none"> <li>• перерыв в торгах обозначается как 1682=2 и идентификатор периода в поле 625.</li> <li>• Переключение между периодами обычно происходит с кратковременным переключением торгового статуса в состояние Нет Торгов, в котором код периода не определен (625=NA)</li> <li>• Порядок смены периодов торгов и расписание определяется Правилами торгов и рыночными условиями</li> <li>• Статусы Нет Торгов и Торги Закрыты означают разные технологические состояния торгов по инструменту. Однако, оба эти статуса означают запрет торговых операций, поэтому им присваиваются одинаковые значения поля 326.</li> </ul>

5509	<i>AuctionIndicator</i>	<i>H</i>	<i>Boolean</i>	'Y' (Да) 'N' (Нет)	Индикатор информирующий, что по данному инструменту проводится аукцион первичного размещения. В настоящее время данные аукционов первичного размещения не публикуются в потоках. Примечания: <ul style="list-style-type: none"> <li>• 5509=N для всех типов аукционов, кроме аукциона первичного размещения</li> <li>• Boolean значения передаются в FAST сообщениях как двоичные целые: 0 для N и 1 для Y</li> </ul>
451	<i>NetChgPrevDay</i>	<i>H</i>	<i>PriceOffset</i>		Изменение цены последней сделки по отношению к цене последней сделки предыдущего торгового дня.
268	<i>NoMDEntries</i>	<i>O</i>	<i>NumInGroup</i>		Количество элементов в группе MDEntryTypes.
=> 269	<i>MDEntryType</i>	<i>O</i>	<i>char</i>	'0' (Котировки на покупку); '1' (Котировки на продажу); '2' (Информация по последней сделке); '3' (Список индексов); '4' (Цена открытия/цена первой сделки); '5' (Цена закрытия/цена последней сделки предыдущего дня);  '7' (Максимальная цена сделки); '8' (Минимальная цена сделки); '9' (Средневзвешенные цены); 'A' (Дисбаланс); 'B' (Объемы сделок); 'J' (Пустой снэпшот) 'N' (Максимальная цена спроса в течение сессии); 'O' (Минимальная цена предложения в течение сессии); 'Q' (Расчетная цена аукциона) 'W' (Цена аукциона закрытия); 'c' (Объем аукциона закрытия); 'f' (Для потоков MSR/MSS - объем рыночных заявок на покупку в аукцион закрытия, в единицах ценных бумаг; для потоков OLR/OLS – рыночная заявка в	Тип рыночных данных. Примечания: <ul style="list-style-type: none"> <li>• Наличие различных типов данных определяется Правилами торгов и зависит от режима торгов и рынка.</li> <li>• Различные каналы вещания имеют подмножества значений типа данных из этого списка</li> <li>• Empty Book (269=J) означает, что по инструменту отсутствуют какие-либо данные. Это сообщение может быть получено без указания кода инструмента и режима (по всему рынку). В этом случае вы должны удалить все накопленные данные и начать процедуру первичного подключения.</li> <li>• Смысл некоторых типов данных зависит от типа рынка (фондовый или валютный)</li> <li>• Переговорные режимы торгов не содержат данных о котировках и активных заявках</li> <li>• Набор значений этого поля может расширяться при обновлениях торговой системы. Рекомендуется разрабатывать код так, чтобы неизвестные значения этого поля и привязанные к ним другие поля игнорировались в коде до тех пор, пока он не будет адаптирован к новым типам данных</li> </ul>

				<p>аукцион закрытия, на покупку );  'g' (Для потоков MSR/MSS - объём рыночных заявок на продажу в аукцион закрытия, в единицах ценных бумаг; для потоков OLR/OLS – рыночная заявка в аукцион закрытия, на продажу );  'i' (Спрос сессии);  'j' (Предложение сессии);  'h' (Цена предторгового периода);  'k' (Цена послеторгового периода);  'l' (Рыночная цена 2); Для валютного рынка – цена валютного фиксинга, рассчитанная за период 11:29-12:00 московского времени  't' (Рыночная цена); Для валютного рынка – цена валютного фиксинга)  'o' (Официальная цена открытия);  'r' (Официальная текущая цена);  'q' (Признаваемая котировка); Для валютного рынка – международная цена валютного фиксинга  'r' (Официальная цена закрытия);  'v' (Совокупный спрос);  'w' (Совокупное предложение);  's' (Цена аукциона крупными пакетами)  'x' (Объём аукциона крупными пакетами);  'y' (Накопленный купонный доход на дату расчетов, в рублях, в пересчете на единицу финансового инструмента)  'u' (Дюрация);  'z' (Все сделки/список обезличенных сделок)</p>	<ul style="list-style-type: none"> <li>• Индексы публикуются в канале MSS/MSR</li> <li>• Данные предыдущего дня обозначаются наличием поля 286</li> </ul>
=> 278	MDEntryID	H	String		<p>Идентификатор элемента MDEntry.  Примечания:</p> <ul style="list-style-type: none"> <li>• Для канала сделок (269=z) это поле содержит строку с биржевым номером сделки</li> <li>• Для канала котировок, это поле содержит строку идентификатора уровня цены</li> <li>• Для канала заявок, это поле содержит строку идентификатора сообщения о добавлении новой заявки, который НЕ связан с биржевым номером</li> </ul>



					заявки в таблице заявок биржи.
=> 270	MDEntryPx	Y	Price		Цена элемента рыночных данных (соответствует заданному типу рыночных данных и относится к текущему элементу MDEntry). Поле условно обязательное, если MDEntryType (269) не является одним из ('A', 'B', 'C', 'J'). Условно обязательно когда MDEntryType (269) = 'Расчетная цена аукциона'
=> 271	MDEntrySize	Y	Qty		Количество или объем элемента рыночных данных (соответствует заданному типу рыночных данных и относится к текущему элементу MDEntry). Поле условно обязательное, если MDEntryType (269) является одним из ('0', '1', '2', 'A', 'B', 'C', 'Q', 'g'). Примечание: для 269=B это поле выражено в количестве единиц финансового инструмента. Для других типов данных это поле выражено в количестве лотов.
=> 272	MDEntryDate	N	UTCDateOnly		Дата, которая относится к данному элементу рыночных данных.
=> 273	MDEntryTime	N	UTCTimeOnly		Время, которое относится к данному элементу рыночных данных.
=> 336	TradingSessionID	N	String		Идентификатор режима торгов SECBOARD Примечание: финансовый инструмент может быть доступен для торгов в разных режимах. Вы должны рассматривать комбинацию Symbol (55) + TradingSessionId (336) как отдельный инструмент с отдельными котировками и таблицами сделок и заявок.
=> 625	TradingSessionSubID	N	String	NA – нет торгов O – период открытия C – период закрытия F – финальный период закрытия N – нормальный период торгов L – аукцион закрытия I – дискретный аукцион D – аукцион крупных пакетов E – период торгов по цене аукциона закрытия	Указывает период торгов  <i>Для инкрементальных обновлений и снэпшотов период торгов указывает на период, в который произошло событие для данного элемента, а не на текущий торговый период.</i>
=> 276	QuoteCondition	N	MultipleValueStrin	'C' (Наилучшая цена)	Список условий, которые характеризуют котировку,

			g		условия между собой разделены пробелами.
=> 277	TradeCondition	H	MultipleValueString	'C' (Расчеты по сделке осуществляются в день заключения сделки) 'J' (Расчеты по сделке осуществляются на следующий день после заключения сделки) 'R' (Цена открытия) 'AJ' (Официальная цена закрытия); '98' (Минимальное значение); '99' (Максимальное значение)	Условия, которые характеризуют сделку или рыночные данные, которые рассчитываются на базе сделки, условия между собой разделены пробелами.
=> 286	OpenCloseSettleFlag	H	MultipleValueString	'4' (Данные предыдущего торгового дня)	Флаг, который идентифицирует тип элемента рыночных данных.
=> 40	OrdType	H	Char	'1'(Рыночная)	Тип заявки. Используется если MDEntryType (269) = 'g', 'f' Примечание: рыночные в аукцион заявки активируются и публикуются после начала аукциона закрытия. Сделки аукциона заключаются по его окончании.
=> 236	Yield	H	Percentage		Доходность, рассчитанная по цене MDEntryPx (270).
=> 64	SettleDate	H	LocalMktDate		Дата расчетов по сделке (SettlementDate) в YYYYMMDD формате Примечания: Для обычных сделок – дата расчетов Для сделок РЕПО – дата расчетов первой части сделки РЕПО
=> 44	Price	H	Price		Ставка РЕПО для сделок РЕПО
=> 423	PriceType	H	int	'1' проценты ставки РЕПО	Указывает на тип цены (ставка РЕПО в процентах) для сделок РЕПО.
=> 5292	BidMarketSize	H	Int		Суммарный объем рыночных заявок в аукционе закрытия на покупку по ожидаемой цене аукциона, выраженный в единицах инструмента.
=> 5293	AskMarketSize	H	Int		Суммарный объем рыночных заявок в аукционе закрытия на продажу, выраженный в единицах инструмента.
=> 5384	AccruedInterestAmt	H	Amt		Объем накопленного купонного дохода.
=> 5459	SettleType	H	Char		Код расчетов по сделкам (269=z)
=> 5510	ChgFromWAPrice	H	PriceOffset		Изменение цены сделки по сравнению со средневзвешенной ценой предыдущего торгового дня.
=> 5558	BuyBackPx	H	Price		Для сделок РЕПО – сумма сделки РЕПО (269=z).
=> 5559	BuyBackDate	H	LocalMktDate		Для сделок РЕПО – дата второй части сделки РЕПО

					(269=z) .
=> 5677	Repo2Px	H	Price		Для сделок РЕПО – сумма второй части сделки РЕПО (269=z).
=> 5791	TotalVolume	H	Amt		Объём денежных средств. Используется если MDEntryType (269)='f' Рыночные в аукцион закрытия заявки на покупку в качестве объема указывают сумму денежных средств. Заявки другого типа содержат количество лотов торгуемого инструмента.
=> 5902	EffectiveTime	H	UTSTimestamp		Время активации заявки. Заявки и уровни цены в котировках, для которых не наступила активация, не участвуют в торгах.
=> 9820	StartTime	H	UTSTimestamp		Время старта аукциона. Используется для аукционов крупных пакетов и дискретных аукционов.
=> 6139	TotalNumOfTrades	H	int		Общее количество сделок, заключенных на протяжении торгового дня.
=> 6143	TradeValue	H	Amt		Объем совершенных сделок.
=> 7017	VolumeIndicator	H	int	'0'(Нет заявок) '1' (Объем заявок не превышает порговой величины N*) '2' (Объем заявок превышает порговую величинуN*)	Индикатор объема заявок крупными пакетами. Используется когда MDEntryType(269)='v' or 'w'.  N(переменная)*- величина индикатора определяется биржей.
=> 9168	OfferNbOr	H	int		Количество заявок на продажу в очереди.
=> 9169	BidNbOr	H	int		Количество заявок на покупку в очереди.
=> 9280	NominalValue	H	float		Объем копировки РЕПО, выраженный в рублях. В режиме торгов РЕПО с Центральным Контрагентом торги проводятся по ставкам РЕПО в качестве цены денежных средств. Для этого режима, цены финансовых инструментов, участвующих сделках как обеспечение, уменьшаются по правилам управления рисками ЦК на величину, зависящую от типа инструмента и объема заявки в лотах. Объемы денежных средств в заявках на каждом уровне ставки РЕПО суммируются и публикуются в этом поле. Для других режимов торгов это поле не публикуется.

=> 9412	<i>OrigTime</i>	<i>H</i>	<i>int</i>		<i>Время транзакции в микросекундах как указано в торговой системе. Данное значение следует прибавить к значению 273 тэга для получения времени транзакции с микросекундной точностью. Поле доступно в каналах сделок и заявок.</i>
=> 10504	<i>OrderSide</i>	<i>H</i>	<i>char</i>		<i>Направленность заявки.</i>
=> 10505	<i>OrderStatus</i>	<i>H</i>	<i>char</i>	<i>'O' (Активная); 'T' (Время активации не наступило).</i>	<i>Текущий статус заявки. Заявки со статусом T неактивны и не участвуют в торгах.</i>
=> 10509	<i>MinCurrPx</i>	<i>H</i>	<i>Price</i>		<i>Минимальная из двух цен: Официальной текущей цены и цены последней сделки, вошедшей в расчёт официальной текущей цены. Используется для определения условий запрета коротких продаж.</i>
=> 10510	<i>MinCurrPxChgTime</i>	<i>H</i>	<i>UTCTimeOnly</i>		<i>Время изменения минимальной текущей цены.</i>

#### 4.3.6 Market Data - Incremental Refresh (X)

Примечания об обработке обновлений:

- Публикация большого объема обновлений в ограниченном по трафику потоке вещания может занять некоторое время. При старте или окончании торговых периодов публикация обновлений происходит параллельно с публикацией сообщений о смене статуса большого числа инструментов. Параллельная публикация может приводить к получению торговых обновлений из последующего периода торгов немного раньше получения изменения периода торгов, или к получению торговых обновлений из предыдущего периода торгов после получения сообщения о смене статуса.
- Для каналов вещания, допускающих добавления, изменения и удаления данных (заявки, котировки) корректное состояние набора данных достигается после окончания обработки всех элементов повторяющихся групп в сообщении.
- Размер FAST сообщения ограничен величиной MTU size, текущее ограничение составляет 1300 байт. Для массовых обновлений, это ограничение может приводить к публикации изменений в нескольких сообщениях подряд. Корректное состояние набора данных может быть достигнуто только после обработки сообщения заведомо меньшего по размеру, чем предельный размер. Промежуточные результаты обработки нескольких сообщений предельного размера могут приводить к перекрещенным котировкам.

Tag	Поле	Наличие	Тип	Допустимые значения	Примечание
<Standard Message Header>		O			Тип сообщения = 'X'
268	NoMDEntries	O	NumInGroup		Количество элементов в группе MDEntryTypes.
=> 279	MDUpdateAction	O	char	'0' (Добавить) '1' (Изменить) '2' (Удалить)	Действие, которое нужно выполнить при обновлении элемента рыночных данных.
=> 269	MDEntryType	У	char	'0' (Котировки на покупку); '1' (Котировки на продажу); '2' (Информация по последней сделке); '3' (Список индексов); '4' (Цена открытия/цена первой сделки); '5' (Цена закрытия/цена последней сделки предыдущего дня); '7' (Максимальная цена сделки); '8' (Минимальная цена сделки); '9' (Средневзвешенные цены); 'A' (Дисбаланс); 'B' (Объемы сделок);  'J' (Пустой снэпшот) 'N' (Максимальная цена спроса в течение сессии); 'O' (Минимальная цена предложения в течение сессии); 'Q' (Расчетная цена аукциона) 'W' (Цена аукциона закрытия); 'c' (Объем аукциона закрытия); 'f' (Для потоков MSR/MSS – объем рыночных заявок на покупку в аукцион закрытия, в единицах ценных бумаг; для потоков OLR/OLS – рыночная заявка в аукцион закрытия, на покупку ); 'g' (Для потоков MSR/MSS – объем рыночных заявок на продажу в аукцион закрытия, в единицах ценных бумаг; для потоков OLR/OLS	Тип рыночных данных. Примечания: <ul style="list-style-type: none"> <li>Наличие различных типов данных определяется Правилами торгов и зависит от режима торгов и рынка.</li> <li>Различные каналы вещания имеют подмножества значений типа данных из этого списка</li> <li>Empty Book (269=J) означает, что по инструменту отсутствуют какие-либо данные. Это сообщение может быть получено без указания кода инструмента и режима (по всему рынку). В этом случае вы должны удалить все накопленные данные и начать процедуру первичного подключения.</li> <li>Смысл некоторых типов данных зависит от типа рынка (фондовый или валютный)</li> <li>Переговорные режимы торгов не содержат данных о котировках и активных заявках</li> <li>Набор значений этого поля может расширяться при обновлениях торговой системы. Рекомендуется разрабатывать код так, чтобы неизвестные значения этого поля и привязанные к ним другие поля игнорировались в коде до тех пор, пока он не будет адаптирован к новым типам данных</li> <li>Индексы публикуются в канале MSS/MSR</li> <li>Данные предыдущего дня обозначаются наличием поля 286</li> </ul>

				<p>– рыночная заявка в аукцион закрытия, на продажу);  <i>'i'</i> (Спрос сессии);  <i>'j'</i> (Предложение сессии);  <i>'h'</i> (Цена предторгового периода);  <i>'k'</i> (Цена послеторгового периода);  <i>'l'</i> (Рыночная цена 2); Для валютного рынка – цена валютного фиксинга, рассчитанная за период 11:29-12:00 московского времени  <i>'m'</i> (Рыночная цена); Для валютного рынка – цена валютного фиксинга</p> <p><i>'o'</i> (Официальная цена открытия);  <i>'p'</i> (Официальная текущая цена);  <i>'q'</i> (Признаваемая котировка); Для валютного рынка – международная цена валютного фиксинга  <i>'r'</i> (Официальная цена закрытия);  <i>'v'</i> (Совокупный спрос);  <i>'w'</i> (Совокупное предложение);</p> <p><i>'s'</i> (Цена аукциона крупными пакетами)  <i>'x'</i> (Объем аукциона крупными пакетами);  <i>'y'</i> (Накопленный купонный доход на дату расчетов, в рублях, в пересчете на единицу финансового инструмента);  <i>'u'</i> (Дюрация);  <i>'z'</i> (Все сделки/список обезличенных сделок)</p>	
=> 278	MDEntryID	H	String		<p>Идентификатор элемента MDEntry.  Примечания:</p> <ul style="list-style-type: none"> <li>• Для канала сделок (269=z) это поле содержит строку с биржевым номером сделки</li> <li>• Для канала котировок, это поле содержит строку идентификатора уровня цены</li> <li>• Для канала заявок, это поле содержит строку идентификатора сообщения о добавлении новой заявки, который НЕ связан с биржевым номером заявки в таблице заявок биржи.</li> </ul>

=> 55	Symbol	O	String		Код/аббревиатура ценной бумаги. В его качестве используется внутренний идентификатор финансового инструмента на MOEX (SECCODE). Примечание: финансовый инструмент с кодом SecCode может быть доступен для торгов в разных режимах. Вы должны рассматривать комбинацию Symbol (55) + TradingsessionId (336) как отдельный инструмент с отдельными котировками и таблицами сделок и заявок.
=> 83	RptSeq	O	int		Порядковый номер обновления по инструменту. Увеличивается на единицу для каждого обновления торговой информации или статуса инструмента.
=> 270	MDEntryPx	Y	Price		Цена элемента рыночных данных (цена соответствует заданному типу рыночных данных, и относится к текущему элементу MDEntry). Поле условно обязательное, когда (279) = New(0) и MDEntryType (269) не является одним из ('A', 'B', 'C', 'J'). Условно обязательно, когда MDEntryType (269) = 'Расчетная цена аукциона'
=> 271	MDEntrySize	Y	Qty		Количество или объем элемента рыночных данных (соответствует заданному типу рыночных данных и относится к текущему элементу MDEntry). Поле условно обязательное, когда MDUpdateAction (279) = New(0) и MDEntryType (269) является одним из ('0', '1', '2', 'A', 'B', 'C', 'Q', 'g').
=> 272	MDEntryDate	H	UTCDateOnly		Дата, которая относится к данному элементу рыночных данных.
=> 273	MDEntryTime	H	UTCTimeOnly		Время, которое относится к данному элементу рыночных данных.
=> 336	TradingSessionID	H	String		Идентификатор режима торгов SECBOARD. Примечание: финансовый инструмент с кодом SecCode может быть доступен для торгов в разных режимах. Вы должны рассматривать комбинацию Symbol (55) + TradingsessionId (336) как отдельный инструмент с отдельными котировками и таблицами сделок и заявок.
=> 625	TradingSessionSubID	H	String	NA – нет торгов O – период открытия C – период закрытия	Указывает период торгов  <i>Примечание: Значение периода торгов является пустым</i>

				F – финальный период закрытия N – нормальный период торгов L – аукцион закрытия I – дискретный аукцион D – аукцион крупных пакетов E – период торгов по цене аукциона закрытия	<i>до начала торгов и после закрытия торгов. Для инкрементальных обновлений и снэпшотов период торгов указывает на период, в который произошло событие для данного элемента, а не на текущий торговый период.</i>
=> 276	QuoteCondition	H	MultipleValueString	'C' (Наилучшая цена)	Список условий, которые характеризуют котировку, условия между собой разделены пробелами.
=> 277	TradeCondition	H	MultipleValueString	'C' (Расчеты по сделке осуществляются в день заключения сделки) 'J' (Расчеты по сделке осуществляются на следующий день после заключения сделки) 'R' (Цена открытия) 'AJ' (Официальная цена закрытия); '98' (Минимальное значение); '99' (Максимальное значение)	Условия, которые характеризуют сделку или рыночные данные, которые рассчитываются на базе сделки, условия между собой разделены пробелами.
=> 286	OpenCloseSettFlag	H	MultipleValueString	'4' (Данные предыдущего торгового дня)	Флаг, который идентифицирует тип элемента рыночных данных.
=> 40	OrdType	H	Char	'1' (Рыночная)	Тип заявки. Используется если MDEntryType (269) = 'g', 'f' Примечание: рыночные в аукцион заявки активируются и публикуются после начала аукциона закрытия. Сделки аукциона заключаются по его окончании.
=> 451	NetChgPrevDay	H	PriceOffset		Изменение цены последней сделки по отношению к цене последней сделки предыдущего торгового дня.
=> 236	Yield	H	Percentage		Доходность, рассчитанная по цене MDEntryPx (270).
=> 64	SettleDate	H	LocalMktDate		<i>Дата расчетов по сделке (SettlementDate) в YYYYMMDD формате</i> <i>Примечания:</i> <i>Для обычных сделок – дата расчетов</i> <i>Для сделок РЕПО – дата расчетов первой части сделки РЕПО</i>
=> 44	Price	H	Price		Ставка РЕПО для сделок РЕПО
=> 423	PriceType	H	int	'1' проценты ставки РЕПО	Указывает на тип цены (ставка РЕПО в процентах) для сделок РЕПО.
=> 5292	BidMarketSize	H	Int		Суммарный объем рыночных заявок в аукционе закрытия на покупку по ожидаемой цене аукциона, выраженный в единицах инструмента.
=> 5293	AskMarketSize	H	Int		Суммарный объем рыночных заявок в аукционе закрытия



					на продажу, выраженный в единицах инструмента.
=> 5384	AccruedInterestAmt	H	Amt		Объем накопленного купонного дохода.
=> 5459	SettlType	H	Char		Код расчетов по сделкам (269=z)
=> 5510	ChgFromWAPrice	H	PriceOffset		Изменение цены сделки по сравнению со средневзвешенной ценой предыдущего торгового дня.
=> 5558	BuyBackPx	N	Price		Для сделок РЕПО – сумма сделки РЕПО (269=z).
=> 5559	BuyBackDate	H	LocalMktDate		Для сделок РЕПО – дата второй части сделки РЕПО (269=z).
=> 5677	Repo2Px	H	Price		Для сделок РЕПО – сумма второй части сделки РЕПО (269=z).
=> 5791	TotalVolume	H	Int		Объем денежных средств. Используется если MDEntryType (269)='f' Рыночные в аукцион закрытия заявки на покупку в качестве объема указывают сумму денежных средств. Заявки другого типа содержат количество лотов торгуемого инструмента.
=> 5902	EffectiveTime	H	UTSTimestamp		Время активации заявки. Заявки и уровни цены в котировках, для которых не наступила активация, не участвуют в торгах.
=> 6139	TotalNumOfTrades	H	int		Общее количество сделок, заключенных на протяжении торгового дня.
=> 6143	TradeValue	H	Amt		Объем совершенных сделок.
=> 7017	VolumeIndicator	H	int	'0'(Нет заявок) '1'(Объем заявок не превышает пороговой величины N*) '2'(Объем заявок превышает пороговую величинуN*)	Индикатор объема заявок крупными пакетами. Используется когда MDEntryType(269)='v' or 'w'.  N(переменная)*- величина индикатора определяется биржей.
=> 9168	OfferNbOr	H	int		Количество заявок на продажу в очереди.
=> 9169	BidNbOr	H	int		Количество заявок на покупку в очереди.
=> 9280	NominalValue	H	float		Объем котировки РЕПО, выраженный в рублях. В режиме торгов РЕПО с Центральным Контрагентом торги проводятся по ставкам РЕПО в качестве цены

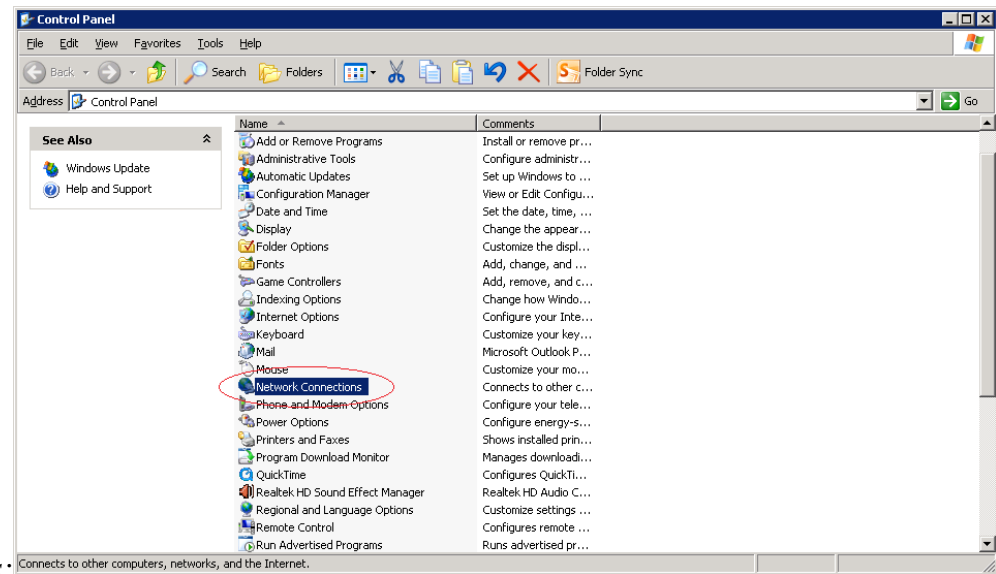
					денежных средств. Для этого режима, цены финансовых инструментов, участвующих сделках как обеспечение, уменьшаются по правилам управления рисками ЦК на величину, зависящую от типа инструмента и объема заявки в лотах. Объемы денежных средств в заявках на каждом уровне ставки РЕПО суммируются и публикуются в этом поле. Для других режимов торгов это поле не публикуется.
=> 9412	<i>OrigTime</i>	<i>N</i>	<i>int</i>		<i>Время транзакции в микросекундах как указано в торговой системе</i> <i>Данное значение следует прибавить к значению 273 тэга для получения времени транзакции с микросекундной точностью. Поле доступно в каналах сделок и заявок.</i>
=> 10504	<i>OrderSide</i>	<i>H</i>	<i>char</i>		<i>Направленность заявки.</i>
=> 10505	<i>OrderStatus</i>	<i>H</i>	<i>char</i>	<i>'O' (Активная);</i> <i>'T' (Время активации не наступило).</i>	<i>Текущий статус заявки. Заявки со статусом T неактивны и не участвуют в торгах.</i>
=>10509	<i>MinCurrPx</i>	<i>H</i>	<i>Price</i>		<i>Минимальная из двух цен: Официальной текущей цены и цены последней сделки, вошедшей в расчёт официальной текущей цены. Используется для определения условий запрета коротких продаж.</i>
=>10510	<i>MinCurrPxChgTime</i>	<i>H</i>	<i>UTCTimeOnly</i>		<i>Время изменения минимальной текущей цены.</i>

## 5. Настройка сетевого соединения

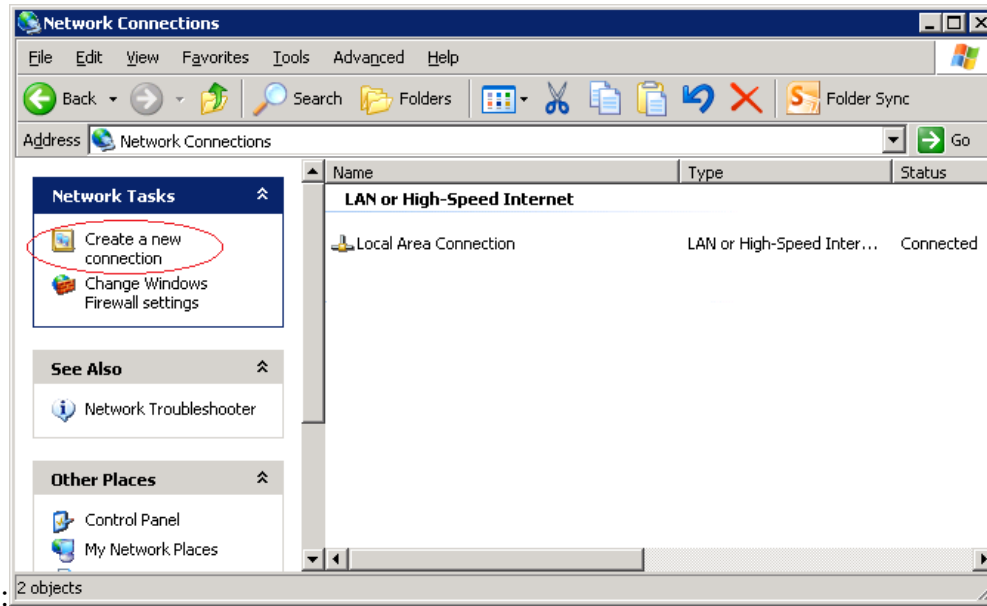
### 5.1. Настройка VPN соединения с МОЕХ на базе Windows XP

Для настройки VPN соединения необходимо выполнить следующее:

1. Убедитесь, что Internet подключен;
2. Нажмите *Start*, перейдите в *Control Panel*;



3. В *Control Panel*, двойной щелчок на *Network Connections*.



4. Выберите *Create a new connection* в *Network Tasks*:


5. В открывшемся помощнике *Network Connection Wizard*, нажмите *Next*:



6. Выберите *Connect to the network at my workplace* и нажмите *Next*:

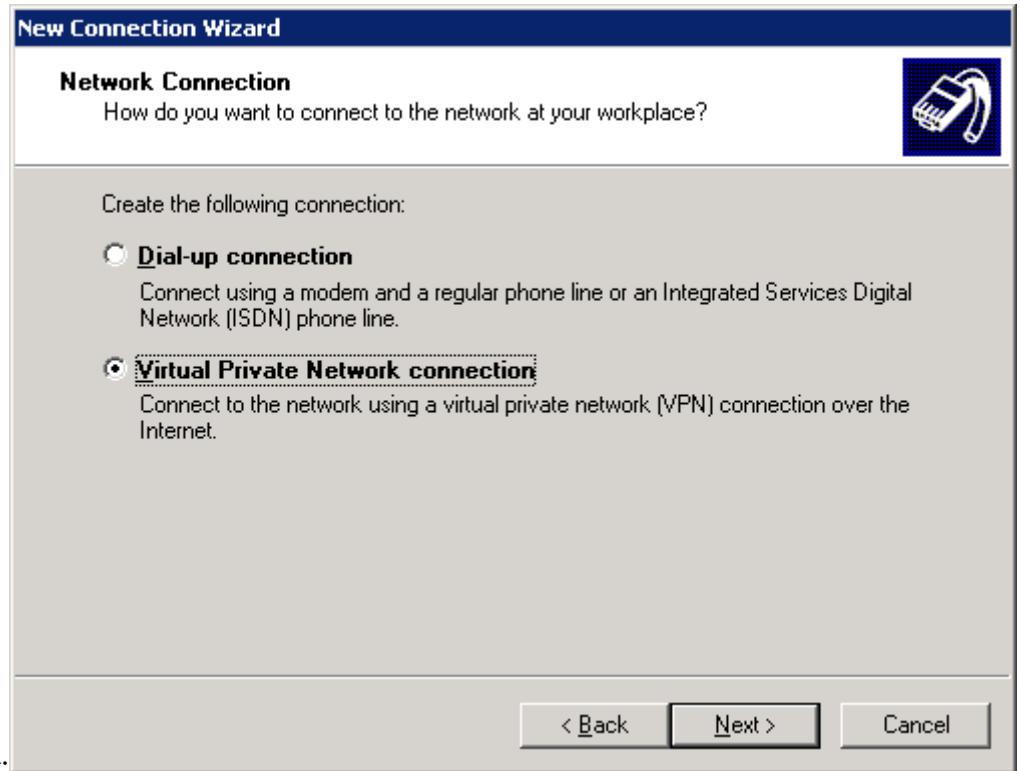
**New Connection Wizard**

**Network Connection Type**  
What do you want to do?



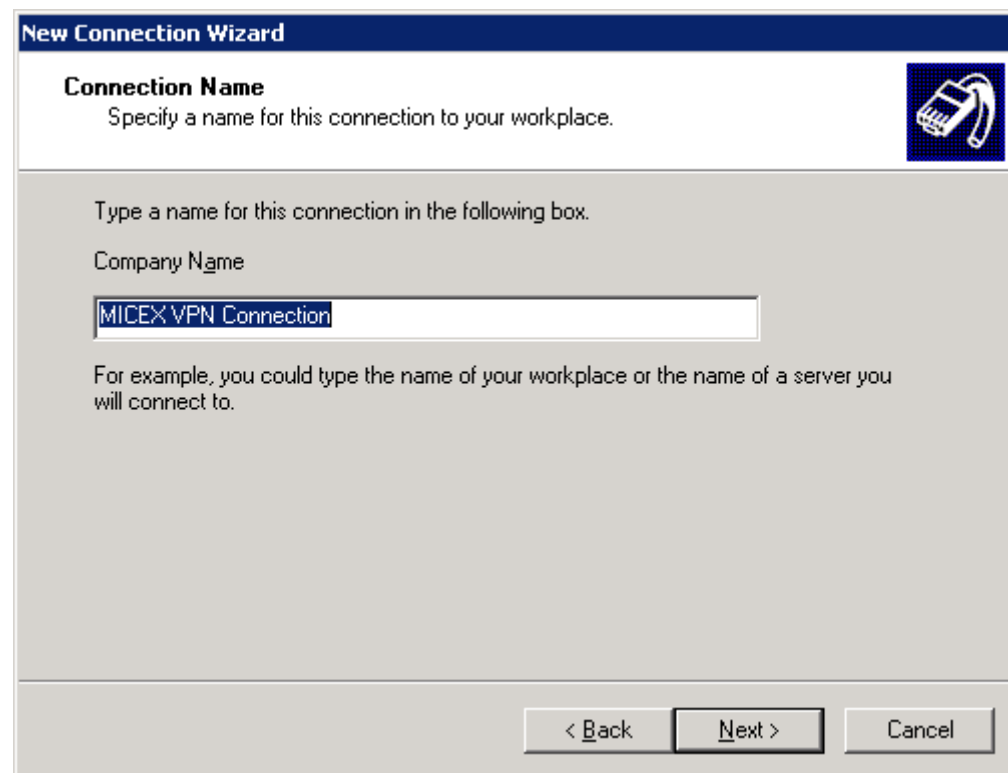
- C**onnect to the Internet  
Connect to the Internet so you can browse the Web and read email.
- C**onnect to the network at my workplace  
Connect to a business network (using dial-up or VPN) so you can work from home, a field office, or another location.
- S**et up an advanced connection  
Connect directly to another computer using your serial, parallel, or infrared port, or set up this computer so that other computers can connect to it.

< Back   Next >   Cancel



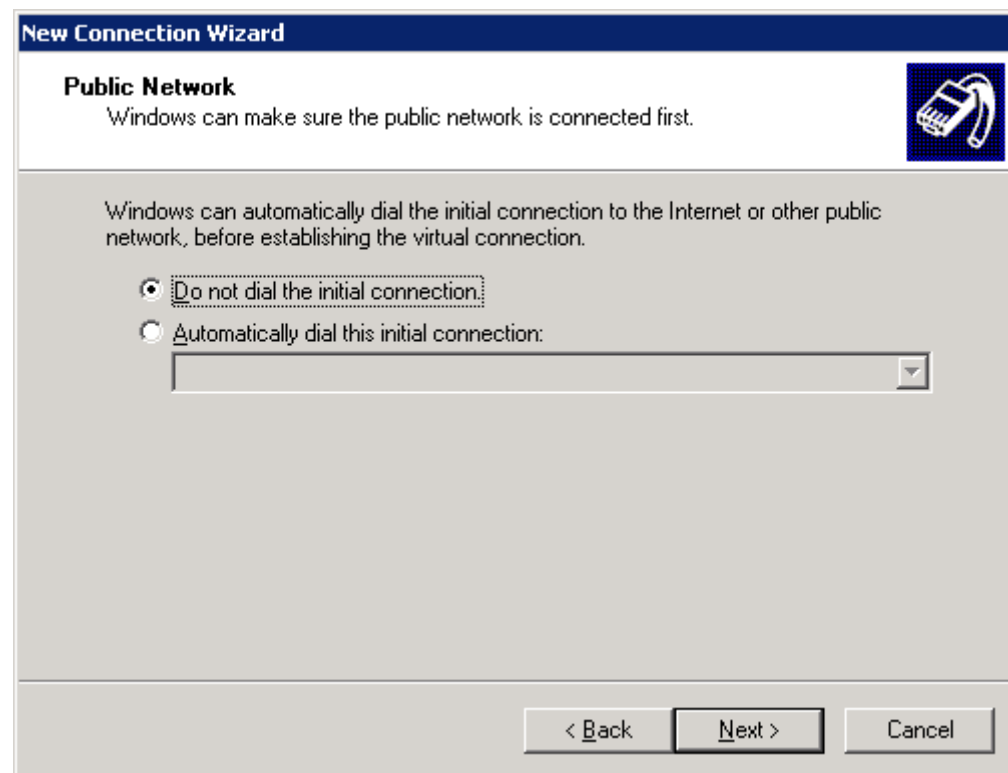
7. Выберите *Virtual Private Network connection* и нажмите *Next*:

8. Введите название в поле *Company Name* (e.g. MOEX VPN Connection), и нажмите *Next*:



9. Выберите *Do not dial the initial connection*, и нажмите *Next*:





10. Введите IP адрес <адрес VPN сервера>, и нажмите *Next*:

## New Connection Wizard

### VPN Server Selection

What is the name or address of the VPN server?



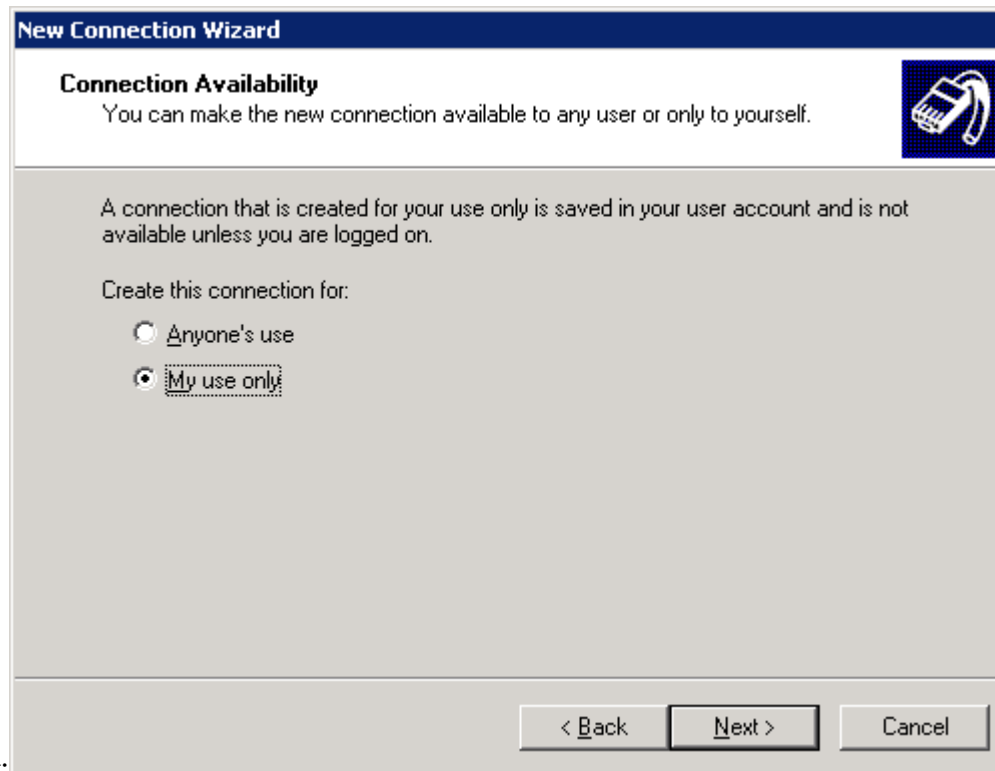
Type the host name or Internet Protocol (IP) address of the computer to which you are connecting.

Host name or IP address (for example, microsoft.com or 157.54.0.1):

< Back

Next >

Cancel



11. Выберите *My use only* и нажмите *Next*:

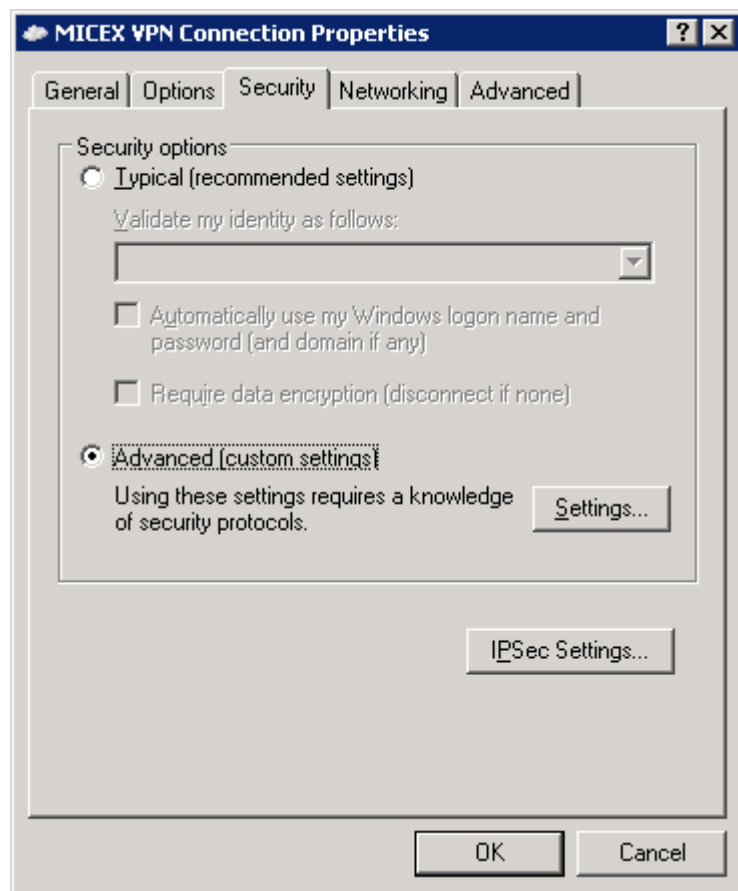


12. Нажмите *Finish*, создание нового соединения завершено:

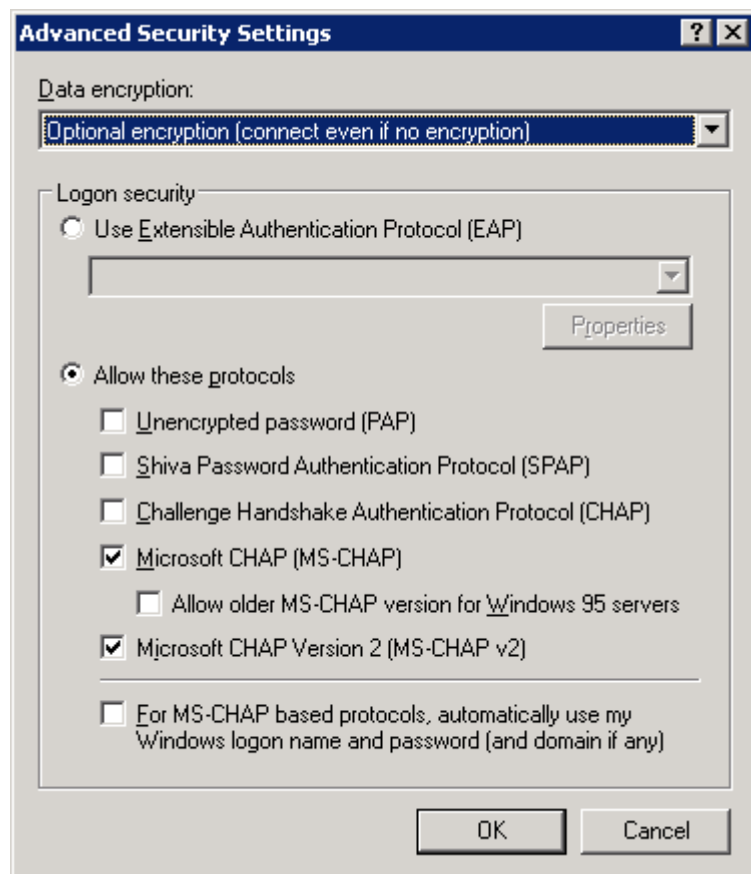


13. Не заполняете поля *User name* и *Password*, нажмите *Properties*

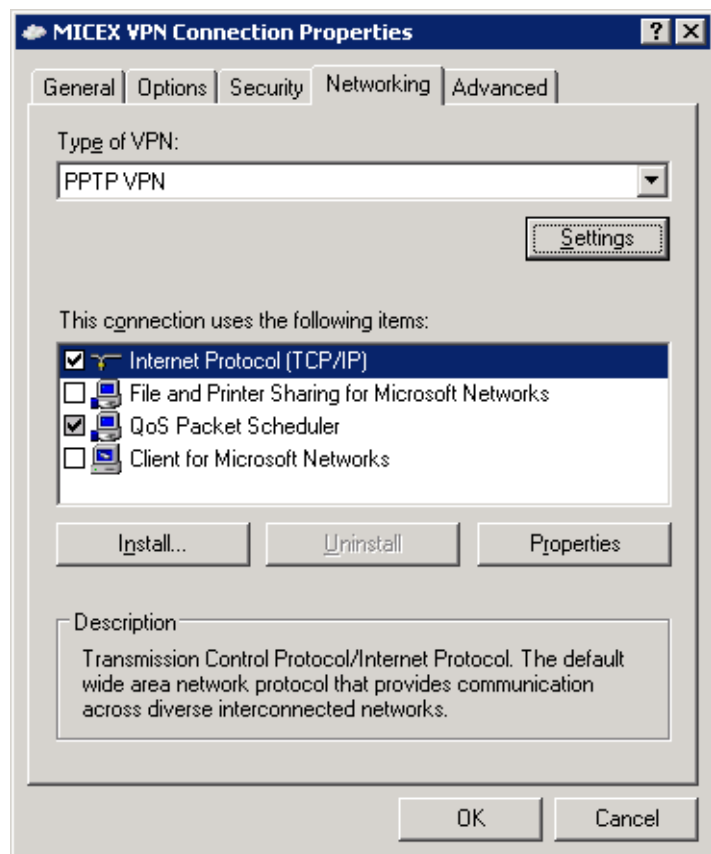
14. Перейдите на закладку *Security*, выберите *Advanced (custom settings)* и нажмите *Settings...*:



15. В выпадающем списке Data encryption выберите *Optional encryption (connect even if no encryption)* и нажмите *OK*:



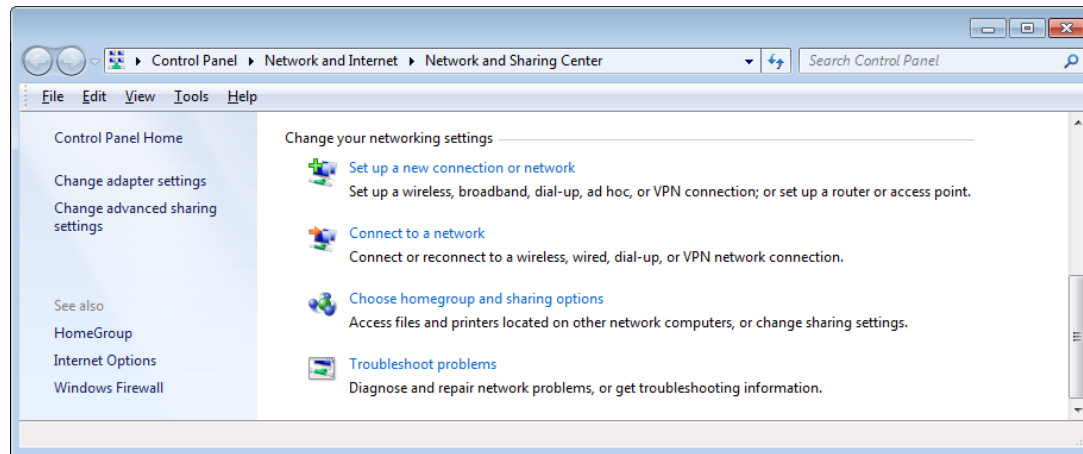
16. На закладке *Networking* в выпадающем списке *Type of VPN* выберите *PPTP VPN* и нажмите *OK*:



## 5.2. Настройка VPN соединения с МОЕХ на базе Windows 7

1. Убедитесь, что Internet подключен.
2. Откройте *Control Panel*→*Network and Internet*→*Network and Share Center* и выберите *Set up a new connection or network*:





3. Выберите *Connect to a workplace* и нажмите *ОК*:



Set Up a Connection or Network



### Choose a connection option



**Connect to the Internet**  
Set up a wireless, broadband, or dial-up connection to the Internet.



**Set up a new network**  
Configure a new router or access point.



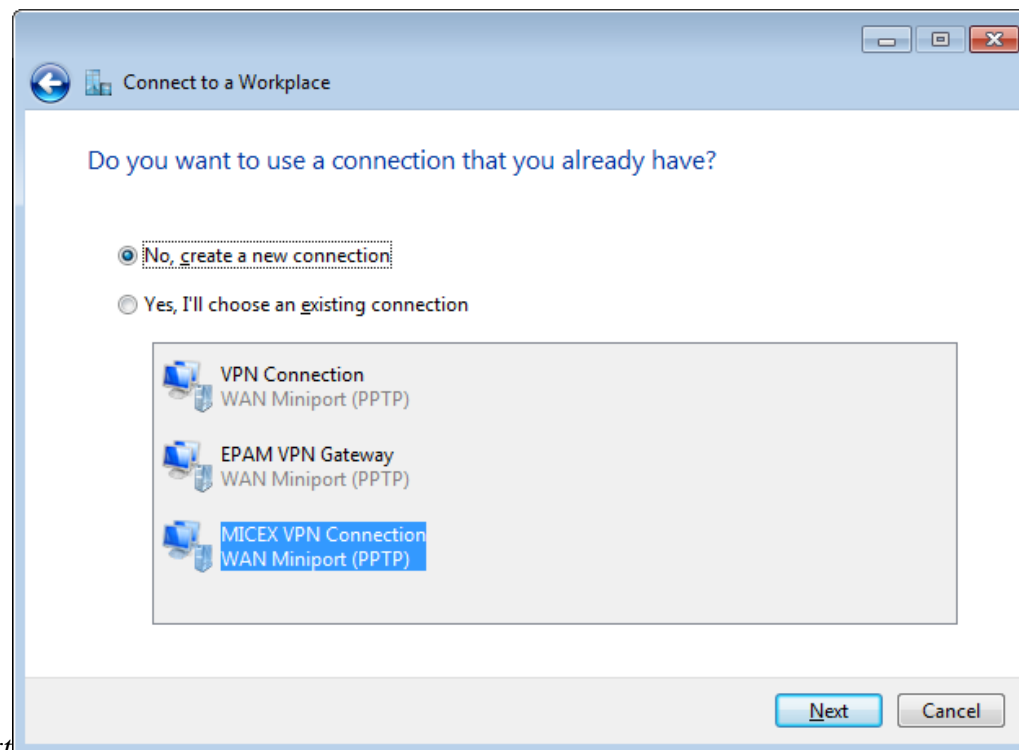
**Connect to a workplace**  
Set up a dial-up or VPN connection to your workplace.



**Set up a dial-up connection**  
Connect to the Internet using a dial-up connection.

Next

Cancel



4. Выберите *No, create a new connection* и нажмите *Next*

5. Нажмите *Use my Internet Connection (VPN)*:



Connect to a Workplace

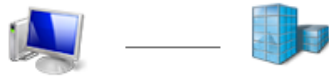


How do you want to connect?

- Use my Internet connection (VPN)  
Connect using a virtual private network (VPN) connection through the Internet.



- Dial directly  
Connect directly to a phone number without going through the Internet.



[What is a VPN connection?](#)

Cancel

6. Введите <адрес VPN сервера> в поле *Internet address*, введите MOEX VPN Connection в поле *Destination name*, выберите *Don't connect now; just set it up so I can connect later* и нажмите *Next*:

Connect to a Workplace

Type the Internet address to connect to

Your network administrator can give you this address.

Internet address:

Destination name:

Use a smart card

Allow other people to use this connection  
This option allows anyone with access to this computer to use this connection.

Don't connect now; just set it up so I can connect later

Next Cancel

7. Оставьте следующее окно без изменений и нажмите *Next*:

Connect to a Workplace

Type your user name and password

User name:

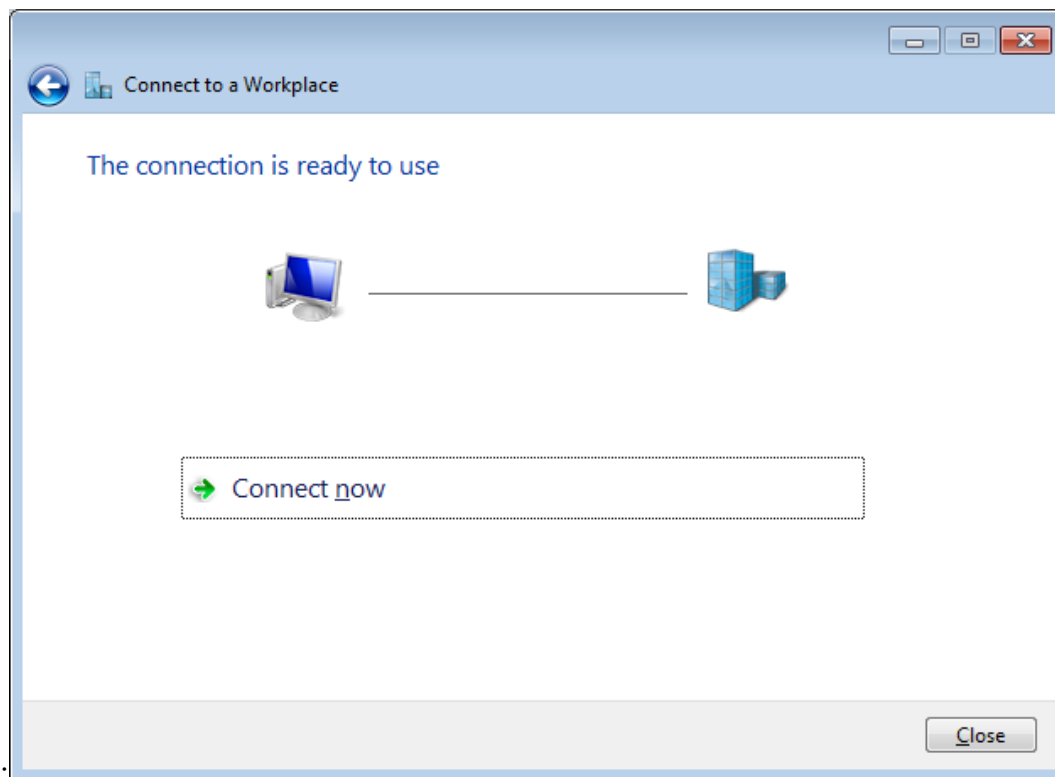
Password:

Show characters

Remember this password

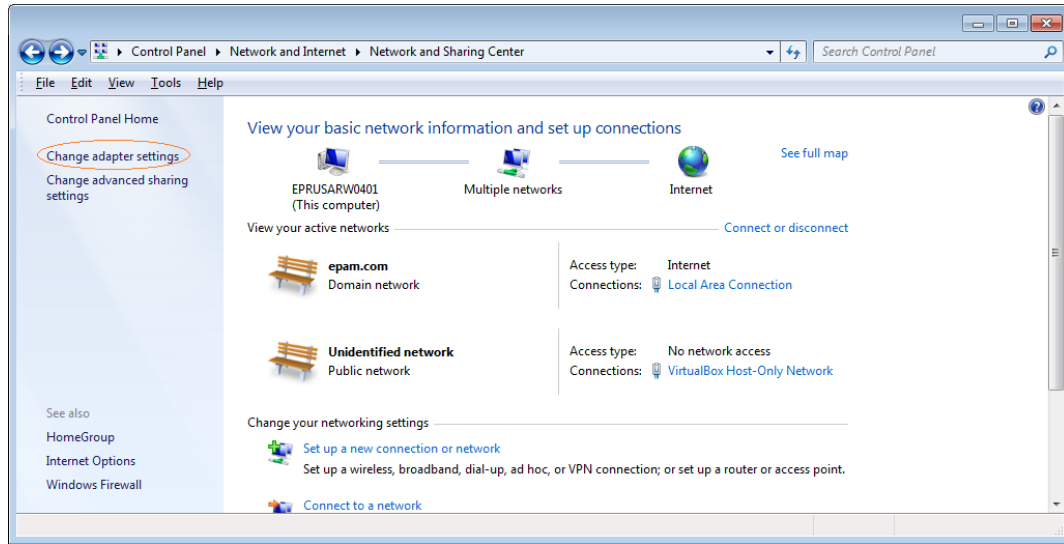
Domain (optional):

Create Cancel

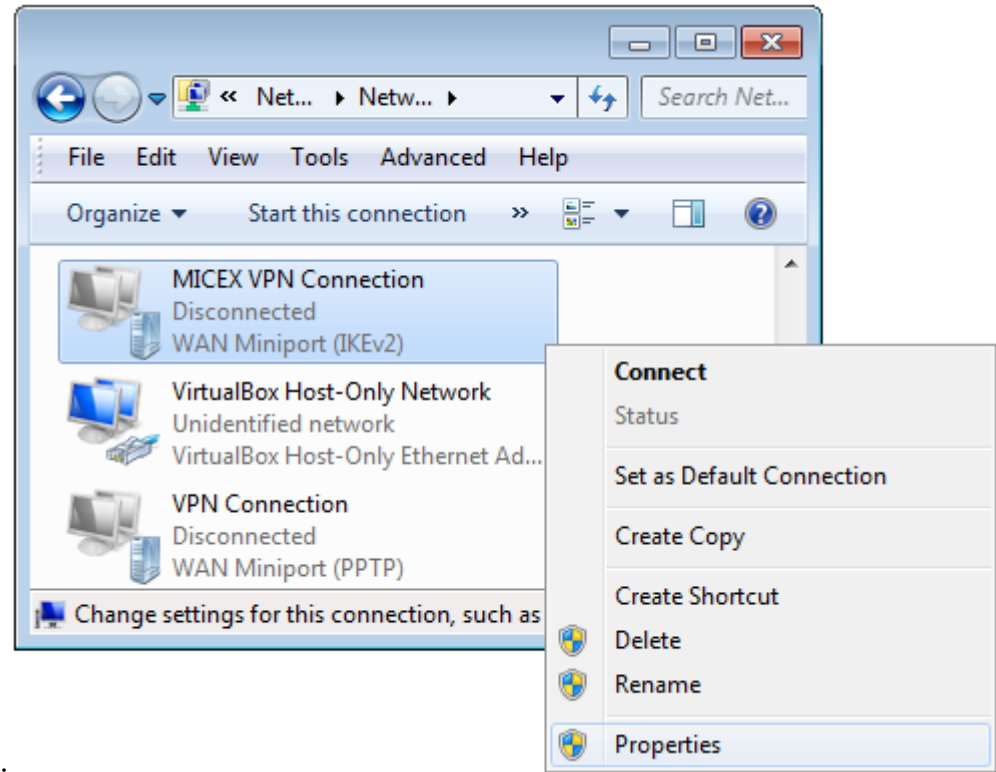


8. Нажмите *Close*:

9. Откройте *Control Panel*→*Network and Internet*→*Network and Share Center* и нажмите *Change adapter setting*:

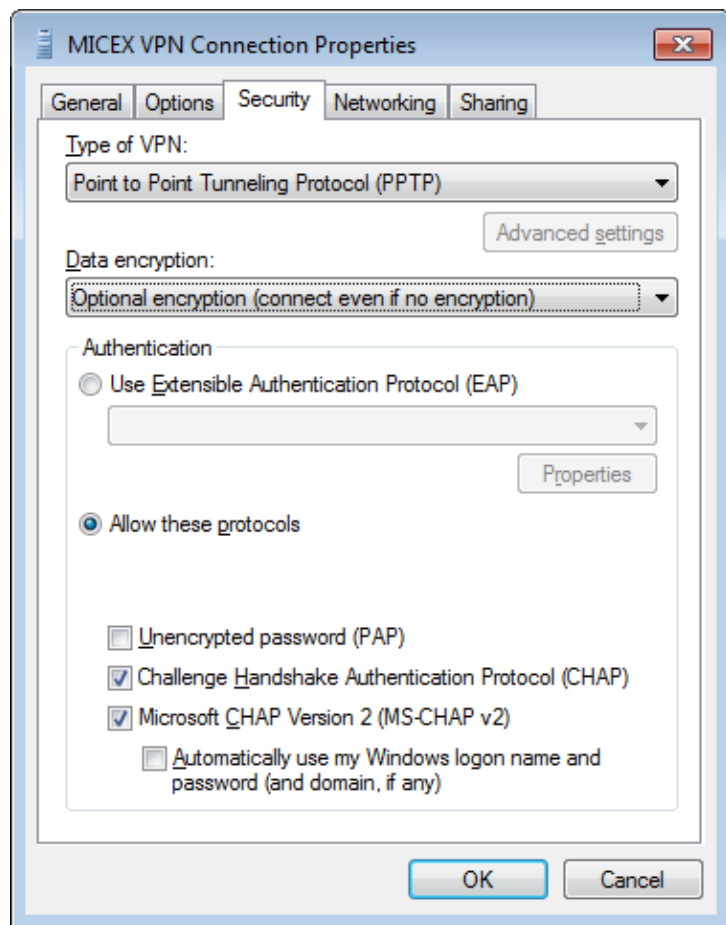






10. Откройте *Properties* для только что созданного соединения:

11. На закладке *Security* выберите в выпадающем списке *Type of VPN - Point to Point Tunneling Protocol (PPTP)*, выберите в выпадающем списке *Data encryption - Optional encryption (connect even if no encryption)* и нажмите *OK*:



### 5.3. Настройка VPN соединения с MOEX на базе OpenSUSE

1. Убедитесь, что Internet подключен;
2. Установите *pptp* клиент, используя следующие команды:

```
sudo zypper install pptp
```

3. Выполните следующую команду:

```
sudo /usr/sbin/pptp-command setup
```

4. Введите '4' и нажмите enter:

```
1.) Manage CHAP secrets
2.) Manage PAP secrets
3.) List PPTP Tunnels
4.) Add a NEW PPTP Tunnel
5.) Delete a PPTP Tunnel
6.) Configure resolv.conf
7.) Select a default tunnel
8.) Quit
?: 4 + <enter>
```

5. Введите '1' и нажмите enter:

```
Add a NEW PPTP Tunnel.
```

```
1.) Other
Which configuration would you like to use?: 1 + <enter>
```

6. Введите 'MOEX\_vpn\_connection' и нажмите enter:

```
Tunnel Name: MOEX_vpn_connection + <enter>
```

7. Введите <адрес VPN сервера> и нажмите enter:

```
Server IP: <адрес VPN сервера> + <enter>
```

8. Введите 'del default' и нажмите enter:

```
route: del default + <enter>
```

9. Введите 'add default gw 1.1.1.1 TUNNEL\_DEV' и нажмите enter:

```
route: add default gw 1.1.1.1 TUNNEL_DEV
```

10. Просто нажмите enter:

```
route: <enter>
```

11. Введите 'test' и нажмите enter:

```
Local Name: test
```

12. Оставьте значения по умолчанию и нажмите enter:

```
Remote Name [PPTP]: <enter>
```

13. Если все было сделано правильно, вы должно увидеть:

```
Adding MOEX_vpn_connection - <адрес VPN сервера> - test - PPTP  
Added tunnel MOEX_vpn_connection
```

14. Введите 'q' и нажмите enter чтобы выйти из помощника по установке.

15. Теперь необходимо сделать несколько изменений в только что созданном файле конфигурации. Для начала необходимо открыть его, выполнив команду:

```
sudo vim /etc/ppp/peers/MOEX_vpn_connection
```

16. Необходимые изменения отмечены **красным**:

```
#  
# PPTP Tunnel configuration for tunnel MOEX_vpn_connection  
# Server IP: <адреса VPN сервера>  
# Route: route del default  
# Route: route add default gw 1.1.1.1 TUNNEL_DEV  
#  
noauth
```

```
#
# Tags for CHAP secret selection
#
name test
remotename PPTP

#
# Include the main PPTP configuration file
#
# file /etc/ppp/options.pptp
```

17. Пожалуйста, не забудьте сохранить файл перед закрытием. На этом все. Теперь необходимо установить VPN соединение, выполнив команду:

```
sudo /usr/sbin/pptp-command start MOEX_vpn_connection
```

На экране должно появиться примерно следующее:

```
Using interface ppp0
Connect: ppp0 <--> /dev/pts/1
local IP address 1.1.1.19
remote IP address 1.1.1.1
Script ?? finished (pid 30023), status = 0x0
Script /etc/ppp/ip-up finished (pid 30032), status = 0x0
Route: add -net 0.0.0.0 gw 1.1.1.1 added
Route: add -net 1.1.1.0 netmask 255.255.255.0 gw 1.1.1.1 added
All routes added.
Tunnel MOEX_vpn_connection is active on ppp0. IP Address: 1.1.1.19
```

18. Чтобы остановить соединение, нужно выполнить следующую команду:

```
sudo /usr/sbin/pptp-command stop
```

19. **Важно:** После того, как VPN соединение будет закрыто необходимо восстановить правила маршрутизации. Иначе следующие попытки установить VPN соединение будут безуспешными.

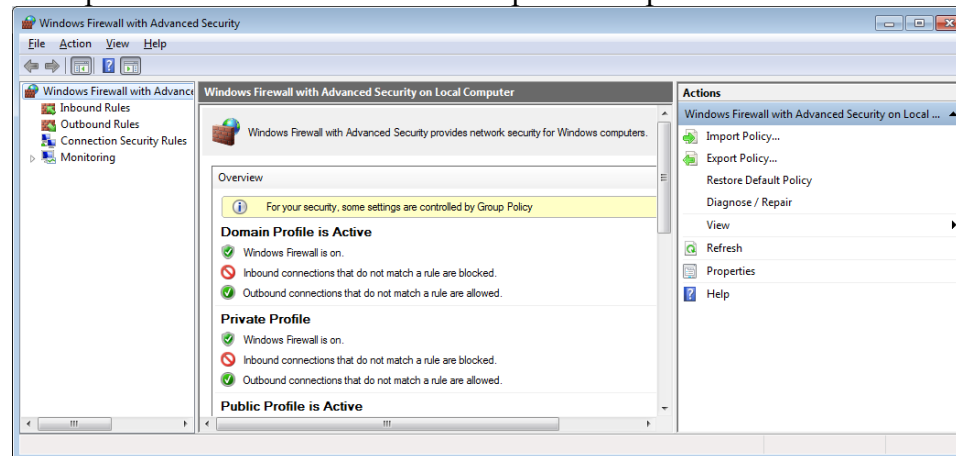
## 5.4. Часто возникающие вопросы и методы их решения

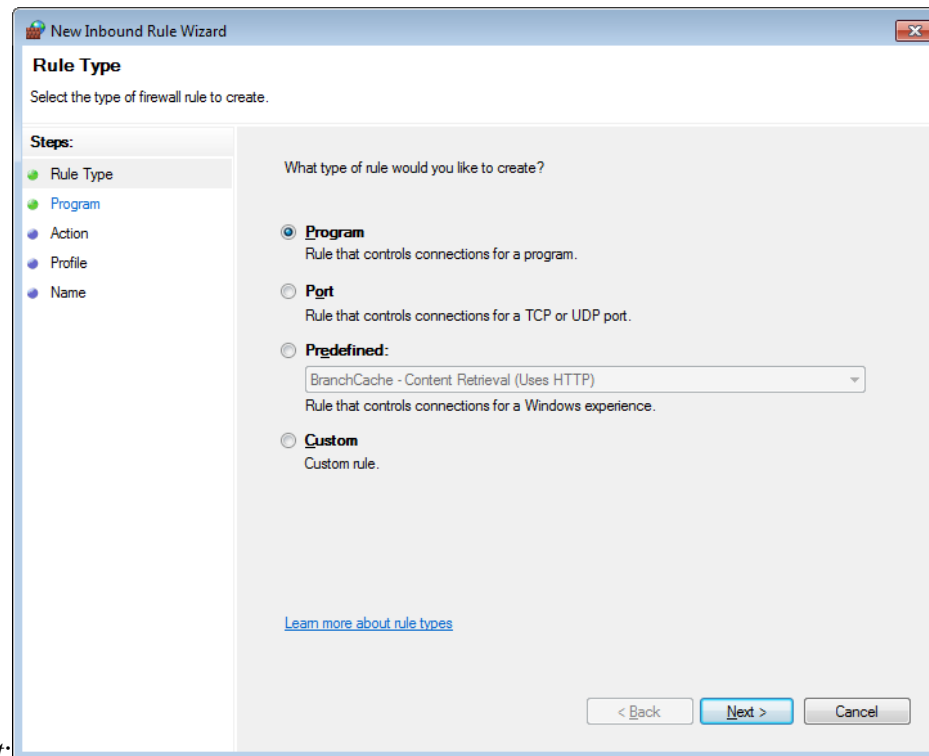
1. VPN соединение установлено, но приложение не получает UDP пакеты (Windows 7)

1.1 Посмотрите на состояние вашего VPN соединения, и проверьте, что количество «Полученных» байт возрастает. Если это не так, обратитесь в службу поддержки МОЕХ.

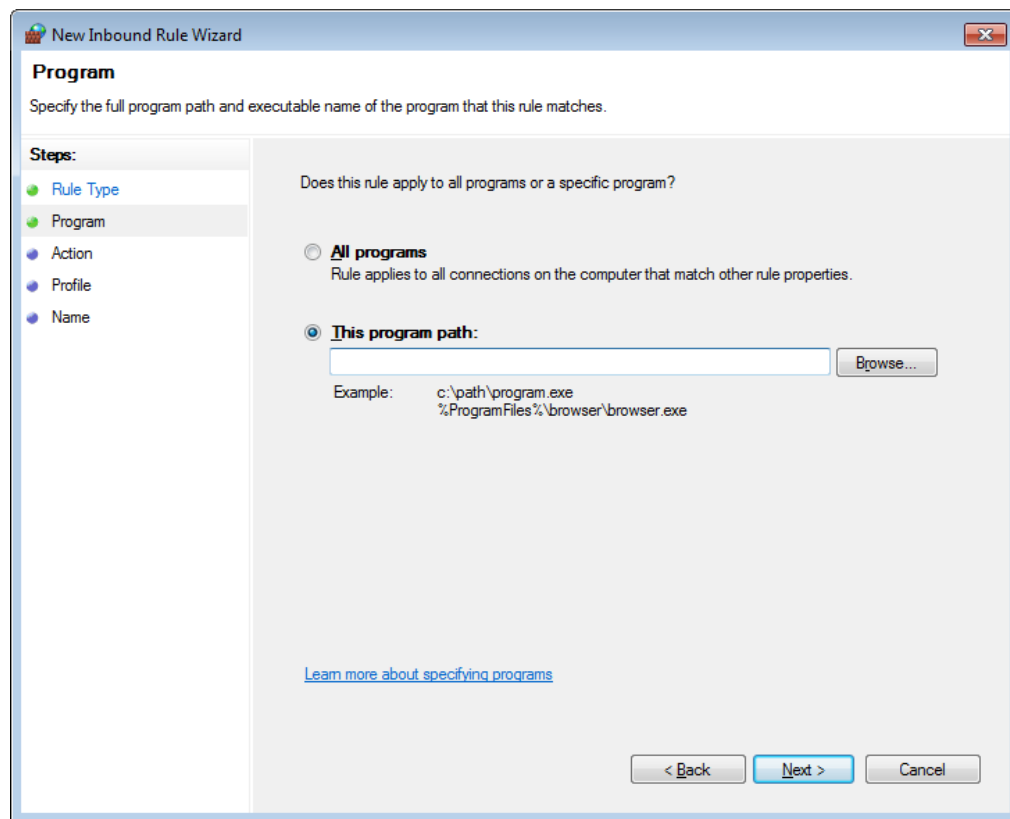
1.2 Проверьте настройки firewall. Временно отключите firewall. Если после этого все «заработает», включите firewall и добавьте следующие настройки:

- ✓ Откройте *Windows Firewall* → *Advanced* настройки;
- ✓ Выберите *Inbound Rules* и в меню справа выберите *New Rule*:





- ✓ Оставьте окно без изменений и нажмите *Next*:
- ✓ В следующем окне необходимо указать путь к программе:



✓ Оставьте следующие окна без изменений:



New Inbound Rule Wizard

**Action**

Specify the action to be taken when a connection matches the conditions specified in the rule.

**Steps:**

- Rule Type
- Program
- **Action**
- Profile
- Name

What action should be taken when a connection matches the specified conditions?

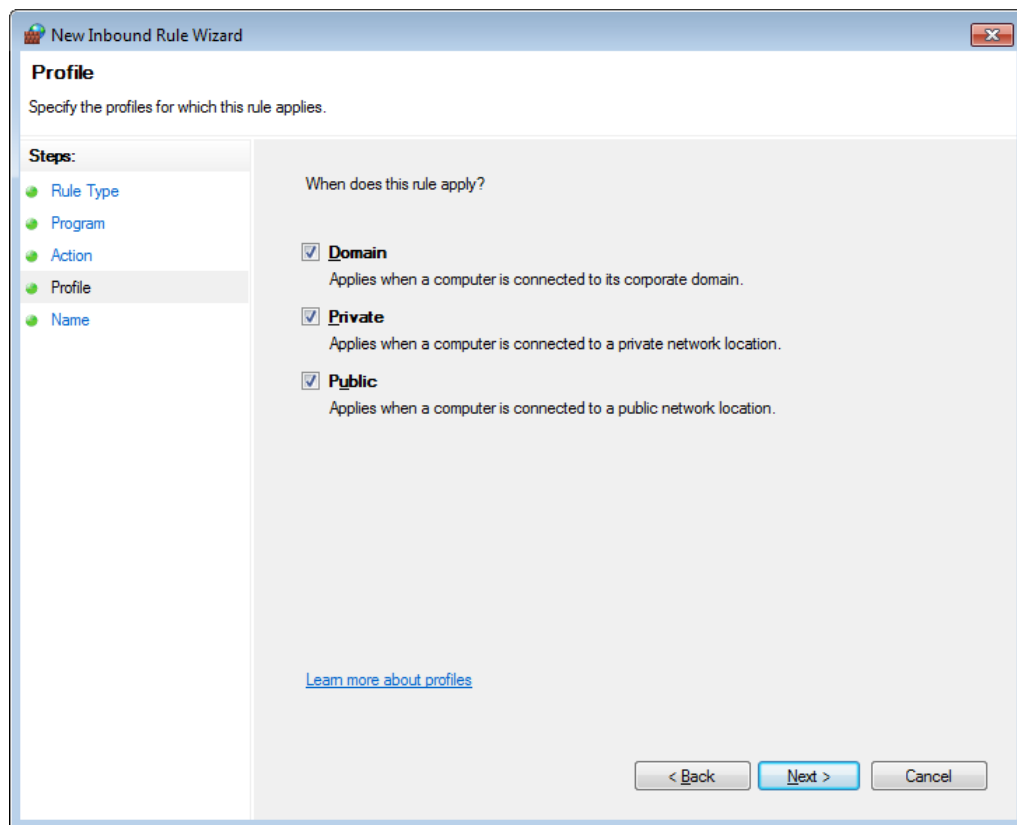
**Allow the connection**  
This includes connections that are protected with IPsec as well as those are not.

**Allow the connection if it is secure**  
This includes only connections that have been authenticated by using IPsec. Connections will be secured using the settings in IPsec properties and rules in the Connection Security Rule node.

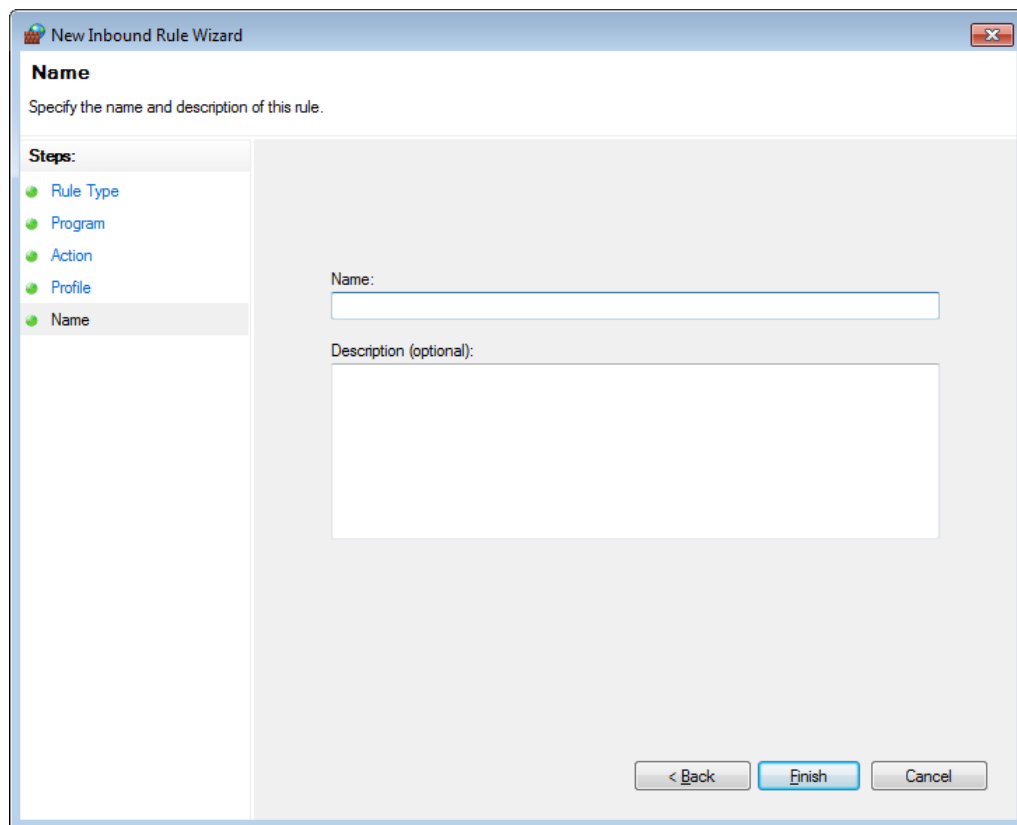
**Block the connection**

[Learn more about actions](#)

< Back   Next >   Cancel



- ✓ Укажите имя создаваемого правила. E.g. MyApplicationRule. Нажмите *Finish*.



## 6. Сертифицированные средства работы

### 6.1. Библиотека FIX Antenna™ от EPAM – B2Bits®

EPAM Systems — крупнейший [разработчик проектного \(заказного\) программного обеспечения](#) и один из ведущих игроков в сфере ИТ-консалтинга в Центральной и Восточной Европе, и Центр компетенций по финансовым рынкам B2BITS®, входящий в состав компании

EPAM Systems, сертифицировали свою высокопроизводительную библиотеку [FIX Antenna™](#) для работы с платформой MOEX Market Data Multicast.

[FIX Antenna™](#) (C++ and .NET) позволяет осуществлять подписку на рыночные данные от MOEX Market Data Multicast прозрачно для программиста, скрывая всю работу по работе с потоками, восстановлению потерянных сообщений, предоставляя программисту объектно-ориентированный API. Пакет включает полную документацию и примеры кода, иллюстрирующие использование [FIX Antenna™](#) с платформой MOEX Market Data Multicast.

### 6.1.1 Quick Start – примеры кода

Ниже представлен код простейшего клиента. Этот код может служить скелетом реального приложения и демонстрирует возможные способы использования [FIX Antenna™](#) для получения данных от MOEX Market Data Multicast.

```
instrument_listener_impl.h:
#pragma once

#include <B2BITS_MOEX_mfix_listeners.h>

namespace mfix_MOEX_client {
class instrument_listener_impl
: public MOEX_mfix::instrument_listener {
public:
virtual bool on_security_definition(const MOEX_mfix::security_description &sec_desc,
                                   const MOEX_mfix::security_id &sec_id,
                                   const MOEX_mfix::symbol &symb,
                                   const std::string &board,
                                   const Engine::FIXMessage &d_msg,
                                   const std::string &channel_id)
{
    //add your processing code here, return your result true or false
    return false;
}

virtual void on_subscribed(const MOEX_mfix::symbol &symb,
                          const std::string &board,
                          MOEX_mfix::mfix_feed_type feed_type)
{
    //add your processing code here
}

virtual void on_unsubscribed(const MOEX_mfix::symbol &symb,
                             const std::string &board,
                             MOEX_mfix::mfix_feed_type feed_type)
{
    //add your processing code here
}
}
```

```

}

virtual void on_increment(const MOEX_mfix::symbol &symb,
                        const std::string &board,
                        const Engine::TagValue &entry,
                        MOEX_mfix::mfix_feed_type feed_type)
{
    //add your processing code here
}

virtual void on_security_status(const MOEX_mfix::symbol &symb,
                              const std::string &board,
                              const Engine::FIXMessage &msg,
                              MOEX_mfix::mfix_feed_type feed_type)
{
    //add your processing code here
}

virtual bool on_natural_refresh(const MOEX_mfix::symbol &symb,
                              const std::string &board,
                              const MOEX_mfix::increments &nr_msgs,
                              MOEX_mfix::mfix_feed_type feed_type)
{
    //add your processing code here, return your result true or false
    return true;
}

virtual void on_snapshot(const MOEX_mfix::symbol &symb,
                        const std::string &board,
                        const MOEX_mfix::snapshots &msgs,
                        MOEX_mfix::mfix_feed_type feed_type)
{
    //add your processing code here
}

virtual void on_recovery_started(const MOEX_mfix::symbol &symb,
                               const std::string &board,
                               MOEX_mfix::mfix_feed_type feed_type)
{
    //add your processing code here, return your result true or false
    return false;
}

virtual void on_recovery_stopped(const MOEX_mfix::symbol &symb,
                                const std::string &board,
                                MOEX_mfix::mfix_recovery_reason reason,
                                MOEX_mfix::mfix_feed_type feed_type)
{
    //add your processing code here
}

```

```

    }

    virtual void on_error(const MOEX_mfix::symbol &symb,
                        const std::string &board,
                        const std::string &error,
                        MOEX_mfix::mfix_feed_type feed_type)
    {
        //add your processing code here
    }
};
}

```

### *application\_listener\_impl.h:*

```
#pragma once
```

```
#include <B2BITS_MOEX_mfix_listeners.h>
```

```

namespace mfix_MOEX_client {
class application_listener_impl
: public MOEX_mfix::MOEX_mfix_application_listener {
public:
    virtual void on_error(const std::string &error)
    {
        //add your processing code here
    }

    virtual void on_process(const Engine::FIXMessage &msg, const std::string &channel_id)
    {
        //add your processing code here
    }

    virtual void on_feed_reset(const std::string &channel_id, MOEX_mfix::mfix_feed_type feed_type)
    {
        //add your processing code here
    }

    virtual void on_heartbeat(const std::string &channel_id, MOEX_mfix::mfix_feed_type feed_type)
    {
        //add your processing code here
    }
};
}

```

### *main.cpp:*

```
#include <iostream>
```

```
#include <B2BITS_FixEngine.h>
```

```
#include <B2BITS_MOEX_mfix_application.h>
```

```

#include "application_listener_impl.h"
#include "instrument_listener_impl.h"

using namespace mfix_MOEX_client;

void subscribe_and_wait(MOEX_mfix::MOEX_mfix_application *app,
                       instrument_listener_impl *ins_listener);

int main(int argc, char *argv[])
{
    MOEX_mfix::MOEX_mfix_application *app = nullptr;
    application_listener_impl *app_listener = nullptr;
    instrument_listener_impl *ins_listener = nullptr;

    try {
        Engine::FixEngine::init("./engine.properties");

        //configure parameters
        MOEX_mfix::MOEX_mfix_application_params app_params;
        app_params.templates_fn_ = "./FIX50SP2.xml";
        app_params.config_xml_ = "./config.xml";

        app_listener = new application_listener_impl();
        app = Engine::FixEngine::singleton()->createMOEXApplication(app_params, app_listener);

        subscribe_and_wait(app, ins_listener);
    } catch (const Utils::Exception &ex) {
        std::cerr<<"Exception: "<<ex.what()<<"\n";
        if (nullptr != app_listener) {
            app_listener->release();
        }

        if (nullptr != ins_listener) {
            ins_listener->release();
        }

        return 100;
    }

    app_listener->release();
    ins_listener->release();

    app->release();

    return 0;
}

void subscribe_and_wait(MOEX_mfix::MOEX_mfix_application *app,
                       instrument_listener_impl *ins_listener)

```

```

{
    //get channels id
    MOEX_mfix::channel_ids channels(app->get_channel_ids());

    //get orderbook feed
    MOEX_mfix::MOEX_feed &order_book_feed = app->get_orderbook_feed();

    ins_listener = new instrument_listener_impl();

    //subscribe to known instrument in channel[1], with market recovery as recovery type
    order_book_feed.subscribe_by_symbol("AFLT", "EQBR", *ins_listener,
                                       channels[1],MOEX_mfix::RM_USE_MARKET_RECOVERY);

    while (true) {
        std::cout<<"Type 'q' for exit\n\n";
        char c;
        std::cin>>c;
        if ('q' == c || 'Q' == c) {
            break;
        }
    }

    order_book_feed.unsubscribe_by_symbol("AFLT", "EQBR", channels[1]);
}

```

## 6.1.2 Обзор API

Для использования библиотеки необходимо подключить следующие заголовки:

```
/include/B2BITS_MOEX_mfix_application.h
```

```
/include/B2BITS_MOEX_mfix_listeners.
```

```
/include/B2BITS_MOEX_mfix_types.h
```

Список классов, структур, объединений и интерфейсов с кратким описанием:

<b>MOEX_mfix::instrument_listener</b>	Слушатель инструментов ( паттерн observer)
<b>MOEX_mfix::MOEX_feed</b>	Представляет MOEX feed (поток - stream)
<b>MOEX_mfix::MOEX_mfix_application</b>	Представляет MOEX mfix application –



	класс с основными коллбэками
<b>MOEX_mfix::MOEX_mfix_application_listener</b>	Представляет MOEX mfix application listener
<b>MOEX_mfix::MOEX_mfix_application_params</b>	Параметры библиотеки
<b>MOEX_mfix::security_definition_listener</b>	Получает описания финансовых инструментов, вещаемых в потоке рыночных данных

### 6.1.2.1.1. *MOEX\_mfix::instrument\_listener*

```
#include <B2BITS_MOEX_mfix_listeners.h>
```

#### Public Member Functions

virtual void	<b>on_subscribed</b> (const symbol &symb, const std::string &board, mfix_feed_type feed_type)=0
	Вызывается при успешной подписке на инструменты.
virtual void	<b>on_unsubscribed</b> (const symbol &symb, const std::string &board, mfix_feed_type feed_type)=0
	Вызывается при успешной отмене подписки на инструменты.
virtual void	<b>on_increment</b> (const symbol &symb, const std::string &board, const Engine::TagValue &entry, mfix_feed_type feed_type)=0
	Вызывается, когда пользователю необходимо очистить данные и заполнить новыми.
virtual void	<b>on_security_status</b> (const symbol &symb, const std::string &board, const Engine::FIXMessage &msg, mfix_feed_type feed_type)=0
	Вызывается, когда пользователю необходимо обновить статус инструмента.
virtual bool	<b>on_natural_refresh</b> (const symbol &symb, const std::string &board, const increments &nr_msgs, mfix_feed_type feed_type)=0
	Вызывается, когда пользователю необходимо очистить данные, при этом

	используется Восстановление по инкрементальным обновлениям. Вернуть true, если данные восстановлены, false в противном случае. Возвращает true, если данные восстановлены, false в противном случае
virtual void	<b>on_snapshot</b> (const symbol &symb, const std::string &board, const snapshots &msgs, mfix_feed_type feed_type)=0
	Вызывается, когда пользователю необходимо очистить данные и заполнить пришедшим снэпшотом.
virtual void	<b>on_recovery_started</b> (const symbol &symb, const std::string &board, mfix_feed_type feed_type)=0
	Вызывается при начале процедуры восстановления.
virtual void	<b>on_recovery_stopped</b> (const symbol &symb, const std::string &board, mfix_recovery_reason reason, mfix_feed_type feed_type)=0
	Вызывается при окончании процедуры восстановления.
virtual void	<b>on_error</b> (const symbol &symb, const std::string &board, const std::string &error, mfix_feed_type feed_type)=0
	Вызывается при ошибке (например, когда пытаются подписаться на инструмент повторно.)

**Примечание:**

Объекты этого класса не следует класть в std::auto\_ptr или в другие умные указатели (за исключением специально разработанных, например Utils::RefCountPtr из состава библиотеки). Объекты должны создаваться оператором new.

**6.1.2.1.2. MOEX\_mfix::MOEX\_feed**

```
#include <B2BITS_MOEX_mfix_application.h>
```

**Public Member Functions**

virtual void	<b>subscribe_by_symbol</b> (const symbol &symb, const std::string &board, instrument_listener &listener, const std::string &channel_id, mfix_recovery_mode recovery=RM_USE_MARKET_RECOVERY)=0
--------------	---

	Подписка на инструмент по его коду
virtual void	<b>unsubscribe_by_symbol</b> (const symbol &symb, const std::string &board, const std::string &channel_id)=0
	Отмена подписки на инструмент по его коду
virtual void	<b>subscribe_all</b> (instrument_listener &listener, const std::string &channel_id, mfix_recovery_mode recovery=RM_USE_MARKET_RECOVERY)=0
	Подписка на все инструменты
virtual void	<b>unsubscribe_all</b> (const std::string &channel_id)=0
	Отмена подписки на все инструменты

### 6.1.2.1.3. *MOEX\_mfix::MOEX\_mfix\_application*

```
#include <B2BITS_MOEX_mfix_application.h>
```

#### Public Member Functions

virtual void	<b>release</b> ()=0
	Освобождение ресурсов, выделенных приложением
virtual MOEX_feed &	<b>get_orderbook_feed</b> () const =0
	Получение ссылки на поток котировок.
virtual MOEX_feed &	<b>get_statistics_feed</b> () const =0
	Получение ссылки на поток рыночной статистики.
virtual MOEX_feed &	<b>get_orders_feed</b> () const =0
	Получение ссылки на поток обезличенных заявок
virtual MOEX_feed &	<b>get_trades_feed</b> () const =0
	Получение ссылки на поток обезличенных сделок

virtual const channel_ids &	<b>get_channel_ids</b> () const =0
	Возвращает идентификаторы канала

#### 6.1.2.1.4. *MOEX\_mfix::MOEX\_mfix\_application\_listener*

```
#include <B2BITS_MOEX_mfix_listeners.h>
```

#### Public Member Functions

virtual void	<b>on_error</b> (const std::string &error)=0 Вызывается при ошибке в MOEX mfix application. Этот метод может быть вызван из другого потока, это следует учитывать при имплементации.
virtual void	<b>on_process</b> (const Engine::FIXMessage &msg, const std::string &channel_id)=0 Вызывается при получении любых сообщений, отличных от X, d и W. Этот метод может быть вызван из другого потока, это следует учитывать при имплементации.
virtual void	<b>on_feed_reset</b> (const std::string &channel_id, mfix_feed_type feed_type)=0 Вызывается, когда происходит «сброс» потока (получено X-сообщение с 269=J).
virtual void	<b>on_heartbeat</b> (const std::string &channel_id, mfix_feed_type feed_type)=0 Вызывается при получении сообщения Heartbeat

#### Примечание:

Объекты этого класса не следует класть в std::auto\_ptr или в другие умные указатели (за исключением специально разработанных, например Utils::RefCountPtr из состава библиотеки). Объекты должны создаваться оператором new.

#### 6.1.2.1.5. *MOEX\_mfix::MOEX\_mfix\_application\_params*

```
#include <B2BITS_MOEX_mfix_application.h>
```

#### Public Types

enum	<b>recovery_type</b> { <b>udp_recovery</b> , <b>tcp_recovery</b> }
------	--

#### Public Attributes

std::string	<b>templates_fn_</b> Путь к FAST-шаблону MFIX Market Data FAST.
std::string	<b>config_xml_</b> Путь к конфигурационному файлу MFIX Market Data Multicast.
size_t	<b>number_of_workers_</b> Количество потоков, используемых для декодирования FAST-сообщений. Значение по умолчанию 4.
size_t	<b>increment_queue_size_</b> Максимальное количество сообщений, которые сохраняются в режиме восстановления для каждого инструмента. Значение по умолчанию 50.
bool	<b>check_udp_sender_</b> Установить в true, если требуется проверять IP-адрес отправителя. Это значение по умолчанию.
std::string	<b>listen_interface_ip_</b> IP сетевого интерфейса, используемого для получения данных. Nullptr или пустая строка означает использование всех интерфейсов – это значение по умолчанию.
size_t	<b>incoming_udp_buffer_size_</b> Размер буфера входящих UDP-пакетов. Следует изменять при многочисленных потерях. Значение по умолчанию 75000000.
size_t	<b>application_message_queue_size_</b> Число сообщений в очереди на обработку. Значение по умолчанию 6553500.
bool	<b>log_incoming_FIX_messages_</b> Установить в true, если требуется запись входящих FIX-сообщений в лог-файл. Значение по умолчанию false.
bool	<b>log_incoming_udp_messages_</b> Установить в true, если требуется запись в файл входящих бинарных FAST-сообщений. Значение по умолчанию false.
std::size_t	<b>hole_pack_delay_</b> Число входящих сообщений с непоследовательными порядковыми номерами, после которого начинается процедура восстановления пропущенных сообщений. Значение по умолчанию 50.
recovery_type	<b>recovery_type_</b>

	<p>Тип восстановления.</p> <p>tcp_recovery использует восстановление по TCP-соединению (поле 34 используется для определения пропусков)</p> <p>udp_recovery использует восстановление из UDP-потока сэпшотов (поле 83 используется для определения пропусков).</p> <p>Значение по умолчанию udp_recovery.</p>
std::string	<p><b>user_login_</b></p> <p>Имя пользователя для установления TCP-соединения для восстановления</p>
std::string	<p><b>user_password_</b></p> <p>Пароль для установления TCP-соединения для восстановления</p>

#### 6.1.2.1.6. *MOEX\_mfix::security\_definition\_listener*

```
#include <B2BITS_MOEX_mfix_listeners.h>
```

#### Public Member Functions

virtual bool	<p><b>on_security_definition</b> (const security_description &amp;sec_desc, const security_id &amp;sec_id, const symbol &amp;symb, const std::string &amp;board, const Engine::FIXMessage &amp;d_msg, const std::string &amp;channel_id)=0</p> <p>Вызывается при получении описания финансового инструмента.</p> <p>Вернуть true, если следует продолжить слушать поток инструментов, false в противном случае.</p>
--------------	---

#### Примечание:

Объекты этого класса не следует класть в std::auto\_ptr или в другие умные указатели (за исключением специально разработанных, например Utils::RefCountPtr из состава библиотеки). Объекты должны создаваться оператором new.