

# MICEX Market Data Multicast FIX/FAST Platform

## *Руководство пользователя*

---

Московская межбанковская валютная биржа

Version 1.0

May 25, 2011

## Содержание

1.	О платформе MICEX Market Data Multicast FIX/FAST .....	4
1.1.	Потоковая передача данных .....	4
1.2.	Инкрементальные сообщения .....	4
1.3.	FIX формат .....	4
1.4.	Кодирование в FAST формат .....	4
1.5.	Получение данных с помощью Multicast .....	5
1.6.	Восстановление данных .....	5
2.	Работа с платформой MICEX Market Data FIX/FAST Multicast .....	6
2.1.	Подключение до старта Торговой системы .....	6
2.2.	Подключение после старта Торговой системы .....	6
2.3.	Обработка дублирующихся данных в потоках А и В .....	6
3.	Функциональность системы .....	8
3.1.	Архитектура системы .....	8
3.2.	FAST формат.....	9
3.2.1	Общее описание .....	9
3.2.2	Кодирование стоп-бита .....	10
3.2.3	Неявное тэгирование .....	10
3.2.4	Возможности кодирования полей .....	10
3.2.5	FAST-шаблон .....	11
3.2.6	Процесс декодирования .....	11
3.2.7	Пример FAST-шаблона .....	12
3.3.	Основные потоки UDP .....	14
3.3.1	Потоки Instrument Definitions.....	14
3.3.2	Потоки OrderBook, Market Statistics, Orders, и Trades.....	14
3.3.3	Потоки Recovery.....	15
3.3.4	Сессии для запроса пропущенных сообщений по TCP .....	15
3.4.	Восстановление пропущенных данных .....	15
3.4.1	Восстановление пропущенных данных из потоков Recovery (UDP) .....	16
3.4.2	Процесс восстановления данных.....	16
3.4.3	Восстановление пропущенных данных по TCP-соединению .....	18
4.	Публичный FIX интерфейс.....	19
4.1.	Группы полей .....	19
4.1.1	Заголовок .....	19
4.1.2	Группа Instrument .....	19
4.1.3	Группа Instrument Leg .....	21
4.1.4	Группа Instrument Extension .....	22

4.1.5	Группа Underlying Instrument .....	22
4.1.6	Группа Market Segment .....	22
4.2.	Сообщения сессионного уровня .....	22
4.2.1	Logon (A) .....	22
4.2.2	Logout (5).....	23
4.2.3	Heartbeat (0).....	23
4.1.	Сообщения бизнес уровня .....	23
4.1.1	Security Definition (d) .....	23
4.1.2	Security Status (f).....	24
4.1.3	Trading Session Status (h) .....	25
4.1.4	Market Data Request (V).....	26
4.1.5	Market Data - Snapshot/Full Refresh (W).....	26
4.1.6	Market Data - Incremental Refresh (X).....	30
5.	Настройка сетевого соединения.....	34
5.1.	Настройка VPN соединения с MICEX на базе Windows XP .....	34
5.2.	Настройка VPN соединения с MICEX на базе Windows 7 .....	41
5.3.	Настройка VPN соединения с MICEX на базе OpenSUSE .....	46
5.4.	Часто возникающие вопросы и методы их решения .....	48
6.	Сертифицированные средства работы.....	51
6.1.	Библиотека FIX Antenna <sup>™</sup> от EPAM – B2Bits®.....	51
6.1.1	Quick Start – примеры кода.....	51
6.1.2	Обзор API .....	54

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

Система MICEX Market Data Multicast FIX/FAST Platform представляет собой новый, высокоэффективный механизм для передачи рыночных данных о торгах на ММВБ. 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 MICEX 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. Потоковая 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.

### 1.2. 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.

### 1.3. FIX формат

Система MICEX 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 .

### 1.4. 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 .

## 1.5. 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 MICEX Market Data Multicast FIX/FAST Platform обеспечивает отправку только одного закодированного в FAST сообщения. FAST сообщение специально формируется таким образом, чтобы соответствовать размеру UDP пакета.

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

## 1.6. Восстановление данных

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

MICEX 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 передаче).

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

### 2.1. Подключение до старта Торговой системы

Фактически клиенты должны подключиться к системе MICEX 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 Поток с ftp-сервера. Конфигурационный файл в формате .xml описывает параметры подключения (адреса multicast, номера портов и т.д.). Скачать файл FAST-шаблона с ftp-сервера. Для получения дополнительной информации см. пункт 3.2.5 с описание 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 Поток с ftp-сервера. Конфигурационный файл в формате .xml описывает параметры подключения (адреса multicast, номера портов и т.д.). Скачать файл FAST-шаблона с ftp-сервера. Для получения дополнительной информации см. пункт 3.2.5 с описание 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 Поток 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. Обработать 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 получаются 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.

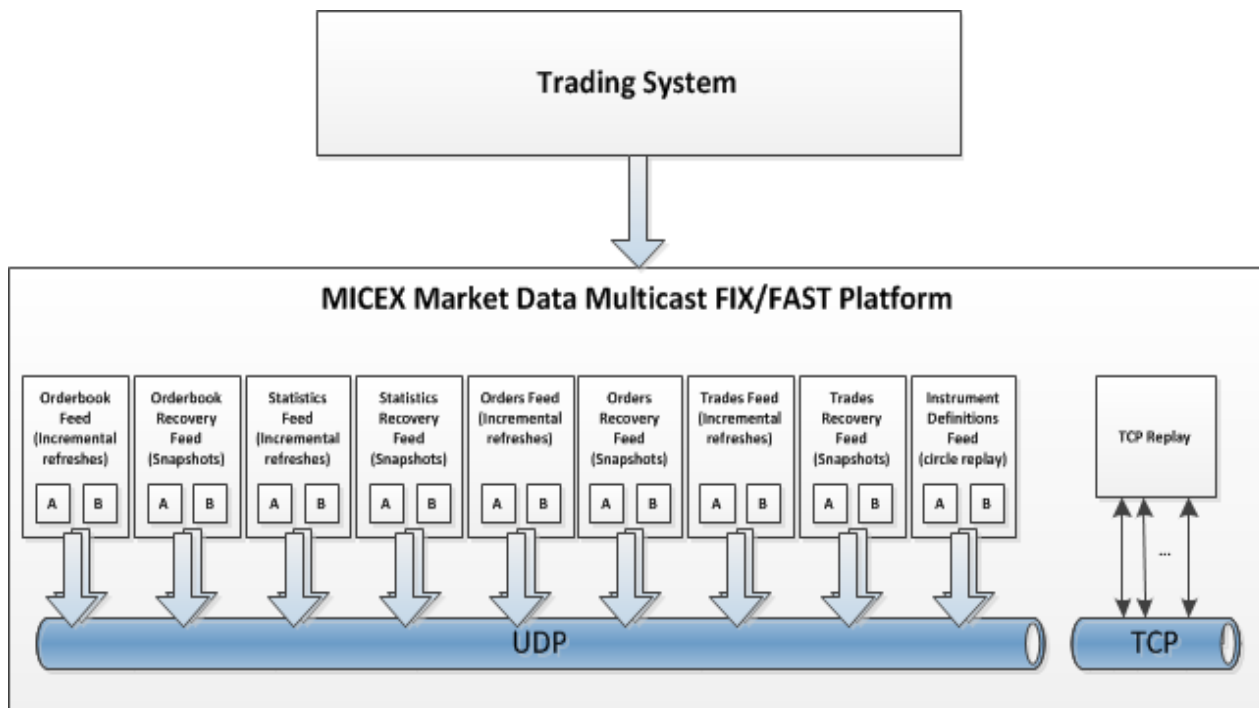
### 3. Функциональность системы

#### 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.
  - 1.3. Потоки распространения информации об изменении статуса финансовых инструментов и сообщений о состоянии соединения с Торговой системой.
2. Потоки восстановления
  - 2.1. Потоки распространения снимков рыночных данных.
  - 2.2. Сессии для запроса пропущенных данных.



MICEX Market Data Multicast FIX/FAST Platform обеспечивает вещание по следующим Потокам:

- Основные потоки:
  - OrderBook Feeds:
    - OrderBook Feed A;
    - OrderBook Feed B.
  - Statistics Feeds:
    - Statistics Feed A;
    - Statistics Feed B.
  - Orders Feeds:
    - Orders Feed A;
    - Orders Feed B.
  - Trades Feeds:



- Trades Feed A;
  - Trades Feed B.
- Потоки Recovery:
  - OrderBook Recovery Feeds:
    - OrderBook Recovery Feed A;
    - OrderBook Recovery Feed B.
  - Statistics Recovery Feeds:
    - Statistics Recovery Feed A;
    - Statistics Recovery Feed B.
  - Orders Recovery Feeds:
    - Orders Recovery Feed A;
    - Orders Recovery Feed B.
  - Trades Recovery Feeds:
    - Trades Recovery Feed A;
    - Trades Recovery Feed B.
- Instruments Definitions Feeds:
  - Instruments Definitions Feed A;
  - Instruments Definitions Feed B.

Помимо трансляции данных в UDP-потоках, MICEX 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
- Statistics Feed
- Orders Feed
- Trades Feed

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;
3. общее количество запрашиваемых в день сообщений ограничено.

## 3.2. FAST формат

### 3.2.1 Общее описание

Все сообщения, отправляемые MICEX 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 информации, в частности, для распространения большого объема данных с минимальной задержкой.

Особенностью распространения данных в информационных потоках от MICEX 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.

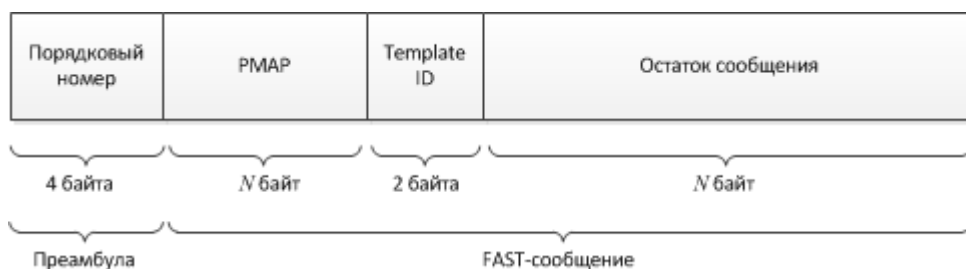


Figure 1

### 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):

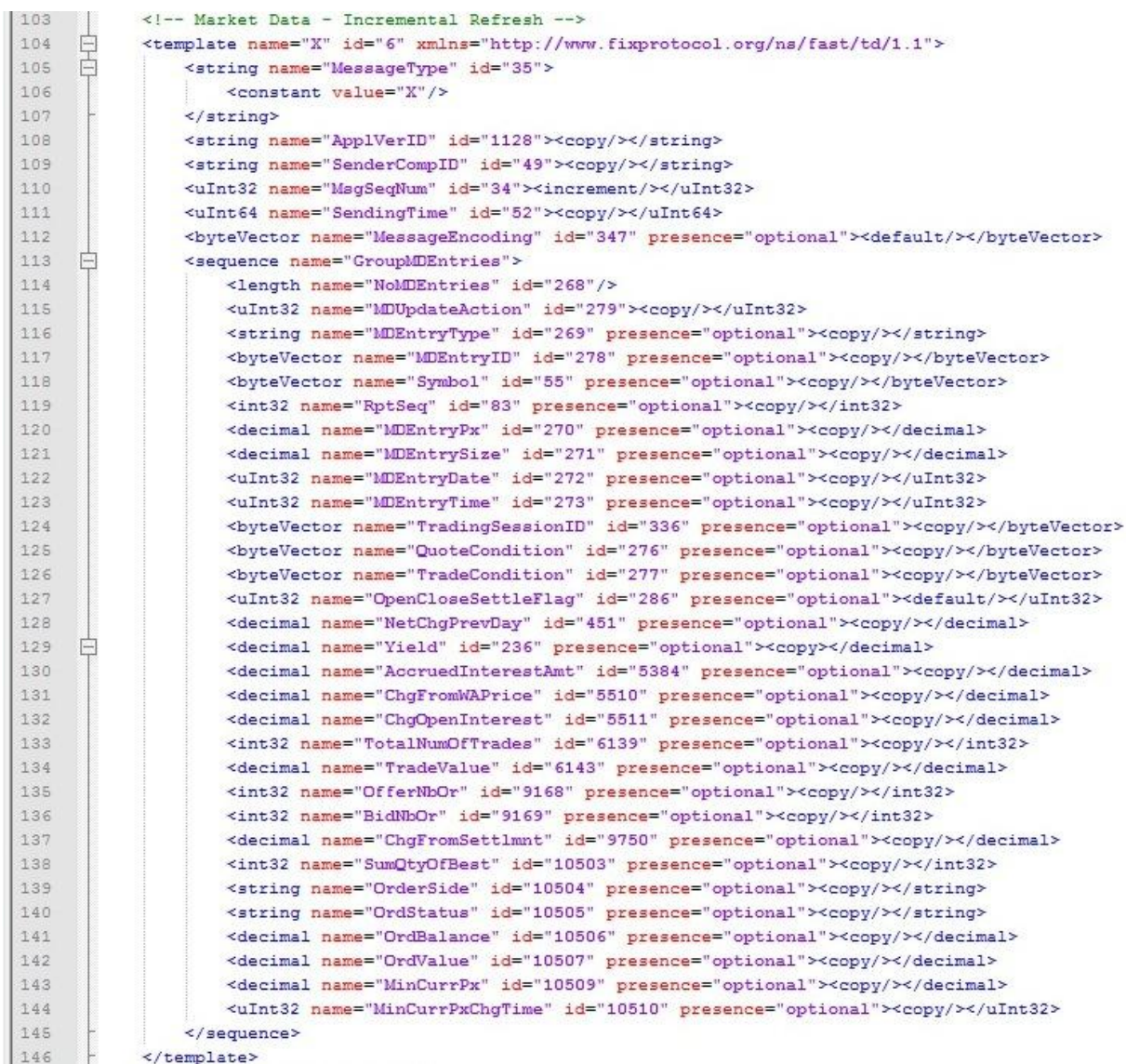


Figure 2

### 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.

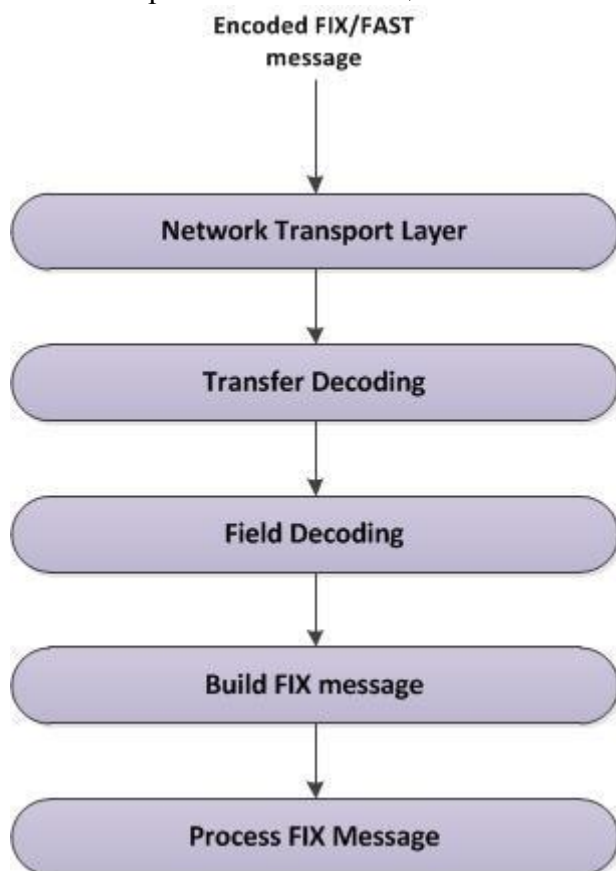


Figure 3

### 3.2.7 Пример FAST-шаблона

Table 1

Line #	Template Syntax	Use and Description
1	<template name="X" id="6" xmlns="http://www.fixprotocol.org/ns/fast/td/1.1">	Идентификатор и название шаблона.
2	<string name="MessageType" id="35"> <constant value="X" /> </string>	MessageType определен как тип данных string, идентификатор = 35.
3	<string name="ApplVerID" id="1128"><copy/></string>	ApplVerID определен как тип данных string, идентификатор = 1128
4	<string name="SenderCompID" id="49"><copy/></string>	SenderCompID определен как тип данных string, идентификатор = 49
5	<uInt32 name="MsgSeqNum" id="34"><increment/></uInt32>	MsgSeqNum определен как тип данных unsigned integer, идентификатор = 34
6	<uInt64 name="SendingTime" id="52"><copy/></uInt64>	SendingTime определен как тип данных unsigned integer, идентификатор = 52
7	<byteVector name="MessageEncoding" id="347" presence="optional"><default/></byteVector>	MessageEncoding определен как тип данных byte vector, идентификатор = 347
8	<sequence name="GroupMDEntries"> <length name="NoMDEntries"	Определение репитинг группы MDEntries. 'NoMDEntries' показывает количество



	id="268"/>	повторяющихся элементов.
9	<uint32 name="MDUpdateAction" id="279" presence="optional"><copy/></uint32>	MDUpdateAction определен как тип данных unsigned integer, идентификатор = 279
10	<string name="MDEntryType" id="269" presence="optional"><copy/></string>	MDEntryType определен как тип данных string, идентификатор = 269
11	<byteVector name="MDEntryID" id="278" presence="optional"><copy/></byteVector>	MDEntryID определен как тип данных byte vector, идентификатор = 278
12	<byteVector name="Symbol" id="55" presence="optional"><copy/></byteVector>	Symbol определен как тип данных byte vector, идентификатор = 55.
13	<int32 name="RptSeq" id="83" presence="optional"><copy/></int32>	RptSeq определен как тип данных signed integer, идентификатор = 83
14	<decimal name="MDEntryPx" id="270" presence="optional"><copy/></decimal>	MDEntryPx определен как тип данных decimal, идентификатор = 270
15	<decimal name="MDEntrySize" id="271" presence="optional"><copy/></decimal>	MDEntrySize определен как тип данных decimal, идентификатор = 271
16	<uint32 name="MDEntryDate" id="272" presence="optional"><copy/></uint32>	MDEntryDate определен как тип данных unsigned integer, идентификатор = 272.
17	<uint32 name="MDEntryTime" id="273" presence="optional"><copy/></uint32>	MDEntryTime определен как тип данных unsigned integer, идентификатор = 273
18	<byteVector name="TradingSessionID" id="336" presence="optional"><copy/></byteVector>	TradingSessionID определен как тип данных byte vector, идентификатор = 336
19	<byteVector name="QuoteCondition" id="276" presence="optional"><copy/></byteVector>	QuoteCondition определен как тип данных byte vector, идентификатор = 276
20	<byteVector name="TradeCondition" id="277" presence="optional"><copy/></byteVector>	TradeCondition определен как тип данных byte vector, идентификатор = 277
21	<byteVector name="OpenCloseSettleFlag" id="286" presence="optional"><copy/></byteVector>	OpenCloseSettleFlag определен как тип данных byte vector, идентификатор = 286
22	<decimal name="NetChgPrevDay" id="451" presence="optional"><copy/></decimal>	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	<decimal name="ChgOpenInterest" id="5511" presence="optional"><copy/></decimal>	ChgOpenInterest определен как тип данных decimal, идентификатор = 5511
26	<int32 name="TotalNumOfTrades" id="6139" presence="optional"><copy/></int32>	TotalNumOfTrades определен как тип данных signed integer, идентификатор = 6139
27	<decimal name="TradeValue" id="6143" presence="optional"><copy/></decimal>	TradeValue определен как тип данных decimal, идентификатор = 6143
28	<decimal name="Yield" id="236" presence="optional"><copy/></decimal>	Yield определен как тип данных decimal, идентификатор = 236
29	<int32 name="OfferNbOr" id="9168" presence="optional"><copy/></int32>	OfferNbOr определен как тип данных signed integer, идентификатор = 9168
30	<int32 name="BidNbOr" id="9169" presence="optional"><copy/></int32>	BidNbOr определен как тип данных signed integer with identifier = 9169
31	<decimal name="ChgFromSettlemnt" id="9750" presence="optional"><copy/></decimal>	ChgFromSettlemnt определен как тип данных decimal, идентификатор = 9750
32	<int32 name="SumQtyOfBest" id="10503" presence="optional"><copy/></int32>	SumQtyOfBest определен как тип данных signed integer, идентификатор = 10503.
33	<string name="OrderSide" id="10504" presence="optional"><copy/></string>	OrderSide определен как тип данных string, идентификатор = 10504.
34	<string name="OrdStatus" id="10505" presence="optional"><copy/></string>	OrdStatus определен как тип данных string, идентификатор = 10505
37	<decimal name="MinCurrPx" id="10509" presence="optional"><copy/></decimal>	MinCurrPx определен как тип данных decimal, идентификатор = 10509.
38	<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 Потоки 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=MICEX|55=VRSB  
P|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=FOND|SMAL|10=  
000|
```

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

Следующие рыночные данные распространяются в отдельных потоках:

- Потоки OrderBook (А и В) – 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).
- Потоки Statistics (А и В) – 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), '6' (Расчетные цены), '7' (Максимальная цена сделки), '8' (Минимальная цена сделки), '9' (Средневзвешенные цены), 'B' (Объемы сделок), 'C' (Открытые позиции), 'N' (Максимальная цена спроса в течение сессии), 'O' (Минимальная цена предложения в течение сессии), 'i' (Спрос сессии), 'j' (Предложение сессии), 'h' (Цена предторгового периода), 'k' (Цена послеторгового периода), 'l' (Рыночная цена 2), 'm' (Рыночная цена), 'n' (Цена последней внесистемной сделки), 'o' (Официальная цена открытия), 'p' (Официальная текущая цена), 't'

(Официальная цена закрытия), 'v' (Совокупный спрос), 'w' (Совокупное предложение), 't' (Объем/Количество внесистемных сделок), 'u' (Дюрация).

- Потоки Orders (A и B) – передают обновления из таблицы ORDERS. Виды обновлений: Add, Change, и Delete. Элементы рыночных данных: 'O' (Котировки на покупку), 'I' (Котировки на продажу).
- 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).

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-потоки инкрементальных обновлений отправляется сообщение Trading Session Status (h).

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

### 3.3.3 Потоки Recovery

В потоках Recovery (OrderBook, Statistics, Orders, Trades) в режиме multicast по протоколу UDP с фиксированной периодичностью распространяются текущие снимки 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 потоки постоянно. К ним необходимо подключаться только в случае необходимости восстановить пропущенную в основных потоках информацию. После восстановления клиенту следует прекратить слушать данные потоки.

### 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-потоков.

В запросе клиент указывает диапазон порядковых номеров для пересылки, а так же идентификатор UDP-потока, из которого запрашивается информация. Максимальное количество сообщений, которое может запросить клиент, ограничено. 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 MICEX 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

Как уже говорилось, 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-пакетов.

Может случиться так, что будут утеряны пакеты из обоих потоков, 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.

MICEX 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 MICEX Market Data Multicast.

Для запроса пропущенных данных клиент должен выполнить следующие действия:

1. Установить TCP-соединение с сервером MICEX 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.

## 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 - сообщение с такими данными будет отклонено.*

### 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 фирмы – отправителя сообщения. Всегда содержит незашифрованные данные. Если сообщение отправляется в ММВБ, это поле должно содержать идентификатор пользователя (USERID), присвоенное брокеру на ММВБ.
34	MsgSeqNum	O	SeqNum		Порядковый номер сообщения. Может быть встроен в блок зашифрованных данных.
52	SendingTime	O	UTCTimestamp		Время передачи сообщения (выражено в формате UTC). Может быть встроено в блок зашифрованных данных.
347	MessageEncoding	H	String(11)	'UTF-8' (Unicode)	Тип кодирования (не ASCII символы). Обязательное, если используется, хотя бы одно из «закодированных» полей.

#### 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 ММВБ (SECCODE).
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); '11' (MUNICIPAL); '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) 'PS' (Акции привилегированные) 'EUSOV' (Еврооблигация) 'BN' (Ценные бумаги, выпущены банком) 'MF' (Паи инвестиционных фондов) 'MLEG' (Multi-leg инструмент) 'MUNI' (Муниципальные облигации)	Тип ценной бумаги.
200	MaturityMonthYear	H	month-year		Месяц и год погашения/срока действия ценной бумаги. Формат: YYYYMM.
541	MaturityDate	H	LocalMktDate		Дата погашения/срок действия ценной бумаги.
224	CouponPaymentDate	H	LocalMktDate		Дата выплаты накопленного купонного дохода.
228	Factor	H	float		Для инструментов с фиксированной доходностью: Коэффициент амортизации для ценных бумаг, обеспеченных активами или пулом ипотек. Для казначейских обязательств (Treasury Inflation Protected Securities) это показатель инфляции. $Qty * Factor * Price = Gross\ Trade\ Amount$ . Для производных ценных бумаг: показатель стоимости контракта, с помощью которого цена должна быть отрегулирована для определения номинальной стоимости одного фьючерсного или опционного контракта.

					$(Qty * Price) * Factor =$ Номинальная стоимость.
202	StrikePrice	H	Price		Цена исполнения опциона.
223	CouponRate	H	Percentage		Процентная ставка, по которой рассчитывается размер купонной выплаты по облигации (купонная ставка).
107	SecurityDesc	H	String		Описание ценной бумаги. На ММВБ это поле определяет наименование финансового инструмента на английском языке.
350	EncodedSecurityDescLen	H	Length		Длина поля EncodedSecurityDesc (351) в байтах.
351	EncodedSecurityDesc	H	data		Зашифрованное (не ASCII символы) значение поля SecurityDesc (107). Формат шифрования указан в поле MessageEncoding (347) в заголовке сообщения.
864	NoEvents	H	NumInGroup		Количество элементов в группе EventType.
=> 865	EventType	H	int	'7' (Последний торговый день) '100' (Первый торговый день)	Тип события. Поле обязательное, если NoEvents (864) > 0.
=> 866	EventDate	H	LocalMktDate		Дата события.
5217	StateSecurityID	H	String		Номер государственной регистрации.
5382	EncodedShortSecurityDescLen	H	Length		Длина поля EncodedShortSecurityDesc (5383) в байтах.
5383	EncodedShortSecurityDesc	H	data		Закодированное (не ASCII символы) краткое наименование ценной бумаги. Формат шифрования указан в поле MessageEncoding (347) в заголовке сообщения.
5556	BaseSwapPx	H	Price		Базовый курс при торговле СВОП инструментам.
5558	BuyBackPx	H	Price		Базовая цена для расчета доходности
5559	BuyBackDate	H	LocalMktDate		Дата, к которой рассчитывается доходность
5677	Repo2Px	H	Price		Цена второй части РЕПО.

#### 4.1.3 Группа Instrument Leg

Table 4

Tag	Поле	Наличие	Тип	Допустимые значения	Примечание
600	LegSymbol	O	String		Код/аббревиатура ценной бумаги для Multileg инструмента.
607	LegProduct	H	int	'4' (CURRENCY)	Тип продукта, с которым связана ценная бумага Multileg инструмента.
608	LegCFICode	H	String		Тип ценной бумаги по стандарту ISO 10962 для Multileg инструмента. CFI код (Classification of Financial Instruments).
609	LegSecurityType	H	String	'FOR' (Валютный контракт)	Тип ценной бумаги Multileg инструмента.
556	LegCurrency	H	Currency		Код валюты, в которой выражена цена элемента рыночных данных.
587	LegSettlType	H	char	'T' (Cash);	Период расчетов по Multileg инструменту.

	<i>e</i>			'2' (Next day).	
--	----------	--	--	-----------------	--

#### 4.1.4 Группа Instrument Extension

Table 5

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 в ценах финансового инструмента)	Тип атрибута ценной бумаги. Поле обязательное, если NoInstrAttrib (870) > 0.
=> 872	InstrAttribValue	H	String		Значение атрибута ценной бумаги (если применимо).

#### 4.1.5 Группа Underlying Instrument

Table 6

Tag	Поле	Наличие	Тип	Допустимые значения	Примечание
311	Underlying Symbol	O	String		Идентификатор финансового инструмента, являющегося базовым активом.

#### 4.1.6 Группа Market Segment

Table 7

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. В качестве идентификатора торговой сессии используется рынок + режим торгов (SuperMarket/SECBOARD).

## 4.2. Сообщения сессионного уровня

#### 4.2.1 Logon (A)

Logon сообщение от пользователя к MICEX:

Table 8

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 MICEX к пользователю:

Table 9

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 10

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

#### 4.2.3 Heartbeat (0)

Table 11

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

### 4.1. Сообщения бизнес уровня

#### 4.1.1 Security Definition (d)

Table 12

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.
711	NoUnderly ings	H	NumInGro up		Количество элементов в группе Underlying Instrument.
=> component block <Underlying Instrument>		H			Информация по финансовому инструменту, который является базовым активом. Поле обязательное, если NoUnderlyings (711) > 0.
555	NoLegs	H	NumInGro up		Количество элементов в группе Instrument Leg
=> component block <Instrument Leg>		H			Информация о частях Multileg инструмента. Поле обязательное, если NoLegs (555) > 0.
15	Currency	H	Currency		Код валюты, в которой выражен номинал ценной бумаги.
component block <Market Segment>		H			Дополнительная информация по финансовому инструменту.
120	SettlCurre ncy	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.
538 5	MarketCod e	H	String		Код рынка, на котором торгуется финансовый инструмент.
969	MinPriceI ncrement	H	float		Минимальный шаг изменения цены.
538 7	MktShareL imit	H	Percentage		Лимит на долю рынка.
538 8	MktShareT hreshold	H	Qty		Минимальный объем, при котором начинает контролироваться лимит на долю рынка.
538 9	MaxOrder sVolume	H	Qty		Лимит объема заявок.
547 0	PriceMvm Limit	H	Price		Лимит изменения цены.
550 8	FaceValue	H	Amt		Номинальная стоимость одной ценной бумаги, в валюте инструмента.
759 5	NoSharesI ssued	H	Qty		Объем выпуска.
919 9	HighLimit	H	Price		Максимально возможная цена.
920 0	LowLimit	H	Price		Минимально возможная цена.
105 08	NumOfDay sToMaturit y	H	int		Количество дней до погашения базового актива.

#### 4.1.2 Security Status (f)

Table 13

Tag	Поле	Наличие	Тип	Допустимые значения	Примечание
-----	------	---------	-----	---------------------	------------



<Standard Message Header>		O			Тип сообщения = 'f'
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 ММВБ (SECCODE).
336	TradingSessionID	H	String		Идентификатор торговой сессии. В качестве идентификатора торговой сессии используется рынок + режим торгов (SuperMarket/SECBOARD).
326	SecurityTradingStatus	H	int	'2' (Торговля приостановлена (перерыв)) '17' (Начало торговой сессии (период открытия)) '18' (Конец торговой сессии (торги закрыты)) '20' (Недоступно для торгов)	Состояние торгов по инструменту.
5509	AuctionIndicator	H	Boolean	'Y' (Да); 'N' (Нет).	Индикатор информирующий, что по данному инструменту проводится аукцион.

#### 4.1.3 Trading Session Status (h)

Table 14

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 ММВБ установлено); '101' (Соединение с ММВБ потеряно); '102' (Соединение восстановлено, торговая система не перезапускалась); '103' (Соединение восстановлено, торговая система перезапускалась).	Статус торговой сессии.
58	Text	H	String		Текстовая строка в свободном формате. Может

					использоваться для комментариев, дополнительной информации касающейся подключения к конкретному рынку ММВБ.
--	--	--	--	--	-------------------------------------------------------------------------------------------------------------

#### 4.1.4 Market Data Request (V)

Table 15

Tag	Поле	Наличие	Тип	Допустимые значения	Примечание
<Standard Message Header>		O			Тип сообщения = 'V'
1180	ApplID	H	String		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).

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

Table 16

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 ММВБ установлено); '101' (Соединение с ММВБ потеряно); '102' (Соединение восстановлено, торговая система не перезапускалась); '103' (Соединение восстановлено, торговая система перезапускалась)	Статус соединения с Торговой системой.

				).	
55	Symbol	O	String		Код/аббревиатура ценной бумаги. В его качестве используется внутренний идентификатор финансового инструмента на ММББ (SECCODE).
893	LastFragment	H	Boolean	'N' (Сообщение не последнее, снэпшот еще не сформирован); 'Y' (Последнее сообщение, снэпшот сформирован).	Индикатор, показывающий, является ли сообщение последним в серии, которая формирует снэпшот по данному инструменту.
1682	MDSecurityTradingStatus	H	int	'2' (Торговля приостановлена (перерыв)); '17' (Начало торговой сессии (период открытия)); '18' (Конец торговой сессии (торги закрыты)); '20' (Недоступно для торгов).	Состояние торгов по инструменту.
5509	AuctionIndicator	H	Boolean	'Y' (Да) 'N' (Нет)	Индикатор информирующий, что по данному инструменту проводится аукцион.
451	NetChgPrevDay	H	PriceOffset		Изменение цены последней сделки по отношению к цене последней сделки предыдущего торгового дня.
268	NoMDEntries	O	NumInGroup		Количество элементов в группе MDEntryTypes.
=> 269	MDEntryType	O	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); '6' (Расчетные цены); '7' (Максимальная цена сделки); '8' (Минимальная цена сделки); '9' (Средневзвешенные цены);	Тип рыночных данных.

				<p>'B' (Объемы сделок);</p> <p>'C' (Открытые позиции);</p> <p>'J' (Пустой снэпиот)</p> <p>'N' (Максимальная цена спроса в течение сессии);</p> <p>'O' (Минимальная цена предложения в течение сессии);</p> <p>'i' (Спрос сессии);</p> <p>'j' (Предложение сессии);</p> <p>'h' (Цена предторгового периода);</p> <p>'k' (Цена послеторгового периода);</p> <p>'l' (Рыночная цена 2);</p> <p>'m' (Рыночная цена);</p> <p>'n' (Цена последней внесистемной сделки);</p> <p>'o' (Официальная цена открытия);</p> <p>'p' (Официальная текущая цена);</p> <p>'r' (Официальная цена закрытия);</p> <p>'v' (Совокупный спрос);</p> <p>'w' (Совокупное предложение);</p> <p>'t' (Объем/Количество внесистемных сделок);</p> <p>'u' (Дюрация);</p> <p>'z' (Все сделки/список обозначенных сделок)</p>	
=> 278	MDEntryID	H	String		Идентификатор элемента MDEntry. Например, используется в качестве идентификатора номера сделки (TRADENO).

=> 270	MDEntryPx	Y	Price		Цена элемента рыночных данных (соответствует заданному типу рыночных данных и относиться к текущему элементу MDEntry). Поле обязательное, если MDEntryType (269) не является одним из ('A', 'B', 'C', 'J').
=> 271	MDEntrySize	Y	Qty		Количество или объем элемента рыночных данных (соответствует заданному типу рыночных данных и относиться к текущему элементу MDEntry). Поле обязательное, если MDEntryType (269) является одним из ('0', '1', '2', 'B', 'C').
=> 272	MDEntryDate	Y	UTCDate Only		Дата, которая относится к данному элементу рыночных данных.
=> 273	MDEntryTime	H	UTCTime Only		Время, которое относится к данному элементу рыночных данных.
=> 336	TradingSessionID	H	String		Идентификатор торговой сессии, к которой относятся рыночные данные. В качестве идентификатора торговой сессии используется рынок + режим торгов (SuperMarket/SECBOARD).
=> 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	OpenCloseSettlFlag	H	MultipleValueString	'4' (Данные предыдущего торгового дня)	Флаг, который идентифицирует тип элемента рыночных данных.
=> 236	Yield	H	Percentage		Доходность, рассчитанная по цене MDEntryPx (270).
=> 5384	AccruedInterestAmt	H	Amt		Объем накопленного купонного дохода.
=> 5510	ChgFromWAPrice	H	PriceOffset		Изменение цены сделки по сравнению со средневзвешенной ценой предыдущего торгового дня.
=> 551	ChgOpenInterest	H	Qty		Изменение количества открытых позиций к предыдущему торговому дню.

1					
=> 613 9	TotalNum OfTrades	H	int		Общее количество сделок, заключенных на протяжении торгового дня.
=> 614 3	TradeValu e	H	Amt		Объем совершенных сделок.
=> 916 8	OfferNbOr	H	int		Количество заявок на продажу в очереди.
=> 916 9	BidNbOr	H	int		Количество заявок на покупку в очереди.
=> 975 0	ChgFromS ettlmnt	H	PriceOffset		Изменение цены последней сделки к расчетной цене предыдущего дня.
=> 105 03	SumQtyOf Best	H	int		Суммарное кол-во лучших котировок.
=> 105 04	OrderSide	H	char		Направленность заявки.
=> 105 05	OrdStatus	H	char	'O' (Активная); 'M' (Исполнена); 'W' (Снята); 'F' (Отклонена партнером); 'R' (Отклонена торговой системой); 'C' (Снята торговой системой); 'T' (Время активации не наступило).	Текущий статус заявки.
=> 105 09	MinCurrP x	H	Price		Минимальная из двух цен: Официальной текущей цены и цены последней сделки, вошедшей в расчёт официальной текущей цены.
=> 105 10	MinCurrP xChgTime	H	UTCTime Only		Время изменения минимальной текущей цены.

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

Table 17

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	Y	char	'0' (Котировки на покупку); '1' (Котировки на продажу);	Тип рыночных данных.

				<p>'2' (Информация по последней сделке);</p> <p>'3' (Список индексов);</p> <p>'4' (Цена открытия/цена первой сделки);</p> <p>'5' (Цена закрытия/цена последней сделки предыдущего дня);</p> <p>'6' (Расчетные цены);</p> <p>'7' (Максимальная цена сделки);</p> <p>'8' (Минимальная цена сделки);</p> <p>'9' (Средневзвешенные цены);</p> <p>'В' (Объемы сделок);</p> <p>'С' (Открытые позиции);</p> <p>'N' (Максимальная цена спроса в течение сессии);</p> <p>'О' (Минимальная цена предложения в течение сессии);</p> <p>'i' (Спрос сессии);</p> <p>'j' (Предложение сессии);</p> <p>'h' (Цена предторгового периода);</p> <p>'к' (Цена постторгового периода);</p> <p>'I' (Рыночная цена 2);</p> <p>'т' (Рыночная цена);</p> <p>'п' (Цена последней внесистемной сделки);</p> <p>'о' (Официальная цена открытия);</p>	
--	--	--	--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

				<p>'p' (Официальная текущая цена);</p> <p>'r' (Официальная цена закрытия);</p> <p>'v' (Совокупный спрос);</p> <p>'w' (Совокупное предложение);</p> <p>'i' (Объем/Количество внесистемных сделок);</p> <p>'u' (Дюрация);</p> <p>'z' (Все сделки/список обозначенных сделок)</p>	
=> 278	MDEntryID	H	String		Идентификатор элемента MDEntry. Например, используется в качестве идентификатора номера сделки (TRADENO).
=> 55	Symbol	O	String		Код/аббревиатура ценной бумаги. В его качестве используется внутренний идентификатор финансового инструмента на ММББ (SECCODE).
=> 83	RptSeq	O	int		Номер обновления инструмента.
=> 270	MDEntryPx	Y	Price		Цена элемента рыночных данных (цена соответствует заданному типу рыночных данных, и относится к текущему элементу MDEntry). Поле обязательное, если MDEntryType (269) не является одним из ('A', 'B', 'C', 'J').
=> 271	MDEntrySize	Y	Qty		Количество или объем элемента рыночных данных (соответствует заданному типу рыночных данных и относится к текущему элементу MDEntry). Поле обязательное, если MDEntryType (269) является одним из ('0', '1', '2', 'B', 'C').
=> 272	MDEntryDate	H	UTCDate Only		Дата, которая относится к данному элементу рыночных данных.
=> 273	MDEntryTime	H	UTCTime Only		Время, которое относится к данному элементу рыночных данных.
=> 336	TradingSessionID	H	String		Идентификатор торговой сессии, к которой относятся рыночные данные. В качестве идентификатора торговой сессии используется рынок + режим торгов (SuperMarket/SECBOARD).
=> 276	QuoteCondition	H	MultipleValueString	'C' (Наилучшая цена)	Список условий, которые характеризуют котировку, условия между собой разделены пробелами.
=> 277	TradeCondition	H	MultipleValueString	<p>'C' (Расчеты по сделке осуществляются в день заключения сделки)</p> <p>'J' (Расчеты по сделке осуществляются на следующий</p>	Условия, которые характеризуют сделку или рыночные данные, которые рассчитываются на базе сделки, условия между собой разделены пробелами.



				день после заключения сделки) 'R' (Цена открытия) 'AJ' (Официальная цена закрытия); '98' (Минимальное значение); '99' (Максимальное значение)	
=> 286	OpenCloseSettleFlag	H	MultipleValueString	'4' (Данные предыдущего торгового дня)	Флаг, который идентифицирует тип элемента рыночных данных.
=> 451	NetChangePreviousDay	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).
=> 5384	AccruedInterestAmt	H	Amt		Объем накопленного купонного дохода.
=> 5510	ChangeFromWAPPrice	H	PriceOffset		Изменение цены сделки по сравнению со средневзвешенной ценой предыдущего торгового дня.
=> 5511	ChangeOpenInterest	H	Qty		Изменение количества открытых позиций к предыдущему торговому дню.
=> 6139	TotalNumberOfTrades	H	int		Общее количество сделок, заключенных на протяжении торгового дня.
=> 6143	TradeVolume	H	Amt		Объем совершенных сделок.
=> 9168	OfferNumberOrder	H	int		Количество заявок на продажу в очереди.
=> 9169	BidNumberOrder	H	int		Количество заявок на покупку в очереди.
=> 9750	ChangeFromSettlement	H	PriceOffset		Изменение цены последней сделки к расчетной цене предыдущего дня.
=> 10503	SumQtyOfBest	H	int		Суммарное кол-во лучших котировок.
=> 10504	OrderSide	H	char		Направленность заявки.
=> 10505	OrderStatus	H	char	'O' (Активная); 'M' (Исполнена); 'W' (Снята); 'F' (Отклонена партнером); 'R' (Отклонена торговой системой); 'C' (Снята торговой системой); 'T' (Время активации не наступило).	Текущий статус заявки.

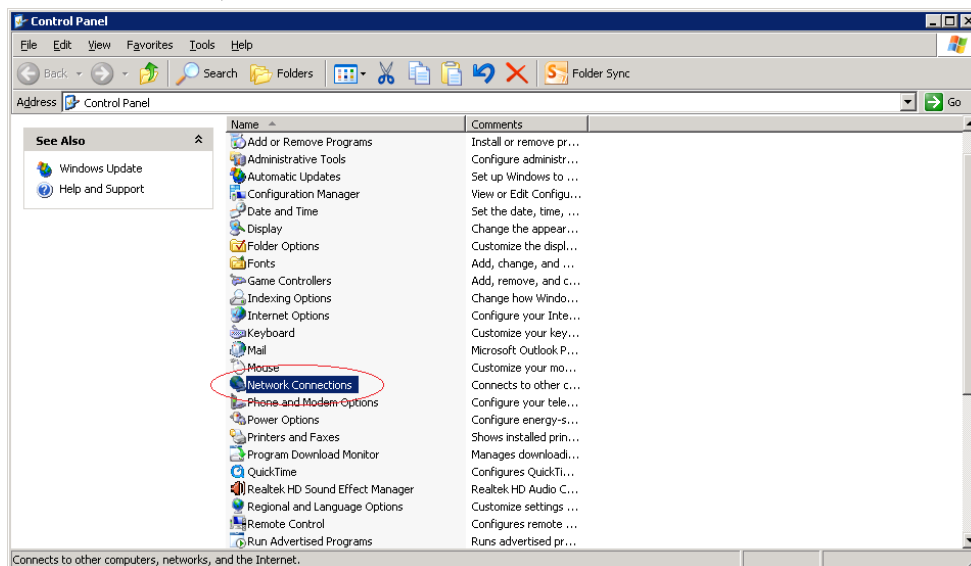
=>10 509	MinCur rPx	H	Price		Минимальная из двух цен: Официальной текущей цены и цены последней сделки, вошедшей в расчёт официальной текущей цены.
=>10 510	MinCur rPxChg Time	H	UTCTime Only		Время изменения минимальной текущей цены.

## 5. Настройка сетевого соединения

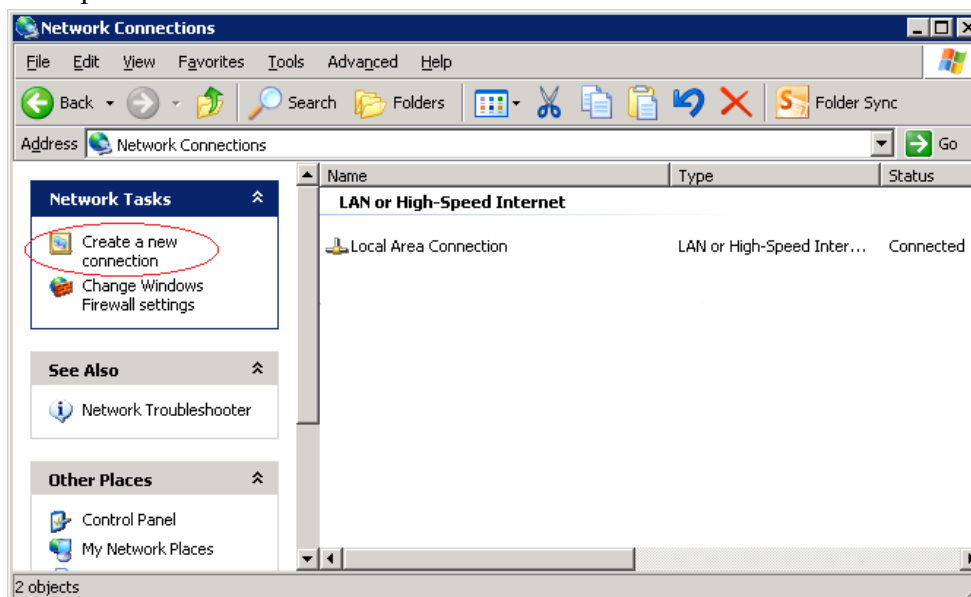
### 5.1. Настройка VPN соединения с MICEX на базе Windows XP

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*:



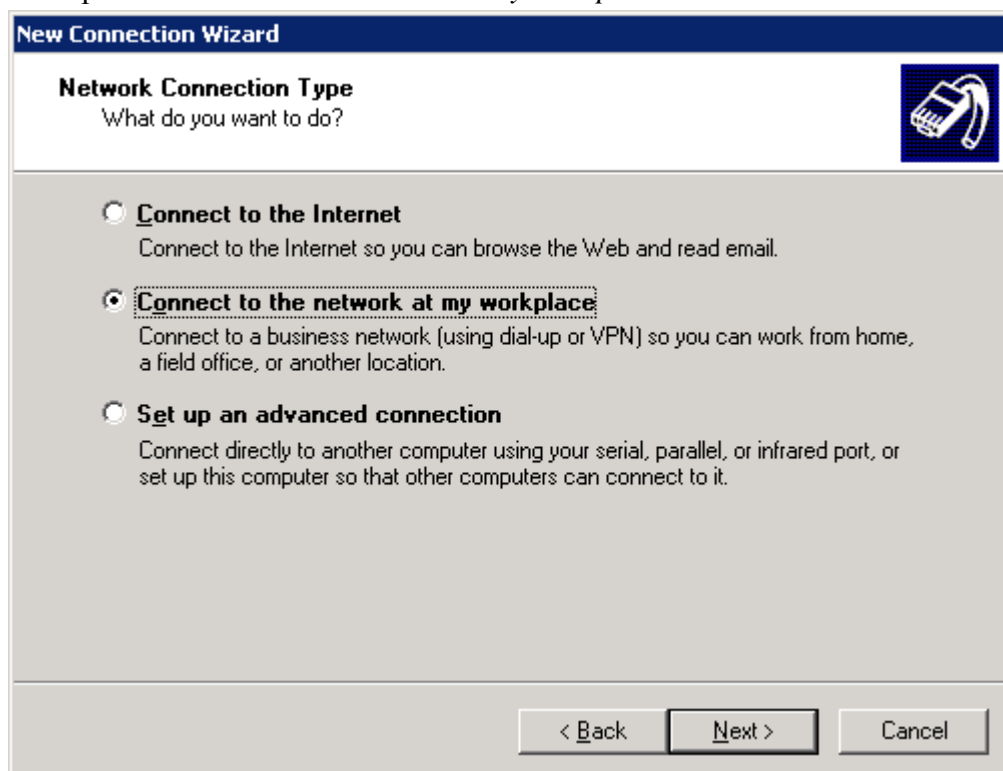
4. Выберите *Create a new connection* in *Network Tasks*:



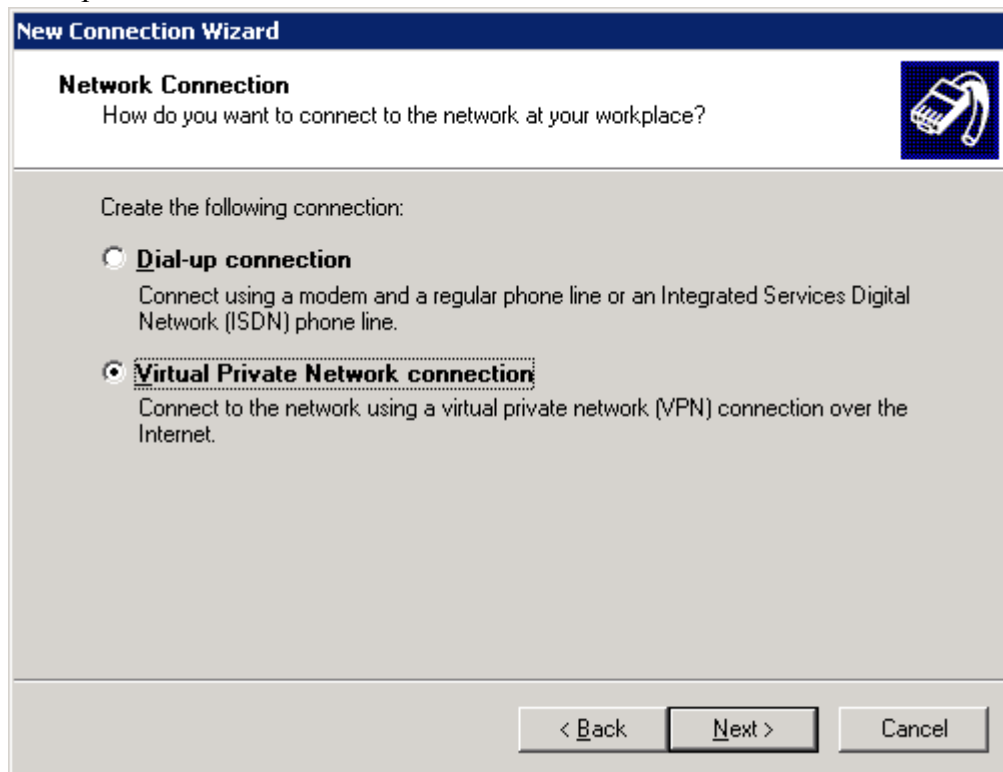
5. В открывшемся помощнике *Network Connection Wizard*, нажмите *Next*:



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



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



The screenshot shows the 'New Connection Wizard' dialog box. The title bar is blue with the text 'New Connection Wizard'. Below the title bar, the section is titled 'Network Connection' with a question 'How do you want to connect to the network at your workplace?'. To the right of the text is a small icon of a network card. The main area contains the instruction 'Create the following connection:' followed by two radio button options. The first option is 'Dial-up connection' with a description: 'Connect using a modem and a regular phone line or an Integrated Services Digital Network (ISDN) phone line.' The second option is 'Virtual Private Network connection' (which is selected) with a description: 'Connect to the network using a virtual private network (VPN) connection over the Internet.' At the bottom right, there are three buttons: '< Back', 'Next >', and 'Cancel'.

**New Connection Wizard**

**Network Connection**  
How do you want to connect to the network at your workplace?

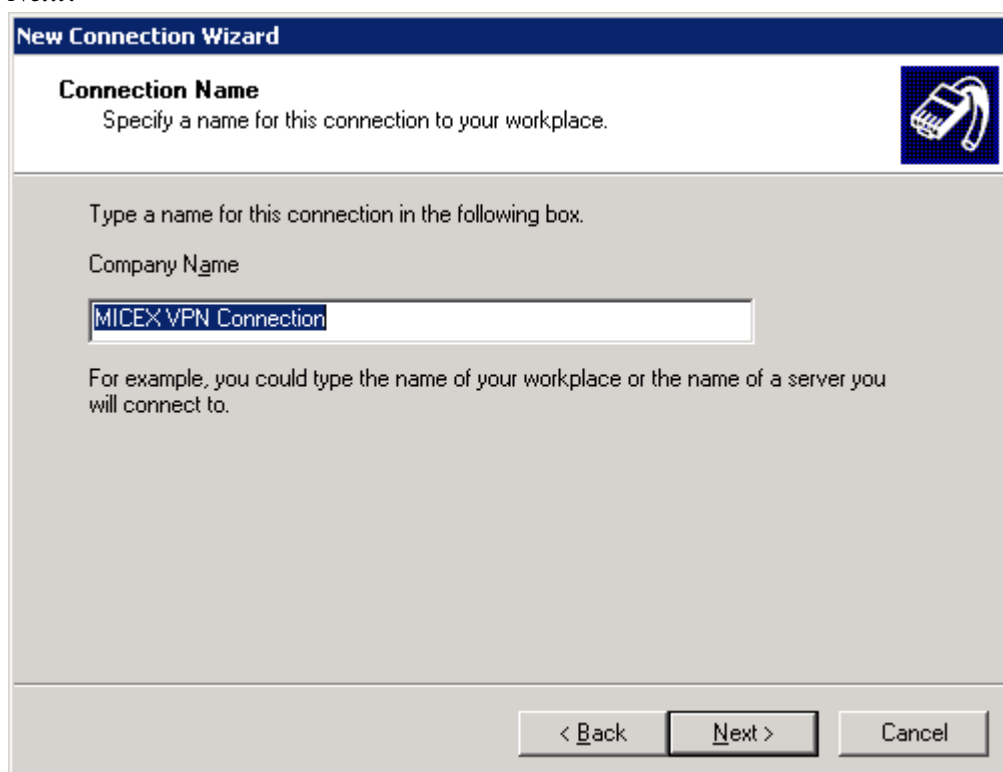
Create the following connection:

☐ **Dial-up connection**  
Connect using a modem and a regular phone line or an Integrated Services Digital Network (ISDN) phone line.

☒ **Virtual Private Network connection**  
Connect to the network using a virtual private network (VPN) connection over the Internet.

< Back   Next >   Cancel

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



The screenshot shows the 'New Connection Wizard' dialog box. The title bar is blue with the text 'New Connection Wizard'. Below the title bar, the section is titled 'Connection Name' with the instruction 'Specify a name for this connection to your workplace.' To the right of the text is a small icon of a network card. The main area contains the instruction 'Type a name for this connection in the following box.' followed by a text input field labeled 'Company Name'. The input field contains the text 'MICEX VPN Connection'. Below the input field, there is a note: 'For example, you could type the name of your workplace or the name of a server you will connect to.' At the bottom right, there are three buttons: '< Back', 'Next >', and 'Cancel'.

**New Connection Wizard**

**Connection Name**  
Specify a name for this connection to your workplace.

Type a name for this connection in the following box.

Company Name

MICEX VPN Connection

For example, you could type the name of your workplace or the name of a server you will connect to.

< Back   Next >   Cancel

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

**New Connection Wizard**

**Public Network**  
Windows can make sure the public network is connected first.

Windows can automatically dial the initial connection to the Internet or other public network, before establishing the virtual connection.

☒ Do not dial the initial connection.

☐ Automatically dial this initial connection:

< Back   Next >   Cancel

10. Введите выданный биржей IP-адрес, и нажмите *Next*:

**New Connection Wizard**

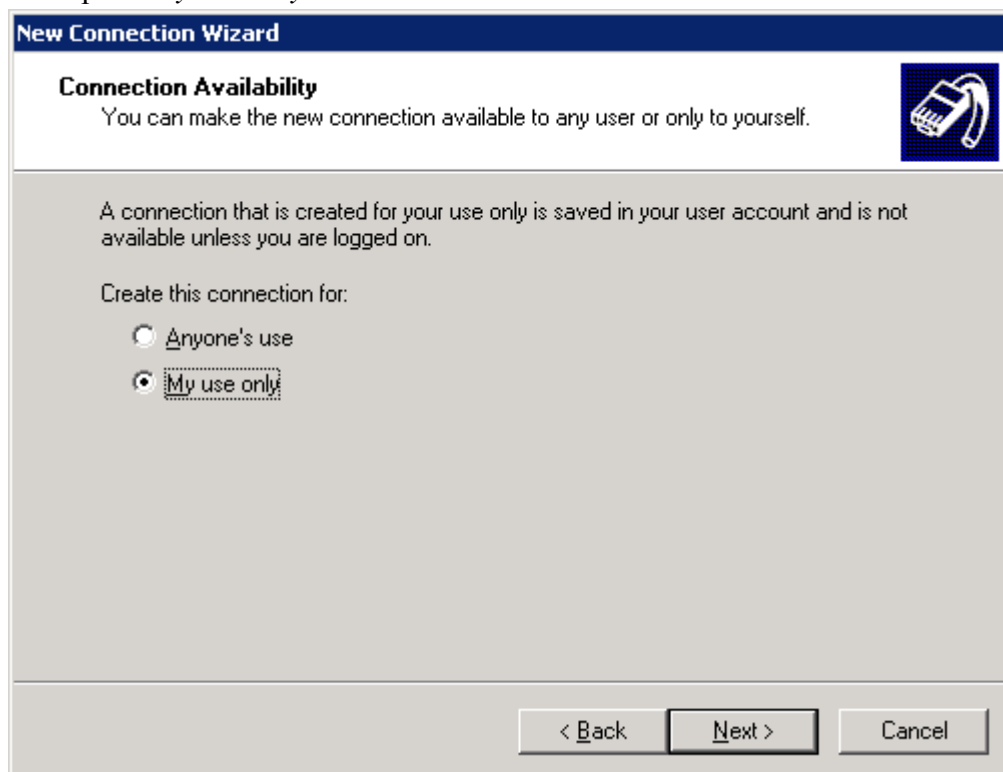
**VPN Server Selection**  
What is the name or address of the VPN server?

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

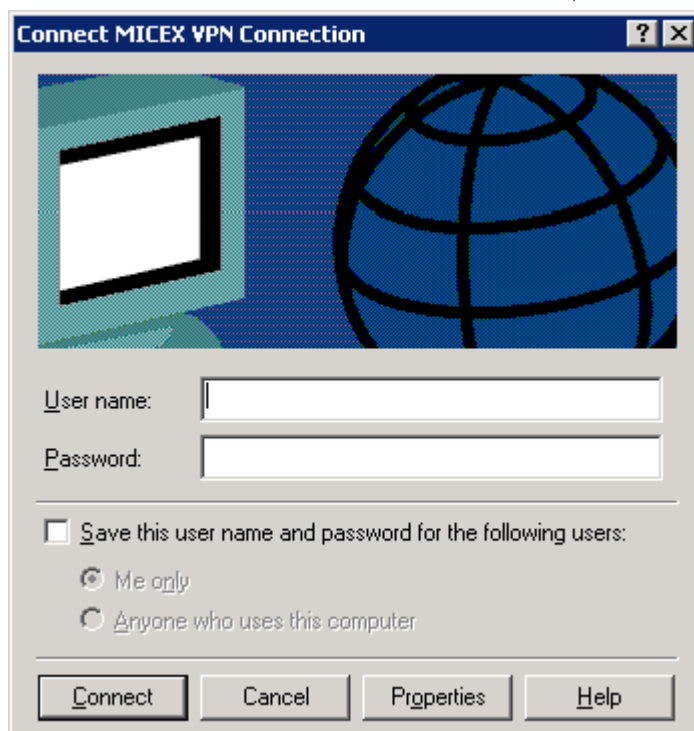
11. Выберите *My use only* и нажмите *Next*:



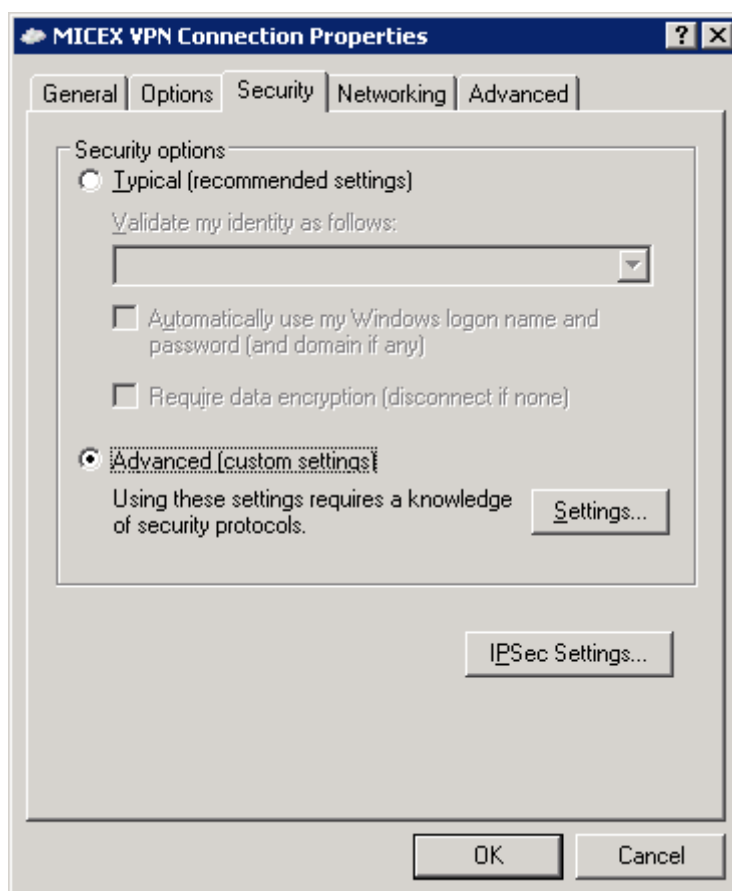
12. Нажмите *Finish*, создание нового соединения завершено:



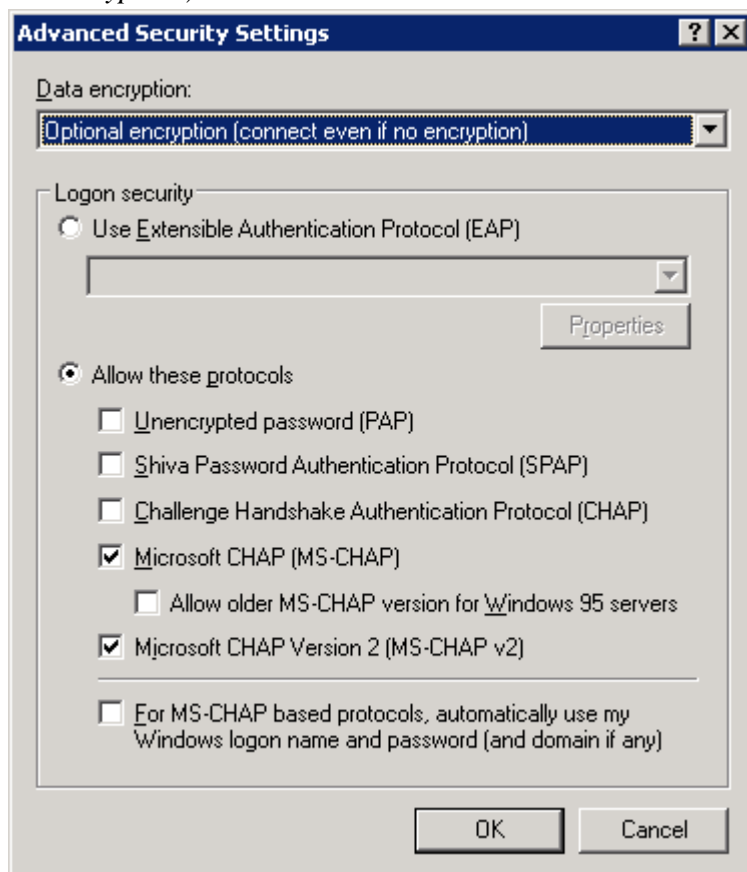
13. Не заполняете поля *User name* и *Password*, нажмите *Properties*



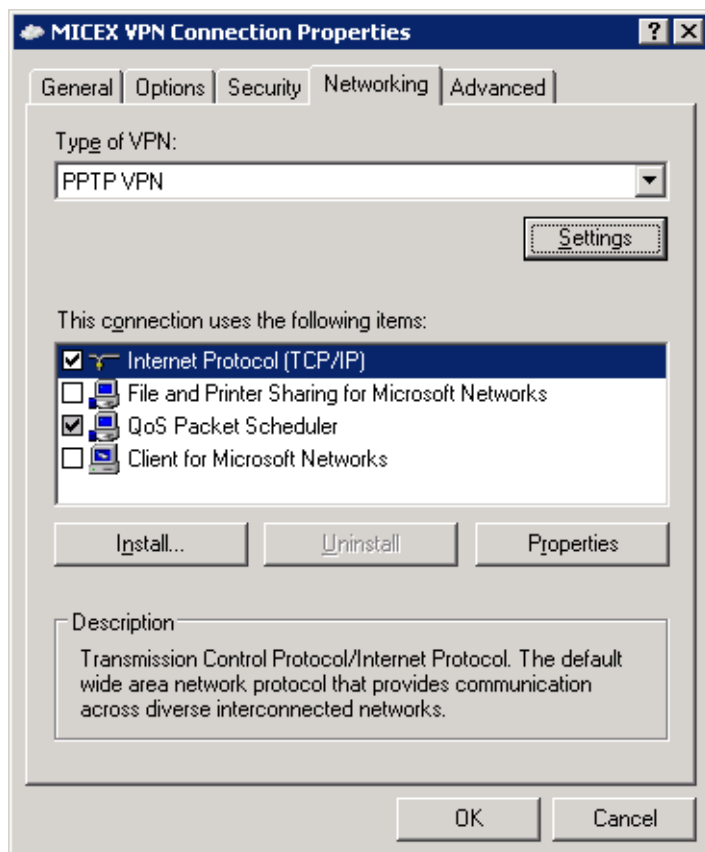
14. Перейдите на закладку *Security*, выберите *Advanced (custom settings)* и нажмите *Settings...*:



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



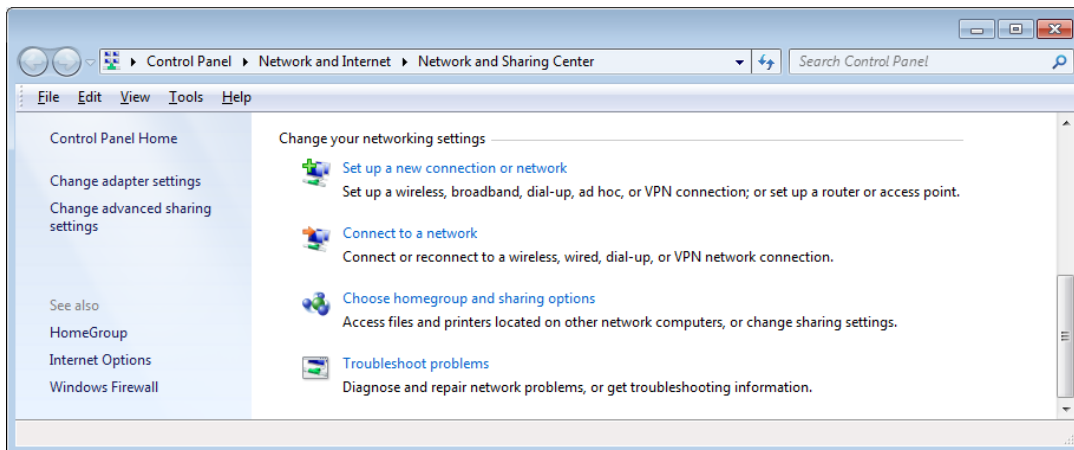
16. На закладке *Networking* в выпадающем списке Type of VPN выберите *PPTP VPN* и нажмите *OK*:



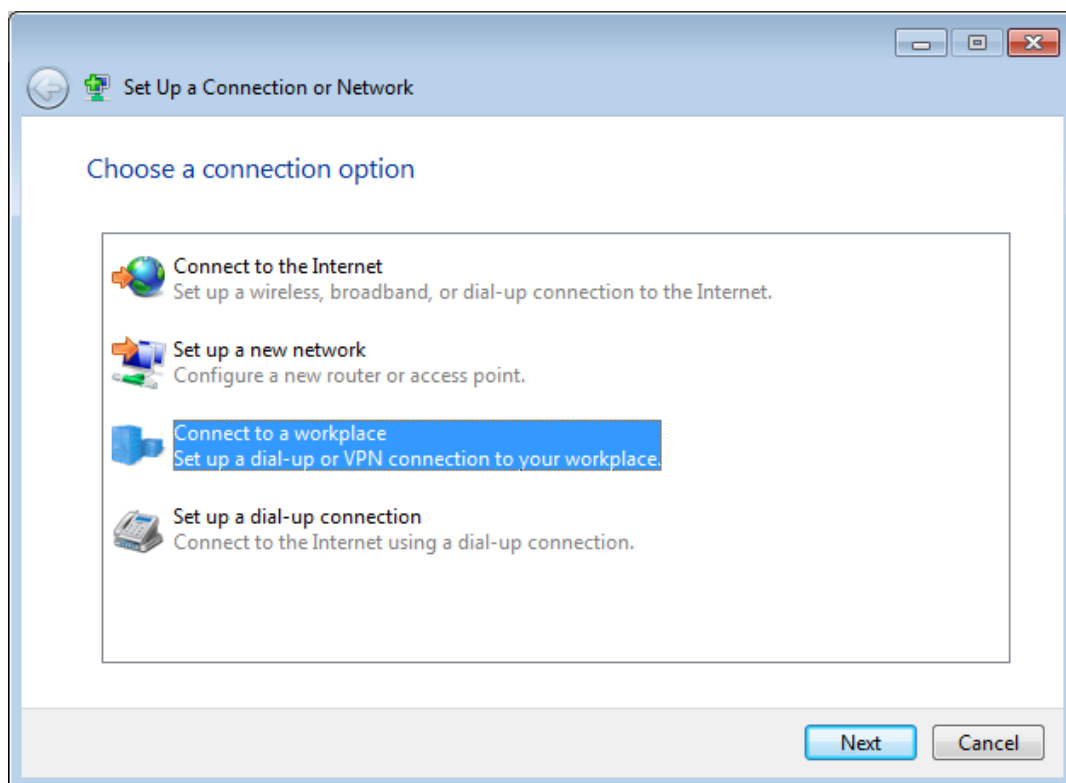


## 5.2. Настройка VPN соединения с MICEX на базе Windows 7

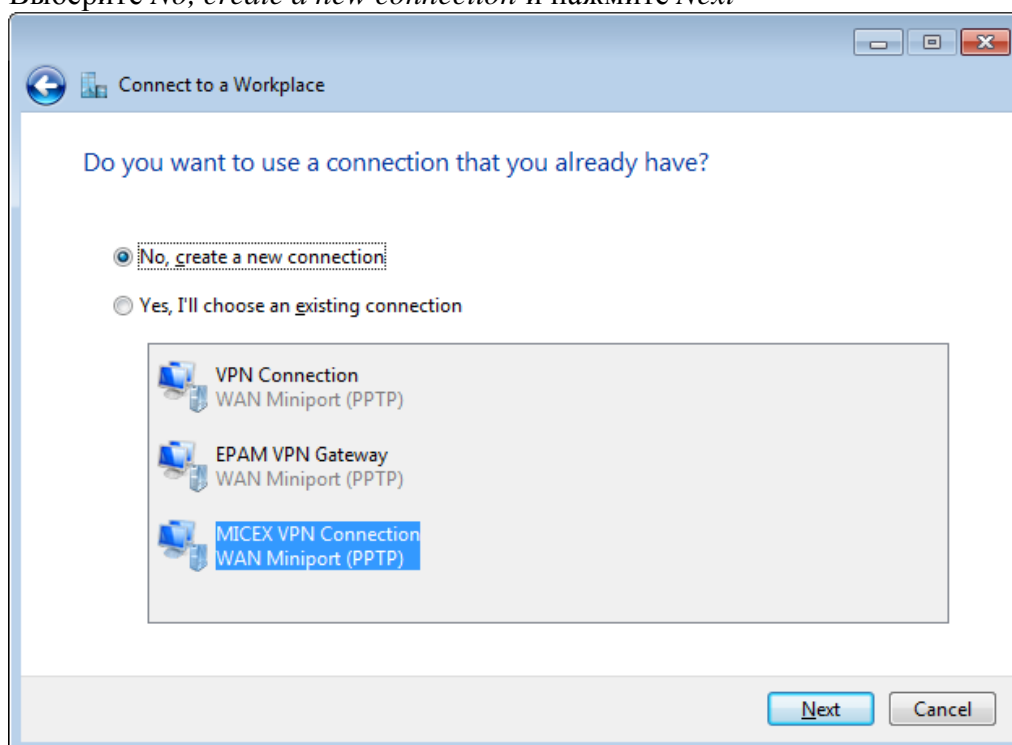
1. Убедитесь, что Internet подключен.
2. Откройте *Control Panel*→*Network and Internet*→*Network and Sharing Center* и выберите *Set up a new connection or network*:



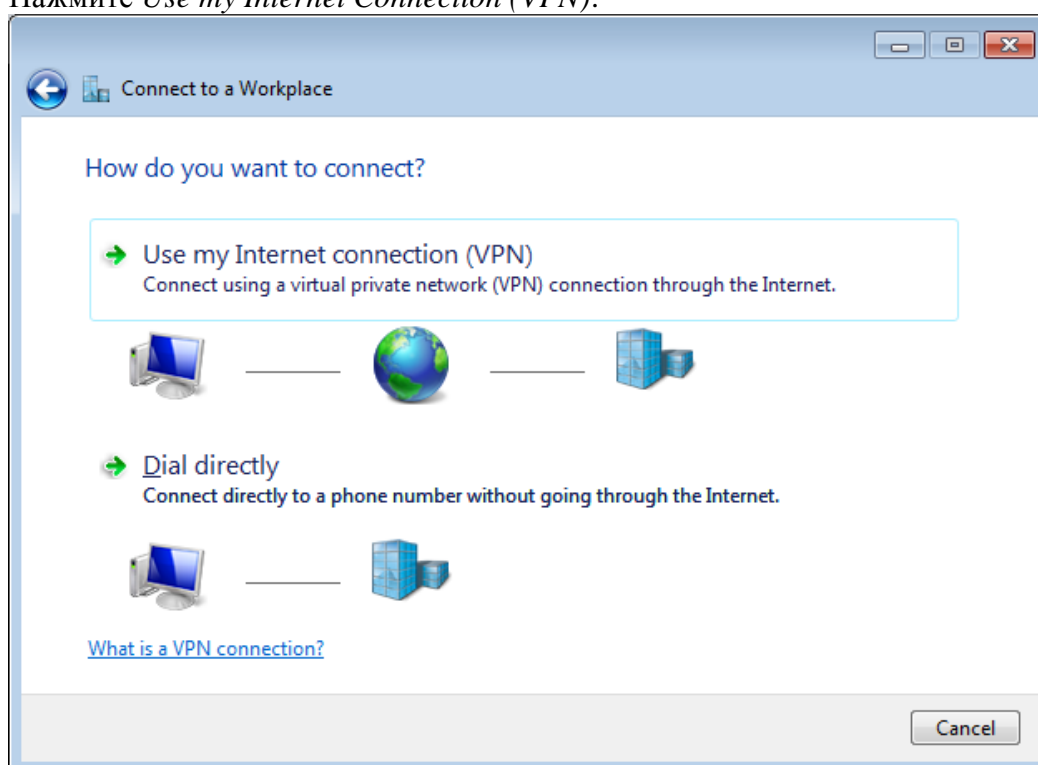
3. Выберите *Connect to a workplace* и нажмите *OK*:



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

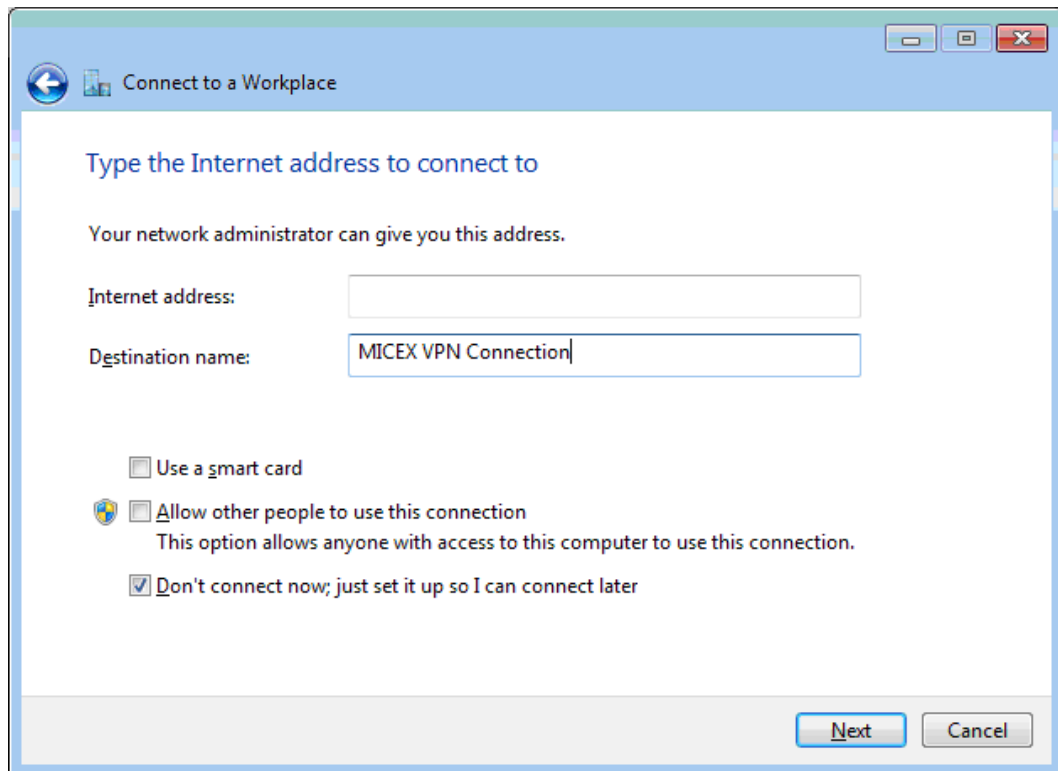


5. Нажмите *Use my Internet Connection (VPN)*:



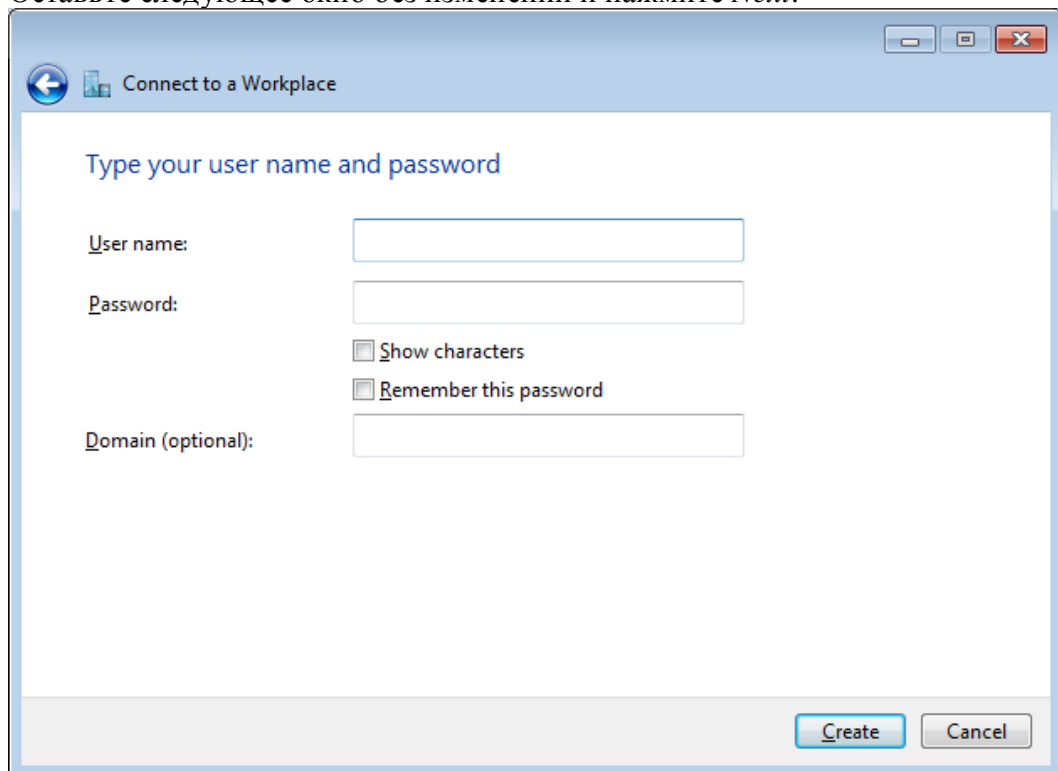
6. Введите выданный биржей IP-адрес в поле *Internet address*, введите MICEX VPN Connection в поле *Destination name*, выберите *Don't connect now; just set it up so I*

can connect later и нажмите *Next*:



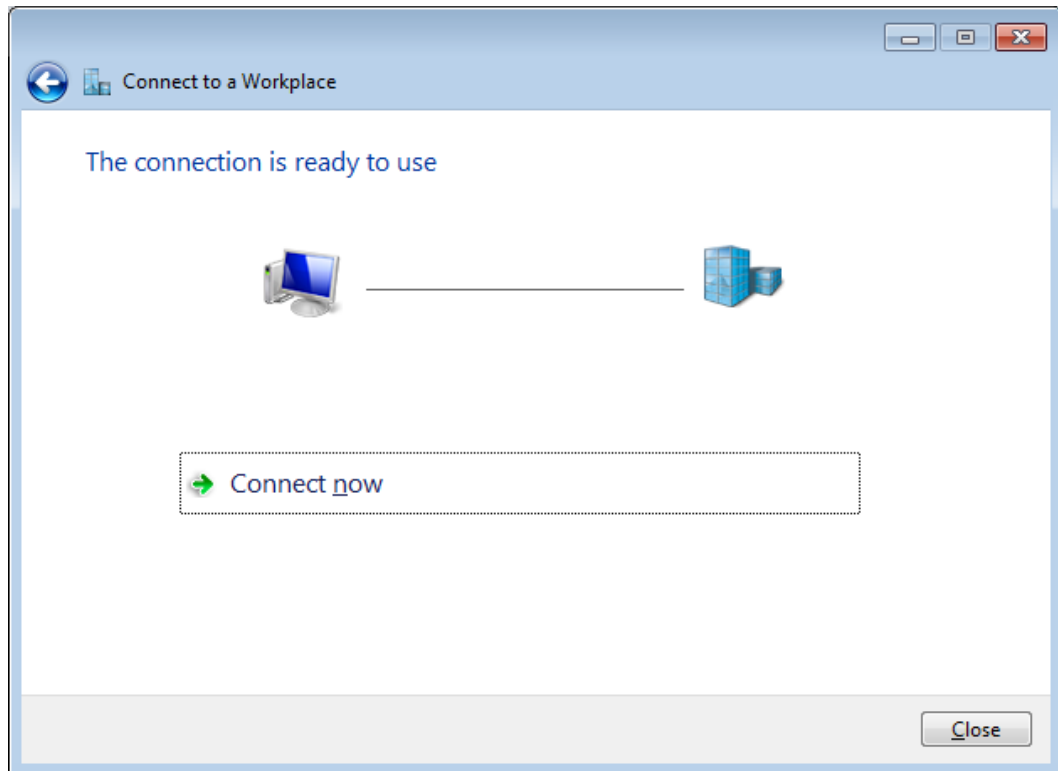
The screenshot shows a Windows XP-style dialog box titled "Connect to a Workplace". The main heading is "Type the Internet address to connect to". Below this, it says "Your network administrator can give you this address." There are two text input fields: "Internet address:" (empty) and "Destination name:" (containing "MICEX VPN Connection"). Below the fields are three checkboxes: "Use a smart card" (unchecked), "Allow other people to use this connection" (unchecked, with a sub-note "This option allows anyone with access to this computer to use this connection."), and "Don't connect now; just set it up so I can connect later" (checked). At the bottom right are "Next" and "Cancel" buttons.

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

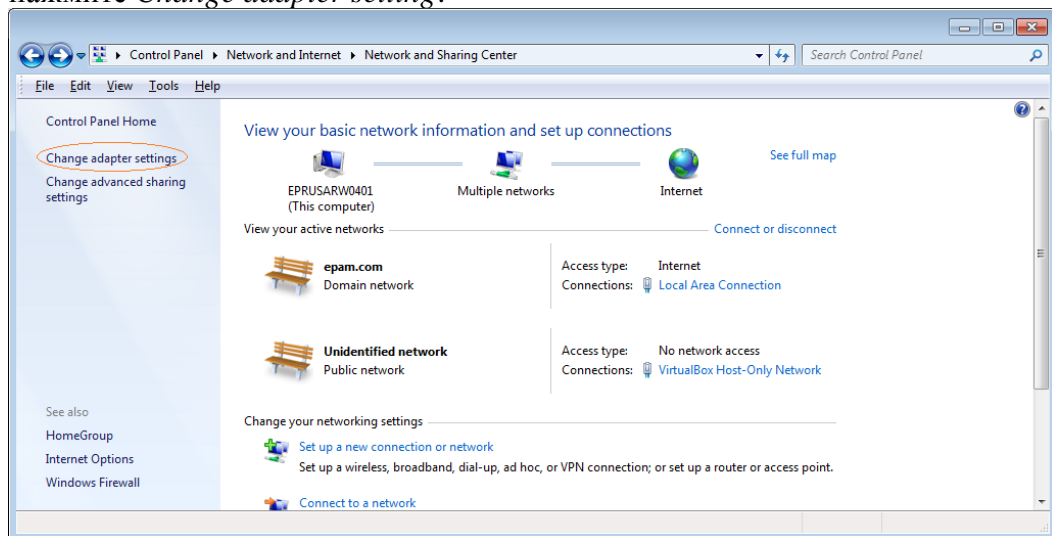


The screenshot shows the same "Connect to a Workplace" dialog box, but at the second step. The heading is "Type your user name and password". There are three text input fields: "User name:", "Password:", and "Domain (optional):". Between the "Password:" and "Domain (optional):" fields are two checkboxes: "Show characters" (unchecked) and "Remember this password" (unchecked). At the bottom right are "Create" and "Cancel" buttons.

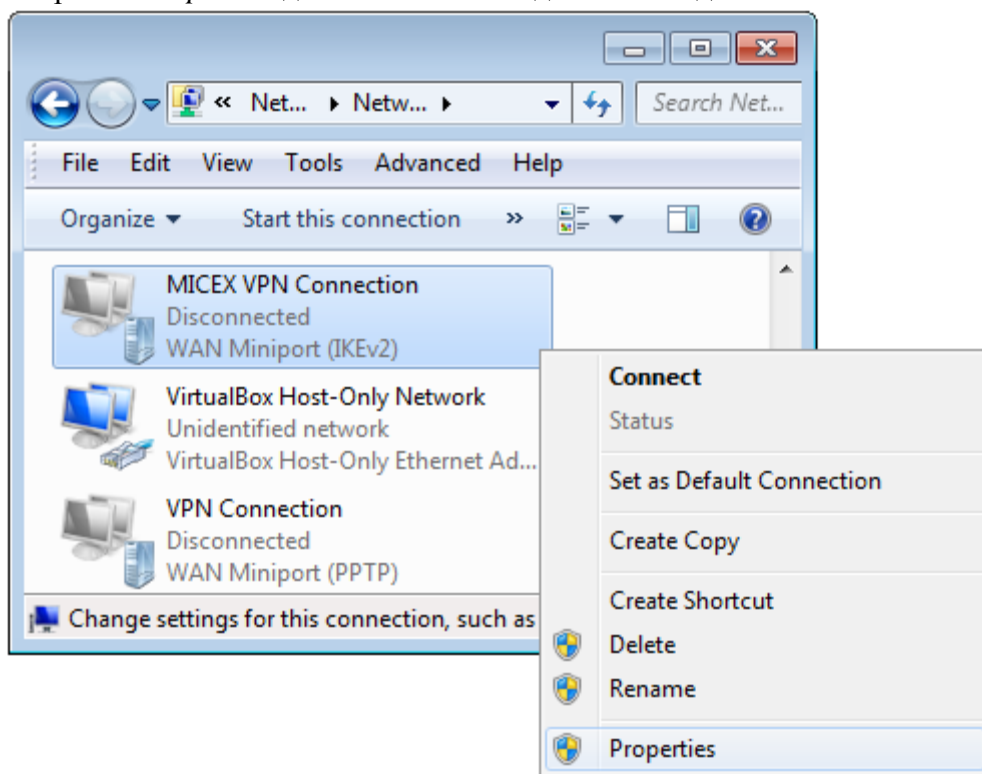
8. Нажмите *Close*:



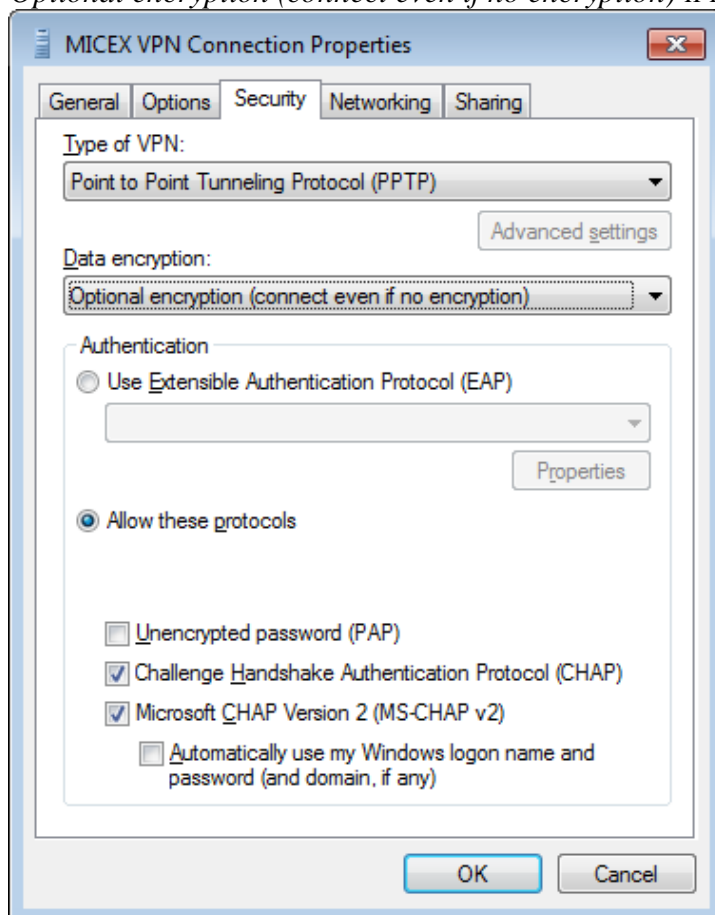
9. Откройте *Control Panel*→*Network and Internet*→*Network and Share Center* и нажмите *Change adapter setting*:



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 MICEX 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 'micex\_vpn\_connection' и нажмите enter:

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

7. Введите выданный биржей IP-адрес и нажмите enter:

```
Server IP: 95.222.333.45 + <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 micex_vpn_connection - 95.222.333.45 - test - PPTP
Added tunnel micex_vpn_connection
```

14. Введите '8' 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/micex_vpn_connection
```

16. Необходимые изменения отмечены **красным**:

```
#
# PPTP Tunnel configuration for tunnel micex_vpn_connection
# Server IP: 95.222.333.45
# 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 micex_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 micex_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.

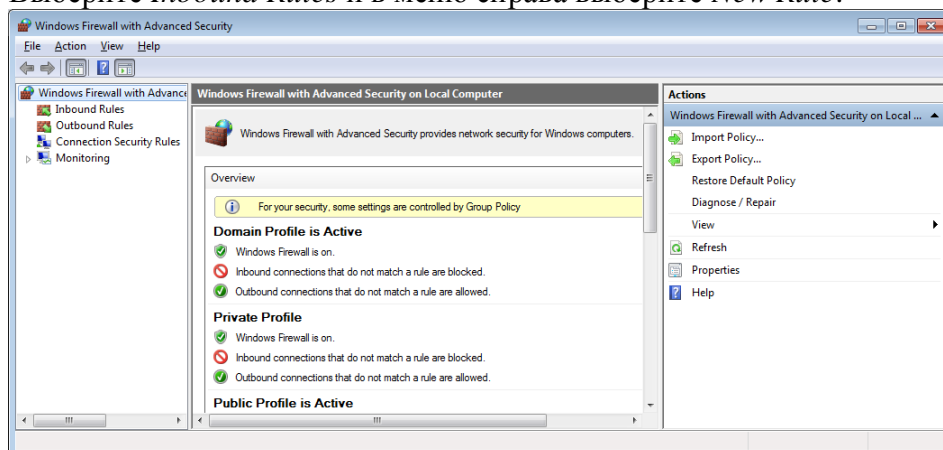
## 5.4. Часто возникающие вопросы и методы их решения

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

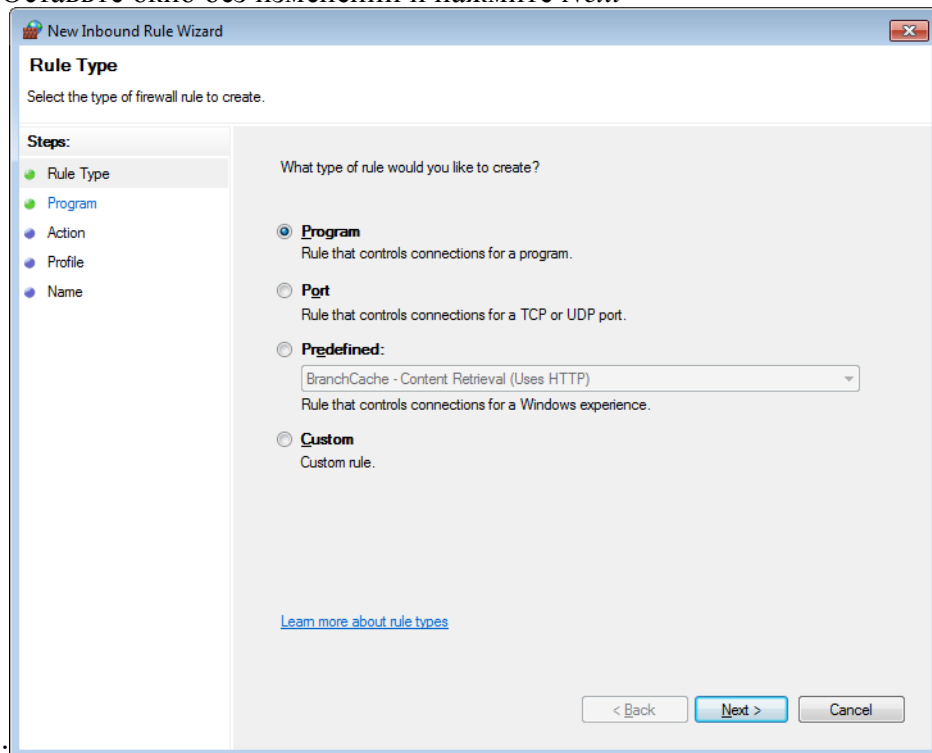
1.1 Посмотрите на состояние вашего VPN соединения, и проверьте, что количество «Полученных» байт возрастает. Если это не так, обратитесь в службу поддержки MICEX.

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:



New Inbound Rule Wizard

### Program

Specify the full program path and executable name of the program that this rule matches.

**Steps:**

- Rule Type
- Program
- Action
- Profile
- Name

Does this rule apply to all programs or a specific program?

☐ All programs  
Rule applies to all connections on the computer that match other rule properties.

☒ This program path:  
  
Browse...

Example: c:\path\program.exe  
%ProgramFiles%\browser\browser.exe

[Learn more about specifying programs](#)

< Back Next > Cancel

✓ 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?

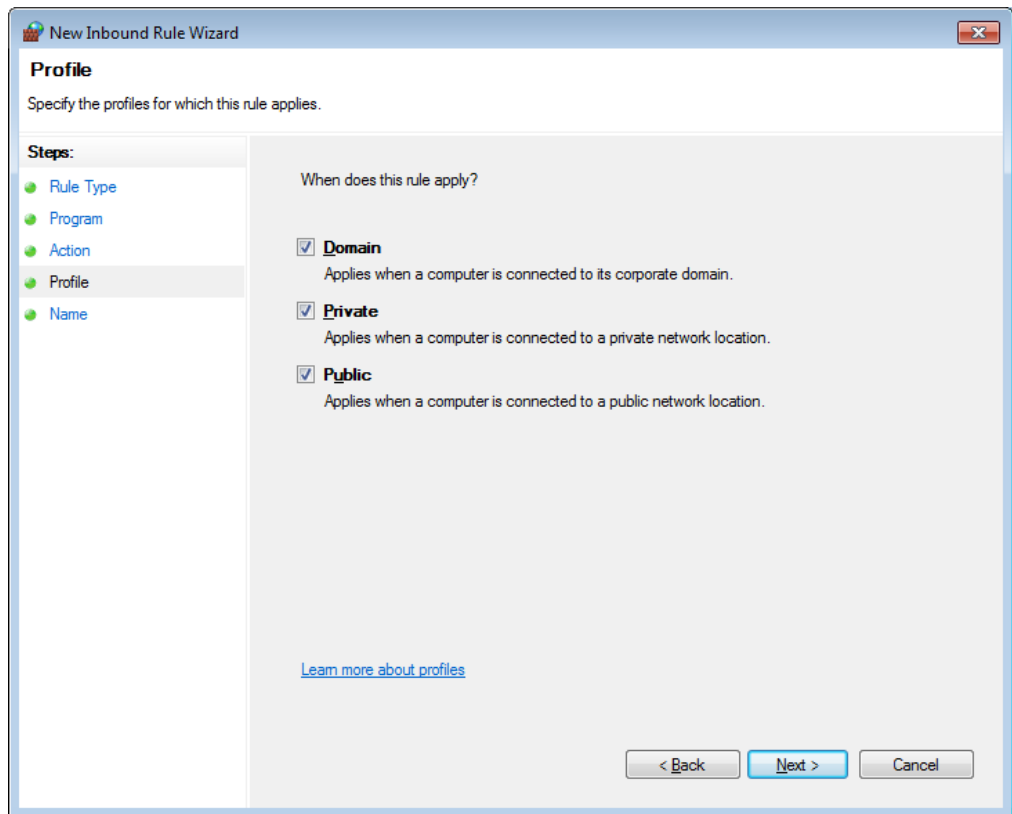
☒ Allow the connection  
This includes connections that are protected with IPsec as well as those are not.

☐ 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.  
Customize...

☐ Block the connection

[Learn more about actions](#)

< Back Next > Cancel



**New Inbound Rule Wizard**

**Profile**

Specify the profiles for which this rule applies.

**Steps:**

- Rule Type
- Program
- Action
- Profile
- Name

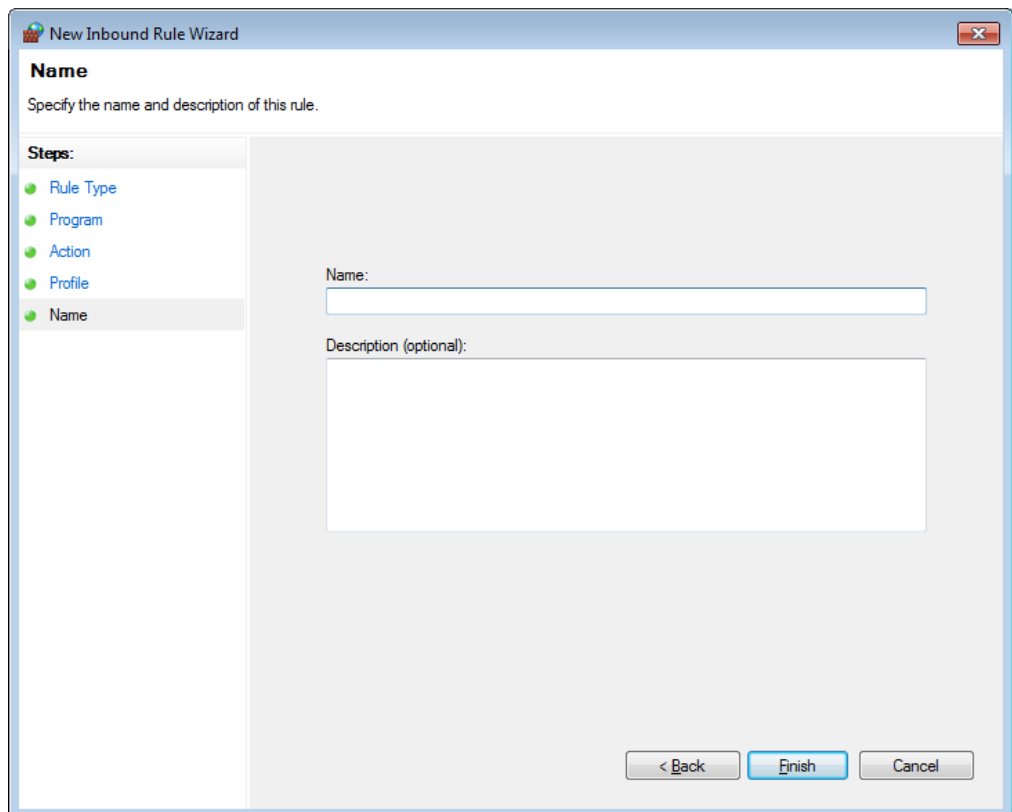
When does this rule apply?

- ☒ **Domain**  
Applies when a computer is connected to its corporate domain.
- ☒ **Private**  
Applies when a computer is connected to a private network location.
- ☒ **Public**  
Applies when a computer is connected to a public network location.

[Learn more about profiles](#)

< Back   Next >   Cancel

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



**New Inbound Rule Wizard**

**Name**

Specify the name and description of this rule.

**Steps:**

- Rule Type
- Program
- Action
- Profile
- Name

Name:

Description (optional):

< Back   Finish   Cancel

## 6. Сертифицированные средства работы

## 6.1. Библиотека FLX 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 MICEX Market Data Multicast.

[FIX Antenna™](#) (C++ and .NET) позволяет осуществлять подписку на рыночные данные от MICEX 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 MICEX 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 MICEX Market Data Multicast.

[illegible]

```

        micex_mfix::mfix_feed_type feed_type)
    {
        //add your processing code here
    }

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

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

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

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

```

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

```

#pragma once

#include <B2BITS_micex_mfix_listeners.h>

namespace mfix_micex_client {
    class application_listener_impl
        : public micex_mfix::micex_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, micex_mfix::mfix_feed_type feed_type)
        {
            //add your processing code here
        }

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

```

```

}

    main.cpp:
#include <iostream>

#include <B2BITS_FixEngine.h>
#include <B2BITS_micex_mfix_application.h>

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

using namespace mfix_micex_client;

void subscribe_and_wait(micex_mfix::micex_mfix_application *app,
                       instrument_listener_impl *ins_listener);

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

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

        //configure parameters
        micex_mfix::micex_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()->createMICEXApplication(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(micex_mfix::micex_mfix_application *app,
                       instrument_listener_impl *ins_listener)
{
    //get channels id
    micex_mfix::channel_ids channels(app->get_channel_ids());

    //get orderbook feed
    micex_mfix::micex_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], micex_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_micex_mfix_application.h
```

```
/include/B2BITS_micex_mfix_listeners.
```

```
/include/B2BITS_micex_mfix_types.h
```

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

<b>micex_mfix::instrument_listener</b>	Слушатель инструментов ( паттерн observer)
<b>micex_mfix::micex_feed</b>	Представляет micex feed (поток - stream)
<b>micex_mfix::micex_mfix_application</b>	Представляет micex mfix application – класс с основными коллбэками
<b>micex_mfix::micex_mfix_application_listener</b>	Представляет micex mfix application listener
<b>micex_mfix::micex_mfix_application_params</b>	Параметры библиотеки
<b>micex_mfix::security_definition_listener</b>	Получает описания финансовых инструментов, вещаемых в потоке рыночных данных

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

```
#include <B2BITS_micex_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::RefCounterPtr из состава библиотеки). Объекты должны создаваться оператором new.

#### 6.1.2.1.2. *micex\_mfix::micex\_feed*

```
#include <B2BITS_micex_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. *micex\_mfix::micex\_mfix\_application*

```
#include <B2BITS_micex_mfix_application.h>
```

#### Public Member Functions

virtual void	<b>release</b> ()=0
	Освобождение ресурсов, выделенных приложением

virtual micex_feed &	<b>get_orderbook_feed</b> () const =0
	Получение ссылки на поток котировок.
virtual micex_feed &	<b>get_statistics_feed</b> () const =0
	Получение ссылки на поток рыночной статистики.
virtual micex_feed &	<b>get_orders_feed</b> () const =0
	Получение ссылки на поток обезличенных заявок
virtual micex_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. *micex\_mfix::micex\_mfix\_application\_listener*

```
#include <B2BITS_micex_mfix_listeners.h>
```

##### Public Member Functions

virtual void	<b>on_error</b> (const std::string &error)=0 Вызывается при ошибке в micex 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 Вызываетс, 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::RefCounterPtr из состава библиотеки). Объекты должны создаваться оператором new.

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

```
#include <B2BITS_micex_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> Тип восстановления. tcp_recovery использует восстановление по TCP-соединению (поле 34 используется для определения пропусков) udp_recovery использует восстановление из UDP-потока снимков (поле 83 используется для определения пропусков). Значение по умолчанию udp_recovery.
std::string	<b>user_login_</b> Имя пользователя для установления TCP-соединения для восстановления
std::string	<b>user_password_</b> Пароль для установления TCP-соединения для восстановления

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

```
#include <B2BITS_micex_mfix_listeners.h>
```

#### Public Member Functions

virtual bool	<b>on_security_definition</b> (const security_description &sec_desc, const security_id &sec_id, const symbol &symb, const std::string &board, const Engine::FIXMessage &d_msg, const std::string &channel_id)=0 Вызывается при получении описания финансового инструмента.
--------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

---

	Вернуть true, если следует продолжить слушать поток инструментов, false в противном случае.
--	---------------------------------------------------------------------------------------------

**Примечание:**

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