



Спецификация протокола FAST Московской Биржи для системы отчета внебиржевых сделок (ОТС-монитор)

Версия 1.18.0

Москва 2023

Содержание

1. Введение	4
1.1. Назначение документа	4
1.2. Общее описание Fast Gate	4
1.2.1. Потокковая передача данных	4
1.2.2. Инкрементальные сообщения	4
1.2.3. FIX формат	4
1.2.4. Кодирование в FAST формат	4
1.2.5. Получение данных с помощью Multicast	4
1.2.6. Восстановление данных	5
2. Сценарии взаимодействия клиентов с Market Data Multicast	6
2.1. Подключение клиента	6
2.2. Обработка дублирующихся данных в потоках А и В	6
3. Функциональность системы	8
3.1. Архитектура системы	8
3.1.1. Основные потоки (UDP)	8
3.1.2. Потоки Recovery (UDP)	8
3.1.3. Потоки Instrument Definitions (UDP)	9
3.1.4. Сообщения в потоках	9
3.1.5. Сессии для запроса пропущенных сообщений (TCP)	9
3.2. FAST формат в информационных потоках	10
3.2.1. Кодирование стоп-бита	10
3.2.2. Неявное тегирование	10
3.2.3. Возможности кодирования полей	11
3.2.4. FAST-шаблон	11
3.2.5. Процесс декодирования	11
3.2.6. Фрагментация сообщений	11
3.2.7. Типы данных	11
3.3. Восстановление пропущенных данных	12
3.3.1. Восстановление пропущенных данных из потоков Recovery (UDP)	12
3.3.2. Восстановление пропущенных данных по TCP-соединению	12
3.4. Сброс номеров сообщений	13
3.5. Шаблоны сообщений	13
3.5.1. OtcMonitorIncrementalRefreshMessage	13
3.5.2. OtcMonitorSnapshotMessage	14
3.5.3. OtcMonitorSecurityDefinition	14
4. Публичный FIX интерфейс	16
4.1. Группы полей	16
4.1.1. Standard Message Header	16
4.2. Сообщения сессионного уровня	16
4.2.1. Logon (A)	16
4.2.2. Logout (5)	16
4.2.3. Heartbeat (0)	17
4.2.4. Sequence Reset (4)	17
4.3. Сообщения бизнес-логического уровня	17
4.3.1. Security Definition (d)	17
4.3.2. Market Data Request (V)	18
4.3.3. Market Data - Snapshot / Full Refresh (W)	18
4.3.4. Market Data - Incremental Refresh (X)	19
5. Ограничения сервиса TCP Recovery (Historical Replay)	21

История изменений

Дата	Версия	Изменения
20.01.2023	1.18.0	<ol style="list-style-type: none"> 1. Спецификация FAST для ОТС-монитора выделена в отдельный документ. 2. Под трансляцию данных из ОТС-монитора добавляются три новых шаблона сообщений: <ul style="list-style-type: none"> • OtcMonitorIncrementalRefreshMessage (id="33") • OtcMonitorSnapshotMessage (id="34") • OtcMonitorSecurityDefinition (id="35") 3. В отчет о сделке - сообщения Market Data - Snapshot / Full Refresh (W) и Market Data - Incremental Refresh (X) добавлены новые поля: <ul style="list-style-type: none"> • SettlCurrency - Валюта денежного обязательства • CFICode - Код CFI ценной бумаги, с которой совершена сделка • TradeVolume - Объем сделки, руб. 4. В сообщении Security Definition добавлены новые поля: <ul style="list-style-type: none"> • CFICode - Код CFI ценной бумаги, с которой совершена сделка • InList - Вхождение ценной бумаги в котировальный список Биржи

1. Введение

1.1. Назначение документа

В данном документе представлено описание протокола Fast для внебиржевых операций (ОТС_монитор).

В данную спецификацию не входят административные и технические аспекты организации сетевого подключения, а также способы обеспечения его безопасности.

1.2. Общее описание Fast Gate

Система **Fast Gate** предназначена для распространения рыночных данных в FAST-формате по протоколу UDP в режиме multicast.

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.

FAST (FIX Adapted for STreaming) протокол разработан FIX Market Data Optimization Working Group на основе протокола FIX для оптимизации электронного обмена финансовой информации, в частности, для распространения большого объема данных с минимальной задержкой. Fast Gate использует версию протокола 1.1: <https://www.fixtrading.org/packages/fast-specification-version-1-1>.

Через FAST Gate для ОТС-монитора раздаются следующие данные:

- описания инструментов;
- анонимные данные об отчетах по внебиржевым сделкам.

1.2.1. Поточковая передача данных

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

1.2.2. 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.

1.2.3. FIX формат

Система **Fast Gate** 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. раздел 4.

1.2.4. 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 протокол 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. раздел 3.2.

1.2.5. 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 обеспечивает отправку в одном UDP пакете только одного закодированного в FAST сообщения. FAST сообщение специально формируется таким образом, чтобы размер UDP пакета не превышал типичного для сети Ethernet значения параметра MTU в 1500 байт.

1.2.6. Восстановление данных

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

Fast Gate обеспечивает восстановление данных 2 способами:

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

2. Сценарии взаимодействия клиентов с Market Data Multicast

В данном разделе описаны процедуры подключения клиентов к каналам Market Data Multicast в различные периоды работы системы, а так же процедуры восстановления потерянных данных.

2.1. Подключение клиента

При подключении к системе для получения полной рыночной информации следует придерживаться следующей процедуры:

1. Скачать файл конфигурации каналов и потоков с ftp-сервера. Конфигурационный файл в формате .xml описывает параметры подключения (IP адреса multicast, номера портов и т.д.).
2. Скачать файл FAST-шаблона с ftp-сервера. Более подробное описание Fast-шаблона см. раздел Fast-шаблон.
3. Получить список инструментов из потока **Instrument Replay**. Начать слушать поток **Instruments Incremental**.
4. Начать слушать потоки **Incremental** и накапливать получаемые сообщения.
5. Начать слушать потоки **Snapshot**. Получить по этим потокам актуальный снэпшот, применить полученный снэпшот. 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. Для каждого инструмента следует отбросить из накопленных данных все сообщения с номерами до значения тэга *369-LastMsgSeqNumProcessed* включительно и применить все оставшиеся. Процесс можно проводить как последовательно (сначала получить снэпшоты по всем инструментам, а потом обрабатывать накопленные обновления), так и параллельно (по мере получения снэпшотов по инструментам обрабатывать накопленные обновления по полученному инструменту).
6. Перестать слушать потоки **Snapshot**.
7. Продолжить обычную обработку потоков инкрементальных обновлений.

2.2. Обработка дублирующихся данных в потоках А и В

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

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

Обработку потоков А и В следует производить по следующему алгоритму:

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

Пример:

Порядок пакетов	Поток А	Поток В
1	34-MsgSeqNum = 59	
2		34-MsgSeqNum = 59
3	34-MsgSeqNum = 60	
4		34-MsgSeqNum = 60
5	34-MsgSeqNum = 62	
6		34-MsgSeqNum = 61
7		34-MsgSeqNum = 62
8	34-MsgSeqNum = 62	
9	34-MsgSeqNum = 63	
10	34-MsgSeqNum = 65	
11		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:

- потоки распространения инкрементальных обновлений рыночных данных;
- потоки распространения описаний финансовых инструментов;
- потоки распространения информации об изменении статуса финансовых инструментов и сообщений о состоянии соединения с Торговой системой.

2. Потоки восстановления:

- потоки распространения снэпшотов рыночных данных;
- сессии для запроса пропущенных данных.

Данные распространяются по набору Каналов, в каждом из которых представлены данные по группе финансовых инструментов. Группировка осуществляется по определённым параметрам, она определяется конфигурацией Системы. За вещание на каждом из Каналов отвечает отдельный экземпляр Market Data Multicast. Один экземпляр Market Data Multicast обеспечивает вещание только одного Канала.

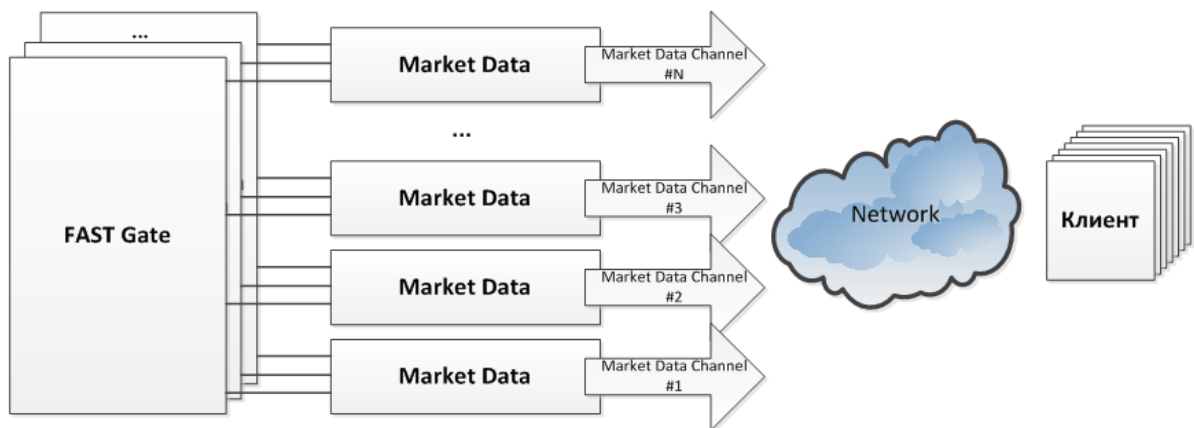


Рис. 1. Каналы распространения рыночных данных

Каждый Канал представляет набор нескольких UDP-потоков (UDP Feeds), по которым идёт непрерывное вещание данных, и TCP-порт доступа, на котором клиент может запросить пропущенные в UDP-потоке сообщения.

Все 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.

Помимо трансляции данных в UDP-потоках, Market Data Multicast 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 (данные доступны за период времени, указанный в конфигурационном файле (но не более чем с начала дня), количество отсылаемых за один раз сообщений и общее количество запрашиваемых в день сообщений ограничено (все ограничения указываются в конфигурационном файле системы)).

3.1.1. Основные потоки (UDP)

Данные по ОТС-сделкам распространяются в основном потоке (incr) OTC-TRADES в режиме multicast по протоколу UDP.

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

3.1.2. Потоки Recovery (UDP)

В потоках Recovery (snap) в режиме multicast по протоколу UDP с фиксированной периодичностью распространяются текущие снэпшоты соответствующих данных в виде FIX-сообщений Market Data - Snapshot/Full Refresh (W), закодированных в формат FAST. Каждое сообщение содержит информацию по одному финансовому инструменту.

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.

3.1.3. Потоки Instrument Definitions (UDP)

В потоках Instrument Replay (inst replay) с фиксированной периодичностью рассылаются описания торговых инструментов в виде FIX-сообщений Security Definition (d), закодированных в формат FAST. Одно сообщение содержит описание одного финансового инструмента. В потоке Instrument Replay сообщение SequenceReset (4) обозначает начало трансляции нового снапшота. Снапшот состоит из описаний торговых инструментов в виде сообщений SecurityDefinition (d).

Данные по OTC инструментам транслируются в потоке OTC-ISSUES.

3.1.4. Сообщения в потоках

В разделе описано, какие сообщения передаются в каждом потоке данных.

Имя потока	Тип потока	Имя шаблона сообщения
OTC-ISSUES	Instrument Replay	Heartbeat (id="6") SequenceReset (id="7") OtcMonitorSecurityDefinition (id="35")
OTC-TRADES	Incremental	Heartbeat (id="6") SequenceReset (id="7") OtcMonitorIncrementalRefreshMessage (id="33")
OTC-TRADES	Snapshot	Heartbeat (id="6") SequenceReset (id="7") OtcMonitorSnapshotMessage (id="34")
OTC-TRADES	Historical Replay	От клиента на шлюз: Logon (FIX MessageType="A") Logout (FIX MessageType="5") Market Data Request (FIX MessageType="V") От шлюза на клиента: Heartbeat (id="6") OtcMonitorIncrementalRefreshMessage (id="33") Logon (id="1000") Logout (id="1001")

3.1.5. Сессии для запроса пропущенных сообщений (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. Максимальное количество сообщений, которое может запросить клиент, ограничено. Запрос отправляется в виде FIX-сообщения Market Data Request (V). Запрос отправляется по TCP-соединению, иницируемому клиентом. Ответные сообщения отправляются клиенту по этому же TCP-соединению в виде FIX-сообщений. По завершению отправки ответных сообщений Market Data Multicast закрывает это TCP-соединение. Все ответные сообщения закодированы в FAST-формат.

Длина каждого сообщения в TCP потоке данных передается как 4-байтное число в начале каждого сообщения:

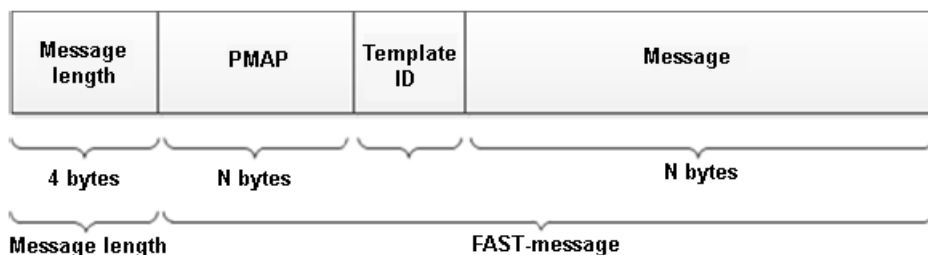


Рис. 2. Структура сообщения в TCP потоке

По завершению отправки сообщений FAST гейт отправляет клиенту FAST сообщение Logout и ожидает ответного FIX сообщения Logout. После завершения FIX сессии TCP соединение закрывается.

Данный сервис должен использоваться лишь в крайнем случае, если другие методы восстановления невозможны. Сервис не обеспечивает высокую производительность. Сервис не доступен для потоков с агрегированной таблицей котировок.

3.2. FAST формат в информационных потоках

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

Особенностью распространения данных в информационных потоках от MOEX Market Data Multicast является то, что перед каждым FAST-сообщением добавляется 4-байтовая преамбула, в которой содержится значение 34-го тэга (SeqNum) следующего за преамбулой FAST-сообщения.

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.

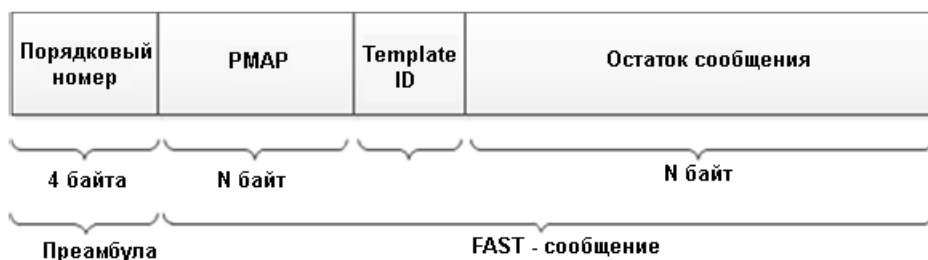


Рис. 3. Структура сообщения

3.2.1. 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.

3.2.2. Неявное тегирование

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.

3.2.3. 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).

3.2.4. 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. Шаблоны сообщений доступны по адресу: <ftp://ftp.moex.com/pub/FAST/Spectra/>.

3.2.5. Процесс декодирования

Процесс декодирования происходит в следующей последовательности:

1. Транспорт: Клиент системы получает закодированное FAST сообщение.
2. Декодирование пакета:
 - определение шаблона;
 - извлечение бинарных закодированных бит;
 - построение соответствия между полученными битами и полями в шаблоне.
3. Декодирование полей: применение операторов для определения значения на основании шаблона.
4. Построение FIX сообщения.
5. Обработка FIX сообщения.

3.2.6. Фрагментация сообщений

Фрагментация сообщений выполняется для того, чтобы размер UDP пакета не превышал типичного для сети Ethernet значения параметра MTU в 1500 байт.

Если в сообщении Market Data - Snapshot / Full Refresh (W) отсутствует тэг 893-LastFragment, то это означает, что снэпшот по инструменту передан в виде одного сообщения. Тэг 893-LastFragment, заполненный значением '0', отмечает в серии фрагментированных сообщений все сообщения, кроме последнего. Тэг 893-LastFragment, заполненный значением '1', отмечает последнее сообщение в снэпшоте по данному инструменту. Поэтому снэпшот по инструменту следует считать полученным, когда получено сообщение с 893-LastFragment = '1'.

Если в сообщении Market Data - Incremental Refresh (X) отсутствует тэг 893-LastFragment, то это означает, что фрагментации не выполнялось и данные находятся в непротиворечивом состоянии после обработки этого сообщения. Тэг 893-LastFragment, заполненный значением '0', отмечает в серии фрагментированных сообщений все сообщения, кроме последнего. Тэг 893-LastFragment, заполненный значением '1', отмечает последнее сообщение в серии. Поэтому данные находятся в непротиворечивом состоянии только после обработки сообщения с 893-LastFragment = '1'.

3.2.7. Типы данных

Поля в шаблоне сообщений могут иметь один из стандартных типов данных, требующих соответствующее декодирование: ASCII строка, Unicode строка, Signed Integer, Unsigned Integer and Decimal. Decimal экспонента и мантисса кодируются как одно составное поле.

FAST не поддерживает типы для передачи timestamps. FAST гейт конвертирует timestamp в подходящий целочисленный тип. После декодирования клиенту следует конвертировать полученное целое число в FIX UTC формат. Время всегда в UTC.

Примеры декодирования timestamp:

FIX Type	FIX Pattern	Sample FIX value	Sample FAST value	FAST field type
UTCTimeOnly	HH:MM:SS.ssssssss (nanoseconds)	18:44:24.123456789	184424123456789	uint64

FIX Type	FIX Pattern	Sample FIX value	Sample FAST value	FAST field type
	HH:MM:SS.ssssssss (nanoseconds)	07:12:13.012345678	71213012345678	uInt64
UTCDateOnly	YYYYMMDD	20080812	20080812	uInt32
UTCTimestamp	YYYYMMDD- HH:MM:SS.sss	20080812-18:23:54.213	20080812182354123	uInt64

3.3. Восстановление пропущенных данных

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

Fast Gate 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.

3.3.1. 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 снимки 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 снимок, 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 снимок. 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 снимков начинается с 1. Поэтому все снимки следует считать полученными, когда приходит сообщение с порядковым номером 1, которое относится к следующему циклу.

Если в сообщении отсутствует тэг 893-LastFragment, то это означает, что снимок по инструменту передан в виде одного сообщения. Иначе тэг 893-LastFragment, заполненный значением '1', отмечает последнее сообщение в снимке по данному инструменту. Поэтому снимок по инструменту следует считать полученным, когда получено сообщение с 893-LastFragment = '1'.

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

Последовательность шагов при восстановлении соответствует шагам 4–7, приведенным в раздел 2.1.

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

3.3.2. Восстановление пропущенных данных по TCP-соединению

Восстановление данных, пропущенных в основных (Incremental) потоках: Indexes, Trades и ORDERS-LOG (поток анонимных заявок и сделок) 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, равно 1000.

Для запроса пропущенных данных клиент должен выполнить следующие действия:

1. Установить TCP-соединение с сервером MOEX Market Data Multicast.
2. Отправить серверу FIX-сообщение Logon(A). В случае успешной авторизации, сервер ответит FAST-сообщением Logon(A).
3. Отправить серверу FIX-сообщение Market Data Request (V), в котором необходимо указать:

а. диапазон порядковых номеров запрашиваемых сообщений – в тэгах 1182-AppBeginSeqNo и 1183-AppEndSeqNo.

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

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

После отправки ответа сервер закрывает соединение.

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

Если в течение 1 секунды от клиента не поступает никакого запроса, то сервер закрывает соединение.

3.4. Сброс номеров сообщений

Ежесуточно происходит очистка Fast Gate от сообщений торговой сессии предыдущего дня и сброс номеров сообщений. После сброса номеров в потоках (incr) транслируется сообщение Sequence Reset с новым номером в поле NewSeqNo. При получении сообщения Sequence Reset клиент должен установить номер сообщений в значение NewSeqNo и сбросить номера RptSeq.

Для Fast Gate установлены следующие технологические перерывы, после завершения которых происходит сброс номеров сообщений:

- ОТС-монитор - с 00:00 MSK по 00:02 MSK;
- тестовые системы для ОТС-монитор - с 00:00 MSK по 00:02 MSK.

Во всех основных потоках (incr) происходит сброс номера сообщений в значение 1 и сброс номера RptSeq в значение 1. При этом транслируется пара сообщений Sequence Reset:

- Sequence Reset: MsgSeqNum=N NewSeqNo[36]=1
- Sequence Reset: MsgSeqNum=N NewSeqNo[36]=N

После трансляции этих сообщений в TCP Recovery доступны FAST сообщения с номерами от 1 до N-1 включительно, содержащие сделки предыдущей вечерней дополнительной торговой сессии. Начальный номер RptSeq клиент может получить одним из двух способов:

- выполнив запрос и обработку сообщений с номерами от 1 до N-1 из сервиса TCP Recovery;
- выполнив подключение к потоку Recovery (UDP) в соответствии с разделом "3.3.1. Восстановление пропущенных данных из потоков Recovery (UDP)".

3.5. Шаблоны сообщений

Под трансляцию данных из ОТС-монитора используются следующие шаблоны сообщений:

- **OtcMonitorIncrementalRefreshMessage (id="33")** - см. раздел 3.5.1
- **OtcMonitorSnapshotMessage (id="34")** - см. раздел 3.5.2
- **OtcMonitorSecurityDefinition (id="35")** - см. раздел 3.5.3

3.5.1. OtcMonitorIncrementalRefreshMessage

Данный шаблон сообщений используется при раздаче обновлений и в сервисе TCP Recovery.

```
<template name="OtcMonitorIncrementalRefreshMessage" id="33">
  <string name="AppVerID" id="1128">
    <constant value="9"/>
  </string>
  <string name="MessageType" id="35">
    <constant value="X"/>
  </string>
  <string name="SenderCompID" id="49">
    <constant value="MOEX"/>
  </string>
  <uint32 name="MsgSeqNum" id="34"/>
  <uint64 name="SendingTime" id="52"/>
  <uint32 name="LastFragment" id="893" presence="optional"/>
  <sequence name="MDEntries">
    <length name="NoMDEntries" id="268"/>
    <uint32 name="MDUpdateAction" id="279"/>
    <string name="MDEntryType" id="269"/>
  </sequence>
</template>
```

```

<string name="Symbol" id="55"/>
<string name="SecurityGroup" id="1151"/>
<uInt32 name="RptSeq" id="83"/>
<int64 name="MDEntryID" id="278"/>
<string name="MDEntryPx" id="270"/>
<int64 name="MDEntrySize" id="271"/>
<uInt32 name="MDEntryDate" id="272" presence="optional"/>
<uInt64 name="MDEntryTime" id="273"/>
<string name="Currency" id="15"/>
<uInt64 name="Revision" id="20018" presence="optional"/>
<string name="OrderSide" id="10504"/>
<string name="SettlCurrency" id="120"/>
<string name="CFICode" id="461"/>
<string name="TradeVolume" id="1020"/>
</sequence>
</template>

```

3.5.2. OtcMonitorSnapshotMessage

Данный шаблон сообщений используется при раздаче снимка.

```

<template name="OtcMonitorSnapshotMessage" id="34">
  <string name="ApplVerID" id="1128">
    <constant value="9"/>
  </string>
  <string name="MessageType" id="35">
    <constant value="W"/>
  </string>
  <string name="SenderCompID" id="49">
    <constant value="MOEX"/>
  </string>
  <uInt32 name="MsgSeqNum" id="34"/>
  <uInt64 name="SendingTime" id="52"/>
  <uInt32 name="LastFragment" id="893" presence="optional"/>
  <uInt32 name="RptSeq" id="83"/>
  <uInt32 name="TotNumReports" id="911"/>
  <uInt32 name="LastMsgSeqNumProcessed" id="369"/>
  <string name="Symbol" id="55"/>
  <string name="SecurityGroup" id="1151"/>
  <sequence name="MDEntries">
    <length name="NoMDEntries" id="268"/>
    <uInt32 name="MDUpdateAction" id="279"/>
    <string name="MDEntryType" id="269"/>
    <int64 name="MDEntryID" id="278"/>
    <string name="MDEntryPx" id="270"/>
    <uInt32 name="MDEntryDate" id="272" presence="optional"/>
    <uInt64 name="MDEntryTime" id="273"/>
    <int64 name="MDEntrySize" id="271"/>
    <string name="Currency" id="15"/>
    <string name="OrderSide" id="10504"/>
    <string name="SettlCurrency" id="120"/>
    <string name="CFICode" id="461"/>
    <string name="TradeVolume" id="1020"/>
  </sequence>
</template>

```

3.5.3. OtcMonitorSecurityDefinition

Данный шаблон сообщений используется при раздаче информации об инструментах.

```

<template name="OtcMonitorSecurityDefinition" id="35">
  <string name="ApplVerID" id="1128">
    <constant value="9"/>
  </string>
  <string name="MessageType" id="35">
    <constant value="d"/>
  </string>
  <string name="SenderCompID" id="49">
    <constant value="MOEX"/>
  </string>
  <uInt32 name="MsgSeqNum" id="34"/>
  <uInt64 name="SendingTime" id="52"/>
  <!-- Total count of SecurityDefinition messages -->

```

```
<uInt32 name="TotNumReports" id="911"/>
<string name="Symbol" id="55"/>
<string name="SecurityDesc" id="107" presence="optional" charset="unicode"/>
<!-- Unique among all instruments; primary key -->
<uInt64 name="SecurityID" id="48"/>
<uInt32 name="SecurityIDSource" id="22">
  <constant value="8"/>
</uInt32>
<string name="SecurityAltID" id="455"/>
<string name="SecurityAltIDSource" id="456"/>
<string name="CFICode" id="461"/>
<string name="MarketID" id="1301">
  <constant value="MOEX"/>
</string>
<string name="MarketSegmentID" id="1300"/>
<sequence name="MDFeedTypes">
  <length name="NoMDFeedTypes" id="1141"/>
  <string name="MDFeedType" id="1022"/>
  <uInt32 name="MarketDepth" id="264" presence="optional"/>
  <uInt32 name="MDBookType" id="1021" presence="optional"/>
</sequence>
<sequence name="InstrumentAttributes">
  <length name="NoInstrAttrib" id="870"/>
  <int32 name="InstrAttribType" id="871"/>
  <string name="InstrAttribValue" id="872" charset="unicode"/>
</sequence>
<decimal name="UnderlyingQty" id="879" presence="optional"/>
<string name="UnderlyingCurrency" id="318" presence="optional"/>
<string name="InList" id="20052"/>
</template>
```

4. Публичный FIX интерфейс

Описание интерфейса базируется на спецификации стандартного протокола FIX версии 5.0 SP2 (<https://www.fixtrading.org/standards/fix-5-0-sp-2>). Предполагается, что пользователь уже знаком с основами этого протокола.

В описании сообщений для каждого поля указаны:

- **Tag** – уникальный идентификатор поля; используется при формировании FIX сообщения;
- **Поле** – имя поля; не используется при формировании FIX сообщения, введено для удобства, в качестве краткого описания;
- **Наличие** – признак, который показывает, должно ли поле обязательно присутствовать в сообщении:
 - **Y** - обязательное поле;
 - **N** - необязательное поле;
 - **C** - обязательно при выполнении условия (указано в графе **Описание**);
- **Описание** - детальное описание поля;
- **Допустимые значения** - дополнительные ограничения.

Символ "*" - признак отличия от стандартного FIX протокола.

4.1. Группы полей

Многие сообщения содержат пересекающийся набор полей. Так, например, каждое сообщение должно содержать поля группы Standard Message Header, несущие служебную информацию.

4.1.1. Standard Message Header

Стандартный заголовок, который должно содержать каждое сообщение.

Tag	Поле	Наличие	Описание	Допустимые значения
34	MsgSeqNum	Y	Порядковый номер сообщения.	
35	MsgType	Y	Определяет тип сообщения.	
49	SenderCompID	Y	Идентификатор отправителя сообщения.	<ul style="list-style-type: none"> • 'MOEX' - Московская биржа • 'ETSC' - Казахская биржа ETS
52	SendingTime	Y	Время отправки сообщения.	
1128	ApplVerID	Y	Идентификатор версии FIX протокола.	'9' (FIX50SP2)

4.2. Сообщения сессионного уровня

4.2.1. Logon (A)

FIX-сообщение, инициирующее установку сессии к сервису TCP Recovery.

Tag	Поле	Наличие	Описание
8	BeginString	Y	Допустимые значения: «FIX.4.4» и «FIXT.1.1».
9	BodyLength	Y	Длина сообщения
35	MsgType	Y	«A»
553	Username	N	Любая строка
554	Password	N	Любая строка
10	Checksum	Y	Контрольная сумма

FAST-сообщение, подтверждающее установку сессии к сервису TCP Recovery.

Tag	Поле	Наличие	Описание
<	Standard Message Header	Y	Тип сообщения 'A'.

4.2.2. Logout (5)

FIX-сообщение, инициирующее завершение сессии с сервисом TCP Recovery.

Tag	Поле	Наличие	Описание
8	BeginString	Y	Допустимые значения: «FIX.4.4» и «FIXT.1.1».
9	BodyLength	Y	Длина сообщения
35	MsgType	Y	«5»
10	Checksum	Y	Контрольная сумма

FAST-сообщение, подтверждающее завершение сессии с сервисом TCP Recovery.

Tag	Поле	Наличие	Описание
<Standard Message Header>		Y	Тип сообщения '5'.
58	Text	N	Причина завершения сессии. В случае отказа в обработке запроса, в поле передается описание ошибки. В случае успешно обработанного запроса передается 'nullValue'.

4.2.3. Heartbeat (0)

Fast Gate отправляет сообщение **HeartBeat** при отсутствии в потоке других сообщений в течении 30 секунд.

Tag	Поле	Наличие	Описание
<Standard Message Header>		Y	Тип сообщения '0'.

4.2.4. Sequence Reset (4)

Tag	Поле	Наличие	Описание
<Standard Message Header>		Y	Тип сообщения '4'.
36	NewSeqNo	Y	Новый номер последовательности.

4.3. Сообщения бизнес-логического уровня

В данном разделе описаны сообщения потоков ОТС-монитора.

Поддерживаются следующие FIX сообщения:

- **Security Definition** – Информация о ценной бумаге.
- **Market Data Request** - Запрос пропущенных данных.
- **Market Data - Snapshot / Full Refresh** – Снапшот с данными.
- **Market Data - Incremental Refresh** – Обновление данных.

4.3.1. Security Definition (d)

Информация о ценной бумаге.

Tag	Поле	Наличие	Описание
<Standard Message Header>		Y	Тип сообщения 'd'
1301	MarketId*	Y	МІС биржи: 'MOEX' - Moscow Exchange
1300	MarketSegmentId*	Y	'Q' - Внебиржевые сделки - ОТС
48	SecurityId	Y	Уникальный числовой код ценной бумаги
22	SecurityIdSource	Y	'8' - Exchange Symbol
55	Symbol	Y	Код ценной бумаги
107	SecurityDesc	N	Наименование эмитента ценной бумаги
455	SecurityAltID*	Y	Символьный код ценной бумаги
456	SecurityAltIDSource*	Y	Класс для SecurityAltID (455): • '4' - ISIN number
461	CFICode	Y	Международный код классификации финансовых инструментов.
870	NoInstrAttrib	Y	=6
=> 871	InstrAttribType	Y	=204

Tag	Поле	Наличие	Описание
=> 872	InstrAttribValue	Y	Государственный регистрационный номер
=> 871	InstrAttribType	Y	=200
=> 872	InstrAttribValue	Y	Общий объем ценных бумаг, выпущенных эмитентом, в штуках
=> 871	InstrAttribType	Y	=205
=> 872	InstrAttribValue	Y	Полное наименование эмитента ценной бумаги либо иного лица, обязанного по ценной бумаге (полное наименование управляющей компании)
=> 871	InstrAttribType	Y	=206
=> 872	InstrAttribValue	Y	Наименование паевого инвестиционного фонда
=> 871	InstrAttribType	Y	=207
=> 872	InstrAttribValue	Y	Вид, категория (тип) ценной бумаги
=> 871	InstrAttribType	Y	=208
=> 872	InstrAttribValue	Y	Вид ценной бумаги
879	UnderlyingQty	N	Номинальная стоимость ценной бумаги
318	UnderlyingCurrency	N	Код валюты номинальной стоимости ценной бумаги
20052	InList	Y	Возможные значения: <ul style="list-style-type: none"> • Y - ценная бумага включена в котировальный список Биржи, • N - ценная бумага не включена в котировальный список Биржи, либо включена в котировальный список Биржи, но является ценной бумагой иностранного эмитента, листинг которой осуществлялся без заключения договора с эмитентом

Символ "*" - признак отличия от стандартного FIX протокола.

4.3.2. Market Data Request (V)

FIX-сообщение для запроса пропущенных данных в сессии к сервису TCP Recovery.

Tag	Поле	Наличие	Описание
8	BeginString	Y	Допустимые значения: <ul style="list-style-type: none"> • FIX.4.4 • FIXT.1.1
9	BodyLength	Y	Длина сообщения.
35	MsgType	Y	"V"
262	MDReqId	Y	Идентификатор запроса.
1182	ApplBegSeqNum	N	Порядковый номер первого запрашиваемого сообщения.
1183	ApplEndSeqNum	N	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 номера сообщения, то ApplEndSeqNum(1183)=0(бесконечность).
10	Checksum	Y	Контрольная сумма.

4.3.3. Market Data - Snapshot / Full Refresh (W)

Снапшот с данными.

Tag	Поле	Наличие	Описание
<Standard Message Header>		Y	Тип сообщения 'W'
893	LastFragment	N	Индикатор, показывающий, является ли сообщение последним в серии сообщений, сформированных по данной ценной бумаге. Допустимые значения: <ul style="list-style-type: none"> • 0 – не последнее сообщение

Tag	Поле	Наличие	Описание
			<ul style="list-style-type: none"> • 1 – последнее сообщение <p>Поле опциональное, при отсутствии поля в сообщении считается, что пакет с сообщением не был фрагментирован.</p>
83	RptSeq	Y	Номер RptSeq последнего инкрементального обновления, вошедшего в текущий снапшот данных для ценной бумаги
911	TotNumReports	Y	Количество сообщений в снапшоте, у которых LastFragment = 1
369	LastMsgSeqNumProcessed	Y	Номер MsgSeqNum последнего сообщения, отправленного в инкрементальный канал на момент генерации текущего снапшота
55	Symbol	Y	Код ценной бумаги
1151	SecurityGroup	Y	= 'ОТС'
268	NoMDEntries	Y	Количество записей MDEntry, входящих в данное сообщение
=>279	MUpdateAction	Y	<p>Тип действия над отчетом о сделке:</p> <ul style="list-style-type: none"> • '0' - New • '1' - Change • '2' - Delete
=>269	MDEntryType	Y	Тип записи: '2' - Trade
=>278	MDEntryID	Y	Регистрационный идентификатор сделки, присвоенный системой ОТС-монитор
=>270	MDEntryPx	Y	Цена одной ценной бумаги
=>15	Currency	Y	Код валюты. Всегда = "RUB"
=>271	MDEntrySize	Y	Количество ценных бумаг
=>272	MDEntryDate	N	Дата заключения сделки.
=>273	MDEntryTime	Y	Время заключения сделки
=>10504	OrderSide	Y	<p>Направленность сделки (покупка 'B' / продажа 'S') Допустимые значения:</p> <ul style="list-style-type: none"> • '1' – покупка (Buy); • '2' – продажа (Sell).
=>120	SettlCurrency	Y	Валюта денежного обязательства
=>461	CFICode	Y	Код CFI ценной бумаги, с которой совершена сделка
=>1020	TradeVolume	Y	Объем сделки, руб.

Символ '*' - признак отличия от стандартного FIX протокола.

4.3.4. Market Data - Incremental Refresh (X)

Обновление данных.

Tag	Поле	Наличие	Описание
	<Standard Message Header>	Y	Тип сообщения 'X'
893	LastFragment	N	<p>Индикатор, показывающий, является ли сообщение последним в серии сообщений, сформированных по данной ценной бумаге.</p> <p>Допустимые значения:</p> <ul style="list-style-type: none"> • '0' – не последнее сообщение • '1' – последнее сообщение <p>Поле опциональное, при отсутствии поля в сообщении считается, что пакет с сообщением не был фрагментирован</p>
268	NoMDEntries	Y	Количество записей MDEntry, входящих в данное сообщение
=>279	MUpdateAction	Y	<p>Тип действия над отчетом о сделке:</p> <ul style="list-style-type: none"> • '0' - New • '1' - Change

Tag	Поле	Наличие	Описание
			• '2' - Delete
=>269	MDEntryType	Y	Тип записи: '2' - Trade
=>55	Symbol	Y	Код ценной бумаги
=>1151	SecurityGroup	Y	= 'ОТС'
=>83	RptSeq	Y	Порядковый номер инкрементального обновления
=>278	MDEntryID	Y	Регистрационный идентификатор сделки удаленной или измененной. При изменениях отчета о сделке у него остаётся прежний Регистрационный идентификатор, присвоенный системой ОТС-монитор при добавлении отчета о сделке.
=>270	MDEntryPx	Y	Цена одной ценной бумаги
=>271	MDEntrySize	Y	Количество ценных бумаг
=>272	MDEntryDate	N	Дата заключения сделки
=>273	MDEntryTime	Y	Время заключения сделки
=>15	Currency	Y	Код валюты. Всегда = 'RUB'
=>20018	Revision	N	Служебное поле подсистемы репликации.
=>10504	OrderSide	Y	Направленность сделки (покупка 'B' / продажа 'S') Допустимые значения: <ul style="list-style-type: none"> • '1' – покупка (Buy); • '2' – продажа (Sell).
=>120	SettlCurrency	Y	Валюта денежного обязательства
=>461	CFICode	Y	Код CFI ценной бумаги, с которой совершена сделка
=>1020	TradeVolume	Y	Объем сделки, руб.

Символ '*' - признак отличия от стандартного FIX протокола.

5. Ограничения сервиса TCP Recovery (Historical Replay)

Следующие ограничения используются для сервиса TCP Recovery (Historical Replay) для потоков OTC-TRADES, ORDERS-LOG с целью снижения нагрузки на него:

Параметр	Значение	Описание
Максимальное количество активных соединений на рынок, на инстанс, на IP адрес	2	Вы можете установить не более указанного количества активных TCP соединений с одного IP адреса. Попытка установить большее количество соединений будет отклонена.
Максимальное количество соединений за день на рынок, на инстанс, на IP адрес	1000	Вы можете установить не более указанного количества TCP соединений с одного IP адреса за день. Дополнительные попытки подключения будут отклонены.
Максимальное количество FAST сообщений в запросе	1000	Запрос отклоняется, если количество запрошенных FAST сообщений превышает указанное значение
Время ожидания запроса Market Data Request (V)	1	Соединение прекращается, если запрос Market Data Request (V) не получен в течение указанного количества секунд после сообщения Logon (A). Сеанс TCP завершается, если от клиента не получено подтверждающего сообщения Logout (5) в течение указанного количества секунд после сообщения Logout (5) с сервера на клиента.