FAST protocol specification

Version 9.12.0

Moscow 2026

Table of Contents

1. Introduction 16
1.1. Document purpose 16
1.2. Fast Gate — Basic information 16
1.2.1. Data streaming approach 16
1.2.2. Incremental messages 16
1.2.3. FIX format 16
1.2.4. Encoding in the FAST format 16
1.2.5. Data receicing via Multicast 17
1.2.6. Data recovery 17
2. Scenarios of client interactions with Market Data Multicast 18
2.1. Connect client 18
2.1.1. Empty snapshot in FO-TRADES stream 18
2.2. Incremental Feeds A and B Arbitration 18
3. System functionality 20
3.1. System architecture 20
3.1.1. Main streams (UDP) 20
3.1.2. Recovery streams (UDP) 21
3.1.3. Instrument Definitions streams (UDP) 21
3.1.4. Messages in streams 21
3.1.5. Sessions for requesting missing messages (TCP) 24
3.2. FAST format — details 24
3.2.1. Stop bit encoding 24
3.2.2. Implicit tagging 25
3.2.3. Fields encoding options 25
3.2.4. FAST-template 25
3.2.5. Decoding overview 25
3.2.6. Message fragmentation 25
3.2.7. Data types 26
3.3. Missing data recovery 26
3.3.1. Recovery missing data using Recovery streams (UDP) 26
3.3.2. Recovering missing data using TCP-connection 27
3.4. Message sequence reset 27
4. FIX protocol message specifications 28
4.1. Field groups 28
4.1.1. Standard Message Header 28
4.2. Session layer messages 28
4.2.1. Logon (A) 28
4.2.2. Logout (5) 28
4.2.3. Heartbeat (0) 29
4.2.4. Sequence Reset (4) 29
4.3. Business logic layer messages 29
4.3.1. Security Definition (d) 29
4.3.2. Security Status (f) 35
4.3.3. Trading Session Status (h) 36
4.3.4. Security Definition Update Report (BP) 37
4.3.5. News (B) 37
4.3.6. Market Data Request (V) 38
4.3.7. Market Data - Snapshot / Full Refresh (W) 38
4.3.8. Market Data - Incremental Refresh (X) 40
4.3.9. DiscreteAuction (U1) 43
4.3.10. SecurityMassStatus (CO) 43
4.3.11. SecurityGroupStatus (f) 44
4.4. Market Fundamentals 45
4.5. The message Empty book (MDEntryType = J) in the Aggregate Book streams 46
4.6. Sample representation of Aggregate Book incremental refresh processing 47
4.7. Instrument IDs 49
4.8. Group statuses of instruments 49
4.8.1. Algorithm for determining trading statuses of instruments by group statuses 49
5. Stream of anonymous orders and trades 52
5.1. Application architecture 52
5.1.1. Streams 52
5.1.2. Messages fragmentation 52
5.1.3. Instrument IDs 52
5.2. Message templates 52
5.2.1. OrdersLogMessage 52
5.2.2. BookMessage 53
5.3. Messages Market Data - Incremental Refresh (X) 53
5.3.1. Adding a new order 53
5.3.2. Deleting an order 54

5.3.3. Partial order matching 55
5.3.4. Full order matching 56
5.3.5. Technical trades 57
5.3.6. Cleaning up active orders pool (specified trading session) 58
5.3.7. Cleaning up active orders pool (all trading sessions) 58
5.4. Messages Market Data - Snapshot / Full Refresh 59
5.4.1. Active orders snapshot 59
5.4.2. Empty active orders snapshot - the message Empty book (MDEntryType = J) 60
5.5. Active orders book 60
6. Synthetic matching 61
6.1. Sample representation of Aggregate Book incremental refresh processing for synthetic matching 61
7. TCP Recovery (Historical Replay) service limitations 65

History of changes

Date	Version	Changes
01.10.2026	9.12.0	1. In the **SecurityDefinition (d)** message, 'SettlPriceOpen' field is deprecated and will be removed in future releases. It is necessary to switch to using the 'SettlPrice' field. 2. A clarification has been added to section **1.1. Document purpose**. 3. Support for fractional lots has been added for instruments. Clients must now expect and support fractional values in the **ContractMultiplier** field of the **SecurityDefinition (d)** message. The description of the **ContractMultiplier** field in the **SecurityDefinition (d)** message has been updated.
10.07.2026	9.9.0	1. In **Market Data - Snapshot / Full Refresh (W)** and **Market Data - Incremental Refresh (X)** messages, changes have been made to **MDEntryType** field: • added new value **'q'** - Current indicative funding rate for the instrument (for one-day futures contracts with auto-prolongation). In points. • added new value **'r'** - Funding rate when the contract price deviates from the underlying asset price beyond the specified level on the contract (for one-day futures contracts with auto-prolongation). In points. 2. The specification versioning has been brought into line with the versions of the Spectra trading system.
07.04.2026	1.30.2	1. In **4.8.1. Algorithm for determining trading statuses of instruments by group statuses**, additions have been made to Section '5', Subsection 'a'.
27.03.2026	1.30.1	1. New 'ClearingSettlPrice' field in **SecurityDefinition (d)** .
16.03.2026	1.30.0	1. In the **SecurityDefinition (d)** message: • The fields MaturityDate and MaturityTime were removed • Added fields: • 'HighLimitPxWeekend' - Upper price limit for option premium during Additional Weekend Trading Session. • 'LowLimitPxWeekend' - Lower price limit for option premium during Additional Weekend Trading Session. • Description of the field 'HighLimitPx' was updated. • Description of the field 'LowLimitPx' was updated. • Changed message template ID from '40' to '47'. 2. In the **Security Status (f)** message : • Added fields: • 'HighLimitPxWeekend' - Upper price limit for option premium during Additional Weekend Trading Session. • 'LowLimitPxWeekend' - Lower price limit for option premium during Additional Weekend Trading Session. • Description of the field 'HighLimitPx' was updated. • Description of the field 'LowLimitPx' was updated. • Changed message template ID from '5' to '48'.
11.03.2026	1.29.3	1. Section '3.4.Message sequence reset' was changed (see sec. 3.4).
04.12.2025	1.29.2	1. All references to SKRIN news have been removed from the specification.
19.11.2025	1.29.1	1. In section **Fast Gate — Basic information**, the line **MOEX Board system** was removed. 2. In sections **FAST-template** and **Message sequence reset**, the substring **MOEX Board** was removed. 3. In section **Security Definition (d)** in table **Moscow Exchange MarketSegmentID values**, the following market segment lines were removed: • SBOARD, ESXXXX;

Date	Version	Changes
		• SBOARDC, ESXXXX; • SBOARDB, ESXXXX
15.08.2025	1.29.0	Changes applied: • In the TradingSessionStatus message (msg id=46): • Message ID was changed from 41 to 46. • The fields TradSesIntermClearingStartTime and TradSesIntermClearingEndTime were removed • Added fields: • SettlSessBegin - Date and time of the start of the settlement session in the format: YYYYMMDDHHMMSSsss • ClrSessBegin - Date and time of the start of the clearing session in the format: YYYYMMDDHHMMSSsss • In the SecurityDefinition message (d) of the Flags field, the purpose of the flag '0x80000' was redefined. It now indicates: Scheduled execution or Execution in Clearing session: • '0' - Clearing session • '1' - Scheduled execution • Description of the field group EventType=7, EventDate, EventTime was updated. • Description of the field group EventType=101, EventDate, EventTime was updated. • Description of the pair of fields MaturityDate, MaturityTime was updated.
05.06.2025	1.28.0	1. The type of field '**SecurityTradingStatus**' has been changed from int32 to uInt32 in message template **SecurityGroupStatus (f)**. Changed message template ID from '39' to '45'.
11.04.2025	1.27.0	1. New fields were added to **SecurityDefinition (d)** message: • **'TradeModelID'** - Trade mode id • **'GroupMask'** - Bit-mask of groups • **'SectionID'** - Identifier of section • **'BaseContractID'** - Identifier of base contract • **'TradePeriodAccess'** - Flags of trading access in the period of session (bit mask) Changed message template ID from '38' to '40'. 2. In the **SecurityDefinition (d)** message, of the **'Flags**' field has stopped broadcasting flags: • '0x1' - Trading in the evening or morning session • '0x40' - Trading in the day session 3. New message **SecurityGroupStatus (f)** - Group status of instruments. 4. New section **4.8. Group statuses of instruments** was added. 5. In the **Trading Session Status (h)** and **Security Definition (d)** messages, new value in the '**TradingSessionID**' field: • '100' - Weekend session. 6. New '**TradePeriodID**' field in **Trading Session Status (h)**. Changed message template ID from '8' to '41'. 7. New '**TradePeriodID**' field in **DiscreteAuction (U1)**. Changed message template ID from '26' to '42'.

Date	Version	Changes
		8. In **Market Data - Snapshot / Full Refresh (W)** and **Market Data - Incremental Refresh (X)** messages, changes have been made to **MDEntryType** field: • changed description of value **'u'** - Funding rate when the contract price deviates from the underlying asset price beyond the specified level on the contract (for one-day futures contracts with auto-prolongation) • changed description of value **'z'** - Dividend amendment for one-day futures contracts with auto-prolongation on the index or the share • added new value **'U'** - Indicative funding rate (for one-day futures contracts with auto-prolongation) • added new value **'Z'** - Current dividend amendment for one-day futures contracts with auto-prolongation on the index or the share • added new value **'m'** - Current market price
29.08.2024	1.24.1	1. New flag "0x1000000000000000 - Trade by Book-or-cancel order (Passive only)" in the '**MDFlags**' field of **Market Data - Incremental Refresh (X)** messages - see sec. 5.3.3 and sec. 5.3.4.
05.07.2024	1.24.0	1. The CFICode description for options was added in the Table 1. **Moscow Exchange MarketSegmentID values** in the **Security Definition (d)** message: • **'OCEFCS'** - European cash-settled futures option Call • **'OPEFCS'** - European cash-settled futures option Put
04.06.2024	1.23.1	1. Section "Message sequence reset" was changed (see sec. 3.4).
01.04.2024	1.23.0	1. The message template ID was changed: • **SecurityDefinition** from '36' to '38' 2. New instruments status was added in the SecurityTradingStatus field in the **Security Definition (d)** and **Security Status (f)** messages: • **'123'** - Session for this instrument is running. You can put orders to close position for this instrument. • **'124'** - The opening auction for this instrument is running. You can put orders to close position for this instrument. 3. The CFICode description for futures was added in the Table 1. **Moscow Exchange MarketSegmentID values** in the **Security Definition (d)**: • **FFSCSX** - Daily futures contract on stock-equity with automatic prolongation(Standardized Financial Future on Stock-Equity, Cash delivery) 4. New value was added in the Table 4. **Decoding characters from the 'CFICode' field for futures and calendar spreads** in the **Security Definition (d)** message: • **S** = Stock-Equities 5. New description for SettlementPrice field was added in the **Security Definition (d)** message: • **SettlPrice** - Settlement price of future or calendar spread at the end of last clearing session. Settlement price (theoretical price of the option) after the last clearing session. 6. The CFICode description for options was added in the Table 1. **Moscow Exchange MarketSegmentID values** in the **Security Definition (d)** message: • **'OPETCS'** - European cash-settled commodity option Put; • **'OCETCS'** - European cash-settled commodity option Call. 7. New underlying asset was added in the Table 3. **Decoding characters from the 'CFICode' field for options** in the **Security Definition (d)** message: • **T** = Commodities
16.01.2024	1.22.0	*Orders with TAS spread are not available in the current version.*

Date	Version	Changes
		Changes applied: 1. In the 'SecurityDefinition' message, new value in the 'UnderlyingFutureID' field: Futures identifier (SecurityID) for TAS (Trade at Settlement) mode. 2. New value was added in the 'TrdType' field: '1005' - Trade Settlement TAS. 3. New flag was added in the 'MDFlags' field: 0x10000 - 'Trade Settlement TAS'.
11.11.2023	1.21.1	1. New section **4.7. Instrument IDs** was added.
05.10.2023	1.21.0	1. New values have been added to the **Security Definition (d)** message for the 'CFICode' field: • 'OPEICS' - European cash-settled index option Put; • 'OCEICS' - European cash-settled index option Cull. 2. New underlying asset has been added to the **Decoding characters from the 'CFICode' field for options**: • I = Index 3. New parameter has been added to the **Adding a new order** and **Deleting an order** in the MDFlags field: • 0x80000000 - Negotiated order matched by reference 4. New parameter has been added to the **Partial order matching** and **Full order matching** in the MDFlags field: • 0x80000000 - Negotiated trade matched by reference 5. Translation of index identifiers and indicative exchange rate identifiers in the SecurityID field in the **DefaultIncrementalRefreshMessage** and **DefaultSnapshotMessage** messages in the INDEX stream has been stopped. 'nullValue' will be broadcast in the field in all messages in the INDEX stream. It is necessary to use the value of the Symbol field as a unique identifier of indices and indicative exchange rates.
25.07.2023	1.20.0	Changes applied: 1. New fields were added to **SecurityDefinition (d)** message: • **InterestRateRiskUp** - Interest risk variable rate on rate up scenario • **InterestRateRiskDown** - Interest risk variable rate on rate down scenario • **RiskFreeRate2** - Risk free interest rate FX2 of the currency pair FX2/FX1 (for currency premium options); dividend yield rate 'q' (for index premium options). • **InterestRate2RiskUp** - Interest rate risk up scenario for RiskFreeRate2 • **InterestRate2RiskDown** - Interest rate risk down scenario for RiskFreeRate2 2. In **SecurityDefinition (d)** message, change for **CFICode** field: • CFI code 'JFTXCC' - Daily futures contract with automatic prolongation (Contract for difference, Cash Settled)" replaced with "'FFCCSX' - Daily futures contract on currency with automatic prolongation (Standardized Financial Future on Currency, Cash delivery) • new CFI code: 'FFICSX' - Daily futures contract on index with automatic prolongation (Standardized Financial Future on Index, Cash delivery) 3. In **SecurityDefinition (d)** message, new value for **SecurityTradingStatus** field: • '20' - Unknown or Invalid status 4. In **Market Data - Snapshot / Full Refresh (W)** and **Market Data - Incremental Refresh (X)** messages, new value for **MDEntryType** field: • **z** - Dividend amendment for perpetual futures on the index or the stock-equities 5. New message **SecurityMassStatus** - Packaged message for accelerated broadcasts of trading statuses of instruments.

Date	Version	Changes
		6. In **DiscreteAuction (MessageType = U1)** message **NoUnderlyings** and **UnderlyingSymbol** fields are optional now. 7. In section **4.3.1. Security Definition (d)**, new table was added - **Decoding characters from the 'CFICode' field for futures and calendar spreads**.
27.06.2023	1.19.2	1. New section in the specification "Empty snapshot in FO-TRADES stream" (see sec. 2.1.1).
27.04.2023	1.19.1	1. New CFI-codes values for currency options (see sec. 4.3.1): • 'OCECCS' - European cash-settled currency option Call • 'OPECCS' - European cash-settled currency option Put
15.04.2023	1.19	1. In **Security Definition (d)** and **Security Status (f)** messages for **SecurityTradingStatus** attribute (see sec. 4.3.1 and sec. 4.3.2): • **'122'** - new value - Session for the instrument is suspended. You can't place orders, but you can cancel orders on this instrument • new description for value **'2'** - Session for all instruments is suspended. You can't place orders, but you can cancel orders. 2. In **SecurityDefinition (d)** message (see sec. 4.3.1): • for **EventType** attribute, value '**100**' is no longer broadcast - Instrument exercise start date • allowed values were changed for **NoEvents** attribute: • '1' - for futures • '2' - for options
15.03.2023	1.18.1	1. Description of Fast Gate OTC-monitor moved from the specification to a separate document - "Moscow Exchange FAST protocol specification for OTC trades report system (OTC-monitor)": OTCMonitor_fastgate_en.pdf and OTCMonitor_fastgate_ru.pdf (see documentation for the OTC-monitor).
19.01.2023	1.18.0	1. The message templates ID were changed: • **OrdersLogMessage** from '14' to '29' • **BookMessage** from '15' to '30' • **DefaultSnapshotMessage** from '25' to '32' • **DefaultIncrementalRefreshMessage** from '24' to '31' 2. In the stream of anonymous orders and trades, in **Market Data - Incremental Refresh (X)** message: • new field **MDFlags2** - Bitmask extension (in addition to the MDFlags field) • new value for **MDFlags** field: 0x2000 - Sign of canceling a passive cross-order - see sec. 5.3.2 3. New field in **Market Data - Snapshot / Full Refresh (W)** and **Market Data - Incremental Refresh (X)** messages: • **TradingSessionSubID** - Trading period: '**S**' – opening auction; '**N**' – normal trading period
15.11.2022	1.17.1	1. New row for **Opening auction price** was added to **Market Fundamentals** tab.
10.10.2022	1.17.0	1. New flag "0x1000000000000000 - Book-or-cancel order (Passive only)" in the '**MDFlags**' field of **Market Data - Incremental Refresh (X)** message - see sec. 5.3.1 and sec. 5.3.2. 2. In the **Market Data - Snapshot / Full Refresh (W)** and **Market Data - Incremental Refresh (X)** messages, new value for the **MDEntryType** field: '**W**' - Opening auction price. 3. New message **DiscreteAuction (MessageType = U1)** transmitted in FUT-INFO stream - sec. 4.3.9 and sec. 3.1.4. 4. In the **SecurityDefinition** and **SecurityStatus** messages, new values in the **SecurityTradingStatus** field:

Date	Version	Changes
		• '119' - the opening auction for the instrument started, you can place and delete orders for this instrument • '121' - the opening auction for this instrument completed 5. In the **Market Data - Incremental Refresh (X)** message, new value in the **MDFlags** field: '0x4000000000000000' - Sign of an order/trade during a discrete auction.
22.08.2022	1.16.1	1. Section 4.5 'Empty book message (MDEntryType = J)' was detailed - see sec. 4.5.
05.07.2022	1.16.0	1. In 'Market Data - Snapshot / Full Refresh (W)' (sec. 4.3.7) and 'Market Data - Incremental Refresh (X)' (sec. 4.3.8) messages, new value for 'TrdType' field: '1004' - exiting a daily futures contract with automatic prolongation. 2. New values of **MDFlags** field for technical trades (sec. 5.3.5): • 0x10000000000000 - The technical trade as a result of exiting a daily futures contract with automatic prolongation (based on the submitted requests) • 0x400000000000000 - The technical trade as a result of forced exiting a daily futures contract with automatic prolongation (realization of unsatisfied demand) • 0x800000000000000 - The technical trade with linked instrument as a result of exiting a daily futures contract with automatic prolongation 3. Off-set technical trades of exiting a perpetual future are marked with TRD-Type[828]='1004'. These trades do not affect instruments' statistics, including last, min, max, close.
29.03.2022	1.15.0	1. The **SecurityDefinition** message template ID was changed from '21' to '23'. 2. The **DefaultIncrementalRefreshMessage** message template ID was changed from '22' to '24'. 3. The **DefaultSnapshotMessage** message template ID was changed from '20' to '25'. 4. New fields were added to **SecurityDefinition** message - sec. 4.3.1: • **ValuationMethod** - Method of setting margin requirements for the option: "FUT" - futures-style; "EQTY" - equity-style • **RiskFreeRate** - Risk-free interest rate • **FixedSpotDiscount** - Sum of discounted values of declared cash flows • **ProjectedSpotDiscount** - Sum of discounted values of forecasted cash flows • **SettlCurrency** - Settlement currency • **UnderlyingBoard** - SECBOARD trading mode ID from ASTS gateway The values in the above fields are transmitted only in option definitions in the OPT-INFO stream. • **NegativePrices** - Sign of limiting negative prices. Values in the field are transmitted only in instrument definitions in the FUT-INFO and OPT-INFO streams. • **DerivativeContractMultiplier** - Coefficient indicating the volume of the underlying asset in the contract quote and strikes of option series. 5. In **SecurityDefinition** message, change for **CFICode** field (sec. 4.3.1): • value 'OCEFPS' and 'OPEFPS' were removed; • description for values 'OCAFPS' and 'OPAFPS' was changed; • new values were added: • OCESCS - European cash-settled Call equity option • OPESCS - European cash-settled Put equity option 6. In **SecurityDefinition** message, transmitted values for the **UnderlyingSymbol** field was changed - sec. 4.3.1.

Date	Version	Changes
		7. In the table for **SecurityDefinition** message, description for **UnderlyingFutureID** field was changed - sec. 4.3.1. 8. In **OrdersLogMessage** message new value for **MDFlags** field: 0x200000000000000 - Passive synthetic order (sec. 5.3.1). 9. New field was added to **DefaultIncrementalRefreshMessage** and **DefaultSnapshotMessage** message template: **IndexCrossRate** - Foreign exchange rate used in calculating the index value (sec. 4.3.7 and sec. 4.3.8). 10 In **SecurityDefinition** message, new value for options is transmitted in the **MinPriceIncrementAmountCurr** field - Value of the minimum increment in foreign currency.
25.03.2022	1.14.0	1. New CFI Code for Futures in Moscow Exchange: 'JFTXCC' - Daily futures contract with automatic prolongation (CFD - Contract for difference, Cash Settled) - sec. 4.3.1. 2. In 'Market Data - Snapshot / Full Refresh (W)' (sec. 4.3.7) and 'Market Data - Incremental Refresh (X)' (sec. 4.3.8) messages, new value for 'MDEntryType' field: 'u' - Swap rate - rate for transferring a position on a CFD futures to the next trading session.
26.08.2021	1.13.1	1. The new section **3.1.4. Messages in streams** was added - sec. 3.1.4. This section describes which messages are transmitted in each data stream.
20.07.2021	1.13.0	1. Broadcasting of **TradingSessionStatus (h)** messages was added to FUT-INFO and OPT-INFO streams - sec. 4.3.3. 2. The **OpenInterest** field (open positions after the trade) was added to **DefaultIncrementalRefreshMessage** message template - sec. 4.3.8. Message identifier was changed from "19" to "22". 3. The **CFICode** field values are detailed in **SecurityDefinition** message - sec. 4.3.1. 4. From version 6.12 in the SPECTRA system, the indicative quotes service for the Derivatives Market (IQS) will no longer be supported. The FAST-IQS service will be disabled in SPECTRA 6.12.
20.03.2021	1.12.0	1. The deprecated streams FUT-BOOK-1, FUT-BOOK-5, FUT-BOOK-20, FUT-BOOK-50, OPT-BOOK-1, OPT-BOOK-5, OPT-BOOK-20, OPT-BOOK-50, FUT-TRADES, OPT-TRADES, IQS-FUT-BOOK-1, IQS-FUT-BOOK-50, IQS-OPT-BOOK-1, IQS-OPT-BOOK-50, IQS-FUT-TRADES, IQS-OPT-TRADES were replaced with new merged streams FO-TRADES, FO-BOOK-1, FO-BOOK-5, FO-BOOK-20, FO-BOOK-50, IQS-TRADES, IQS-BOOK-1, IQS-BOOK-50. Since version Spectra 6.9 you will no be able to work with deprecated streams. 2. The group of fields with **'EventType'=5** (Instrument trading start date) is no longer broadcast in the **SecurityDefinition (d)** message for futures, calendar spreads and options. 3. In FUT-INFO, OPT-INFO, IQS-FUT-INFO, IQS-OPT-INFO streams in **SecurityDefinition (d)** messages, the **MarketSegmentID** field value was replaced with "**D**". 4. The description of market segments was changed (field **MarketSegmentId**) in the following sections: • **Trading Session Status (h)** - sec. 4.3.3 • **News (B)** - sec. 4.3.5 • **Security Definition (d) -** sec. 4.3.1 5. The description of **20008 Flags** field was changed in **SecurityDefinition (d)** message - sec. 4.3.1.
20.09.2020	1.11.0	1. Streams FUT-BOOK-1, FUT-BOOK-5, FUT-BOOK-20, FUT-BOOK-50, OPT-BOOK-1, OPT-BOOK-5, OPT-BOOK-20, OPT-BOOK-50, FUT-TRADES, OPT-TRADES, IQS-FUT-BOOK-1, IQS-FUT-BOOK-50, IQS-OPT-BOOK-1, IQS-OPT-BOOK-50, IQS-FUT-TRADES, IQS-OPT-TRADES are deprecated and replaced in the specification with new merged streams FO-TRADES, FO-BOOK-1, FO-BOOK-5, FO-BOOK-20, FO-BOOK-50, IQS-TRADES, IQS-BOOK-1, IQS-BOOK-50. **It is recommended to switch to using the new merged streams**. 2. In the current protocol version, the **MDFeedTypes** field in the **SecurityDefinition** message contains the names of the old streams to maintain backward compatibility. In the next versions of the FAST protocol, all merged streams will be added to **MDFeedTypes** and all deprecated streams will be removed from **MDFeedTypes** in the **SecurityDefinition** messages.

Date	Version	Changes
		3. In the FUT-INFO, OPT-INFO, IQS-FUT-INFO, IQS-OPT-INFO streams, in the **Security-Definition** messages, the **MarketSegmentID** field value will be replaced with "**D**". In the xml-descriptions of the FUT-INFO, OPT-INFO, IQS-FUT-INFO, IQS-OPT-INFO stream configurations, the value of the **marketID** attribute will be replaced with value "**D**". A new code "**D**" was added to the Moscow Exchange market segment code table - sec. 4.3.1. 4. New optional fields were added to **Security Definition (d)** message - sec. 4.3.1: • **MinPriceIncrementAmountCurr** - value of the minimum increment in USD • **SettlPriceOpen** - settlement price at the start of the session.
03.09.2020	1.10.1	1. New section **Synthetic matching** was added - sec. 6.
10.04.2020	1.10.0	1. Description of **MDFlags** field was changed in the message **Market Data - Snapshot / Full Refresh** for active orders snapshot - sec. 5.4.1. 2. List of possible values for **MDFlags** field was changed. 3. Section **7. TCP Recovery (Historical Replay) service limitations** was changed. 4. New value **0x200000000000 - Synthetic ord-er** was added in the description of **MD-Flags** field in the **Market Data - Incremental Refresh (X)** messages in stream of anonymous orders - sec. 5.3. 5. Changes of the **Market Data - Snapshot / Full Refresh (W)** and **Market Data - Incremental Refresh (X)** messages: • new **MDEntrySyntheticSize** field (Synthetic liquidity volume) was added; • description of **MDEntryType** field was changed. 6. Section **Market Fundamentals** was changed- sec. 4.4.
05.12.2019	1.9.0	1. Changed description for field **Flags** in the message **Security Definition (d)** - see sec. 4.3.1. 2. Added new value ('2002' - Trade originated in RFS) for parameter **TrdType** in the following tables: • Market Data - Snapshot / Full Refresh (W) - see sec. 4.3.7 • Market Data - Incremental Refresh (X) - see sec. 4.3.8 3. Changed description of field **MDFlags** in tables: • Adding a new order (sec. 5.3.1), • Deleting an order (sec. 5.3.2), • Partial order matching (sec. 5.3.3), • Full order matching (sec. 5.3.4),
02.08.2019	1.8.0	1. Deleted section **Feature of message processing in the stream INDEXES**. 2. Changed description for field **MDFlags** in the messages **Market Data - Incremental Refresh (X)** - see sec. 5.3. 3. Section **Message templates** - **OrdersBook** renamed to **BookMessage** and **OrdersLog** renamed to **OrdersLogMessage** - see sec. 5.2. 4. Section **Anonymous quotes stream** - **LogMessage** renamed to **QuotesLogMessage**. 5. Section **Market Data - Snapshot / Full Refresh (W)** - new description for field **TotNum-Reports** - see sec. 4.3.7. 6. Deleted section **5.3.6 - Trade types**. 7. Section **Status values in field MDFlags** - new values for field: • 0x20000000000 – The active side in the trade. The order that led to the trade when added to the order- book

Date	Version	Changes
		• 0x40000000000 – The passive side in the trade. The order from the order-book involved in the trade
29.07.2019	1.7.2	1. Added new section **Feature of message processing in the stream INDEXES**.
27.06.2019	1.7.1	1. Added new value ('2001' - Trade originated in IQS) for parameter **TrdType** in the following tables: • Market Data - Snapshot / Full Refresh (W) - see sec. 4.3.7 • Market Data - Incremental Refresh (X) - see sec. 4.3.8 • Indicative trades and general information.
05.12.2018	1.7.0	1. Fields **LowLimitPx**, **HighLimitPx** are discontinued for Option instruments, and now available for Futures and calendar spreads only. The changes mentioned are applied to the following messages: • **Security Definition (d)** - see sec. 4.3.1. • **Security Status (f)** - see sec. 4.3.2. 2. As data from the Ukrainian Exchange are no more transmitted, all information regarding the Ukrainian Exchange have been removed from the current document.
22.12.2017	1.4.4	1. For table **Security Definition (d)** added field **Flags** - see sec. 4.3.1. 2. Added new section **Indicative Quote System of Derivatives market**.
22.08.2017	1.4.3	1. Updated sections **Fast Gate — Basic information** (раздел 1.2), **Message sequence reset** (sec. 3.4) and **FIX protocol message specifications** (sec. 4).
23.06.2017	1.4.2	1. Added new section **Message sequence reset** (sec. 3.4) .
09.06.2017	1.4.1	1. Updated section **TCP Recovery (Historical Replay) service limitations** (sec. 7).
18.05.2017	1.4.0	1. Added new section **TCP Recovery (Historical Replay) service limitations** (sec. 7) . 2. For table **Security Definition (d)** added field **UnderlyingFutureID** - sec. 4.3.1. 3. For sections **Security Definition (d)** (sec. 4.3.1) and **Security Status (f)** (sec. 4.3.2): • updated description of field **SecurityTradingStatus**: • '19' - Not traded on this market
20.04.2017	1.3.2	1. Added description on FAST message into sections **Logon (A)** (sec. 4.2.1) and **Logout (5)** (sec. 4.2.2). 2. Added description on new fields into section **Market Data Request (V)** (sec. 4.3.6).
16.01.2017	1.3.1	1. Changed description of field **MDEntryTradeType** for sections **Market Data - Snapshot / Full Refresh (W)** (sec. 4.3.7) and **Market Data - Incremental Refresh (X)** (sec. 4.3.8) .
26.10.2016	1.3.0	1. Changed names of data types in FAST-message template (see sec. 3.2.7): • UTC Time Only to **UTCTimeOnly** • UTC Date Only to **UTCDateOnly** • UTC Timestamp to **UTCTimestamp** 2. Changed description of type **UTCTimeOnly** - see sec. 3.2.7. 3. Updated templates for messages **OrdersLog** and **OrdersBook** - see sec. 5.2.1 and sec. 5.2.2. 4. Added new fields (31) **LastPx** and (32) **LastQty** into tables: • Adding a new order (sec. 5.3.1), • Deleting an order (sec. 5.3.2), • Partial order matching (sec. 5.3.3), • Full order matching (sec. 5.3.4), • Technical trades (sec. 5.3.5), • Cleaning up active orders pool (specified trading session) (sec. 5.3.6),

Date	Version	Changes
		• Cleaning up active orders pool (all trading sessions) (sec. 5.3.7). 5. Changed description of field **MDEntrySize** in tables: • Adding a new order (sec. 5.3.1), • Technical trades (sec. 5.3.5). 6. Changed description of field **MDEntryPx** in tables: • Partial order matching (sec. 5.3.3), • Full order matching (sec. 5.3.4), • Technical trades (sec. 5.3.5). 7. Changed description of message **Empty book** (sec. 4.5). 8. Changed description of message **SecurityID** of table **Security Definition (d)** (sec. 4.3.1). 9. Updated section **Instrument IDs** (sec. 5.1.3). 10. For sections **Market Data - Snapshot / Full Refresh (W)** (sec. 4.3.7) and **Market Data - Incremental Refresh (X)** (sec. 4.3.8): • for field **MDEntryType**, deleted types: • 'm' - Market Price • 'p' - Official Current Price; • new values added for field **MDEntryTradeType**: • for <repo type> for Ukrainian Exchange: **BR18** - Repo near leg and **BR19** - repo far leg; • updated description of field **MDEntryDate**; • changed description of field **MDEntryID**. 11. For table **Standard Message Header** updated description of field **SenderCompID** - see sec. 4.1.1. 12. Updated description of message **Heartbeat** - see sec. 4.2.3. 13. For table **News (B)**, added field **LastFragment** - see sec. 4.3.5. 14. For table **Active orders snapshot**, deleted description of bit: *0x1000 End-of-transaction bit* of field **MDFlags** - see sec. 5.4.1. 15. Updated section **Recovering missing data using TCP-connection** - see sec. 3.3.2.
04.10.2016	1.2.4	1. Added field ‘MDEntrySize’ in section 'Deleting an order' (sec. 5.3.2),
21.03.2016	1.2.3	1. The name of document was changed to 'FAST protocol specification'. 2. The section 'Stream of anonymous orders' renamed to 'Stream of anonymous orders and trades' - sec. 5. 3. The section 'Trade on a leg of multi leg instrument' renamed to 'Technical trades' - sec. 5.3.6. 4. Added section 'Trade types' - . 5. Description of field ‘MDEntryType’ has been changed in sections: • Partial order matching (sec. 5.3.3), • Full order matching (sec. 5.3.4), • Technical trades (sec. 5.3.5) to: Record type:

Date	Version	Changes
		• **0** - Bid • **1** - Ask 6. Deleted field ‘MDEntrySize’ in section 'Deleting an order' (sec. 5.3.2). 7. Added field ‘Revision’ in sections: • Adding a new order (sec. 5.3.1), • Deleting an order (sec. 5.3.2), • Partial order matching (sec. 5.3.3), • Full order matching (sec. 5.3.4), • Technical trades (sec. 5.3.5), • Cleaning up active orders pool (specified trading session) (sec. 5.3.6), • Cleaning up active orders pool (all trading sessions) (sec. 5.3.7), 8. Added field ‘Revision’ in message template 'OrdersLog' - sec. 5.2.1.
03.02.2016	1.2.2	1. Section ‘4.3.1. Security Definition (d)’ – added some new info in field ‘LegRatioQty’ description. 2. Added section ‘5.3.5. Trade on a leg of multi leg instrument’. 3. Description of field ‘MDFlags’ has been changed in sections ‘5.3.3 Partial order matching’ and ‘5.3.4. Full order matching’.
11.01.2016	1.2.2	1. Added description for stream of anonymous orders. - see sec. 5.
02.12.2015	1.2.1	1. Added description of a method for obtaining authentic instrument ID from field 'SecurityID'.
21.08.2015	1.2.0	1. The channel INDEX no more transmits exchange rates with Symbol = USD and Symbol = USR. Instead, please use the exchange rate with Symbol = USD/RUB. 2. The messages Market Data - Incremental Refresh (X) now contain the optional field Revision (tag 20018) - a service field of the replication subsystem, which is used for matching FAST messages with messages transmitted via the CGate. The field is transmitted only for quotations, trades and market fundamentals of the Derivatives market. 3. FAST messages are now generated in a size not bigger then that of the MTU parameter, i.e. 1500 bytes, which is typical for Ethernet networks. The messages Market Data - Snapshot / Full Refresh (W) and Market Data - Incremental Refresh (X) now contain the optional field LastFragment (tag 893). For more details see the new section '3.2.6. Message fragmentation'. 4. Fields ContractMultiplier, InstrAttribType = 203 are no more transmitted in the messages Security Definition (d) in the channel OTC-ISSUES. MarketSegmentID field's value is now 'Q'. 5. Message templates do not now contain the compression operators copy, delta and increment. 6. The messages Market Data - Snapshot / Full Refresh (W) and Market Data - Incremental Refresh (X) now contain the optional field OrderSide: side of the initiating order in trade. 7. The channels FUT-BOOK-1, FUT-BOOK-5, FUT-BOOK-20, FUT-BOOK-50, OPT-BOOK-1, OPT-BOOK-5, OPT-BOOK-20, OPT-BOOK-50 in snapshot now transmit the message J (Empty book) when there are now orders for the instrument available. 8. Packet format of the TCP Recovery now matches that of the ASTS FAST. The first 4 bytes of each FAST message now contain its length. 9. Prices in the channel OTC-TRADES are transmitted in the field MDEntryPx in Russian roubles for all trades. The field Currency = RUB.
05.02.2015	1.1.0	1. The market segment **RTS Board** renamed to **MOEX Board** - section 4.3.1. **Security Definition (d)** in table **Moscow Exchange MarketSegmentID values**. 2. In section 4.3.1. **Security Definition (d)** in table **Ukrainian Exchange MarketSegmentID values**:

Date	Version	Changes

		• the market segments deleted: **SA**, **SBOND**;
		• the market segment **SBONDG** renamed to "State bonds";
		• the market segment **SGTS** renamed to "Shares and investment securities";
		• new market segments added:
		  • **SGTF** - Foreign shares and depositary receipts
		  • **SBONDL** - Municipal bonds
		  • **SBONDE** - Eurobonds (Ukraine)

1. Introduction

1.1. Document purpose

This document overviews the FAST protocol specifications.

This document does not cover administrative and technical aspects of network connection. Also, this document does not cover security support aspect.

This specification does not include a description of the SPECTRA trading system. The SPECTRA trading system is described in **p2gate_en.pdf** (https://ftp.moex.com/pub/ClientsAPI/Spectra/CGate/prod/docs/) in section '2. SPECTRA system overview'.

1.2. Fast Gate — Basic information

The **Fast Gate** system is used for distributing market data in the FAST-format via the UDP protocol in the multicast mode.

This approach combines the FIX protocol structure and message syntax with the FAST protocol dataflow optimization benefits. Also, it provides possibilities for fast and reliable data distribution to multiple clients of the UDP ptotocol.

The FAST (FIX Adapted for STreaming) protocol is a FIX based protocol developed by FIX Market Data Optimization Working Group in order to optimize financial data exchange performance and reduce latency in distributing large amounts of data. Fast Gate uses the protocol version 1.1: https://www.fixtrading.org/packages/fast-specification-version-1-1.

The **Fast Gate** system allows to distribute anonymous data to clients from:

- SPECTRA trading system;
- MOEX index replica.

The **Fast Gate** system distributes data from the following organizations:

- MOEX (Moscow Exchange);
- ETSC (Eurasian Trade System Commodity Exchange).

The distributed market data include:

- trades;
- quotes;
- market fundamentals;
- instruments and instruments status;
- trading session status;
- indexes;
- news;
- log of anonymous orders.

The servise is used for distributing data to large vendors as well as to brokers, traders, etc.

1.2.1. Data streaming approach

Using of the data streaming approach allows to transmit data from sender to recepient without breaking it into separate messages. The new approach allows to combine several events into a single message which leads to higher data transfer speed and reduce latency time.

1.2.2. Incremental messages

Using of incremental messages allows to significantly reduce amount of transmitted data. Only the data changed due to the market events are transmitted; also, minimal number of commands are used for refreshing data: 'add new record', 'change record', 'delete record'.

1.2.3. FIX format

The **Fast Gate** system uses the FIX mesages format and syntax. Each message consists of header, message body and trailer. Fields are separated with the ASCII symbol — <SOH>.

For more information see sec. 4 .

1.2.4. Encoding in the FAST format

The FAST (FIX Adapted for STreaming) protocol is the FIX based protocol developed by FIX Market Data Optimization Working Group in order to optimize financial data exchange performance and reduce latency in distributing large amounts of data.

The following features are used for data compression:

- implicit tagging;
- fields encoding options;
- usage of PMap;
- stop-bit encoding;
- usage of binary encoding method.

In most cases, the FAST format encoding rules are negotiated between counterparties by exchanging XML-templates.

For more information see sec. 3.2 .

1.2.5. Data receicing via Multicast

For data distribution, the UDP protocol is used in order to distribute data to more than one client at once.

A single UDP packet may contain several FIX messages in the FAST format. Although, currently the system does not provide a possibility to send more than one FAST-coded message via a single UDP packet. In order to match the restriction, FAST messages are generated in a size not bigger then that of the MTU parameter, i.e. 1500 bytes, which is typical for Ethernet networks.

1.2.6. Data recovery

It is extremely important to clients to be able to recover data instantly in case of any data loss.

Fast Gate provides 2 methods of data recovery:

- recovering big amounts of data by sending snapshots (for example, for the clients connected to the system after the trading session start);
- recovering small amounts of data via TCP-connection (for example, in case of message loss during sending).

2. Scenarios of client interactions with Market Data Multicast

This section covers different scenarios of clients connection to the Market Data Multicast feeds. Also, this section covers loss data recovery procedures details.

2.1. Connect client

When client starts listening to MOEX Market Data Multicast FIX/FAST Platform, it should keep the following procedure:

1. Download the actual multicast IP addresses configuration file from ftp. Configuration file is the XML - file describing the connectivity parameters (feeds, multicast addresses, ports, etc.).

2. Download the FAST template from ftp.

3. Start listening to the **Instruments Incremental** stream and accumulate received messages.

4. Start listening to the **Instrument Replay** stream. Obtain the list of instruments from **Instrument Replay**. Apply the messages accumulated in step 3 to the instrument list.

5. Start listening to the **Incremental** feeds and queue received data.

6. Start listening to the **Snapshot** feeds. Receive and apply actual market data snapshot. In each Market Data - *Snapshot/Full Refresh (W)* tag *369-LastMsgSeqNumProcessed* is equal to tag *34-MsgSeqNum* of the last message Market Data - *Incremental Refresh (X)* of the appropriate stream included in the snapshot. The refresh number of each instrument within the tag *83-RptSeq* of the message Market Data - *Snapshot/Full Refresh (W)* is equal to number of incremental refresh in the tag *83-RptSeq* which corresponds to *MDEntry* of the last message Market Data - *Incremental Refresh (X)*, included into the snapshot. For each instrument, it is necessary to omit all messages with numbers through *369-LastMsgSeqNumProcessed* tag number and apply all that are left. The procedure can be both sequential or parallel. I.e., you can either receive snapshots for all instruments and then process the accumulated data or you can process data after receiving each snapshot.

7. Stop listening to the **Snapshot** feeds.

8. Continue receiving and normal processing incremental data.

2.1.1. Empty snapshot in FO-TRADES stream

If the snapshot for the instrument is empty, then it is not transmitted in the stream.

If there is no snapshot for the instrument, it means that there are no active orders and/or trades for the instrument (there is no last trade, all prices and volumes in the statistics for the instrument are equal to 0).

In this case, you need to use all the data received in the Incremental stream. If there are no messages for the instrument, then there are no active orders and/or trades for this instrument. If there are messages for the instrument in the stream, then we initialize the prices and volumes in the statistics for the instrument from them.

As soon as orders appear in the order-book and trades are possibly completed, this will be broadcast in the Incremental stream and the client will receive up-to-date data.

2.2. Incremental Feeds A and B Arbitration

Data in all UDP Feeds are disseminated in two identical feeds (A and B) on two different multicast IPs. It is strongly recommended that client receive and process both feeds because of possible UDP packet loss. Processing two identical feeds allows one to statistically decrease the probability of packet loss

It is not specified in what particular feed (A or B) the message appears for the first time. To arbitrate these feeds one should use the message sequence number found in Preamble or in tag 34 - MsgSeqNum. Utilization of the Preamble allows one to determine message sequence number without decoding of FAST message.

Processing messages from feeds A and B should be performed using the following algorithm:

1. Listen feeds A and B.

2. Process messages according to their sequence numbers.

3. Ignore a message if one with the same sequence number was already processed before.

4. If the gap in sequence number appears, this indicates packet loss in both feeds (A and B). Client should initiate one of the Recovery process. But first of all client should wait a reasonable time, perhaps the lost packet will come a bit later due to packet reordering . UDP protocol can’ t guarantee the delivery of packets in a sequence .

Example:

Packet order	Feed A	Feed B
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

Messages are received from Feed A and Feed B.

1. Receive message # 59 from Feed A, process it.
2. Receive message #59 from Feed B, discard it, because this message was processed before from Feed A.
3. Receive message # 60 from Feed A, process it.
4. Receive message # 60 from Feed B, discard it, because this message was processed before from Feed A.
5. Receive message # 62 from Feed A, discard it and wait for message #61.
6. Receive message # 61 from Feed B, process it.
7. Receive message # 62 from Feed B, process it.
8. Receive message # 62 from Feed A, discard it, because this message was processed before from Feed B.
9. Receive message # 63 from Feed A, process it.
10. Receive message # 65 from Feed A, discard it and wait for message #64.
11. Receive message # 65 from Feed B, discard it and wait for message #64.
12. Begin recovery process, because gap is detected. Message #64 is missed.

3. System functionality

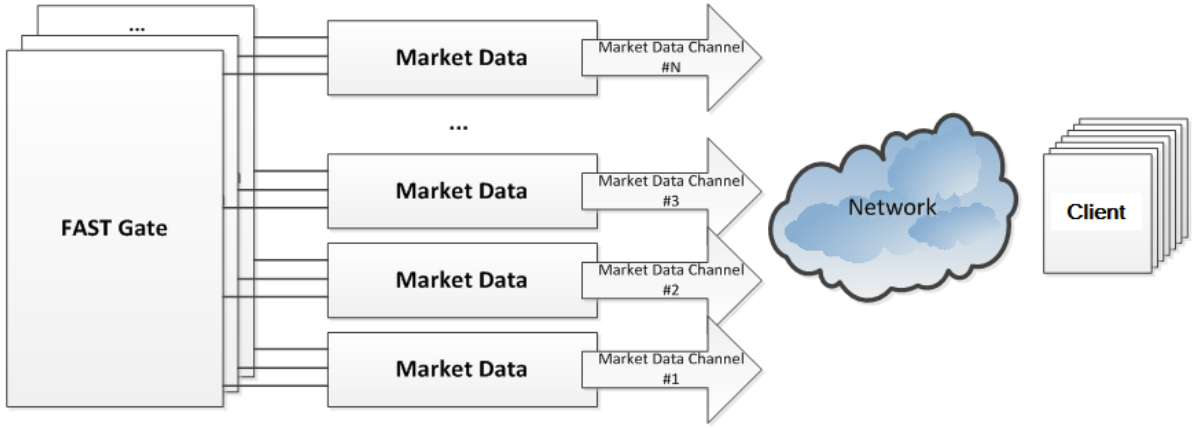
3.1. System architecture

UDP channels used to transfer market data from MOEX. UDP channels are also used for recovery process, TCP connection is used to replay sets of lost messages, already published in one of UDP Channels.

Following feeds are used in the system:

1. Basic:
   - Market Data Incremental Refresh feeds;
   - Instrument Definition feed;
   - Data distribution feed for instrument status change and Trading System connection status.
2. Recovery feeds:
   - Market Recovery feed;
   - TCP Replay session.

Data are distributed via group of feeds, each of that contain data for financial instrument group. The instruments are grouped by the Trading System according to particular parameters. The dedicated Market Data Multicast instance is responsible for distribution in each Feed. A single Market Data Multicast instance is responsible for a single Feed data distribution.



Pic. 1. Market Data distribution feeds

Each feed is a) a bunch of several UDP-feeds with continuous data distribution; b) TCP-port which is used for requesting messages missed in the UDP-feed.

All streams are transmitted using the UDP multicast protocol and every stream is transmitted using a dedicated multicast address. The A and B streams transmit the same data in order to decrease the probability for missing UDP-packets.

Apart from transmitting data in UDP streams, Market Data multicast can accept incoming TCP connections for letting clients request missing data. Clients can request missing messages using one of the next UDP streams (data are available for a period of time specified in the configuration file (not earlier than from beginning of the day), number of messages to be sent at one is limited, number of requests per day is limited, too. All limits are specified in the system configuration file.

3.1.1. Main streams (UDP)

The main streams (incr) in the multicast mode with the UDP protocol is used to distribute the following market data:

- Streams FO-TRADES, FO-BOOK-5, FO-BOOK-20, FO-BOOK-50 — aggregate Order-book data refresh;
- Streams FO-TRADES — trade table and Derivatives market statistics data refresh;
- INDEX stream — index tables data, currency exchange rates and their fixings;
- NEWS stream — Derivatives market related news;
- SPOT-BOOK-1, SPOT-BOOK-20 - updates for aggregated quotations table;
- SPOT-TRADES - updates for trades table and market statistics;
- ORDERS-LOG - anonymous orders.

The data are distributed as FIX-messages Market Data - Incremental Refresh (X) coded in the FAST format. Each message may contain refresh data for several financial instruments.

Any change in trading session status will result in issuing a message Trading Session Status (h) into the Incremental Refresh UDP streams: FO-BOOK-1, FO-BOOK-5, FO-BOOK-20, FO-BOOK-50, FO-TRADES.

3.1.2. Recovery streams (UDP)

The Recovery (snap) streams in the multicast mode with the UDP protocol are used to periodically distribute the current snapshot of the corresponding data as FIX-messages Market Data - Snapshot/Full Refresh (W) coded in the FAST format. Each message contains data for a single instrument only. In Recovery (snap) streams, the SequenceReset (4) message indicates the start of a new snapshot broadcasting.

It is not necessary for clients to be constantly connected to these streams. After receiving the missing data, it is recommended to disconnect from these streams.

Messages Trading Session Status (h), containing data on the trading session status, are transmitted in the end part of each snapshot within the following UDP streams: FO-BOOK-1, FO-BOOK-5, FO-BOOK-20, FO-BOOK-50, FO-TRADES.

3.1.3. Instrument Definitions streams (UDP)

The Instrument Replay (inst replay) streams are used to periodically distribute the trading session status and descriptions of financial instruments as TradingSessionStatus (h) and Security Definition (d) FIX messages coded in the FAST format. Each message contains description for a single financial instrument. The TradingSessionStatus (h) message is broadcast in FUT-INFO and OPT-INFO streams. In the Instrument Replay stream, the SequenceReset(4) message marks the start of broadcasting a new snapshot. A spashot consists of trading instruments descriptions in the form of SecurityDefinition (d) messages and a trading session status TradingSessionStatus (h). Upon receipt of a full snapshot, it should be considered that all instruments descriptions for a given trading session were received.

In case of instrument status change, collateral volume change or price limits change, the Instrument Incremental (inst incr) stream transmits the Security Status (f) FIX-messages.

Transmitted data:

- FUT-INFO stream — futures, calendar spreads, collateral futures for equity options;
- OPT-INFO stream — options;
- SPOT-INFO stream – shares, bonds, commodities.

For the indexes, the Instrument Definitions stream is missing, meaning that Security Definition (d) messages are not being transmitted for the indexes.

3.1.4. Messages in streams

This section describes which messages are transmitted in each data stream.

Stream name	Stream type	Message template name
ORDERS-LOG	Incremental	Heartbeat (id="6") SequenceReset (id="7") OrdersLogMessage (id="29")
ORDERS-LOG	Snapshot	Heartbeat (id="6") SequenceReset (id="7") BookMessage (id="30")
ORDERS-LOG	Historical Replay	**From client to gateway:** Logon (FIX MessageType="A") Logout (FIX MessageType="5") Market Data Request (FIX MessageType="V") **From gateway to client:** Heartbeat (id="6") OrdersLogMessage (id="29") Logon (id="1000") Logout (id="1001")
FUT-INFO	Instrument Replay	Heartbeat (id="6") SequenceReset (id="7")

Stream name	Stream type	Message template name
		SecurityDefinition (id="47") TradingSessionStatus (id="46") DiscreteAuction (id="42") SecurityGroupStatus (id="45")
FUT-INFO	Instrument Incremental	Heartbeat (id="6") SequenceReset (id="7") SecurityStatus (id="48") TradingSessionStatus (id="46") DiscreteAuction (id="42") SecurityMassStatus (id="37") SecurityGroupStatus (id="45")
SPOT-INFO	Instrument Replay	Heartbeat (id="6") SequenceReset (id="7") SecurityDefinition (id="47") TradingSessionStatus (id="46")
SPOT-INFO	Instrument Incremental	Heartbeat (id="6") SequenceReset (id="7") SecurityStatus (id="48") TradingSessionStatus (id="46")
OPT-INFO	Instrument Replay	Heartbeat (id="6") SequenceReset (id="7") SecurityDefinition (id="47") TradingSessionStatus (id="46") SecurityGroupStatus (id="45")
OPT-INFO	Instrument Incremental	Heartbeat (id="6") SequenceReset (id="7") SecurityStatus (id="48") SecurityDefinitionUpdateReport (id="4) TradingSessionStatus (id="46") SecurityMassStatus (id="37") SecurityGroupStatus (id="45")
FO-TRADES FO-BOOK-1 FO-BOOK-5 FO-BOOK-20 FO-BOOK-50 SPOT-BOOK-1 SPOT-BOOK-20 SPOT-TRADES	Incremental	Heartbeat (id="6") SequenceReset (id="7") DefaultIncrementalRefreshMessage (id="31") TradingSessionStatus (id="46")
FO-TRADES	Snapshot	Heartbeat (id="6")

Stream name	Stream type	Message template name
FO-BOOK-1 FO-BOOK-5 FO-BOOK-20 FO-BOOK-50 SPOT-BOOK-1 SPOT-BOOK-20 SPOT-TRADES		SequenceReset (id="7") DefaultSnapshotMessage (id="32") TradingSessionStatus (id="46")
FO-TRADES SPOT-TRADES	Historical Replay	**From client to gateway:** Logon (FIX MessageType="A") Logout (FIX MessageType="5") Market Data Request (FIX MessageType="V") **From gateway to client:** Heartbeat (id="6") DefaultIncrementalRefreshMessage (id="31") TradingSessionStatus (id="46") Logon (id="1000") Logout (id="1001")
INDEX	Incremental	Heartbeat (id="6") SequenceReset (id="7") DefaultIncrementalRefreshMessage (id="31")
INDEX	Snapshot	Heartbeat (id="6") SequenceReset (id="7") DefaultSnapshotMessage (id="32")
INDEX	Historical Replay	**From client to gateway:** Logon (FIX MessageType="A") Logout (FIX MessageType="5") Market Data Request (FIX MessageType="V") **From gateway to client:** Heartbeat (id="6") DefaultIncrementalRefreshMessage (id="31") Logon (id="1000") Logout (id="1001")
NEWS	Incremental	Heartbeat (id="6") SequenceReset (id="7") News (id="9")
NEWS	Historical Replay	**From client to gateway:** Logon (FIX MessageType="A") Logout (FIX MessageType="5") Market Data Request (FIX MessageType="V") **From gateway to client:**

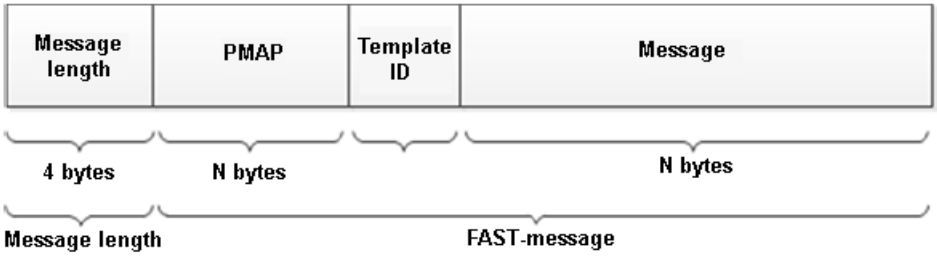
Stream name	Stream type	Message template name
		Heartbeat (id="6") News (id="9") Logon (id="1000") Logout (id="1001")

3.1.5. Sessions for requesting missing messages (TCP)

This service allows to request the resend of missing messages within a specified range of numbers.

The request contains a range of message (numbers) to resend. The request is sent as the Market Data Request (V) FIX-message using the client-initiated TCP-connection. The respond messages are sent to the client as FIX-messages coded in the FAST format using the same TCP-connection. Upon completion of sending, Market Data Multicast closes this TCP-connection. Please note, that maximum number of messages to resend is limited.

The first 4 bytes of each message transmitted in a TCP stream contain its length. Little-endian byte order is used for encoding this number.



Pic. 2. Message structure in TCP stream

When all FAST messages have been sent out, the gateway sends the message Logout to the FAST client, expecting the message Logout from the client in respond. Finishing FIX session also causes TCP session to close.

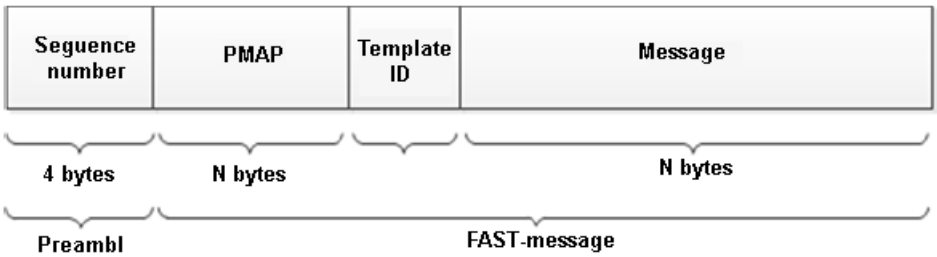
Please also note, that this service should be used only when all other methods are unavailable. This service does not provide high performance and is not available for streams containing aggregated Book-order data.

3.2. FAST format — details

All messages sent by MOEX Market Data Multicast are in the FIX-format coded in the FAST (FIX Adapted for STreaming) protocol. The FAST protocol was developed by FIX Market Data Optimization Working Group in order to optimize financial data flow via distributing bigger amounts of data with less latency.

A specific feature of data distribution via the MOEX Market Data Multicast streams is, that there is a 4-bytes preamble added before every FAST-message. The preamble contains the 34-th tag (SeqNum) value. The 34-th tag is located right after the preamble. Little-endian byte order is used for encoding the preamble.

It allows to receive the message sequence number (both when processing messages from the A and B streams and in case of missing messages) without decoding the FAST-message itself; this leads to time saving during processing of streams.



Pic. 3. Message structure

3.2.1. Stop bit encoding

Encoding stop bit is a constitutive procedure of FAST. The coding allows to exclude redundancy on the data field link layer using the stop bit instead of the standard byte separator. In FAST, stop bit is used instead of the standard FIX-separator (<SOH> byte); therefore, 7 bits of every byte are used for data transmission while the 8th bit indicates the field end.

3.2.2. Implicit tagging

According to the FIX protocol standards, every message is as: **Tag = Value <SOH>**, where:

Tag — number of the field, which is now transmitted;

Value — actual value in this field;

<SOH> – ASCII symbol, used as a separator.

Example:

35=x|268=3 (message header) 279=0|269=2|270=9462.50|271=5|48=800123|22=8 (trade) 279=0|269=0|270=9462.00|271=175|1023=1|48=800123|22=8|346=15 (new bid 1) 279=0|269=0|270=9461.50|271=133|1023=2|48=800123|22=8|346=12 (new bid 2)

FAST allows to avoid this redundancy by using a template which describes the whole message structure. This method is called 'implicit tagging', as FIX tags become implicit parts of the transmitted data. FAST-template exchanges the 'Tag=Value' syntax with 'implicit tagging' according to the following rules:

- tags numbers are not transmitted in message but specified in the template;
- sequence of fields in the message is alike to one of the tags in the template;
- the template specifies a structured bunch of fields with their operators.

3.2.3. Fields encoding options

FAST operates as a state machine, which must 'know' all values to store in memory each moment of time. FAST compares the current field value with the previous one and decides how to act:

- use the constant specified in the template as a new value;
- use the default value (in case of absence of a new field value).

3.2.4. FAST-template

A FAST template corresponds to the FIX message type and uniquely identifies order of fields in each message.

The template also includes syntax indicating the type of field and transfer decoding to apply. Each FAST message contains template ID which is used for decoding.

Message templates are available through links below.

- T1 test environment: https://ftp.moex.com/pub/FAST/Spectra/test/templatesT1/templates.xml
- T0 test environment: https://ftp.moex.com/pub/FAST/Spectra/test/templatesT0/templates.xml
- Derivatives market trading system, indexes: https://ftp.moex.com/pub/FAST/Spectra/prod/templates/MOEX/templates.xml
- ETSC (Eurasian Trade System Commodity Exchange): https://ftp.moex.com/pub/FAST/Spectra/prod/templates/ETSC/templates.xml

3.2.5. Decoding overview

Below is the order of decoding procedure:

1. Transport. A client receives an encoded FAST message.
2. Packet decoding:
   - identification of a template;
   - withdrawal of binary encoded bits;
   - determining correspondences between the received bits and template fields.
3. Fields decoding: using operators to determine value according to the template.
4. Generation of FIX-message.
5. Processing the FIX-message.

3.2.6. Message fragmentation

In order to prevent UDP packets from exceeding MTU size of 1500 bytes (typical for Ethernet networks), messages are fragmented into several parts.

If the message Market Data - Snapshot / Full Refresh (W) does not contain the tag 893-LastFragment, it means that snapshot was transmitted as a single message. All fragmented messages except the last one contain the tag 893-LastFragment = 0. The last fragmented messages

contains the tag 893-LastFragment = 1. Therefore, receiving a message with the tag 893-LastFragment = 1 indicates that snapshot has been completely transmitted.

If the message Market Data - Incremental Refresh (X) does not contain the tag 893-LastFragment, it means that messages have not been fragmented, and the book is consistent after processing the message. All fragmented messages except the last one contain the tag 893-LastFragment = 0. The last fragmented messages contains the tag 893-LastFragment = 1. Therefore, receiving a message with the tag 893-LastFragment = 1 indicates that the book is consistent.

3.2.7. Data types

A field within a FAST template will have one of the standard Data Types indicating the required decoding action: ASCII string, Unicode string, Signed Integer, Unsigned Integer and Decimal. Decimal exponent and mantissa will be encoded as a single, composite field.

FAST does not natively support timestamps. FAST gate will convert the timestamp to an integer value depending on the field type. The decoding application should convert the integer to the FIX UTC format after decoding. Time is always displayed in UTC.

Samples of timestamps encoding:

FIX Type	FIX Pattern	Sample FIX value	Sample FAST value	FAST field type
UTCTimeOnly	HH:MM:SS.sssssssss (nanoseconds)	18:44:24.123456789	184424123456789	uInt64
	HH:MM:SS.sssssssss (nanoseconds)	07:12:13.012345678	71213012345678	uInt64
UTCDateOnly	YYYYMMDD	20080812	20080812	uInt32
UTCTimestamp	YYYYM-MDD-HH:MM:SS.sss	20080812-18:23:54.213	20080812182354123	uInt64

3.3. Missing data recovery

MOEX Market Data Multicast FIX/FAST Platform disseminates Market Data in all feeds over two UDP subfeeds: Feed A and Feed B. In Feeds A and B the identical messages are sent. It lowers the probability of packets loss and provides the first level of protection against missed messages.

Sometimes, messages may be missed on both feeds, requiring a recovery process to take place. Message loss can be detected using the FIX message sequence numbers (tag MsgSeqNum (34)), which are also found in the Preamble. The message sequence number is an incrementing number; therefore, if a gap is detected between messages in the tag MsgSeqNum (34) value, or the Preamble sequence number, this indicates a message h as been missed. In addition, tag RptSeq (83) can be used to detect a gap between the messages at the instrument level. In this case client system should assume that market data maintained in it is no longer correct and should be synchronized to the latest state using one of the recovery mechanisms.

MOEX Market Data Multicast FIX/FAST Platform offers several options for recovering missed messages and synchronizing client system to the latest state. Market Recovery process together with Instruments Replay Feed is the recommended mechanism for recovery. TCP Replay provides less performance mechanism recommended only for emergency recovering of small amount of lost messages when other mechanisms cannot be used for some reason. Instrument level sequencing and natural refresh can be utilized to supplement the recovery process.

3.3.1. Recovery missing data using Recovery streams (UDP)

This recovery method is preferable to use for large - scale data recovery and for late joiners. Recovery feeds contains Market Data - Snapshot/Full Refresh (W) messages. The sequence number (LastMsgSeqNumProcessed(369)) in the Market Data - Snapshot/Full Refresh (W) message corresponds to the sequence number (MsgSeqNum(34)) of the last Market Data - Incremental Refresh (X) message in the corresponding feed. Instrument level sequence number (RptSeq(83)) in Market Data - Snapshot/Full Refresh (W) message correspond to the sequence number (RptSeq(83)) in the MDEntry from last Market Data - Incremental Refresh (X) message. Thus, tag MsgSeqNum(34) shows the gap at the messages level, tag RptSeq(83) shows gap at the instrument level.

After value of RptSeq(83) tag from Market Data - Incremental Refresh (X) becomes more than value of RptSeq(83) tag from Market Data - Incremental Refresh (X) , market data becomes actual.

After value of MsgSeqNum(34) from Market Data - Incremental Refresh (X) message becomes more than value of tag LastMsgSeqNumProcessed(369) from Market Data - Snapshot/Full Refresh (W) message, market data becomes actual.

Messages sequence numbers begins from '1' and ends with the SequenceReset (4) message in each cycle. Therefore, all snapshots should be considered received when the SequenceReset (4) message arrives. After the SequenceReset (4) message a message with sequence number 1 comes, which refers to the next snapshot sending cycle.

If a message does not contain the tag 893-LastFragment, it means that snapshot was transmitted as a single message. Otherwise, the last fragmented messages contains the tag 893-LastFragment = 1. Therefore, receiving a message with the tag 893-LastFragment = 1 indicates that snapshot has been completely transmitted.

Clients should keep queuing real - time data until all missed data is recovered. The recovered data should then be applied prior to data queued.

Consequence of recovery is equal to that described in sec. 2.1 (steps 4 - 7).

Since clients have retrieved recovery data, it is recommended to stop listening Market Recovery feeds.

3.3.2. Recovering missing data using TCP-connection

If there any market data missing in incremental streams Indexes, Trades and ORDERS-LOG (anonymous orders and trades), it can be recovered over the TCP historical replay component using the sequence number range. TCP Replay is a low performance recovery option and should only be used if other options are unavailable or for small - scale data recovery. Number of messages which can be requested by client during TCP connection is limited to 1000.

To request missing data, you should do the following:

1. Establish TCP connection with MOEX Market Data Multicast.
2. Send FIX message Logon(A) with sequence numder 1 to server. After successful authorization server sends the FAST - encoded Logon(A) message.
3. Send Market Data Request (V) message with:
   a. Range of sequence numbers - ApplBegSeqNum(1182) and ApplEndSeqNum (1183) tags.

If request is correct, server sends FAST messages according to requested sequence numbers.

If request is incorrect, server sends FAST Logout (5) message with reject reason.

After server responses, the connection is closed.

Server will process only first user request, second and others will be ignored. If the server does not receive Market Data Request within an established timeout interval after logon, the connection is closed.

Recovery channel has 1 second incoming request timeout.

3.4. Message sequence reset

Every 24 hours, the Fast Gate is being cleaned up from the last day trading session messages, and its message sequences are being reset. When the message sequences have been reset, a message 'Sequence Reset' with a new value in the field 'NewSeqNo' will be transmitted in the (incr) streams. Upon receiving the message 'Sequence Reset', the client is to set the message number value to that transmitted in the message 'NewSeqNo', and reset 'RptSeq' numbers.

Below is the break time schedule for Fast Gate. In the end of each break time, message sequence numbers will be reset:

- MOEX Derivatives Market - 0:00 AM (Moscow Time) till 04:40 AM (Moscow Time);
- ETSC (Eurasian Trade System Commodity Exchange) - 0:00 AM (Moscow Time) till 06:00 AM (Moscow Time);
- Derivatives market test environment - 0:00 AM (Moscow Time) till 04:40 AM (Moscow Time).

For all the main (incr) streams, excluding the FO-TRADES and ORDERS-LOG streams, the message sequence number will be set to 1, and 'RptSeq' number will be set to 1. In the main stream (incr) FO-TRADES, the message sequence number will be set to N, where N is greater than or equal to 1, and the 'RptSeq' number will be set to number greater than or equal to 1, with a pair of Sequence Reset messages transmitted:

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

After transmission of these messages, TCP Recovery provides FAST messages with numbers from 1 to N-1 inclusive, containing trades from previous Additional Weekend Trading Sessions, if any. The initial 'RptSeq' number can be obtained by the member in one of two ways:

- By requesting and processing messages with numbers from 1 to N-1 from the TCP Recovery service;
- By connecting to the Recovery stream (UDP) in accordance with section "3.3.1. Recovery missing data using Recovery streams (UDP)".

In the main ORDERS-LOG stream (incr) the message sequence number will be set to 1 and 'RptSeq' number will be set to 0. This is due to the fact that after a reset, one Empty Book message is transmitted to the ORDERS-LOG stream (incr) for the entire market. See the details in the "5.3.7. Cleaning up active orders pool (all trading sessions)". After Empty Book message has been transmitted, order-books are available in the ORDERS-LOG stream (snap) for all instruments with the value RptSeq=0. Next order-book update in the ORDERS-LOG stream (incr) have the value RptSeq=1.

4. FIX protocol message specifications

The protocol message specifications description below is based on the standard FIX protocol specification v. 5.0 SP2 (https://www.fixtrading.org/standards/fix-5-0-sp-2). It is recommended for users to read some general information about the protocol before commencing with this specification.

Each field is described below:

- **Tag** – the unique field ID, used for generating a FIX message.
- **Field** – the field name, not used for generating FIX messages and described for your reference only.
- **Mandatory** – a field attribute: specifies whether the field in message is mandatory or optional.
  - Y - mandatory field;
  - N - optional field;
  - C - mandatory, if meets the condition (see 'Details').
- **Details** – detailed description of the field.
- **Allowable values** - additional limitations.

The "*****" symbol - flag of difference from the standard FIX protocol.

4.1. Field groups

Many messages contain the same fields. For example, the 'Standard Message Header' group fields contain some administrative information and are mandatory for every message.

4.1.1. Standard Message Header

The standard header, mandatory for every message.

Tag	Field	Mandatory	Details	Available values
34	MsgSeqNum	Y	Message sequence number	
35	MsgType	Y	Message type	
49	SenderCompID	Y	Message sender ID	• 'MOEX' - Moscow Exchange • 'ETSC' - Kazakhstan Exchange (ETS)
52	SendingTime	Y	Message sending time	
1128	ApplVerID	Y	FIX protocol version ID	"9" (FIX50SP2)

4.2. Session layer messages

4.2.1. Logon (A)

A FIX message which is used to initiate a session establishment to the service TCP Recovery.

Tag	Field	Mandatory	Details
8	BeginString	Y	Allowable values: 'FIX.4.4' and 'FIXT.1.1'.
9	BodyLength	Y	Message length.
35	MsgType	Y	'A'
553	Username	N	Any string
554	Password	N	Any string
10	CheckSum	Y	Checksum.

A FAST message which is used to confirm a session establishment to the service TCP Recovery.

Tag	Field	Mandatory	Details
<Standard Message Header>		Y	Message type 'A'.

4.2.2. Logout (5)

A FIX message which is used to initiate a session closure with the service TCP Recovery.

Tag	Field	Mandatory	Details
8	BeginString	Y	Allowable values: 'FIX.4.4' and 'FIXT.1.1'.
9	BodyLength	Y	Message length.
35	MsgType	Y	'5'
10	CheckSum	Y	Checksum.

A FAST message which is used to confirm a session closure with the service TCP Recovery.

Tag	Field	Mandatory	Details
<Standard Message Header>		Y	Message type '5'.
58	Text	N	Reason for ending the session. In case of refusal to process the request, a description of the error is transmitted in the field. If the request is successfully processed, 'nullValue' is transmitted.

4.2.3. Heartbeat (0)

The message **HeartBeat** is sent by FastGate when there were no messages sent in the stream within a 30 seconds time interval.

Tag	Field	Mandatory	Details
<Standard Message Header>		Y	Message type '0'.

4.2.4. Sequence Reset (4)

Tag	Field	Mandatory	Details
<Standard Message Header>		Y	Message type '4'.
36	NewSeqNo	Y	New sequence number.

4.3. Business logic layer messages

This section describes messages of all streams, excluding messages of the anonymous orders and trades stream (ORDERS-LOG) (see sec. 5 below).

The following FIX messages are supported:

- **Security Definition** – Information on instrument.
- **Security Status** – Status and price limit change, change of collateral volume for instrument.
- **Trading Session Status** – Trading session status.
- **Security Definition Update Report** – Volatility and theoretical prices for options.
- **News** – SPECTRA system administrator's messages.
- **Market Data Request** - Missed data request.
- **Market Data - Snapshot / Full Refresh** – Data snapshot (for example, the Order-book full status).
- **Market Data - Incremental Refresh** – Data refresh.
- **DiscreteAuction** - Opening auction data.
- **SecurityMassStatus** - Packaged message for accelerated broadcasts of trading statuses of instruments.
- **SecurityGroupStatus** - Group status of instruments.

4.3.1. Security Definition (d)

Information on instrument.

Tag	Field	Mandatory	Details
<Standard Message Header>		Y	Message type 'd'
911	TotNumReports	Y	Total messages number in the current list
1301	MarketId*	Y	Exchange MIC: • 'MOEX' - Moscow Exchange • 'ETSC' - ETS Eurasian Trading System Commodity Exchange
48	SecurityID	C	Instrument unique ID.

Tag	Field	Mandatory	Details
			ID uniqueness is guaranteed within the market segment specified by the field MarketSegmentId.
22	SecurityIDSource	C	"8" - Exchange Symbol
55	Symbol	N	Symbol code of the instrument
107	SecurityDesc	N	Instrument name
167	SecurityType	N	Multileg type - 'MLEG' — calendar spread
461	CFICode	N	Financial instrument class according to ISO-10962. Valid values are shown in the table below. Note that the value of the CFICode field is provided for informational purposes only and does not confirm its official assignment. According to the international standard ISO 10962, CFI code assignment is performed by the agency responsible for ISIN code assignment.
231	ContractMultiplier	N	Units of underlying asset in instrument. Fractional values are possible.
969	MinPriceIncrement	N	Minimum price step.
1146	MinPriceIncrementAmount	N	Price step cost.
15	Currency	N	Quotation currency.
1148	LowLimitPx*	N	• Lower price limit. Futures and calendar spreads only. • Lower price limit for option premium
1149	HighLimitPx*	N	• Upper price limit. Futures and calendar spreads only. • Upper price limit for option premium.
1300	MarketSegmentId*	N	Market segments. Valid values are shown in the table below.
336	TradingSessionId	N*	Trading session type: • '1' - Day session • '3' - Morning session • '5' - Evening session • '100' - Weekend session
5842	ExchangeTradingSessionId*	N	Trading session ID.
5678	Volatility*	N	Option volatility.
20006	TheorPrice*	N	Option theoretical price.
20007	TheorPriceLimit*	N	Option theoretical price (limits adjusted).
20002	InitialMarginOnBuy*	N	• futures — buyer collateral • options — underlying collateral for buying futures-style option
20000	InitialMarginOnSell*	N	• futures — seller collateral • options — underlying collateral for one uncovered position
20001	InitialMarginSyntetic*	N	Underlying collateral for one covered position (RUB). Options only.
326	SecurityTradingStatus*	N	Instrument trading status: • '21' - Session for the instrument is scheduled. You can't place orders, but you can cancel orders on this instrument. • '17' - Session for the instrument is opened. You can place and cancel orders on this instrument. • '2' - Session for all instruments is suspended. You can't place orders, but you can cancel orders. • '18' - Session for the instrument is ended. You can't place and cancel orders on this instrument. • '119' - The opening auction on the instrument is started. You can place and cancel orders on this instrument. • '121' -The opening auction on the instrument is ended. You can't place and cancel orders on this instrument.

Tag	Field	Mandatory	Details
			• '122' - Session for the instrument is suspended. You can't place orders, but you can cancel orders on this instrument. • '20' - Unknown or Invalid status • '123' - Session for this instrument is running. You can put orders to close position for this instrument. • '124' - The opening auction for this instrument is running. You can put orders to close position for this instrument.
711	NoUnderlyings	N	=1
=> 311	UnderlyingSymbol	N	Valid field values: • for futures and calendar spreads - underlying asset code is broadcast; • for options on futures - futures code is broadcast; • for equity options - SECCODE code from ASTS gateway is broadcast for getting a market data of a share. The combination of SECCODE + SECBOARD should be considered as a separate instrument with separate quotes and tables of deals and orders.
=>20045	UnderlyingBoard	N	SECBOARD trading mode ID from ASTS gateway: • for futures and calendar spreads - 'nullValue' is broadcast; • for options on futures - 'nullValue' is broadcast; • for equity options - SECBOARD trading mode ID from ASTS gateway is broadcast. The combination of SECCODE + SECBOARD should be considered as a separate instrument with separate quotes and tables of deals and orders.
=> 309	UnderlyingSecurityID	N	Futures instrument ID
=> 2620	UnderlyingFutureID	N	Option series identifier 'option_series_id', applicable to options. Futures identifier (SecurityID) for TAS (Trade at Settlement) mode.
1141	NoMDFeedTypes	N*	Number of duplicated blocks
=>1022 =>264 =>1021	MDFeedType MarketDepth MDBookType	N N N	Feed type. Order-book depth. Order-book type: • '1' — Top of Book • '2' — Price Depth
555	NoLegs	N	=2
=> 600	LegSymbol	N	Symbol code of the multi-leg instrument
=> 602	LegSecurityID	N	Multi-leg instrument code
=> 623	LegRatioQty	N	Quantity ratio. Value of field ‘LegRatioQty’ indicates both amount and direction of a multi leg instrument, i.e. if the field ‘LegRatioQty’ contains a value greater than 0, then the multi leg instrument has the same direction as the multi leg order, while a value less than 0 indicates a direction of this multi leg instrument opposite to that of the multi leg order. The absolute value of the field ‘LegRatioQty’ multiplied by multi leg instrument amount in the order allows to obtain the instrument amount value for field ‘LegSymbol’.
455	SecurityAltID*	N	Instrument symbol code.
456	SecurityAltIDSource*	N	Class for SecurityAltID (455): • '8' — Exchange Symbol • '4' — ISIN number
864	NoEvents	N	• '1' — for futures • '2' — for options
=>865	EventType	N	EventType=7. Last trading day.

Tag	Field	Mandatory	Details
=>866 =>1145	EventDate EventTime		In the EventDate field, the date is provided in the format YYYYMMDD, in the MSK time zone. In the EventTime field, the date and time are provided in the format YYYYMMDDHHMISSsss, with the date and time in the UTC time zone.
=>865 =>866 =>1145	EventType EventDate EventTime	N	EventType=101. Expiration date and time of the instrument. In the EventDate field, the expiration date of the instrument is provided in the format YYYYMMDD, in the MSK time zone. In the EventTime field, the date and time of expiration during trading (by schedule) are provided, in the format YYYYMMDDHHMISSsss, in the UTC time zone. The field is used only for expiration by schedule: when the flag "0x80000" in the Flags field has the value "1" - scheduled execution.
870	NoInstrAttrib	N	=3
=> 871 => 872	InstrAttribType InstrAttribValue	N N	=203 Instrument type by SWIFT.
=> 871 => 872	InstrAttribType InstrAttribValue	N N	=204 State registration number.
=> 871 => 872	InstrAttribType InstrAttribValue	N N	=200 Total number of securities by issuer, in units.
202	StrikePrice	N	Strike price.
20005	QuotationList	N	Quotation List.
879	UnderlyingQty	N	Security nominal value.
318	UnderlyingCurrency	N	Code of currency of the security nominal value.

20008	Flags*	N	The field is a bit mask.
			The field can take the following values for futures and calendar spreads:
			• '0x10' - Sign of anonymous trading
			• '0x20' - Sign of non-anonymous trading
			• '0x100' - Sign of multileg-instrument;
			• '0x40000' - Sign of collateral instrument
			• '0x80000' - Scheduled execution or execution in clearing session:
			  • '0' - clearing session
			  • '1' - scheduled execution
			The field can take the following values for options:
			• '0x10' - Sign of anonymous trading
			• '0x20' - Sign of non-anonymous trading

20040	MinPriceIncrementAmountCurr	N	Value of the minimum increment in foreign currency. The 'nullValue' value is transmitted into the field in the SPOT-INFO stream.
20041	SettlPriceOpen	N	The field is deprecated and will be removed in future releases. It is necessary to switch to using the 'SettlPrice' field. The 'nullValue' is always transmitted into the field.
1197	ValuationMethod	N	Method of setting margin requirements for the option:: • "FUT" - futures-style mark-to-market • "EQTY" - equity-style
1190	RiskFreeRate	N	Risk-free interest rate.
20042	FixedSpotDiscount	N	Sum of discounted values of declared cash flows .
20043	ProjectedSpotDiscount	N	Sum of discounted values of forecasted cash flows.

Tag	Field	Mandatory	Details
120	SettlCurrency	N	Settlement currency.
20044	NegativePrices	N	Sign of limiting negative prices: • 0 - Futures prices, price limits and option strikes are limited to positive values; • 1 - Futures prices, price limits and option strikes are not limited.
1266	DerivativeContractMultiplier	N	Coefficient indicating the volume of the underlying asset in the contract quote and strikes of option series. Broadcast for options only.
20053	InterestRateRiskUp	N	Interest risk variable rate on rate up scenario.
20054	InterestRateRiskDown	N	Interest risk variable rate on rate down scenario.
20055	RiskFreeRate2	N	Risk free interest rate FX2 of the currency pair FX2/FX1 (for currency premium options); dividend yield rate 'q' (for index premium options).
20056	InterestRate2RiskUp	N	Interest rate risk up scenario for RiskFreeRate2.
20057	InterestRate2RiskDown	N	Interest rate risk down scenario for RiskFreeRate2.
730	SettlPrice	N	Settlement price of future or calendar spread at the end of last clearing session. Settlement price (theoretical price of the option) after the last clearing session.
20063	TradeModeID	Y	The trading mode to which an instrument belongs. One instrument can only belong to one trading mode. The bit number, counting from 1. • '1' - Futures, anonymous • '3' - Options, anonymous • '14' - Calendar spreads, anonymous • '17' - Collateral, negotiated • '18' - One-day futures with auto-prolongation, anonymous
20060	GroupMask	Y	Bitmask of instrument groups. Bits (flags) with the numbers of the groups to which the instrument belongs are set. One instrument can belong to several groups. • '0x1' - Free • '0x2' - HaltGroupMask overflow • '0x4' - Prohibition on trading in the additional trading session • '0x10' - Intraday clearing session settlement (delivery) • '0x20' - Halts related to discrete auctions
20061	SectionID	Y	Section identifier: • '0' - All sections. • '1' - Stock section. • '2' - Commodity Section. • '3' - Money Section.
20062	BaseContractID	Y	Underlying futures contract identifier
20065	TradePeriodAccess	Y	Flags of trading access in the period of session (bit mask): • '0x1' - Day session. • '0x2' - Evening session. • '0x4' - Weekend session. • '0x8' - Morning session.
20071	HighLimitPxWeekend	N	Upper price limit for option premium during Additional Weekend Trading Session.

Tag	Field	Mandatory	Details
			On a regular trading day (not during Additional Weekend Trading Session), the field contains a null value.
20072	LowLimitPxWeekend	N	Lower price limit for option premium during Additional Weekend Trading Session. On a regular trading day (not during Additional Weekend Trading Session), the field contains a null value.
2528	ClearingSettlPrice	N	Settlement price for the future clearing session.

***** - differs from the standard FIX protocol.

Table 1. Moscow Exchange MarketSegmentID values

MarketId	MarketSegmentId	CFICode	SecurityType	Description
MOEX	D	FXXXSX FFXCSX FCXCSX FXXCSX FFXPSX FCXPSX FXXPSX FFCCSX FFICSX FFSCSX		Futures: • 'FXXXSX' - undefined type of futures contract (Standardized Unknown Future, Unknown delivery) • 'FFXCSX' - Cash-settled Futures on the stock and money sections of the market (Standardized Financial Future, Cash delivery) • 'FCXCSX' - Cash-settled Futures on the commodity and NAMEX sections of the market (Standardized Commodity Future, Cash delivery), one-day futures contract on commodities (gold) with auto-prolongation. • 'FXXCSX' - Cash-settled Futures otherwise (Standardized Unknown Future, Cash delivery) • 'FFXPSX' - Deliverable Futures on the stock and money sections of the market (Standardized Financial Future, Physical delivery) • 'FCXPSX' - Deliverable Futures on the commodity and NAMEX sections of the market (Standardized Commodity Future, Physical delivery) • 'FXXPSX' - Deliverable Futures otherwise (Standardized Unknown Future, Physical delivery) • 'FFCCSX' - Daily futures contract on currency with automatic prolongation (Standardized Financial Future on Currency, Cash delivery) • 'FFICSX' - Daily futures contract on index with automatic prolongation (Standardized Financial Future on Index, Cash delivery) • 'FFSCSX' - Daily futures contract on stock-equity with automatic prolongation (Standardized Financial Future on Stock-Equity, Cash delivery)
MOEX	D	FMXXSX	MLEG	Calendar spreads
MOEX	D	OCAFPS OPAFPS OCESCS OPESCS OCECCS OPECCS OCEICS OPEICS OCETCS OPETCS OCEFCS OPEFCS		Options: • 'OCAFPS' - American deliverable futures option Call • 'OPAFPS' - American deliverable futures option Put • 'OCESCS' - European cash-settled equity option Call • 'OPESCS' - European cash-settled equity option Put • 'OCECCS' - European cash-settled currency option Call • 'OPECCS' - European cash-settled currency option Put • 'OCEICS' - European cash-settled index option Cull • 'OPEICS' - European cash-settled index option Put • 'OCETCS' - European cash-settled commodity option Call • 'OPETCS' - European cash-settled commodity option Put • 'OCEFCS' - European cash-settled futures option Call • 'OPEFCS' - European cash-settled futures option Put

MarketId	MarketSegmentId	CFICode	SecurityType	Description
MOEX	I			Indexes

Table 2. ETS MarketSegmentID values

MarketId	MarketSegmentId	CFICode	SecurityType	Description
ETSC	SAG_CLS	ESXXXX		Quote driven market, agricultural products
ETSC	SAG_CLSN	ESXXXX		Quote driven market, agricultural products (without value added tax)
ETSC	SOL_CLS	ESXXXX		Section of trade in oil products
ETSC	SOL_CLSN	ESXXXX		Section of trade in oil products (without value added tax)
ETSC	SMT_CLS	ESXXXX		Section of trade in metals
ETSC	SAG_AA	ESXXXX		Agricultural products
ETSC	SMT_AA	ESXXXX		Manufactured goods

Table 3. Decoding characters from the 'CFICode' field for options

Char 1 Category	Char 2 Group	Char 3 Scheme	Char 4 Underlying Asset	Char 5 Delivery	Char 6 Stdized/Non-Std
O=Options	**C**=Call **P**=Put	**A**=American **E**=European	**S**=Stock-Equities **F**=Futures **C**=Currencies **I**=Index **T**=Commodities	**P**=Physical **C**=Cash	**S**=Standardized terms (maturity date, strike price, contract size)

Table 4. Decoding characters from the 'CFICode' field for futures and calendar spreads

Symbol 1 *Category*	Symbol 2 *Group*	Symbol 3 *Underlying Asset*	Symbol 4 *Delivery*	Symbol 5 *Stdized/Non-Std*	Symbol 6
F=Future	**F**=Financial **C**=Commodities **X**=Unknown	**C**=Currencies **I**=Index **S**=Stock-Equities **X**=Unknown	**P**=Physical **C**=Cash **X**=Unknown	**S**=Standardized terms (maturity date, strike price, contract size)	**X**=Unknown

4.3.2. Security Status (f)

The message is transmitted at change of instrument status, price limits or collateral volume.

Tag	Field	Mandatory	Details
<Standard Message Header>		Y	Message type 'f'
48	SecurityID	C	Instrument numerical code
22	SecurityIDSource	C	'8' — Exchange Symbol
55	Symbol	N	Symbol code of the instrument
326	SecurityTradingStatus*	N	Instrument trading status: • '21' - Session for the instrument is scheduled. You can't place orders, but you can cancel orders on this instrument. • '17' - Session for the instrument is opened. You can place and cancel orders on this instrument. • '2' - Session for all instruments is suspended. You can't place orders, but you can cancel orders. • '18' - Session for the instrument is ended. You can't place and cancel orders on this instrument. • '119' - The opening auction on the instrument is started. You can place and cancel orders on this instrument. • '121' -The opening auction on the instrument is ended. You can't place and cancel orders on this instrument.

Tag	Field	Mandatory	Details
			• '122' - Session for the instrument is suspended. You can't place orders, but you can cancel orders on this instrument. • '20' - Unknown or Invalid status • '123' - Session for this instrument is running. You can put orders to close position for this instrument. • '124' - The opening auction for this instrument is running. You can put orders to close position for this instrument.
1148	LowLimitPx*	N	• Lower price limit. Futures and calendar spreads only. • Lower price limit for option premium
1149	HighLimitPx*	N	• Upper price limit. Futures and calendar spreads only. • Upper price limit for option premium.
20002	InitialMarginOnBuy*	N	• futures — buyer collateral • options — underlying collateral for buying futures-style option
20000	InitialMarginOnSell*	N	• futures — seller collateral • options — underlying collateral for one uncovered position
20001	InitialMarginSyntetic*	N	Underlying collateral for one covered position (RUB). Options only.
20071	HighLimitPxWeekend	N	Upper price limit for option premium during Additional Weekend Trading Session. On a regular trading day (not during Additional Weekend Trading Session), the field contains a null value.
20072	LowLimitPxWeekend	N	Lower price limit for option premium during Additional Weekend Trading Session. On a regular trading day (not during Additional Weekend Trading Session), the field contains a null value.

4.3.3. Trading Session Status (h)

The message is transmitted at the start and in the end of trading sessions.

Tag	Field	Mandatory	Details
<Standard Message Header>		Y	Message type 'h'
336	TradingSessionId	Y	Trading session type: • '1' - Day session • '3' - Morning session • '5' - Evening session • '100' - Weekend session
5842	ExchangeTradingSessionID*	N	Trading session ID
340	TradSesStatus	Y	Trading session state: • '4' - Trading is scheduled • '2' - Trading is running • '1' - Trading is suspended • '3' - Trading is completed because the time is up
1368	TradSesEvent	N	Trading session events: • '0' — Trading resumed after a night technological break • '1' — Start and end of trading session • '3' — Trading session status change
1301	MarketId	N*	Exchange MIC:

Tag	Field	Mandatory	Details
			• 'MOEX' — Moscow Exchange • 'ETSC' — ETS Eurasian Trading System Commodity Exchange
1300	MarketSegmentId	N*	Market segments: • 'D' — Futures, Options, Calendar spreads • 'S*' — Securities, Bonds, Commodities
342	TradSesOpenTime	N	Trading session open time and date
344	TradSesCloseTime	N	Trading session close time and date
20066	TradePeriodID	Y	Identifies the trade period.
20069	SettlSessBegin	Y	Date and time of the start of the settlement session in the format: YYYYMMDDHHMMSSsss
20070	ClrSessBegin	Y	Date and time of the start of the clearing session in the format: YYYYMMDDHHMMSSsss

***** - differs from the standard FIX protocol.

4.3.4. Security Definition Update Report (BP)

Options volatility and theoretical prices.

Tag	Field	Mandatory	Details
<Standard Message Header>		Y	MESSAGE TYPE 'BP'
48	SecurityID	C	Instrument numeric code
22	SecurityIDSource	C	'8' — Exchange Symbol
5678	Volatility*	N	Option volatility.
20006	TheorPrice*	N	Option theoretical price.
20007	TheorPriceLimit*	N	Option theoretical price (limits adjusted)

***** - differs from the standard FIX protocol.

4.3.5. News (B)

SPECTRA system administrator's messages.

Tag	Field	Mandatory	Details
<Standard Message Header>		Y	Message type 'B'
893	LastFragment	N	This field indicates whether the message is the last in the series generated for a news message with NewsID. Allowable values: • **0** – not the last message • **1** – last message The field is non-mandatory. When absent, the message packet is considered as non-fragmented.
1472	NewsID	N	News ID
42	OrigTime	N	News date and time
1474	LanguageCode	N	Language
61	Urgency	N	Urgency
148	Headline	Y	Header text
33	NoLinesOfText	Y	=1
=>58	Text	Y	Message text body. The string is transmitted by FAST gate in UTF-8 character set
1301	MarketId	N	Exchange MIC: • 'MOEX' — Moscow Exchange • 'ETSC' — ETS Eurasian Trading System Commodity Exchange
1300	MarketSegmentId	N	Market segments:

Tag	Field	Mandatory	Details
			• 'F' — futures

4.3.6. Market Data Request (V)

A FIX message which is used to request missing data in the session to the service TCP Recovery.

Tag	Field	Mandatory	Details
8	BeginString	Y	Allowable values: • FIX.4.4 • FIXT.1.1
9	BodyLength	Y	Message length
35	MsgType	Y	"V"
262	MDReqId	Y	Request ID
1182	ApplBegSeqNum	N	Sequence number of the first requested message.
1183	ApplEndSeqNum	N	Sequence number of the last requested message. If a single message is requested, then ApplBegSeqNum(1182)=ApplEndSeqNum(1183). If all messages are requested (no more than total messages sent) after a particular message number, then ApplEndSeqNum(1183)=0(infinity).
10	CheckSum	Y	Checksum

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

Data snapshot.

Tag	Field	Mandatory	Details
<Standard Message Header>		Y	Message type 'W'
893	LastFragment	N	Indicates the last message in the message group for the instrument. Allowable values: • **0** – not the last message • **1** – the last message The field is not mandatory. If a message does not contain this field, it means that the packet with message has not been fragmented.
911	TotNumReports	Y	The number of messages in the snapshot, which have 'LastFragment '= 1
83	RptSeq	Y	The 'RptSeq' number of the last incremental update included in the current market data snapshot for instrument.
369	LastMsgSeqNumProcessed	N	The 'MsgSeqNum' of the last message sent into incremental feed at the time of the current snapshot generation.
48	SecurityID	N	Instrument numeric code. The 'nullValue' value is transmitted into the field in the INDEX stream.
22	SecurityIDSource	C	'8' — Exchange Symbol
55	Symbol	N	Symbol code of the instrument, currency exchange rate name, index name.
1151	SecurityGroup	N	Not used
268	NoMDEntries	Y	Number of 'MDEntry' records in the current message.
=>20003	MDEntryTradeType	N	MDEntryTradeType is sent for Spot market trades. The tag value format is <trade type><repo type><trade status>. Possible trade types: • 'Q' - quote-based transaction • 'B' - two-sided transaction • 'A' - auction-based transaction • 'I' - IPO transaction Possible repo types:

Tag	Field	Mandatory	Details
			• ' ' (space) - regular trade • 'R' - repurchase agreement
=>269	MDEntryType	Y	Record type: • '0' — Bid • '1' — Ask • '2' — Trade • '3' — Index Value • '4' — Opening Price • '5' — Closing Price • '6' — Settlement Price • '7' — Trading Session High Price • '8' — Trading Session Low Price • '9' — Trading Session VWAP Price • 'B' — Cumulative Trade Volume • 'C' — Open Interest • 'v' — Total bid volume with synthetic liquidity • 'w' — Total offer volume with synthetic liquidity • 's' — Market Capitalization • 'J' — Empty book • 'x' — Total bid volume without synthetic liquidity • 'y' — Total offer volume without synthetic liquidity • 'u' - Funding rate when the contract price deviates from the underlying asset price beyond the specified level on the contract (for one-day futures contracts with auto-prolongation) • 'W' - Opening auction price • 'z' - Dividend amendment for one-day futures contracts with auto-prolongation on the index or the share • 'U' - Indicative funding rate (for one-day futures contracts with auto-prolongation) • 'Z' - Current dividend amendment for one-day futures contracts with auto-prolongation on the index or the share • 'm' - Current market price • 'q' - Current indicative funding rate for the instrument (for one-day futures contracts with auto-prolongation). In points. • 'r' - Funding rate when the contract price deviates from the underlying asset price beyond the specified level on the contract (for one-day futures contracts with auto-prolongation). In points.
=>5842	ExchangeTradingSessionId*	N	Trading session ID
=>278	MDEntryID	N	Trade ID
=>264	MarketDepth	N	Market depth
=>270	MDEntryPx	N	Price, rate and index values.
=>271	MDEntrySize	N	Volume, quantity.
=>1023	MDPriceLevel	N	Price level.
=>272	MDEntryDate	N	Record last change date.

Tag	Field	Mandatory	Details
			Value NULL indicates the current date.
=>273	MDEntryTime	N	Record last change time.
=>346	NumberOfOrders	N	• Sell orders quantity (for records with MDEntryType=v (Total bid volume)*) • Buy orders quantity (for records with MDEntryType=w (Total offer volume)*)
=>828	TrdType	C	Trade type: • '0' — Market trade • '22' — Negotiated trade • '45' — Option exercise • '1000' — Futures exercise (standard method). Non-deliverable futures exercise • '1003' — Option expiration • '2002' - Trade originated in RFS • '1004' - exiting a daily futures contract with automatic prolongation • '1005' - Trade Settlement TAS The field is mandatory for records with MDEntryType=2 (Trade).
=>20017	MDFlags*	N	• '0x1' — Flag of fixing for the main clearing session • '0x100' — Flag of fixing
=>15	Currency	N	Currency code
=>10504	OrderSide	N	Side of the aggressive order in trade. Transmitted only for the Derivatives market. Allowable values: • '1' – buy order (Buy) • '2' – sell order (Sell)
=>20039	MDEntrySyntheticSize	C	Synthetic liquidity volume. The field is mandatory for records in the streams FO-BOOK-1, FO-BOOK-5, FO-BOOK-20, FO-BOOK-50 – aggregate Order-book data refresh.
=>10514	IndexCrossRate	N	Foreign exchange rate used in calculating the index value. Published in the INDEX stream only and only in the messages with MDEntryType = '3' (Index Value) only.
=>625	TradingSessionSubID	N	Trading period: • 'S' – opening auction • 'N' – normal trading period Published only in the FO-TRADES stream and only in messages with MDEntryType = '2' - Trade.

***** - differs from the standard FIX protocol.

4.3.8. Market Data - Incremental Refresh (X)

Data refresh.

Tag	Field	Mandatory	Details
<Standard Message Header>		Y	Message type 'X'
893	LastFragment	N	Indicates the last message in the message group for the instrument. Allowable values: • '0' – not the last message • '1' – the last message

Tag	Field	Mandatory	Details
			The field is not mandatory. If a message does not contain this field, it means that the packet with message has not been fragmented.
268	NoMDEntries	Y	Number of 'MDEntry' records in the current message.
=>83	RptSeq	Y	Incremental refresh sequence number
=>279	MDUpdateAction	Y	Incremental refresh type: • '0' — New • '1' — Change • '2' — Delete
=>20003	MDEntryTradeType	N	MDEntryTradeType is sent for Spot market trades. The tag value format is <trade type><repo type><trade status>. Possible trade types: • 'Q' - quote-based transaction • 'B' - two-sided transaction • 'A' - auction-based transaction • 'I' - IPO transaction Possible repo types: • ' ' (space) - regular trade • 'R' - repurchase agreement
=>31	LastPx	N	Used for MDUpdateAction=2 only. Last exchange trade price.
=>269	MDEntryType	Y	Record type: • '0' — Bid • '1' — Ask • '2' — Trade • '3' — Index Value • '4' — Opening Price • '5' — Closing Price • '6' — Settlement Price • '7' — Trading Session High Price • '8' — Trading Session Low Price • '9' — Trading Session VWAP Price • 'B' — Cumulative Trade Volume • 'C' — Open Interest • 'v' — Total bid volume with synthetic liquidity • 'w' — Total offer volume with synthetic liquidity • 's' — Market Capitalization • 'J' — Empty book • 'x' — Total bid volume without synthetic liquidity • 'y' — Total offer volume without synthetic liquidity • 'u' - Funding rate when the contract price deviates from the underlying asset price beyond the specified level on the contract (for one-day futures contracts with auto-prolongation)

Tag	Field	Mandatory	Details
			• 'W' - Opening auction price • 'z' - Dividend amendment for one-day futures contracts with auto-prolongation on the index or the share • 'U' - Indicative funding rate (for one-day futures contracts with auto-prolongation) • 'Z' - Current dividend amendment for one-day futures contracts with auto-prolongation on the index or the share • 'm' - Current market price • 'q' - Current indicative funding rate for the instrument (for one-day futures contracts with auto-prolongation). In points. • 'r' - Funding rate when the contract price deviates from the underlying asset price beyond the specified level on the contract (for one-day futures contracts with auto-prolongation). In points.
=>48	SecurityID	N	Instrument numeric code. The 'nullValue' value is transmitted into the field in the INDEX stream.
=>22	SecurityIDSource	C	'8' — Exchange Symbol
=>5842	ExchangeTradingSessionId*	N	Trading session ID
=>278	MDEntryID	N	Trade ID
=>264	MarketDepth	N	Market depth
=>270	MDEntryPx	N	Price, rate and index values.
=>271	MDEntrySize	N	Volume, quantity.
=>1023	MDPriceLevel	N	Price level.
=>272	MDEntryDate	N	Record last change date. Value NULL indicates the current date.
=>273	MDEntryTime	N	Record last change time.
=>346	NumberOfOrders	N	• Sell orders quantity (for records with MDEntryType=v (Total bid volume)*). • Buy orders quantity (for records with MDEntryType=w (Total offer volume)*).
=>828	TrdType	C	Trade type: • '0' — Market trade • '22' — Negotiated trade • '45' — Option exercise • '1000' — Futures exercise (standard method). Non-deliverable futures exercise • '1003' — Option expiration • '2002' - Trade originated in RFS • '1004' - exiting a daily futures contract with automatic prolongation • '1005' - Trade Settlement TAS The field is mandatory for records with MDEntryType=2 (Trade).
=>55	Symbol	N	Symbol code of the instrument, currency exchange rate name, index name.
=>20017	MDFlags*	N	• '0x1' — Flag of fixing for the main clearing session • '0x100' — Flag of fixing
=>15	Currency	N	Currency code.
=>1151	SecurityGroup	N	Not used

Tag	Field	Mandatory	Details
=>20018	Revision	N	Service field of the replication subsystem. The field is transmitted only for quotations, trades and market fundamentals of the Derivatives market.
=>10504	OrderSide	N	Side of the aggressive order in trade. Transmitted only for the Derivatives market. Allowable values: • '1' – buy order (Buy) • '2' – sell order (Sell)
=>20039	MDEntrySyntheticSize	C	Synthetic liquidity volume. The field is mandatory for records in the streams FO-BOOK-1, FO-BOOK-5, FO-BOOK-20, FO-BOOK-50 – aggregate Order-book data refresh.
=>746	OpenInterest	N	Open interest - the number of positions on the instrument in the market after the trade. This field is mandatory for records with MDEntryType = 2 (Trade) in the FO-TRADES stream.
=>10514	IndexCrossRate	N	Foreign exchange rate used in calculating the index value. Published in the INDEX stream only and only in the messages with MDEntryType = '3' (Index Value) only.
=>625	TradingSessionSubID	N	Trading period: • 'S' – opening auction • 'N' – normal trading period Published only in the FO-TRADES stream and only in messages with MDEntryType = '2' - Trade.

***** - differs from the standard FIX protocol.

4.3.9. DiscreteAuction (U1)

The message contains data about the opening auction (discrete auction).

Tag	Field	Mandatory	Details
<Standard Message Header>		Y	Message type 'U1'
20046	TradSesCloseTimeFrom	Y	The beginning of the time interval during which the discrete auction will stop accepting orders
20047	TradSesCloseTimeTill	Y	The end of the time interval during which the discrete auction will stop accepting orders
342	TradSesOpenTime	Y	The start time for accepting orders for a discrete auction
21002	AuctionID	Y	Discrete auction ID
5842	ExchangeTradingSessionID	Y	Trading session number
20048	EventIDOpen	Y	The number of the synchro event of the beginning of the order acceptance for a discrete auction
20049	EventIDClose	Y	The number of the synchro event of the end of the order acceptance for a discrete auction
711	NoUnderlyings	N	The number of items in the group
311	=> UnderlyingSymbol	N	Underlying asset code
20066	TradePeriodID	Y	Identifies the trade period.

4.3.10. SecurityMassStatus (CO)

Packaged message for accelerated broadcasts of trading statuses of instruments. Messages are fragmented when the maximum size specified in the gateway configuration is reached. Each packet broadcasts a repeating group of fields containing the instrument ID and its status. The first packet in a series is marked with the 'LastFragment'=0 flag. The last packet is marked with the 'LastFragment'=1 flag.

Tag	Field	Mandatory	Details
<Standard Message Header>		Y	Message type 'CO'
146	NoRelatedSym	Y	Number of statuses.
=>48	SecurityID	Y	Instrument numerical code

Tag	Field	Mandatory	Details
=>22	SecurityIDSource	C	'8' — Exchange Symbol
=>55	Symbol	Y	Symbol code of the instrument
=>326	SecurityTradingStatus*	Y	Instrument trading status: • '21' - Session for the instrument is scheduled. You can't place orders, but you can cancel orders on this instrument. • '17' - Session for the instrument is opened. You can place and cancel orders on this instrument. • '2' - Session for all instruments is suspended. You can't place orders, but you can cancel orders. • '18' - Session for the instrument is ended. You can't place and cancel orders on this instrument. • '119' - The opening auction on the instrument is started. You can place and cancel orders on this instrument. • '121' -The opening auction on the instrument is ended. You can't place and cancel orders on this instrument. • '122' - Session for the instrument is suspended. You can't place orders, but you can cancel orders on this instrument. • '20' - Unknown or Invalid status

Symbol '*****' - sign differs from the standard FIX protocol.

4.3.11. SecurityGroupStatus (f)

Group status of instruments. Is used for accelerated broadcasts of trading statuses of instruments. For more detailed information on the algorithm for determining the SecurityGroupStatus (f) message, refer to the section "sec. 4.8".

Tag	Field	Mandatory	Details
<Standard Message Header>		Y	Message type 'f'
20064	SecurityGroupID	Y	Instrument group ID.
20058	HaltType	Y	Halt type (group status of instruments): • '0' - The SecurityGroupStatus (f) message with HaltType=0 (Empty Halts). The client should clear all group statuses on their side and must NOT reuse previously sent SecurityGroupIDs. • '1' - Trading session level. • '2' - Section level. • '3' - Section + Trading mode level. • '4' - Underlying futures contract level. • '5' - Instruments within the trading mode. • '6' - Instruments within the trading mode + Underlying futures contract level. • '7' - Instrument group level. • '8' - Instrument group + Underlying futures contract level.
20059	TradeModeMask	Y	Trading mode (Bitmask with one set bit): • '0' (All bits cleared) - All trading modes • '0x1' - Futures, anonymous. • '0x4' - Options, anonymous. • '0x2000' - Calendar spreads, anonymous. • '0x10000' - Collateral, negotiated. • '0x20000' - One-day futures with auto-prolongation, anonymous.

Tag	Field	Mandatory	Details
20060	GroupMask	Y	Instrument group (Bitmask with one set bit). • '0' (All bits cleared) - All groups • '0x1' - Free • '0x2' - HaltGroupMask overflow • '0x4' - Prohibition on trading in the additional trading session • '0x10' - Intraday clearing session settlement (delivery) • '0x20' - Halts related to discrete auctions
20061	SectionID	Y	Section identifier: • '0' - All sections. • '1' - Stock section. • '2' - Commodity Section. • '3' - Money Section.
20062	BaseContractID	Y	Underlying futures contract identifier
326	SecurityTradingStatus*	Y	Group mode: • '18' (Not available for trading (end of session)) - All prohibited. • '2' (Trading halt) - All prohibited except for order cancellation. • '123' (Close position) - Position increase prohibited. • '119' (Opening auction) - Order collection during the opening auction. • '17' (Ready to trade (start of session)) - All allowed. Modes are listed in order of decreasing priority (strictness of restriction).
60	TransactTime	Y	Time of status change. A 64-bit unsigned integer encoding the date and time in UTC: YYYYMMDDHHMMSSsss.

***** - differs from the standard FIX protocol.

4.4. Market Fundamentals

FastGate does not broadcast the value 0 in the MDEntryPx and MDEntrySize fields for all Market Data Entries, except for Market Data Entries with MDEntryType: 'u', 'z', 'U', 'Z', 'q', 'r' for one-day futures contracts with auto-prolongation. After receiving the list of instruments assigned to the new trading session, the client should initialize the MDEntryPx and MDEntrySize fields to 0 for all Market Data Entries as described in the table below. Subsequently, the MDEntryPx and MDEntrySize fields will either receive non-zero values, or remain at 0 after the trading session end. The fields may remain at 0 if there are no orders or trades for the instrument during the trading session.

There are several aggregate values reflecting instrument market during a session, which may be changed by market events:

Market Data Entry type	MDEntryType	MDEntryPx	MDEntrySize	Market
Opening Price	4	•	–	SIFOC
Closing Price	5	•	–	SIFOC
Settlement Price	6	•	–	SIFO
Trading Session High Price	7	•	–	SIFOC
Trading Session Low Price	8	•	–	SIFOC
Trading Session VWAP Price	9	•	–	SIFO
Cumulative Trade Volume	B	•	•	SIFOC
Open Interest	C	–	•	FO
Market capitalization	s	•	–	I
Total bid volume with synthetic liquidity	v	–	•	FOC
Total offer volume with synthetic liquidity	w	–	•	FOC
Total bid volume without synthetic liquidity	x	–	•	FOC
Total offer volume without synthetic liquidity	y	–	•	FOC

Market Data Entry type	MDEntryType	MDEntryPx	MDEntrySize	Market
Opening auction price	W	•	–	F
Funding rate when the contract price deviates from the underlying asset price beyond the specified level on the contract (for one-day futures contracts with auto-prolongation)	u	•	–	F
Dividend amendment for one-day futures contracts with auto-prolongation on the index or the share	z	•	–	F
Indicative funding rate (for one-day futures contracts with auto-prolongation)	U	•	–	F
Current dividend amendment for one-day futures contracts with auto-prolongation on the index or the share	Z	•	–	F
Current market price	m	•	–	FOC
Current indicative funding rate for the instrument (for one-day futures contracts with auto-prolongation). In points.	q	•	–	F
Funding rate when the contract price deviates from the underlying asset price beyond the specified level on the contract (for one-day futures contracts with auto-prolongation). In points.	r	•	–	F

Markets:

- **F** – Futures,
- **O** – Options,
- **C** - Calendar spreads,
- **I** – Indexes,
- **S** – Securities, Bonds, Commodities.

Message **Cumulative Trade Volume** contains **MDEntryPx = 0** for calendar spreads.

Exceptions from FIX standards:

- 'Trade Volume' messages contain:
  - **MDEntrySize** – daily number of shares or contracts traded (standard FIX meaning);
  - **MDEntryPx** – accumulated daily turnover expressed in the currency of the instrument;
- 'Open Interest' messages contain:
  - **MDEntrySize** – number of shares or contracts;
- 'Market capitalization' messages contain:
  - **MDEntryPx** – index equities capitalization;
- 'Total bid volume with synthetic liquidity/ Total offer volume with synthetic liquidity' messages contain:
  - **MDEntrySize** – total number of shares or contracts with synthetic liquidity.
- 'Total bid volume without synthetic liquidity/ Total offer volume without synthetic liquidity' messages contain:
  - **MDEntrySize** – total number of shares or contracts without synthetic liquidity.

4.5. The message Empty book (MDEntryType = J) in the Aggregate Book streams

The message Empty Book obliges a client to delete data on a certain instrument.

The field **SecurityID** contains the instrument ID.

The field **ExchangeTradingSessionID** remains empty.

The message is sent on session's close, or on technical breaks via the main streams (incr): FO-BOOK-1, FO-BOOK-5, FO-BOOK-20, FO-BOOK-50, SPOT-BOOK-1, SPOTBOOK-20.

The stream Recovery (snap): FO-BOOK-1, FO-BOOK-5, FO-BOOK-20, FO-BOOK-50, SPOT-BOOK-1, SPOT-BOOK-20 transmits the message **J (Empty book)** in case there is no orders added on the certain instrument. The absence of messages on the instrument in the specified Recovery (snap) streams means that there are no orders for this instrument.

4.6. Sample representation of Aggregate Book incremental refresh processing

Sample representation of Aggregate Book will be used in this section:

	Bid		Offer	
Level	Quantity	Price	Quantity	Price
1	100	2411.00	30	2412.00
2	500	2410.50	90	2413.00
3	950	2410.00	400	2413.50
4	500	2409.00	500	2414.00
5	300	2408.50	320	2414.50

Some notes regarding this view of Aggregate Book:

- price levels are sorted by both **Price** and **Price Level** columns;
- Bid table is sorted by Price descending;
- Offer table is sorted by price ascending.

Update of Existing Record

When new order is entered or existing order was filled or cancelled, some **Price Level** will be affected. Update message will be sent only if affected level matches 5-deep book subscription.

Here's an excerpt of update messages that refers to existing **Market Data Entry**, reflecting new order entered 'Buy 20 GAZP @ 2409.00':

Tag	Field	Description
279	MDUpdateAction	1=Change
48	SecurityID	12345
22	SecurityIDSource	8 = Exchange Symbol
269	MDEntryType	0 = Bid
1023	MDPriceLevel	4
270	MDEntryPx	2409.00
271	MDEntrySize	520
264	MarketDepth	5

Order Book should be updated:

	Bid		Offer	
Level	Quantity	Price	Quantity	Price
1	100	2411.00	30	2412.00
2	500	2410.50	90	2413.00
3	950	2410.00	400	2413.50
4	**520**	2409.00	500	2414.00
5	300	2408.50	320	2414.50

New Record

When a new order is submitted, resulting in creation of a new price level, **Market Data Entry** with **MDUpdateAction=New** will be sent to create/insert a new price.

For example:

Two orders were entered simultaneously on instrument GAZP (with security id 12345): 'Sell 40 by 2412.50' and 'Sell 20 by 2412.50'. This corresponds to new Price Level 2. All other price levels (formerly 2-5) should be implicitly shifted down (to positions 3-6); excess price level (6) should be removed implicitly – no corresponding messages will be sent.

This will result in **Market Data Entry** being sent as part of Market Data Incremental Refresh message:

Tag	Field	Description
279	MDUpdateAction	0 = New
48	SecurityID	12345
22	SecurityIDSource	8
1023	MDPriceLevel	2
269	MDEntryType	1 = Offer
270	MDEntryPx	2412.50
271	MDEntrySize	60
264	MarketDepth	5

5-deep Order Book should be updated accordingly:

	Bid		Offer	
Level	Quantity	Price	Quantity	Price
1	100	2411.00	30	2412.00
2	500	2410.50	**60**	**2412.50**
3	950	2410.00	90	2413.00
4	520	2409.00	400	2413.50
5	300	2408.50	500	2414.00
			~~320~~	~~2414.50~~

Delete of Existing Record

When existing order was filled or cancelled, some **Price Level** will be deleted.

For example: all orders were deleted on level “Offer 90, 2413.00”. All price levels (formerly 4-5) should be implicitly shifted up (to positions 3-4). There was offer orders out of 5-deep book.

This will result in **Market Data Entries** being sent as part of **Market Data Incremental Refresh** message:

Tag	Field	Description
279	MDUpdateAction	2 = Delete
48	SecurityID	12345
22	SecurityIDSource	8
1023	MDPriceLevel	3
269	MDEntryType	1 = Offer
270	MDEntryPx	2413.00
271	MDEntrySize	0
264	MarketDepth	5
279	MDUpdateAction	0 = New
48	SecurityID	12345
22	SecurityIDSource	8
1023	MDPriceLevel	5
269	MDEntryType	1 = Offer
270	MDEntryPx	2414.50
271	MDEntrySize	320
264	MarketDepth	5

5-deep Order Book should be updated accordingly:

Level	Bid		Offer	
	Quantity	Price	Quantity	Price
1	100	2411.00	30	2412.00
2	500	2410.50	60	2412.50
3	950	2410.00	~~90~~	~~2413.00~~
4	520	2409.00	400	2413.50
5	300	2408.50	500	2414.00
			320	2414.50

4.7. Instrument IDs

The instrument IDs are transmitted in fields of messages Security Definition (d), Security Status (f), Security Definition Update Report (BP), Market Data - Snapshot / Full Refresh (W), Market Data - Incremental Refresh (X):

- SecurityID - Instrument numeric code;
- Symbol - Instrument symbol code.

Table describes which field to use as the Instrument ID in each stream.

Stream name	Instrument ID
ORDERS-LOG	SecurityID
FUT-INFO OPT-INFO SPOT-INFO	SecurityID Symbol
FO-TRADES FO-BOOK-1 FO-BOOK-5 FO-BOOK-20 FO-BOOK-50 SPOT-BOOK-1 SPOT-BOOK-20 SPOT-TRADES	SecurityID
INDEX	Symbol
NEWS	Is missing
OTC-ISSUES	Symbol
OTC-TRADES	Symbol

4.8. Group statuses of instruments

Statuses for instrument groups (the SecurityGroupStatus message) and individual instrument statuses (the SecurityStatus and SecurityMassStatus messages) are distributed independently and duplicate each other. The client can use either option. To minimize latency, instrument group statuses are recommended.

Fast Gate distributes instrument statuses by groups (message SecurityGroupStatus (f)). An instrument can be subject to multiple group statuses, consequently, one instrument can belong to several groups. Distribution of statuses by groups (many instruments, few groups) helps save traffic, as there is no need to receive updates for all instruments, and speeds up the calculation of the instrument status.

4.8.1. Algorithm for determining trading statuses of instruments by group statuses

An instrument belongs to one or more groups. Each group has a trading status in the SecurityTradingStatus field of the SecurityGroupStatus (f) message. The final instrument status is the most restrictive (strictest) status among all groups to which the instrument belongs. Trading statuses of instrument group:

- '18' - All prohibited.
- '2' - All prohibited except for order cancellation.

- '123' - Position increase prohibited.
- '119' - Order collection during the opening auction.
- '17' - All allowed.

Statuses are listed in order of decreasing priority (strictness of restriction).

The client can independently calculate the trading status of an instrument based on the Security Definition (d) and SecurityGroupStatus (f) messages, following the algorithm:

1. The client subscribes to the topics of the Instrument Incremental stream and saves received messages SecurityGroupStatus (f).
2. The client receives a snapshot of instruments (Security Definition (d) message, one per instrument) from the topics of the Instrument Replay stream. In the message, the connections to the Underlying futures contract, sections, trading mode, and the instrument group are transmitted in BaseContractID, SectionID, TradeModeID, GroupMask fields. The client fixes the list of instruments and their connections, which remains unchanged during trading.
3. The client receives group status snapshots of instruments (SecurityGroupStatus (f) messages) from the Instrument Replay stream topics.
4. The client constructs the instrument_state container based on records from the snapshots of SecurityDefinition (d) and SecurityGroupStatus (f) messages. Container elements include:
   - SecurityID – Unique instrument identifier.
   - TradeModeMask - The trading mode to which an instrument belongs. The value from the TradeModeID field is converted into a mask with one set bit (flag).
   - GroupMask – Mask where bits (flags) with the numbers of the groups to which the instrument belongs are set.
   - SectionID - Section identifier.
   - BaseContractID - Underlying futures contract identifier.
   - map<SecurityGroupID, SecurityTradingStatus> security_groups_map container. map — is a sorted associative container that holds key-value pairs with unique keys.
5. The handling of SecurityGroupStatus (f) messages is performed as described below.

   For each received message, the type of group status is determined using the HaltType field, and the corresponding processing is carried out:

   a. '0' - EmptyHalts.

      The message is transmitted in the topics of the Instrument Incremental stream.

      The client should clear the security_groups_map container within all instruments in the instrument_state. Then wait for receipt of instrument group statuses in the topics of the Instrument Incremental stream.

   b. '1' - Trading session level.

      The client should add an element to the security_groups_map within all instruments in instrument_state, containing SecurityGroupID and SecurityTradingStatus. If such an element already exists in the container, the client should update the SecurityTradingStatus field within the element using the value taken from the SecurityGroupStatus (f) message being processed.

   c. '2' - Section level.

      The client should identify all instruments in the instrument_state container where the value of SectionID matches the corresponding value from the SecurityGroupStatus (f) Smessage and add an element with SecurityGroupID and SecurityTradingStatus to their security_groups_map container. If such an element already exists in the container, the client should update the SecurityTradingStatus field within the element using the value taken from the SecurityGroupStatus (f) message being processed.

   d. '3' - Section + Trading mode level.

      The client should identify all instruments in the instrument_state container where the values of SectionID and TradeModeMask match the corresponding values from the SecurityGroupStatus (f) message and add an element with SecurityGroupID and SecurityTradingStatus to their security_groups_map container. If such an element already exists in the container, the client should update the SecurityTradingStatus field within the element using the value taken from the SecurityGroupStatus (f) message being processed.

   e. '4' - Underlying futures contract level.

      The client should identify all instruments in the instrument_state container where the value of BaseContractID matches the corresponding value from the SecurityGroupStatus (f) message and add an element with SecurityGroupID and SecurityTradingStatus to their security_groups_map container. If such an element already exists in the container, the client should update the SecurityTradingStatus field within the element using the value taken from the SecurityGroupStatus (f) message being processed.

   f. '5' - Instruments within the trading mode.

The client should identify all instruments in the instrument_state container where the value of TradeModeMask matches the corresponding value from the SecurityGroupStatus (f) message and add an element with SecurityGroupID and SecurityTradingStatus to their security_groups_map container. If such an element already exists in the container, the client should update the SecurityTradingStatus field within the element using the value taken from the SecurityGroupStatus (f) message being processed.

g. '6' - Instruments within the trading mode + Underlying futures contract level.

   The client should identify all instruments in the instrument_state container where the values of TradeModeMask and BaseContractID match the corresponding values from the SecurityGroupStatus (f) message and add an element with SecurityGroupID and SecurityTradingStatus to their security_groups_map container. If such an element already exists in the container, the client should update the SecurityTradingStatus field within the element using the value taken from the SecurityGroupStatus (f) message being processed.

h. '7' - Instrument group level.

   The client should identify all instruments in the instrument_state container where the GroupMask field includes a set bit from the GroupMask field (instrument_state.GroupMask & SecurityGroupStatus.GroupMask != 0) of the SecurityGroupStatus (f) message and add an element with SecurityGroupID and SecurityTradingStatus to their security_groups_map container. If such an element already exists in the container, the client should update the SecurityTradingStatus field within the element using the value taken from the SecurityGroupStatus (f) message being processed.

i. '8' - Instrument group + Underlying futures contract level.

   The client should identify all instruments in the instrument_state container where the GroupMask field includes a set bit from the GroupMask field (instrument_state.GroupMask & SecurityGroupStatus.GroupMask != 0) of the SecurityGroupStatus (f) message, and the value of BaseContractID matches the corresponding value from the SecurityGroupStatus (f) message and add an element with SecurityGroupID and SecurityTradingStatus to their security_groups_map container. If such an element already exists in the container, the client should update the SecurityTradingStatus field within the element using the value taken from the SecurityGroupStatus (f) message being processed.

6. If the client has received SecurityGroupStatus (f) messages during snapshot reading, they should process the messages saved in step 1 as described in step 5.

7. The client then switches to the topics of the Instrument Incremental stream. The client receives and processes SecurityGroupStatus (f) messages according to the algorithm described in step 5.

8. Determination of the current instrument status is performed as follows.

   Within the container instrument_state, the client iterates through the elements of the container map<SecurityGroupID, SecurityTradingStatus> security_groups_map and determines the current instrument status as the most restrictive (strictest) in order of decreasing priority based on the field values of SecurityTradingStatus.

5. Stream of anonymous orders and trades

The service provides the following operations:

- distributing the entire order book using protocol UDP Multicast;
- distributing active orders snapshot book using protocol UDP Multicast;
- rendering service TCP Recovery

The service distributes messages using protocol FAST (see sec. 5.2); all messages are coded according to the templates **OrdersLogMessage**, **BookMessage** (see sec. 5.3).

5.1. Application architecture

5.1.1. Streams

The service distributes data as follows:

- the main streams distribute messages Market Data - Incremental Refresh (X) as incremental refreshes for entire order book;
- the data refresh streams distribute messages Market Data - Snapshot / Full Refresh (W) as actual snapshots containing active orders on instruments;
- the history of entire order book refreshes for the current trading session is available via the missed messages request session (TCP Recovery (Historical Replay)).

The following FIX messages are supported:

- **Market Data Request** - request for missed data;
- **Market Data - Snapshot/ Full Refresh** – active orders snapshot;
- **Market Data - Incremental Refresh** – incremental refresh of entire order book.

5.1.2. Messages fragmentation

The message fragmentation is used when sending data via:

- main streams (Incremental);
- recovery streams (Snapshot).

You can find more information about message fragmentation above (see sec. 3.2.6). Also, the first message in an active orders snapshot contains the tag **7944-RouteFirst** with value **1**.

5.1.3. Instrument IDs

The unique instrument IDs from the trading system SPECTRA are transmitted in fields **SecurityID** of messages **Market Data - Incremental Refresh** and **Market Data - Snapshot/ Full Refresh**.

5.2. Message templates

There are two certain message templates used for distributing the entire order book data:

- **OrdersLogMessage** - see sec. 5.2.1
- **BookMessage** - see sec. 5.2.2

5.2.1. OrdersLogMessage

This template is used for data refresh purpose. Also, it is used by the service TCP Recovery.

```
<template name="OrdersLogMessage" id="29">
        <string name="ApplVerID" 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"/>
        <sequence name="MDEntries">
            <length name="NoMDEntries" id="268"/>
            <uInt32 name="MDUpdateAction" id="279"/>
            <string name="MDEntryType" id="269"/>
            <int64 name="MDEntryID" id="278" presence="optional"/>
            <uInt64 name="SecurityID" id="48" presence="optional"/>
            <uInt32 name="SecurityIDSource" id="22">
              <constant value="8"/>
            </uInt32>
            <uInt32 name="RptSeq" id="83" presence="optional"/>
            <uInt32 name="MDEntryDate" id="272" presence="optional"/>
            <uInt64 name="MDEntryTime" id="273"/>
            <decimal name="MDEntryPx" id="270" presence="optional"/>
            <int64 name="MDEntrySize" id="271" presence="optional"/>
            <decimal name="LastPx" id="31" presence="optional"/>
            <int64 name="LastQty" id="32" presence="optional"/>
            <int64 name="TradeID" id="1003" presence="optional"/>
            <uInt32 name="ExchangeTradingSessionID" id="5842" presence="optional"/>
            <int64 name="MDFlags" id="20017" presence="optional"/>
            <int64 name="MDFlags2" id="20050" presence="optional"/>
            <uInt64 name="Revision" id="20018" presence="optional"/>
        </sequence>
    </template>
```

5.2.2. BookMessage

This template is used for distributing snapshots.

```
<template name="BookMessage" id="30">
        <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="LastMsgSeqNumProcessed" id="369"/>
        <uInt32 name="RptSeq" id="83" presence="optional"/>
        <uInt32 name="LastFragment" id="893"/>
        <uInt32 name="RouteFirst" id="7944"/>
        <uInt32 name="ExchangeTradingSessionID" id="5842"/>
        <uInt64 name="SecurityID" id="48" presence="optional"/>
        <uInt32 name="SecurityIDSource" id="22">
            <constant value="8"/>
        </uInt32>
        <sequence name="MDEntries">
            <length name="NoMDEntries" id="268"/>
            <string name="MDEntryType" id="269"/>
            <int64 name="MDEntryID" id="278" presence="optional"/>
            <uInt32 name="MDEntryDate" id="272" presence="optional"/>
            <uInt64 name="MDEntryTime" id="273"/>
            <decimal name="MDEntryPx" id="270" presence="optional"/>
            <int64 name="MDEntrySize" id="271" presence="optional"/>
            <int64 name="TradeID" id="1003" presence="optional"/>
            <int64 name="MDFlags" id="20017" presence="optional"/>
            <int64 name="MDFlags2" id="20050" presence="optional"/>
        </sequence>
    </template>
```

5.3. Messages Market Data - Incremental Refresh (X)

5.3.1. Adding a new order

Adds an order into the pool of active orders.

Tag	Field	Manda-tory	Details
279	MDUpdateAction	Y	Incremental refresh type: "**0**" (New).
269	MDEntryType	Y	Record type: • **0** - Bid; • **1** - Ask.
278	MDEntryID	Y	Order ID
48	SecurityID	Y	Instrument numeric code.
83	RptSeq	Y	Incremental refresh sequence number.
272	MDEntryDate	N	Record last change date. Value NULL in field MDEntryDate indicates the current date.
273	MDEntryTime	Y	Record last change time. Format: **HHMMSSssssssss**.
270	MDEntryPx	Y	Order price.
271	MDEntrySize	Y	Volume, quantity.
31	LastPx	N	Is missing
32	LastQty	N	Is missing
1003	TradeID	N	Is missing
5842	ExchangeTradingSessionID	Y	Trading session ID
20017	MDFlags	Y	The field is a bit mask: • 0x01 - Day order • 0x02 - IOC order • 0x04 - OTC order • 0x1000 - End of transaction bit • 0x80000 - FOK order • 0x100000 - The record results from replacing the order • 0x4000000 - Negotiated order • 0x8000000 - Multi-leg order • 0x80000000 - Negotiated order matched by reference • 0x200000000000 - Synthetic order • 0x400000000000 - Order originated in RFS • 0x2000000000000000 - Passive synthetic order • 0x1000000000000000 - Book-or-cancel order (Passive only) • 0x4000000000000000 - Sign of an order/trade during a discrete auction
20018	Revision	Y	Service field (replication system)
20050	MDFlags2	Y	Bitmask extension (in addition to the MDFlags field)

5.3.2. Deleting an order

Deletes an order from the pool of active orders according to the order's ID.

Tag	Field	Manda-tory	Details
279	MDUpdateAction	Y	Incremental refresh type: "**2**" (Delete).
269	MDEntryType	Y	Record type: • **0** - Bid; • **1** - Ask.
278	MDEntryID	Y	Order ID
48	SecurityID	Y	Instrument numeric code.

Tag	Field	Mandatory	Details
83	RptSeq	Y	Incremental refresh sequence number.
272	MDEntryDate	N	Record last change date. Value NULL in field MDEntryDate indicates the current date.
273	MDEntryTime	Y	Record last change time. Format: **HHMMSSssssssss**.
270	MDEntryPx	Y	Order price.
271	MDEntrySize	Y	Volume to delete.
31	LastPx	N	Is missing
32	LastQty	N	Is missing
1003	TradeID	N	Is missing
5842	ExchangeTradingSessionID	Y	Trading session ID
20017	MDFlags	Y	The field is a bit mask: • 0x01 - Day order • 0x02 - IOC order • 0x04 - OTC order • 0x1000 - End of transaction bit • 0x2000 - Sign of canceling a passive cross-order • 0x80000 - FOK order • 0x100000 - The record results from replacing the order • 0x200000 - The record results from cancelling the order • 0x400000 - The record results from mass cancelling • 0x4000000 - Negotiated order • 0x8000000 - Multi-leg order • 0x80000000 - Negotiated order matched by reference • 0x20000000 - Flag of cancelling the left balance of the order because of a cross-trade • 0x100000000 - The record results from cancelling an order via ‘Cancel on Disconnect’ service. • 0x400000000000 - Order originated in RFS • 0x1000000000000000 - Book-or-cancel order (Passive only) • 0x4000000000000000 - Sign of an order/trade during a discrete auction
20018	Revision	Y	Service field (replication system)
20050	MDFlags2	Y	Bitmask extension (in addition to the MDFlags field)

5.3.3. Partial order matching

When an order is partly matched into a trade, its attributes in the pool of active order change.

Tag	Field	Mandatory	Details
279	MDUpdateAction	Y	Incremental refresh type: "**1**" (Change).
269	MDEntryType	Y	Record type: • **0** - Bid • **1** - Ask
278	MDEntryID	Y	Order ID
48	SecurityID	Y	Instrument numeric code.
83	RptSeq	Y	Incremental refresh sequence number.
272	MDEntryDate	N	Record last change date. Value NULL in field MDEntryDate indicates the current date.

Tag	Field	Mandatory	Details
273	MDEntryTime	Y	Record last change time. Format: **HHMMSSssssssss**.
270	MDEntryPx	Y	Order price.
271	MDEntrySize	Y	Instrument units left in order
31	LastPx	Y	Matched trade price.
32	LastQty	Y	Trade volume.
1003	TradeID	Y	Trade ID
5842	ExchangeTradingSessionID	Y	Trading session ID
20017	MDFlags	Y	The field is a bit mask: • 0x1 - Trade by Day-order • 0x2 - Trade by IOC-order • 0x4 – OTC-trade including clearing, negotiated, multi-leg, RFS trades • 0x1000 - End of transaction bit • 0x80000 - Trade by Fill-or-kill order • 0x4000000 - Negotiated trade • 0x8000000 - Multi-leg trade. Applied to all multi-leg transactions • 0x80000000 - Negotiated trade matched by reference • 0x20000000000 – The active side in the trade. The order that led to the trade when added to the order-book • 0x40000000000 – The passive side in the trade. The order from the order-book involved in the trade • 0x400000000000 - Trade originated in RFS • 0x1000000000000000 - Trade by Book-or-cancel order (Passive only) • 0x4000000000000000 - Sign of an order/trade during a discrete auction
20018	Revision	Y	Service field (replication system)
20050	MDFlags2	Y	Bitmask extension (in addition to the MDFlags field)

5.3.4. Full order matching

After an order is fully matched into a trade, this order will be deleted from the pool of active orders.

Tag	Field	Mandatory	Details
279	MDUpdateAction	Y	Incremental refresh type. If deleted, then "**2**" (Delete).
269	MDEntryType	Y	Record type: • **0** - Bid • **1** - Ask
278	MDEntryID	Y	Order ID
48	SecurityID	Y	Instrument numeric code.
83	RptSeq	Y	Incremental refresh sequence number.
272	MDEntryDate	N	Record last change date. Value NULL in field MDEntryDate indicates the current date.
273	MDEntryTime	Y	Record last change time. Format: **HHMMSSssssssss**.
270	MDEntryPx	Y	Order price.
271	MDEntrySize	N	Is missing
31	LastPx	Y	Matched trade price.
32	LastQty	Y	Trade volume.

Tag	Field	Mandatory	Details
1003	TradeID	Y	Trade ID
5842	ExchangeTradingSessionID	Y	Trading session ID
20017	MDFlags	Y	The field is a bit mask: • 0x1 - Trade by Day-order • 0x2 - Trade by IOC-order • 0x4 – OTC-trade including clearing, negotiated, multi-leg, RFS trades • 0x1000 - End of transaction bit • 0x80000 - Trade by Fill-or-kill order • 0x4000000 - Negotiated trade • 0x8000000 - Multi-leg trade. Applied to all multi-leg transactions • 0x80000000 - Negotiated trade matched by reference • 0x20000000000 – The active side in the trade. The order that led to the trade when added to the order-book • 0x40000000000 – The passive side in the trade. The order from the order-book involved in the trade • 0x200000000000 - Synthetic order • 0x400000000000 - Trade originated in RFS • 0x1000000000000000 - Trade by Book-or-cancel order (Passive only) • 0x4000000000000000 - Sign of an order/trade during a discrete auction
20018	Revision	Y	Service field (replication system)
20050	MDFlags2	Y	Bitmask extension (in addition to the MDFlags field)

5.3.5. Technical trades

Detailed information on trade types is given in the document **'SPECTRA Plaza-2 gate'** in section **2.4.3. 'Trade types, created upon exercising and expiration of futures and options**' - http://ftp.moex.com/pub/ClientsAPI/Spectra/CGate/prod/docs/p2gate_en.pdf.

Tag	Field	Mandatory	Details
279	MDUpdateAction	Y	Incremental refresh type: "**0**" (New).
269	MDEntryType	Y	Record type: • **0** - Bid • **1** - Ask
278	MDEntryID	Y	Order ID
48	SecurityID	Y	Instrument numeric code.
83	RptSeq	Y	Incremental refresh sequence number.
272	MDEntryDate	N	Record last change date. Value NULL in field MDEntryDate indicates the current date.
273	MDEntryTime	Y	Record last change time. Format: **HHMMSSssssssss**.
270	MDEntryPx	N	Is missing.
271	MDEntrySize	N	Is missing
31	LastPx	Y	Matched trade price.
32	LastQty	Y	Trade volume.
1003	TradeID	Y	Trade ID
5842	ExchangeTradingSessionID	Y	Trading session ID
20017	MDFlags	Y	The field is a bit mask: • 0x4 – OTC-trade including clearing, negotiated, multi-leg, RFS trades

Tag	Field	Mandatory	Details
			• 0x8 – Position transfer between BFs • 0x20 – Option exercise trade • 0x80 – Flag of instrument expiration (exercise for futures, expiration for options) • 0x1000 - End of transaction bit • 0x4000 – Far leg transaction • 0x10000 - Trade Settlement TAS • 0x800000 – Option expiration trade • 0x2000000 – Off-book clearing trade. Applied to all clearing trades • 0x8000000 – Multi-leg trade • 0x40000000 – Futures exercise trade • 0x400000000000 - Trade originated in RFS • 0x10000000000000 - The technical trade as a result of exiting a daily futures contract with automatic prolongation (based on the submitted requests) • 0x400000000000000 - The technical trade as a result of forced exiting a daily futures contract with automatic prolongation (realization of unsatisfied demand) • 0x800000000000000 - The technical trade with multi-leg instrument as a result of exiting a daily futures contract with automatic prolongation
20018	Revision	Y	Service field (replication system)
20050	MDFlags2	Y	Bitmask extension (in addition to the MDFlags field)

5.3.6. Cleaning up active orders pool (specified trading session)

After receiving the message, all orders of the specified trading session are to be deleted on the client side.

Tag	Field	Mandatory	Details
279	MDUpdateAction	Y	Incremental refresh type: "**0**" (New).
269	MDEntryType	Y	Record type: "**J**" (Empty Book).
278	MDEntryID	N	Is missing
48	SecurityID	N	Is missing
83	RptSeq	N	Is missing
272	MDEntryDate	N	Record last change date. Value NULL in field MDEntryDate indicates the current date.
273	MDEntryTime	Y	Record last change time. Format: **HHMMSSssssssss**.
270	MDEntryPx	N	Is missing
271	MDEntrySize	N	Is missing
31	LastPx	N	Is missing
32	LastQty	N	Is missing
1003	TradeID	N	Is missing
5842	ExchangeTradingSessionID	Y	Trading session ID
20017	MDFlags	N	Is missing
20018	Revision	Y	Service field (replication system)
20050	MDFlags2	N	Is missing

5.3.7. Cleaning up active orders pool (all trading sessions)

After receiving the message, all orders are to be deleted on the client side. After that, you should perform steps 4-7 in the section **Connect client** - see sec. 2.1.

Tag	Field	Mandatory	Details
279	MDUpdateAction	Y	Incremental refresh type: "**0**" (New).
269	MDEntryType	Y	Record type: "**J**" (Empty Book).
278	MDEntryID	N	Is missing
48	SecurityID	N	Is missing
83	RptSeq	N	Is missing
272	MDEntryDate	N	Record last change date. Value NULL in field MDEntryDate indicates the current date.
273	MDEntryTime	Y	Record last change time. Format: **HHMMSSssssssss**.
270	MDEntryPx	N	Is missing
271	MDEntrySize	N	Is missing
31	LastPx	N	Is missing
32	LastQty	N	Is missing
1003	TradeID	N	Is missing
5842	ExchangeTradingSessionID	N	Is missing
20017	MDFlags	N	Is missing
20018	Revision	N	Is missing
20050	MDFlags2	N	Is missing

5.4. Messages Market Data - Snapshot / Full Refresh

5.4.1. Active orders snapshot

Snapshots are distributed as one or several messages (for each instrument).

Tag	Field	Mandatory	Details
<Standard Message Header>		Y	Message type 'W'
369	LastMsgSeqNumProcessed	Y	The 'MsgSeqNum' of the last message sent into incremental feed at the time of the current snapshot generation.
83	RptSeq	Y	The 'RptSeq' number of the last incremental update included in the current market data snapshot for instrument.
893	LastFragment	Y	Indicates the last message in the message group for the instrument. Allowable values: • **0** – not the last message • **1** – the last message
7944	RouteFirst	Y	Indicates whether the message is the first of the ones generated for the given instrument or not. Allowable values: • **0** – not the first message • **1** – the first message
5842	ExchangeTradingSessionId	Y	Trading session ID
48	SecurityID	Y	Instrument numeric code
22	SecurityIDSource	C	**8** - Exchange Symbol
268	NoMDEntries	Y	Number of 'MDEntry' records in the current message.
=>269	MDEntryType	Y	Record type: • **0** - Bid • **1** - Ask
=>278	MDEntryID	Y	Order ID.
=>272	MDEntryDate	N	Record last change date.
=>273	MDEntryTime	Y	Record last change time. Format: **HHMMSSssssssss**.
=>270	MDEntryPx	Y	Order price.

Tag	Field	Mandatory	Details
=>271	MDEntrySize	Y	Volume, quantity.
=>1003	TradeID	C	None if no trade was matched, else contains last trade ID.
=>20017	MDFlags	Y	The field is a bit mask: • 0x01 - Day order • 0x04 - OTC order • 0x4000000 - Negotiated order
=>20050	MDFlags2	Y	Bitmask extension (in addition to the MDFlags field)

5.4.2. Empty active orders snapshot - the message Empty book (MDEntryType = J)

If the active orders snapshot is empty, and a trading session has not yet started, the snapshot will not be distributed. If the active orders snapshot gets empty during a trading session, it will be distributed in the form of the message below.

Tag	Field	Mandatory	Details
<Standard Message Header>		Y	Message type 'W'
369	LastMsgSeqNumProcessed	Y	The 'MsgSeqNum' of the last message sent into incremental feed at the time of the current snapshot generation.
83	RptSeq	Y	The 'RptSeq' number of the last incremental update included in the current market data snapshot for instrument.
893	LastFragment	Y	Indicates the last message in the message group for the instrument. Allowable values: • **1** – the last message
7944	RouteFirst	Y	Indicates whether the message is the first of the ones generated for the given instrument or not. Allowable values: • **1** – the first message
5842	ExchangeTradingSessionId	Y	Trading session ID.
48	SecurityID	Y	Instrument numeric code.
22	SecurityIDSource	C	**8** - Exchange Symbol
268	NoMDEntries	Y	Number of 'MDEntry' records in the current message.
=>269	MDEntryType	Y	Record type: • **J** - Empty Book
=>278	MDEntryID	N	Is missing
=>272	MDEntryDate	N	Record last change date.
=>273	MDEntryTime	Y	Record last change time. Format: **HHMMSSssssssss**.
=>270	MDEntryPx	N	Is missing
=>271	MDEntrySize	N	Is missing
=>1003	TradeID	N	Is missing
=>20017	MDFlags	N	Is missing
=>20050	MDFlags2	N	Is missing

5.5. Active orders book

To make order-book for active orders from the Incremental stream, only orders and trades in which the flag (0x4 (NonQuote) - an OTC order or an OTC trade) is not set should be processed in OrdersLogMessage messages.

When receiving an active order-book from the Snapshot stream in BookMessage messages, only orders in which the flag (0x4 (NonQuote) - an OTC order) is not set should be processed.

It should be taken into account that if a user builds their own order-book from the ORDERS-LOG and compares it with the data transmitted in the FO-BOOK-N streams (FO-BOOK-1, FO-BOOK-5, FO-BOOK-20, FO-BOOK-50), the order-books will differ. These differences are caused by the transmission of synthetic liquidity in the FO-BOOK-N, which the client needs to calculate independently in the case of ORDERS-LOG. See section "6. Synthetic matching".

6. Synthetic matching

Synthetic matching - the trades formation on the basis of orders from different order-books (order-books of different instruments). The purpose of synthetic matching is to increase the instruments liquidity by combining several order-books. For example, synthetic matching allows match calendar spread orders not only with a counter order inside this instrument order-book, but also with orders from order-books of its futures-legs. Thus, the calendar spread order takes into account the counter interest from other order-books of their legs.

Detailed information on synthetic matching is given in the document **'SPECTRA Plaza-2 gate'** in section **2.9. 'Synthetic matching'** - http://ftp.moex.com/pub/ClientsAPI/Spectra/CGate/prod/docs/p2gate_en.pdf.

Below, in this section, an example of accounting for synthetic liquidity when processing incremental updates of the order-books in the FO-BOOK-1, FO-BOOK-5, FO-BOOK-20, and FO-BOOK-50 streams is provided. The synthetic liquidity in the order-book from the ORDERS-LOG stream, if necessary, the client must calculate on their side. Descriptions are available at the following links:

- https://www.moex.com/ru/spreads/;
- https://fs.moex.com/files/21418.

6.1. Sample representation of Aggregate Book incremental refresh processing for synthetic matching

For example, we have three instruments:

- SecurityID=111 - RTS-3.18-6.18 - calendar spread;
- SecurityID=222 - RTS-3.18 - near future;
- SecurityID=333 - RTS-6.18 - far future.

The Aggregate Books for these instruments are completely empty.

The price level updating by a regular order

A regular order is an order submitted by a user. Such an order enters the order-book and forms a price level with normal liquidity in it.

Participant A added a buy order for the RTS-3.18 instrument at a price of 127.050 in the volume of 10 contracts.

A new price level is created in the order book as a result of placing an order.

Here's an excerpt of update messages that refers to existing Market Data Entry:

Tag	Field	Description
279	MDUpdateAction	0 = New
48	SecurityID	222
22	SecurityIDSource	8
269	MDEntryType	0 = Bid
270	MDEntryPx	127,050
271	MDEntrySize	10
264	MarketDepth	5
20039	MDEntrySyntheticSize	0

Order Books for instruments will be updated:

RTS-3.18				
Synth. size	Size	Price	Size	Synth. size
0	10	127,05		

RTS-6.18				
Synth. size	Size	Price	Size	Synth. size

RTS-3.18-6.18				
Synth. size	Size	Price	Size	Synth. size

The price level updating by a synthetic order

Participant B added a sell order for the RTS-6.18 instrument at a price of 128.150 in the volume of 20 contracts.

A synthetic sell order for the calendar spread RTS-3.18-6.18 at a price of 128.150 - 127.050 = 1,100 in the volume of 10 contracts is automatically generated based on two orders for futures, forming a new price level with synthetic liquidity in the order book for calendar spread.

Here's an excerpt of update messages that refers to existing Market Data Entry:

Tag	Field	Description
279	MDUpdateAction	0 = New
48	SecurityID	333
22	SecurityIDSource	8
269	MDEntryType	1 = Ask
270	MDEntryPx	128,150
271	MDEntrySize	20
264	MarketDepth	5
20039	MDEntrySyntheticSize	0
279	MDUpdateAction	0 = New
48	SecurityID	111
22	SecurityIDSource	8
269	MDEntryType	1 = Ask
270	MDEntryPx	1,100
271	MDEntrySize	10
264	MarketDepth	5
20039	MDEntrySyntheticSize	10

Order Books for instruments will be updated:

RTS-3.18				
Synth. size	Size	Price	Size	Synth. size
0	10	127,05		

RTS-6.18				
Synth. size	Size	Price	Size	Synth. size
		128,15	20	0

RTS-3.18-6.18				
Synth. size	Size	Price	Size	Synth. size
		1,1	10	10

Updating of the price level with synthetic liquidity by a regular order

Participant D added a sell order for the RTS-3.18-6.18 calendar spread at a price of 1,100 in the volume of 5 contracts.

The price level in the order book for the calendar spread will be updated as a result of this order.

Here's an excerpt of update messages that refers to existing Market Data Entry:

Tag	Field	Description
279	MDUpdateAction	1 = Change
48	SecurityID	111
22	SecurityIDSource	8
269	MDEntryType	1 = Ask
270	MDEntryPx	1,100
271	MDEntrySize	15
264	MarketDepth	5
20039	MDEntrySyntheticSize	10

Order Books for instruments will be updated:

RTS-3.18				
Synth. size	Size	Price	Size	Synth. size
0	10	127,05		

RTS-6.18				
Synth. size	Size	Price	Size	Synth. size
		128,15	20	0

RTS-3.18-6.18				
Synth. size	Size	Price	Size	Synth. size
		1,1	15	10

Performing synthetic matching

Participant C added a buy order for the RTS-3.18-6.18 calendar spread at a price of 1,100 in the volume of 10 contracts.

In the process of synthetic matching, two synthetic orders will be generated for near and far futures:

- for the instrument RTS-3.18 to sell at a price of 127,050 in the volume of 10 contracts;
- for the instrument RTS-6.18 to buy at a price of 128,150 in the volume of 10 contracts.

As a result of synthetic matching, orders for three instruments are executed to the trades.

The orders have been executed to the trades, so the corresponding price levels will be deleted from the order books or updated.

Here’s an excerpt of update messages that refers to existing Market Data Entry for price level deleting:

Tag	Field	Description
279	MDUpdateAction	2 = Delete
48	SecurityID	222
22	SecurityIDSource	8
269	MDEntryType	0 = Bid
270	MDEntryPx	127,050
271	MDEntrySize	0
264	MarketDepth	5
20039	MDEntrySyntheticSize	0
279	MDUpdateAction	1 = Change
48	SecurityID	333
22	SecurityIDSource	8
269	MDEntryType	1 = Ask
270	MDEntryPx	128,150
271	MDEntrySize	10
264	MarketDepth	5
20039	MDEntrySyntheticSize	0
279	MDUpdateAction	1 = Change
48	SecurityID	111
22	SecurityIDSource	8
269	MDEntryType	1 = Ask
270	MDEntryPx	1,100
271	MDEntrySize	5
264	MarketDepth	5
20039	MDEntrySyntheticSize	0

Order Books for instruments will be updated:

RTS-3.18				
Synth. size	Size	Price	Size	Synth. size

RTS-6.18				
Synth. size	Size	Price	Size	Synth. size
		128,15	10	0

RTS-3.18-6.18				
Synth. size	Size	Price	Size	Synth. size
		1,1	5	0

7. TCP Recovery (Historical Replay) service limitations

The following limitations are applied to the TCP Recovery service for streams SPOT-TRADES, ORDERS-LOG in order to lower the load:

Parameter	Value	Details
Maximum active connections, per market, per instance, per IP address	2	You can establish no more than indicated number active TCP connection from single IP address. An attempt to make more connections will be rejected
Maximum connections count, per market, per instance, per day, per IP address	1000	You can make no more than indicated number of tcp connections per IP address per day. Extra connection attempts will be rejected
Maximum number of messages to request	1000	TCP replay request is rejected if a number of requested messages is greater than indicated value
Marketdata request timeout, seconds	1	Connection is terminated with logout message if marketdata request is not received within indicated number of seconds since logon message. TCP session is terminated if no confirming logout is received after server-side logout.

The following limitations are applied to the TCP Recovery service for streams FO-TRADES, INDEX, NEWS, in order to lower the load:

Parameter	Value	Details
Maximum active connections, per market, per instance, per IP address	2	You can establish no more than indicated number active TCP connection from single IP address. An attempt to make more connections will be rejected
Maximum connections count, per market, per instance, per day, per IP address	15000	You can make no more than indicated number of tcp connections per IP address per day. Extra connection attempts will be rejected
Maximum number of messages to request	1000	TCP replay request is rejected if a number of requested messages is greater than indicated value
Marketdata request timeout, seconds	1	Connection is terminated with logout message if marketdata request is not received within indicated number of seconds since logon message. TCP session is terminated if no confirming logout is received after server-side logout.