

Moscow Exchange

FIX-protocol for Request for Stream service

User guide

Moscow Exchange
Version 1.0.1
June 8th 2020

Contents

Overview	5
Document purpose	5
Service description	6
Document history	7
FIX component blocks	7
Standard message header	7
Parties	8
Standard message trailer	9
FIX session-level messages	9
Logon (A).....	9
Logout (5)	10
Heartbeat (0)	11
Test Request (1)	11
Resend Request (2)	12
Sequence Reset (4).....	13
Reject (3).....	14
FIX session establishment and termination scenario	15
Establish connection	15
Resend messages mechanism.....	16
Session status check	16
Reset sequence numbers	17
Close (Terminate) session	17
Reestablish session after failure	17

Messages common for both role: Maker and Taker	19
Security List Request (x).....	19
Security List (y)	19
Messages from Client to Server. Role: Maker	20
RFQ Request (AH).....	20
QuoteRequestReject (AG)	20
Quote ('S').....	21
QuoteCancel ('Z')	22
Messages from Server to Client. Role: Maker	23
Quote Request ('R')	23
Quote Response ('AJ').....	24
Execution Report ('8')	25
Quote ('S').....	26
Quote Status Report ('AI').....	27
Messages from Client to Server. Role: Taker	28
Quote Request ('R')	28
Quote Response ('AJ').....	29
Messages from Server to Client. Role:Taker	30
Quote ('S').....	30
Quote Request Reject ('AG').....	31
Quote Response ('AJ').....	31
Execution Report ('8')	32

Overview

Document purpose

The document below describes the FIX protocol provided by the Moscow Exchange for connection to Request for Stream of FX Market. The description based on the standard FIX protocol (Financial Information Exchange, <http://www.fixtrading.org>, version 4.4) specification. It assumed that users have basic knowledge about FIX standard. The specification does not contain neither technical nor administrative details on network connection or security protection methods.

RFS FIX Server supports only messages, component blocks and fields that described in this document.

Note that all fields which required or conditionally required by FIX 4.4 standard but absent in MOEX Interface specification are optional and will be ignored by MOEX. All field values, which are valid according to FIX 4.4 standard but are not described in this document, will be considered as invalid and messages with such values will be rejected.

Each message or component block is represented as table, where each row is a message field or component block. The following characteristics described for each field:

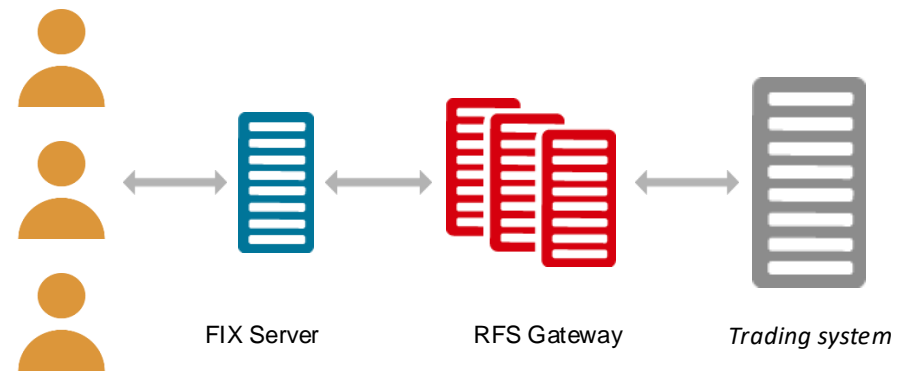
- Tag – unique field identifier.
- Name – field name.
- Required – shows whether the field is required or not in appropriate message or component block.
 - 'Y' – tag is required (mandatory);
 - 'N' – tag is not required (optional);
 - 'C' – tag is conditionally required.
 - 'Y*' – tag is required by MOEX, but not required by the standard FIX 4.4 protocol;
 - 'N*' – tag is not required by MOEX but required by the standard FIX 4.4 protocol;
 - 'C*' – tag is conditionally required by MOEX, but not required by the standard FIX 4.4 protocol.
- Type – field type.
- Valid values – list of valid tag values;
- Comments – comments, additional information for the tag.

Service description

Base service functionality:

- User authorization \ authentication in Trading System
- Initiation of RFS auctions
- Receiving and sending of quotes
- Trade confirmation

Schematically the work of the service is shown in the diagram below.



Document history

Issue	Date	Description
1.0.1	08.06.2020	Field Anonymity(6101) added to the message QuoteRequest.

FIX component blocks

Standard message header

A standard header precedes each administrative or application message. The header identifies the message type, length, destination, sequence number, origination point and time.

N.B. Our service does not support incoming messages with 43(PossDupFlag)=Y or 97(PossResend)=Y tags.

Tag	Field name	Required	Type	Valid values	Comments
8	BeginString	N	String (7)	'FIX.4.4'	Identifies beginning of new message and protocol version. Always unencrypted, must be first field in message.
9	BodyLength	N	Length		Message length, in bytes, forward to the CheckSum field. Always unencrypted, must be second field in message.
35	MsgType	N	String (10)		Defines message type. Always unencrypted, must be third field in message.
49	SenderCompID	N	String (12)		Assigned value used to identify firm sending message. Always unencrypted. If this message sent to MOEX, then it should contain USERID assigned to a trader by MOEX.

					If this message sent from MOEX, then it will contain the MOEX server identifier. This parameter is given by MOEX
56	TargetCompID	N	String		Assigned value used to identify receiving firm. Always unencrypted. If this message sent from MOEX, then it will contain USERID assigned to a trader by MOEX. If this message sent to MOEX, then it should contain the MOEX server identifier. This parameter is given by MOEX
34	MsgSeqNum	N	SeqNum		Message sequence number.
52	SendingTime	N	UTCTimestamp		Time of message transmission (expressed in UTC). Field format is YYYYMMDD-HH:MM:SS.sss.
122	OrigSendingTime	Y	UTCTimestamp		Original time of message transmission when transmitting messages as the result of resend request (expressed in UTC). Field format is YYYYMMDD-HH:MM:SS.sss. Required for message resend as a result of a resend request.

Parties

The <Parties> component block is used to define parties of order or trade. For each party the following fields should be defined: PartyID, PartyIDSource, PartyRole.

- PartyID (448) = <client code>, PartyIDSource (447) = 'D', PartyRole (452) = '3' – specifies client;

Tag	Field name	Required	Type	Valid values	Comments
453	NoPartyID	Y*	NumInGroup		Number of repeating PartyID group entries.
448	=> PartyID	C	String (12)		Party identifier/code. Required if PartyIDSource is specified. Required if NoPartyIDs > 0. User, firm, client can be defined as party of order/trader. Note: this field must contain MOEX CLIENTCODE value

					that is assigned by broker to a client. For own broker's accounts this value is ignored and not returned in execution reports.
447	=> PartyIDSource	C	char	'D' (Proprietary/Custom code)	Identifies class or source of the PartyID (448) value. Required if PartyID (448) is specified. Note: applicable values depend upon PartyRole (452) specified. Only constant value 'D' is used.
452	=> PartyRole	C	int	'1' (Executing Firm); '3' (Client ID); '12' (Executing Trader); '17' (Contra Firm).	Identifies the type or role of the PartyID (448) specified. '1' is used for firm; '3' is used for client in New Order – Single (D) message; '12' is used for user (trader, broker). '17' is used for counterparty.

Standard message trailer

A standard trailer terminates each message, administrative or application. The trailer is used to segregate messages and contains the three-digit character representation of the Checksum value.

Tag	Field name	Required	Type	Valid values	Comments
10	Checksum	N	String(3)		Three byte, simple checksum. Always unencrypted, always last field in message.

FIX session-level messages

The session level messages are used to establish, close (terminate), support FIX session, manage session status and opportunity to reestablish it after failure.

Logon (A)

The logon message is used to initiate FIX session and to confirm the establishing of it. The logon message must be the first message sent by the application requesting to initiate a FIX session. The Logon reply normally can take up to 3 seconds to receive. Please allow 3 seconds waiting time for reply

Note: SenderCompID cannot be used for more than single connection to any MOEX service.

Tag	Field name	Required	Type	Valid values	Comments
<Standard Message Header>		Y			MsgType = 'A'
98	EncryptMethod	Y	int	'0' (None)	Method of encryption. Always unencrypted. Note: Encryption is not supported by MOEX.
108	HeartBtInt	Y	int		Heartbeat interval (seconds). Values must fit in limits between 1 and 60. If HeartBtInt field value is out of range, then Logout (35=5) message is sent with text description of error.
141	ResetSeqNumFlag	N	Boolean	'Y' (Yes) 'N' (No)	Indicates if the both sides of the FIX session should reset sequence numbers. Default value is 'Y'
554	Password	Y*	String(8)		User password. The maximum length is 8 characters. Required field at MOEX
1409	SessionStatus	N	Char	'0' (Session active) '5' (Wrong password or user ID) '7' (Logons are not allowed \ UserID already in use)	Status of the request to change the password. Required if the server generates the message.
6936	LanguageID	N	Char	'R' (Russian) 'E' (English)	Language of Trading System messages.
<Standard Message Trailer>		Y			

Logout (5)

The logout message initiates or confirms the termination of a FIX session.

Tag	Field name	Required	Type	Valid values	Comments
	<Standard Message Header>	Y			MsgType = '5'
58	Text	N	String		Logout reason.
	<Standard Message Trailer>	Y			

Heartbeat (0)

The Heartbeat monitors the status of the communication link and identifies when the last of a string of messages was not received.

Heartbeats issued as the result of Test Request must contain the TestReqID transmitted in the Test Request message. This is useful to verify that the Heartbeat is the result of the Test Request and not as the result of a regular timeout.

Tag	Field name	Required	Type	Valid values	Comments
	<Standard Message Header>	Y			MsgType = '0'
112	TestReqID	N	String		Identifier included in Test Request (1) message to be returned in resulting Heartbeat (0).
	<Standard Message Trailer>	Y			

Test Request (1)

The test request message forces a heartbeat from the opposing application. The test request message checks sequence numbers or verifies communication line status. The opposite application responds to the Test Request with a Heartbeat containing the TestReqID.

The TestReqID verifies that the opposite application is generating the heartbeat as the result of Test Request and not a normal timeout. The opposite application includes the TestReqID in the resulting Heartbeat. Any string can be used as the TestReqID (one suggestion is to use a timestamp string).

Tag	Field name	Required	Type	Valid values	Comments
	<Standard Message Header>	Y			MsgType = '1'
112	TestReqID	Y	String		Identifier included in Test Request (1) message to be returned in resulting Heartbeat (0).
	<Standard Message Trailer>	Y			

Resend Request (2)

Receiving application sends the resend request to initiate the retransmission of messages. This function is utilized if a sequence number gap is detected, if the receiving application lost a message, or as a function of the initialization process. The resend request can be used to request a single message, a range of messages or all messages subsequent to a particular message.

N.B. Our service support Resend Request message but there is no resending for Quote and Quote Request messages.

N.B. Our service does not support incoming messages with 43(PossDupFlag)=Y или 97(PossResend)=Y tags.

Tag	Field name	Required	Type	Valid values	Comments
	<Standard Message Header>	Y			MsgType = '2'
7	BeginSeqNo	Y	SeqNum		Message sequence number of first message in range to be resent.

16	EndSeqNo	Y	SeqNum		Message sequence number of last message in range to be resent. If request is for a single message BeginSeqNo (7) = EndSeqNo (16). If request is for all messages subsequent to a particular message, EndSeqNo (16) = '0' (representing infinity).
<Standard Message Trailer>		Y			

Sequence Reset (4)

The Sequence Reset message has two modes: Gap Fill mode and Reset mode.

Gap Fill mode is used in response to a Resend Request when one or more messages must be skipped over for the following reasons:

- During normal resend processing, the sending application may choose not to send a message (e.g. an aged order).
- During normal resend processing, a number of administrative messages are skipped and not resent (such as Heart Beats, Test Requests).

Gap Fill mode is indicated by GapFillFlag (tag 123) field = "Y". If the GapFillFlag field is present (and equal to "Y"), the MsgSeqNum should conform to standard message sequencing rules (i.e. the MsgSeqNum of the Sequence Reset GapFill mode message should represent the beginning MsgSeqNum in the GapFill range because the remote side is expecting that next message sequence number).

Tag	Field name	Required	Type	Valid values	Comments
<Standard Message Header>		Y			MsgType = '4'
123	GapFillFlag	N	Boolean	Y' (Gap Fill message, MsgSeqNum field valid) 'N' (Sequence Reset, ignore MsgSeqNum)	Indicates that the Sequence Reset (4) message is replacing administrative or application messages which will not be resent.
36	NewSeqNo	Y	SeqNum		New sequence number.
<Standard Message Trailer>		Y			

Reject (3)

The reject message should be issued when a message is received but cannot be properly processed due to a session-level rule violation. An example of when a reject may be appropriate would be the receipt of a message with invalid basic data (e.g. MsgType=&) which successfully passes CheckSum and BodyLength checks. Generation and receipt of a Reject message indicates a serious error that may be the result of faulty logic in either the sending or receiving application.

Tag	Field name	Required	Type	Valid values	Comments
	<Standard Message Header>	Y			MsgType = '3'
45	RefSeqNum	Y	SeqNum		MsgSeqNum (34) of rejected message.
371	RefTagID	N	int		The tag number of the FIX field being referenced.
372	RefMsgType	N	String(10)		The MsgType (35) of the FIX message being referenced.
373	SessionRejectReason	N	int	'0' (Invalid tag number) '1' (Required tag missing) '2' (Tag not defined for this message type) '3' (Undefined tag) '4' (Tag specified without a value) '5' (Value is incorrect (out of range) for this tag) '6' (Incorrect data format for value) '7' (Decryption problem) '8' (Signature problem) '9' (CompID problem) '10' (SendingTime accuracy problem) '11' (Invalid MsgType) '12' (XML validation error) '13' (Tag appears more than once) '14' (Tag specified out of required order) '15' (Repeating group fields out of order) '16' (Incorrect NumInGroup count for repeating group)	Code to identify reason for reject.

				'17' (Non "data" value includes field delimiter) '99' (Other)	
58	Text	N	String		Message to explain reason for rejection.
<Standard Message Trailer>		Y			

FIX session establishment and termination scenario

Establish connection

The FIX client (initiator) sends a Logon message with SenderCompID and Password for Trading System in order to establish connection with server (acceptor). The acceptor will authenticate the identity of the initiator by examining the Logon message. The Logon message will contain the data necessary to support the previously agreed upon authentication method. If the initiator is successfully authenticated, the acceptor responds with a Logon message. If authentication fails, the session acceptor shuts down the connection and sending message to indicate the reason of failure. The session initiator may begin to send messages immediately following the Logon message, however, the acceptor may not be ready to receive them. The initiator must wait for the confirming Logon message from the acceptor before declaring the session fully established.

After the initiator has been authenticated, the acceptor will respond with a confirming Logon message. The initiator side will use the Logon message being returned from the acceptor as confirmation that a FIX session has been established. The confirming Logon message from MOEX normally can take up to 3 seconds to receive. Please allow 3 seconds waiting time for reply.

After authentication, the initiator and acceptor must synchronize their messages through interrogation of the MsgSeqNum field before sending any queued or new messages. A comparison of the MsgSeqNum in the Logon message to the internally monitored next expected sequence number will indicate any message gaps. Likewise, the initiator can detect gaps by comparing the acknowledgment Logon message's MsgSeqNum to the next expected value. The section on message recovery later in this document deals with message gap handling.

Comments: FIX client should send Logon message with MsgSeqNum (34) = 1 each new day. FIX client should send Logon message with MsgSeqNum (34) = sequence number of the last message in out log + 1 establishing the next session on the same day.

If FIX client sends to server a Logon (A) message with ResetSeqNumFlag='Y', then it will not receive Execution Reports (8) for events which took place before the session is established.

Resend messages mechanism

During initialization, or in the middle of a FIX session, message gaps may occur which are detected via the tracking of incoming sequence numbers. The following section provides details on how to recover messages.

As previously stated, each FIX participant (FIX client or FIX server) must maintain two sequence numbers for each FIX session, each for incoming and outgoing messages. Each message is assigned a unique sequence number, which is incremented after the message. Likewise, every received message has a unique sequence number and the incoming sequence counter is incremented after each message.

If the incoming sequence number is greater than expected, it indicates that messages were missed and retransmission of the messages is requested via the Resend Request (2) message.

Each side of connection expects to receive message with sequence number, which is equal to sequence number of the last message in his out log + 1. In this case, the side, which detects gaps, should send Resend Request (2) message with a range of missed messages.

The resend request can be used to request a single message, a range of messages or all messages subsequent to a particular message.

- To request a single message: BeginSeqNo = EndSeqNo;
- To request a range of messages: BeginSeqNo = first message of range, EndSeqNo = last message of range;
- To request all messages subsequent to a particular message: BeginSeqNo = first message of range, EndSeqNo = 0 (represents infinity).

N.B. Our service support Resend Request message but there is no resending for Quote and Quote Request messages.

Session status check

The Heartbeat monitors the status of the communication link and identifies cases when the last message of a string was not received. During periods of message inactivity, FIX applications will generate Heartbeat messages at regular time intervals. The heartbeat monitors the status of the communication link and identifies incoming sequence number gaps. The session initiator using the HeartBtInt field in the Logon message declares the Heartbeat Interval. The heartbeat interval timer should be reset after every message is transmitted (not just heartbeats). The HeartBtInt value should be agreed upon by the two firms and specified by the Logon initiator and echoed back by the Logon acceptor. Note that the same HeartBtInt value is used by both sides, the Logon "initiator" and Logon "acceptor".

When either end of a FIX connection has not sent any data for [HeartBtInt] seconds, it will transmit a Heartbeat message. When either end of the connection has not received any data for (HeartBtInt + "some reasonable transmission time") seconds, it will transmit a test request message. If there is still no heartbeat message received after (HeartBtInt + "some reasonable transmission time") seconds then the connection should be considered lost and corrective action be initiated. If HeartBtInt is set to zero no regular heartbeat messages will be generated. Note that a test request message can still be sent independent of the value of the HeartBtInt, which will force a Heartbeat message.

Heartbeats issued as the result of Test Request must contain the TestReqID transmitted in the Test Request message. This is useful to verify that the Heartbeat is the result of the Test Request and not as the result of a regular timeout.

Reset sequence numbers

MOEX automatically resets sequence numbers (MsgSeqNum) at the start of each day. It means that sequence numbers of messages should start from 1 each new day.

FIX client (initiator) may request to reset sequence number of messages (MsgSeqNum (34)) during a trading day. In this case, it is recommended for the initiator to send a TestRequest and wait for a Heartbeat in response to ensure there are no sequence number gaps. Once the Heartbeat has been received, the initiator should send a Logon with ResetSeqNumFlag set to Y and with MsgSeqNum of 1. The acceptor should respond with a Logon with ResetSeqNumFlag set to Y and with MsgSeqNum of 1. At this point new messages from either side should continue with MsgSeqNum of 2. It should be noted that once the initiator sends the Logon with the ResetSeqNumFlag set, the acceptor must obey this request and the message with the last sequence number transmitted "yesterday" may no longer be available.

In case FIX server cannot correctly resend missed messages via Sequence Reset – Gap Fill mode, for example after an unrecoverable application failure, it may request to increase sequence number of messages via sending Sequence Reset (2) message with GapFillFlag (123) = N (Sequence Reset) and NewSeqNo (36) = <new sequence number>. Note that the use of Sequence Reset – Reset may result in the possibility of losing messages.

Close (Terminate) session

In order to close FIX session FIX client should send Logout (5) message.

The logout message initiates or confirms the termination of a FIX session. Disconnection without the exchange of logout messages should be interpreted as an abnormal condition. Before actually closing the session, the logout initiator should wait for the opposite side to respond with a confirming logout message. This gives the remote end a chance to perform any Gap Fill operations that may be necessary. The session may be terminated if the remote side does not respond in an appropriate timeframe.

After sending the Logout message, the logout initiator should not send any messages unless requested to do so by the logout acceptor via a ResendRequest.

Reestablish session after failure

There are certain mechanisms of FIX session reestablishment:

1. In case connection was broken but FIX client didn't lose its logs the following steps should be taken in order to reestablish FIX session:
 - a. Send Logon (A) message with sequence number (MsgSeqNum (34)) = sequence number of the last message in out log + 1;
 - b. If FIX server confirms logon and sends Logon (A) message with sequence number greater than expected, then send Resend Request (2) message with a range of missed messages;
 - c. In this case FIX server resends all missed messages to FIX client.

2. In the case of serious failure when FIX client lost his logs the following steps should be taken in order to reestablish FIX session:
 - a. The first way:
 - i. Send Logon (A) message with sequence number (MsgSeqNum (34)) = 1 and ResetSeqNumFlag (141) = 'Y';
 - ii. If FIX server confirms logon and sends Logon (A) message with MsgSeqNum (34)) = 1 and ResetSeqNumFlag (141) = 'Y', then send Order Status Request (H) for each order in question.
 - b. The second way:
 - i. Send Logon (A) message with sequence number (MsgSeqNum (34)) = 1;
 - ii. If FIX server confirms logon and sends Logon (A) message with Text (58) = "MsgSeqNum too low, expecting X but received Y" send Logon (A) message with sequence number (MsgSeqNum (34)) = X;
 - iii. Send Resend Request (2) message with a range of missed messages;
 - iv. In this case FIX server resends all missed messages to FIX client.
3. In order to get order status for particular order Order Status Request (H) message with ClrOrdID or OrderID fields should be sent.

Messages common for both role: Maker and Taker

Security List Request (x)

Message to request list of available security definitions.

Tag	Field name	Required	Type	Valid values	Comments
	<Group «Header»>	Y			Message type = 'x'
320	SecurityReqID	Y	String		Unique ID of the request message.
559	SecurityListRequestType	N*	Integer	'0'	Identifies the type of Security List Request.

Security List (y)

Security List message is used to return a list of securities requested in Security List Request(x) message.

Tag	Field name	Required	Type	Valid values	Comments
	<Group «Header»>	Y			Message type= 'x'
320	SecurityReqID	Y	String		Unique ID of the request message.
322	SecurityResponseID	Y	String		Identifier for Security List message.
560	SecurityRequestResult	Y	Integer	0 – correct request	Result of Security Request message.
146	NoRelatedSym	N	NumInGroup		Specifies the number of repeating symbols (instruments) specified.
=> 336	TradingSessionID	N	String		Identifier for Trading Session which contains MOEX security board (SECBOARD).
=> 55	Symbol	N	String(12)		Ticker symbol. The MOEX internal instrument identifier, SECCODE
=> 460	Product	N	Char	'4'	Indicates the type of product the security is associated with.

=> 561	RoundLot	N	Qty		The trading lot size of a security.
--------	----------	---	-----	--	-------------------------------------

Messages from Client to Server. Role: Maker

RFQ Request (AH)

RFQ request message issued by liquidity providers to subscribe to RFS auctions quote requests.

Tag	Field name	Required	Type	Valid values	Comments
	<Group «Header»>	Y			Message type = 'AH'
644	RFQReqID	Y	String(20)		RFQ Request ID - used to identify an RFQ Request message. Should be unique during session.
146	NoRelatedSym	Y	Integer	1	Specifies the number of repeating symbols (instruments) specified.
=> 55	Symbol	Y	String(12)		Ticker symbol. The MOEX internal instrument identifier, SECCODE. User able to send 55=* to subscribe all securities.
=> 336	TradingSessionID	Y	String		Identifier for Trading Session which contains MOEX security board (SECBOARD). User able to send 336=* to subscribe all available boards.
263	SubscriptionRequestType	Y	Char	'1' - Subscribe; '2' - Unsubscribe;	Request type.

QuoteRequestReject (AG)

The Quote Request Reject (AG) message is used to refuse participation in some RFS auction.

Tag	Field name	Required	Type	Valid values	Comments
-----	------------	----------	------	--------------	----------

<Group «Header»>		Y			Message type = 'AG'
131	QuoteReqID	Y	String(20)		Unique auction ID generated by the trading system.
658	QuoteRequestRejectReason	N*	Integer	'99' - Other	Reject reason.
146	NoRelatedSym	Y	Integer	1	Specifies the number of repeating symbols (instruments) specified.
=> 55	Symbol	Y	String(12)		Ticker symbol. The MOEX internal instrument identifier, SECCODE
=> 336	TradingSessionID	Y	String(4)		Identifier for Trading Session which contains MOEX security board (SECBOARD).

Quote ('S')

The Quote (S) message is used by the liquidity providers as the response to a Quote Request (R) message.

Liquidity provider able to send quotes in both directions buy and sell in one message. Combination of 132\133 + 134\135 without 54 tag should be used in this case.

Tag	Field name	Required	Type	Valid values	Comments
<Group «Header»>		Y			Message type = 'S'
131	QuoteReqID	Y	String(20)		Unique auction ID generated by the trading system.
117	QuoteID	Y	String(20)		Unique (during auction) quote id.
54	Side	C	Char	'1' – Buy; '2' – Sell;	Quote direction. User able to send bidirectional quotes with specific price and quantity for each direction. In this case user shouldn't send tag 54. Instead tags 132\134 and 133\135 should be presented.
55	Symbol	Y	String(12)		Ticker symbol. The MOEX internal instrument identifier, SECCODE
460	Product	N	Char		Indicates the type of product the security is associated with.
336	TradingSessionID	Y	String(4)		Identifier for Trading Session which contains MOEX security board (SECBOARD).
«Parties»		N			Parties of the order. Contains client code.

1	Account	Y*	String		Trade account.
132	BidPx	C	Price		Bid price.
133	OfferPx	C	Price		Offer price.
134	BidSize	C	Qty		Quantity of bid
135	OfferSize	C	Qty		Quantity of offer
62	ValidUntilTime	N	UTCTimestamp		Indicates expiration time of quote.

QuoteCancel ('Z')

Liquidity providers are able to use Quote Cancel (Z) message to withdraw their quotes.

Tag	Field name	Required	Type	Valid values	Comments
	<Group «Header»>	Y			Message type = 'Z'
131	QuoteReqID	C	String(20)		Unique auction ID generated by the trading system. If tag not specified quote will be canceled by the condition from 298 tag.
117	QuoteID	Y*	String(20)		Arbitrary string (the maximum length is 20 characters). Tag is required under the standard protocol FIX 4.4, but MOEX does not support it.
298	QuoteCancelType	Y	Char	'1' - Cancel for Symbol '4' - Cancel All Quotes	If tag 131 not specified quote will be canceled by the condition from 298 tag. If tag 131 presented, 298 tag still should be present as a mandatory under the protocol standard. It is possible to cancel quotes by symbol. In this case next tags 295, 55, 336 should be presented.
295	NoQuoteEntries	C	NumInGroup		The number of quote entries for a QuoteSet.
=>55	Symbol	C	String(12)		Ticker symbol. The MOEX internal instrument identifier, SECCODE
336	TradingSessionID	C	String		Identifier for Trading Session which contains MOEX security board (SECBOARD).

Usage examples:

Cancel quote by auction number:

Auction number 131=296. Tag 298 specified but ignored by the service.

8=FIX.4.4; 9=91; 35=Z; 34=2; 49=MD9222100001; 56=MFIXRFSId; 52=20200114-07:56:31.000; 117=rand_str; 131=296; 298=1; 10=249;

Cancel quote by the symbol:

Tag 131 is absent. Tags 295, 55, 336 specified.

8=FIX.4.4; 9=120; 35=Z; 34=2; 49=MD9222100001; 56=MFIXRFSId; 52=20200114-07:58:20.000; 117=rand_str; 298=1; 295=1; 55=EUR_RUB__TOD; 336=RFSP; 460=4; 10=095;

Cancel all maker quotes in all auctions.

8=FIX.4.4; 9=83; 35=Z; 34=2; 49=MD9222100001; 56=MFIXRFSId; 52=20200114-08:07:06.000; 117=rand_str; 298=4; 10=136;

Messages from Server to Client. Role: Maker

Quote Request ('R')

Server send Quote Request (R) message to liquidity provider after RFS auction initiation.

Tag	Field name	Required	Type	Valid values	Comments
	<Group «Header»>	Y			Message type = 'R'
115	OnBehalfOfCompID	N	String(12)		Liquidity consumer disclosure. Alias in case of anonymous auction, firm code if not.
131	QuoteReqID	Y	String(20)		Unique auction ID generated by the trading system.
644	RFQReqID	Y	String(20)		Quote subscription ID.
21002	AuctionID	Y	String(20)		Unique auction ID generated by the trading system.

537	QuoteType	N	Integer	1 – Quote 3 – Counter	537=3 is a confirmation of statement that liquidity consumer initiate auction in his\her own interest.
146	NoRelatedSym	Y	Integer	1	Specifies the number of repeating symbols (instruments) specified.
=> 55	Symbol	Y	String(12)		Ticker symbol. The MOEX internal instrument identifier, SECCODE
=> 336	TradingSessionID	Y	String		Identifier for Trading Session which contains MOEX security board (SECBOARD).
=> 460	Product	Y	Char	'4'	Indicates the type of product the security is associated with.
=> «Parties»		N			Parties of the order. Contains client code.
=> 854	QtyType	N	Integer	'1' – In lots	Type of quantity specified in a quantity field 38 (OrderQty). Always in lots for MOEX.
=> 54	Side	N	Char	'1' (Buy) '2' (Sell)	Quote direction. Absence of this quote considered as a bidirectional quote.
=> 38	OrderQty	Y	Qty		Quantity expressed in lots.
=> 1	Account	N	String(12)		Trading account. Presented in only case when liquidity consumer also subscribed to the auction as a liquidity provider.
=> 126	ExpireTime	Y	UTCTimestamp		Auction expiration time. Format YYYYMMDD-HH:MM:SS.

Quote Response ('AJ')

Server send Quote Response (AJ) message after the RFS auction ends.

Tag	Field name	Required	Type	Valid values	Comments
	<Group «Header»>	Y			Message type = 'AJ'
693	QuoteRespID	Y	String(20)		Unique auction ID generated by the trading system.
117	QuoteID	Y			Best quote update number
21002	AuctionID	Y	String(20)		Unique auction ID generated by the trading system.
537	QuoteType	N	Integer	1 – Quote 3 – Counter	537=3 is a confirmation of statement that liquidity consumer initiate auction in his\her own interest.

460	Product	N	Char		Indicates the type of product the security is associated with.
854	QtyType	N	Integer	'1' – In lots	Type of quantity specified in a quantity field 38 (OrderQty). Always in lots for MOEX.
694	QuoteRespType	Y	Char	'1' – Hit '3' – Expired '6' – Pass	Identifies the type of Quote Response (AJ).
336	TradingSessionID	Y	String		Identifier for Trading Session which contains MOEX security board (SECBOARD).
55	Symbol	Y	String(12)		Ticker symbol. The MOEX internal instrument identifier, SECCODE
54	Side	Y	Char	'1' – Buy; '2' – Sell;	Quote direction.
38	OrderQty	Y	Qty		Quantity expressed in lots.
1	Account	N	String(12)		Trading account.
11	ClOrdId	N	String(20)		Additional information entered by the trading firm

Execution Report ('8')

Execution report (8) message include data about matched RPS trade.

Tag	Field name	Required	Type	Valid values	Comments
	<Group «Header»>	Y			Message type = '8'
37	OrderID	Y	String(20)		Unique quote ID generated by the system.
693	QuoteRespID	Y	String(20)		Unique auction ID generated by the trading system.
526	SecondaryClOrdID	N	String(20)		A reference field, which may be used as a back-feed by an external system.
115	OnBehalfOfCompID	N	String(12)		Liquidity consumer disclosure.
21002	AuctionID	Y	String(20)		Unique auction ID generated by the trading system.
17	ExecID	N	String(20)		Trade ID. Unique for MOEX.
	<Group "Parties">	N			
150	ExecType	Y	Char	'F' – Trade	Type of Execution report (8) message.
39	OrdStatus	Y	Char	'2' – Filled	Order status.
1	Account	Y	String(12)		Trading account

55	Symbol	Y	String(12)		Ticker symbol. The MOEX internal instrument identifier, SECCODE
336	TradingSessionID	Y	String(4)		Identifier for Trading Session which contains MOEX security board (SECBOARD).
460	Product	N	Char		Indicates the type of product the security is associated with.
54	Side	Y	Char	'1' – Buy; '2' – Sell;	Quote direction.
38	OrderQty	Y	Qty		Quantity expressed in lots.
44	Price	Y	Price		Order price.
381	GrossTradeAmt	N	Amount		Trade volume expressed in rubbles
60	TransactTime	Y	UTCTimestamp		Trade registration time.
64	SettlDate	N	LocalMktDate		Specific date of trade settlement (SettlementDate) in YYYYMMDD format
15	Currency	N	Currency		Currency code.
423	PriceType	Y	Integer	2 – Per unit	Code to represent the price type. Always 2 for MOEX.
854	QtyType	N	Integer	'1' – In lots	Type of quantity specified in a quantity field. Always 1 for MOEX.
151	LeavesQty	N*	Qty		Quantity open for further execution (order balance). Always equal zero.
14	CumQty	Y*	Qty		Currently executed shares for chain of orders. Always equal to 38 tag value.
6	AvgPx	Y*	Price		Calculated average price of all fills on this order. Always equal to 44 tag value.
32	LastQty	Y*	Qty		Trade Quantity. Always equal to 38 tag value.
31	LastPx	Y*	Price		Trade Price. Always equal to 44 tag value.

Quote ('S')

After the auction initiation, liquidity provider begins to receive Quote messages with best quotes. Some specific case is empty book quote. Service send it when there is no any quote in the auction. Signs of the empty book quote are 132=0, 133=0, 134=0,135=0 tags.

Tag	Field name	Required	Type	Valid values	Comments
	<Group «Header»>	Y			Message type = 'S'
131	QuoteReqID	Y	String(20)		Unique auction ID generated by the trading system.
693	QuoteRespID	Y	String(20)		Unique auction ID generated by the trading system.

21002	AuctionID	Y	String(20)		Unique auction ID generated by the trading system.
117	QuoteID	Y	String(20)		Unique identifier for quote.
55	Symbol	Y	String(12)		Ticker symbol. The MOEX internal instrument identifier, SECCODE
336	TradingSessionID	Y	String(4)		Identifier for Trading Session which contains MOEX security board (SECBOARD).
132	BidPx	N	Price		Bid price.
133	OfferPx	N	Price		Offer price.
134	BidSize	Y	Qty		Quantity of bid quote expressed in lots.
135	OfferSize	Y	Qty		Quantity of offer quote expressed in lots.
537	QuoteType	N	Integer	1 – Quote 3 – Full amount quote	537=3 is a confirmation of statement that liquidity consumers initiate auction in his\her own interest.
854	QtyType	N	Integer	'1' – In lots	Type of quantity specified in a quantity field. Always 1 for MOEX.
423	PriceType	Y	Integer	2 – Per unit	Code to represent the price type. Always 2 for MOEX.
126	ExpireTime	Y	UTCTimestamp		Auction expiration time

Quote Status Report ('AI')

Quote status report message (AI) is used to

- response on liquidity provider Quote messages.
- inform liquidity provider about quote status changes
- as a report to Quote Cancel message if there were issued any problems

Tag	Field name	Required	Type	Valid values	Comments
	<Group «Header»>	Y			Message types = 'AI'
117	QuoteID	Y	String(20)		Unique identifier for quote.
131	QuoteReqID	Y	String(20)		Unique auction ID generated by the trading system.
297	QuoteStatus	Y	Integer	0 – Accepted 5 – Canceled 6 – Canceled by the system	Quote status.

				7 – Expired 8 – Error 9 – Quote not found 10 – Pending 11 – Pass	
55	Symbol	Y	String(12)		Ticker symbol. The MOEX internal instrument identifier, SECCODE
336	TradingSessionID	Y	String(4)		Identifier for Trading Session which contains MOEX security board (SECBOARD).
58	Text	N	String		TS transaction answer

Messages from Client to Server. Role: Taker

Quote Request ('R')

Liquidity consumer send Quote Request (R) message to initiate RFS auction.

Tag	Field name	Required	Type	Valid values	Comments
	<Group «Header»>	Y			Message type = 'R'
131	QuoteReqID	Y	String(20)		Auction ID generated by liquidity consumer. Must be unique during session.
537	QuoteType	N	Integer	1 – Quote 3 – Full amount quote	537=3 is a confirmation of statement that liquidity consumers initiate auction in his\her own interest.
11	ClOrdId	N	String(20)		Additional information entered by the trading firm
526	SecondaryClOrdID	N	String(20)		A reference field, which may be used as a back-feed by an external system.
146	NoRelatedSym	Y	Integer	1	Specifies the number of repeating symbols (instruments) specified.
=> 55	Symbol	Y	String(12)		Ticker symbol. The MOEX internal instrument identifier, SECCODE
=> 336	TradingSessionID	Y	String		Identifier for Trading Session which contains MOEX security board (SECBOARD).
=> 460	Product	Y	Char		Indicates the type of product the security is associated with.
=> «Parties»		N			Parties of the order. Contains client code.

=> 54	Side	N	char	'1' (Buy) '2' (Sell)	Quote direction. Absence of this quote considered as a bidirectional quote.
=> 38	OrderQty	Y	Qty		Quantity in lots.
=> 1	Account	Y	String(12)		Trade account.
=> 126	ExpireTime	Y	UTCTimestamp		Auction expiration time.
=> 6101	Anonymity	N	Char	'Y' – Yes 'N' – No.	Indicates whether a Firm wants to remain anonymous during auction.. По умолчанию 'Y'. Если 'N' – провайдер ликвидности получит код фирмы потребителя ликвидности в тэге OnBehalfOfCompID(115).

Quote Response ('AJ')

Liquidity consumer send Quote Response (AJ) message to finish auction. Message could be considered as a request to finish auction or as a hit to match quote.

Tag	Field name	Required	Type	Valid values	Comments
<Group «Header»>		Y			Message type = 'AJ'
693	QuoteRespID	Y	String(20)		Quote request message ID.
694	QuoteRespType	Y	Char	'1' = Hit '6' = Pass	Quote response message type.
336	TradingSessionID	Y	String(4)		Identifier for Trading Session which contains MOEX security board (SECBOARD).
55	Symbol	Y	String(12)		Ticker symbol. The MOEX internal instrument identifier, SECCODE
54	Side	C	char	'1' (Buy) '2' (Sell)	Quote direction.
38	OrderQty	C	Qty		Quantity expressed in lots.
460	Product	Y	Char		Indicates the type of product the security is associated with.
44	Price	C	Price		Quote price.
1	Account	C	String(12)		Trading account.

11	ClOrdId	N	String(20)	Quote number from the quote message presented by liquidity provider.
-----------	---------	---	------------	--

Messages from Server to Client. Role:Taker

Quote ('S')

After the auction initiation, liquidity consumer begins to receive Quote messages with best quotes from liquidity providers. Some specific case is empty book quote. Service send it when there is no any quote in the auction. Signs of the empty book quote are 132=0, 133=0, 134=0,135=0 tags.

Tag	Field name	Required	Type	Valid values	Comments
<Group «Header»>		Y			Message type = 'S'
131	QuoteReqID	Y	String(20)		Auction ID from the Quote request message.
693	QuoteRespID	Y	String(20)		Auction ID from the Quote request message.
21002	AuctionID	Y	String(20)		Unique auction ID generated by the trading system.
117	QuoteID	Y	String(20)		Best quote update number
<Group "Parties">		N			
1	Account	Y	String(12)		Trading account
55	Symbol	Y	String(12)		Ticker symbol. The MOEX internal instrument identifier, SECCODE
336	TradingSessionID	Y	String(4)		Identifier for Trading Session which contains MOEX security board (SECBOARD).
460	Product	Y	Char	'4'	Indicates the type of product the security is associated with.
132	BidPx	N	Price		Bid price.
133	OfferPx	N	Price		Offer price.
134	BidSize	Y	Qty		Quantity of bid quote expressed in lots.
135	OfferSize	Y	Qty		Quantity of offer quote expressed in lots.
38	OrderQty	Y	Qty		Quantity expressed in lots.

537	QuoteType	N	Integer	1 – Quote 3 – Full amount quote	537=3 is a confirmation of statement that liquidity consumers initiate auction in his\her own interest.
854	QtyType	Y	Integer	'1' – In lots	Type of quantity specified in a quantity field. Always 1 for MOEX.
423	PriceType	Y	Integer	2 – Per unit	Code to represent the price type. Always 2 for MOEX.
126	ExpireTime	Y	UTCTimestamp		Auction expiration time

Quote Request Reject ('AG')

If there were any problems with auction initiation, liquidity consumer get Quote Request Reject (AG) message.

Tag	Field name	Required	Type	Valid values	Comments
<Group «Header»>		Y			Message type = 'AG'
58	Text	N	String		Error text.
131	QuoteReqID	Y	String(20)		Auction ID from the Quote request message.
658	QuoteRequestRejectReason	Y	Integer	'10'	Rejection reason.
146	NoRelatedSym	Y	Integer	1	Specifies the number of repeating symbols (instruments) specified.
=> 55	Symbol	Y	String(12)		Ticker symbol. The MOEX internal instrument identifier, SECCODE
=> 336	TradingSessionID	Y	String(4)		Identifier for Trading Session which contains MOEX security board (SECBOARD).
58	Text	Y	String		Reject reason.

Quote Response ('AJ')

Quote Response (AJ) message is used to report liquidity consumer about the end of auction.

Tag	Field name	Required	Type	Valid values	Comments
<Group «Header»>		Y			Message type = 'AJ'
693	QuoteRespID	Y	String(20)		Quote response message ID.

117	QuoteID	Y			Best quote update number
21002	AuctionID	Y	String(20)		Unique auction ID generated by the trading system.
694	QuoteRespType	Y	Char	'1' = Hit '3' – Expired '6' - Pass	Identifies the type of Quote Response.
336	TradingSessionID	Y	String(4)		Identifier for Trading Session which contains MOEX security board (SECBOARD).
55	Symbol	Y	String(12)		Ticker symbol. The MOEX internal instrument identifier, SECCODE
460	Product	N	Char		Indicates the type of product the security is associated with.
537	QuoteType	N	Integer	1 – Quote 3 – Full amount quote	537=3 is a confirmation of statement that liquidity consumers initiate auction in his\her own interest.
854	QtyType	Y	Integer	'1' – In lots	Type of quantity specified in a quantity field. Always 1 for MOEX.
54	Side	Y	Char	'1' – Buy; '2' – Sell;	Quote direction.
38	OrderQty	Y	Qty		Quantity expressed in lots.
1	Account	N	String(12)		Trading account.
11	CIOrdId	N	String(20)		Additional information entered by the trading firm

Execution Report ('8')

Execution report (8) message include data about RPS trade.

Tag	Field name	Required	Type	Valid values	Comments
	<Group «Header»>	Y			Message type = '8'
37	OrderID	Y	String(20)		Unique quote ID generated by the system.
693	QuoteRespID	Y	String(20)		Quote response message ID.
526	SecondaryCIOrdID	N	String(20)		A reference field, which may be used as a back-feed by an external system.
854	QtyType	Y	Integer	'1' – In lots	Type of quantity specified in a quantity field. Always 1 for MOEX.
21002	AuctionID	Y	String(20)		Unique auction ID generated by the trading system.
17	ExecID	Y	String(20)		Trade ID. Unique for MOEX.

<Group "Parties">		N			
150	ExecType	Y	Char	'F' – Trade '8' – Reject	Type of Execution report (8) message.
39	OrdStatus	Y	Char	'2' – Filled '8' - Rejected	Order status.
1	Account	N	String(12)		Trading account
55	Symbol	Y	String(12)		Ticker symbol. The MOEX internal instrument identifier, SECCODE
460	Product	Y	Char		Indicates the type of product the security is associated with.
336	TradingSessionID	Y	String(4)		Identifier for Trading Session which contains MOEX security board (SECBOARD).
54	Side	Y	Char	'1' – Buy; '2' – Sell;	Quote direction.
38	OrderQty	Y	Qty		Quantity expressed in lots.
44	Price	Y	Price		Order price.
381	GrossTradeAmt	N	Amount		Trade volume expressed in rubbles
60	TransactTime	N	UTCTimestamp		Trade registration time.
64	SettlDate	N	LocalMktDate		Specific date of trade settlement (SettlementDate) in YYYYMMDD format
151	LeavesQty	N*	Qty		Quantity open for further execution (order balance). Always equal zero.
14	CumQty	N*	Qty		Currently executed shares for chain of orders. Always equal to 38 tag value.
6	AvgPx	N*	Price		Calculated average price of all fills on this order. Always equal to 44 tag value.
32	LastQty	N*	Qty		Trade Quantity. Always equal to 38 tag value.
31	LastPx	N*	Price		Trade Price. Always equal to 44 tag value.
15	Currency	N	Currency		Currency code.
423	PriceType	Y	Integer	2 – Per unit	Code to represent the price type. Always 2 for MOEX.
103	OrdRejReason	N	Integer	99 – Other	Reject reason.
58	Text	N	String		TS transaction answer

QuoteStatusReport ('AI')

Quote status report message (AI) is used to inform liquidity consumer about negative reaction to the Quote Request Reject message.

Tag	Поле	Наличие	Тип	Допустимые значения	Примечание
	<Группа «Header»>	О			Тип сообщения = 'AI'
117	QuoteID	О	String(20)		Идентификатор котировки провайдера ликвидности.
131	QuoteReqID	О	String(20)		Идентификатор аукциона.
297	QuoteStatus	О	Integer	9 – аукцион не найден	Статус аукциона.
55	Symbol	О	String(12)		Код инструмента.
336	TradingSessionID	О	String(4)		Код режима торгов.
58	Text	Н	String		Текст ответа ТС