Moscow Exchange public FIX 4.4 interface specification for Securities and FX markets

Revision 4.1

March 24, 2016

Contents

1 Introduction

Moscow Exchange (MOEX) Interface specification contains the description of FIX 4.4 protocol messages, component blocks and fields which are supported by MFIX Transactional Server. This Interface specification is valid for Moscow Exchange FX and Securities (Main and T+2) markets only. For other markets please refer to the www.moex.com web site.

Permanent location of this document and updates is ftp://ftp.moex.com/pub/FIX/ASTS/docs/

This specification is based on FIX 4.4 standard (http://fixprotocol.org/) and adapted to MOEX's purposes. It's assumed that users have basic knowledge about FIX standard.

Only messages, component blocks and fields which are described in this document are supported by MFIX Transactional Server. *Note that all fields which are 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 aren't 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 are described for each field:
- Tag – unique field identifier.
- Name – field name.
- Req'd – 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.

Note:

Fields which are marked with italics are custom and defined additionally to standard fields for MOEX purposes.

Service MFIX Transactional consists of three separate services:

- MFIX Trade which is used to submit orders, cancel orders, request order status and receive execution reports for each order;

- MFIX Trade Capture which is used to receive executed trades, including negotiated and REPO deals and corresponding executions.

- MFIX Drop Copy which is used to receive copies of all execution reports available for the user.

MFIX Trade, MFIX Trade Capture, and MFIX Drop Copy services are served by separate servers (connections differ by IP addresses and port numbers) and have different access levels:

- MFIX Trade service delivers only messages which are related to orders of appropriate user.

- MFIX Trade Capture and MFIX Drop Copy services deliver data according to user permissions at Trading System. For example, user with 'firm manager' permissions can get all trades or executions which belong to his firm.

1.1 Document History

Issue	Date	Description
1.0	May 25, 2010	Original version of this document
2.0	December 12, 2012	Added support for negotiated deals in Trade Capture service
3.3	March 25, 2013	Added support for REPO with CCP and T+2 trading modes. Clarifications added.
3.3.1	October 10, 2013	Fixing document errors and adding clarifications per users' feedback. In Execution Report, and in Trade Capture report, added OptionSettlType (5459) field that contains MOEX settlement code for trades. **Important**: added support for Order Cancel/Replace FIX message.
3.3.2	October 23, 2013	FIX messages of types 35=F (Order Cancel Request) and 35=G (Order Cancel Replace Request) will be rejected via the Reject message (35=3) if the ClOrdID field (11) of these

		messages starts with a hash (pound) symbol '#'. You can use '#' symbol in any position of this string, except the first one.
3.3.3	November 1, 2013	Added clarifications to the description of trading system response to the Order Cancel and Order Cancel Replace messages. Added comments to the sections Order Cancel Request, Order Cancel Replace request, Execution report, Cancel Reject.
3.3.4	January 24, 2014	Order Cancel Replace description is updated to highlight differences between FX and Equities markets. Added corrections for market orders in Closing auction behavior Added field CalculatedCcyLastQty (1056) Trade Capture Reports that specifies the trade quantity expressed in the units of securities
3.3.5	March 12, 2014	Added public order ID from FAST UDP multicast marketdata OLR (full book) channel as MDEntryID (278) field in the Execution Report messages (Equities market only). Added TransactTime (60) and OrigTime (9412) fields to Order Cancel Reject, Order Mass Cancel Report, and to Execution Report / Rejected messages. Added Order Status (39) field and Too late to cancel '0' value to the CxlRejReason (102) field in Order Cancel Reject message.
3.3.6.	April 1, 2014	OrigClOrdID (41) and OrderID (37) fields specifications changed to conditionally required in Order Cancel Request and Order Cancel Replace request. ParentID (9580) field that indicates the showcase trade number of REPO or currency swap trade has been added to Execution report and Trade Capture report messages for linked spot trades. Message format description of Execution report for cancellation of partially filled order for failed Cancel Replace is corrected Order Cancel Replace execution rules are identical at FX and equities market, including the use of CancelOrigOnReject (9619)=Y / N field.
3.3.7.	May 13,2014	In MFIX Transactional FX market has the following changes: • Added functionality Cancel On Disconnect. Added field CancelOnDisconnect (COD) (6867) Logon (A). Added value '100' = ' (Cancel on disconnect)' to

		ExecRestatementReason (378) Execution Report (35=8) ', Order Status (39) = 4. Updated logic of sending Execution Report for canceled orders: • If an order is canceled by owner, then Execution report will contain tags ClOrdID and OrigClOrdID(41). • In all other cases Execution Report (35=8) will contain ClOrderID (11) tag of original order. • Pending Cancel (150=6) Execution Report message is not sent for cancellation of order that is canceled with no execution..
4.0	March 25, 2015	• Cancel On Disconnect feature has been enabled for subscribed users only in always ON state. Tag 6867 is accepted in logon message, but it does not alter COD On/Off state. • Commission data and Misc Fees fields are removed from Execution Report / Trade for MFIX Trade service. These fields are available according to user permissions in MFIX Trade Capture and MFIX Drop Copy services. • New value 'S' for the trading period tag (625) has been added for opening auction period. Value 625=OPEN description is updated to refer to opening period. • IOC and FOK orders behavior is identical at all markets • Market orders are not allowed for FX swap instruments • Document is corrected to remove errors in data types • Clarifications added
	September 23, 2015	• Added field LanguageID (6936) Logon (A), available codes: 'R' - Russian, 'E' - English
4.1	March 24, 2016	• Order status request (35=H) message is no longer supported and will be rejected. • Field 60 format description for Trade Capture report is corrected to reflect actual state of having only whole second precision. • REPO with CCP Execution Report for Trade has been fixed to send showcase trade and both legs for MFIX Trade service (bug fix). • Clarifications added

1.1 Important recommendation for developers:

In some cases server initiated messages may be extended by optional fields that are not required for use by most of clients. These modifications may happen with short notice. It is strongly recommended to design your client application being able to ignore unknown fields in server initiated messages.

As an example, the field StipulationValue (6636) has been added in Execution Report with trade to indicate a trade that resulted in iceberg order replenishment. See also new fields CoverID, CurrencyCode, InstitutionID, ClientAccID, TradeIssueDate, OrigOrderID that were added in ver. 3.3.1.

2 FIX components blocks

2.1 Standard Message Header

Each administrative or application message is preceded by a standard header. The header identifies the message type, length, destination, sequence number, origination point and time.

The standard message header format is as follows:

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

Tag	Field name	Req'd	Type	Valid values	Comments
					If this message is sent to MOEX, then it should contain the MOEX server identifier. This parameter is given by MOEX
34	MsgSeqNum	Y	SeqNum		Integer message sequence number.
43	PossDupFlag	N	Boolean	'Y' (Possible duplicate) 'N' (Original transmission)	Indicates possible retransmission of message with this sequence number. Required for retransmitted messages.
97	PossResend	N	Boolean	'Y' (Possible resend) 'N' (Original transmission)	Indicates that message may contain information that has been sent under another sequence number.
52	SendingTime	Y	UTCTimestamp		Time of message transmission (expressed in UTC). YYYYMMDD-HH:MM:SS.sss
122	OrigSendingTime	N	UTCTimestamp		Original time of message transmission when transmitting messages as the result of resend request (expressed in UTC). Required for message resent as a result of a resend request.

2.2 Standard Message Trailer

Each message, administrative or application, is terminated by a standard trailer. The trailer is used to segregate messages and contains the three digit character representation of the Checksum value.

The standard message trailer format is as follows:

Tag	Field name	Req'd	Type	Valid values	Comments
10	CheckSum	Y	String(3)		Three byte, simple checksum. Always unencrypted, always last field in message.

2.3 Instrument

The <Instrument> component block describes security which is traded the exchange.

The <Instrument> component block format is as follows:

Tag	Field name	Req'd	Type	Valid values	Comments
55	Symbol	Y	String (12)		Ticker symbol. The MOEX internal instrument identifier, SecCode
460	Product	N	int	'4' (Currency)	Indicates the type of product the security is associated with. (Used only for Currency (FX) market)
461	CFICode	N	String	'MRCXXX' (Misc, Referential Instrument, Currency [FOR])	Indicates the type of security using ISO 10962 standard, Classification of Financial Instruments (CFI code) values. (Used only for Currency market)
167	SecurityType	C	String	'FXSPOT' (FX Spot); 'FXSWAP' (FX Swap); 'FXFWD'(FX Forward); 'FXBKT' (FXBKT) 'REPO' (REPO with CCP)	Indicates type of security. Conditionally required for REPO with CCP order: you should use 167=REPO tag value, otherwise REPO with CCP order will be rejected.

2.4 Yield Data

The <Yield Data> component block describes security yield.
The <Yield Data> component block format is as follows:

Tag	Field name	Req'd	Type	Valid values	Comments
236	Yield	N	Percentage		Yield calculated at trade price. Used only for Equities market.

2.5 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) = <user id>, PartyIDSource (447) = 'D', PartyRole (452) = '12' – specifies user (trader, broker);
- PartyID (448) = <firm id>, PartyIDSource (447) = 'D', PartyRole (452) = '1' – specifies firm;
- PartyID (448) = <client code>, PartyIDSource (447) = 'D', PartyRole (452) = '3' – specifies client;

- PartyID (448) = <counterparty firm id>, PartyIDSource (447) = 'D', PartyRole (452) = '17' - specifies counterparty firm.

The <Parties> component block format is as follows:

Tag	Field name	Req'd	Type	Valid values	Comments
453	NoPartyID	N	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(2)	'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).

2.6 Commission Data

The <Commission Data> component block defines total trade commission.

The <Commission Data> component block format is as follows:

Tag	Field name	Req'd	Type	Valid values	Comments
12	Commission	N	Amt (16,2)		Commission, expressed in unit of currency as defined by trading rules for each instrument

Tag	Field name	Req'd	Type	Valid values	Comments
13	CommType	N	String	'3' (absolute (total monetary amount)); *100 (points per quick settlement instruction).*	Commission type Equities

2.7 Order Qty Data

The <Order Qty Data> component block defines number of lots in order or trade.
The <Order Qty Data> component block format is as follows:

Tag	Field name	Req'd	Type	Valid values	Comments
38	OrderQty	Y*	Qty (10)		Quantity ordered, expressed in number of lots. Lot size is different for different Symbol + Board combinations and should be determined from the marketdata feeds.
152	CashOrderQty	Y*	Qty		Specifies the approximate "monetary quantity" for the order. Note that unless otherwise specified, only one of CashOrderQty (152) or OrderQty (38), should be specified.

2.8 TrdRegTimestamps

The <TrdRegTimestamps> component block is used to represent time when order was registered in Trading System.
The <TrdRegTimestamps> component block format is as follows:

Tag	Field name	Req'd	Type	Valid values	Comments
768	NoTrdRegTimestamps	N	NumInGroup		Number of repeated TrdRegTimestamp group entries.
769	TrdRegTimestamp	C	UTCTimestamp		Traded / Regulatory timestamp value. E.g. Order creation time. Required if NoTrdRegTimestamps (768) >0.

Tag	Field name	Req'd	Type	Valid values	Comments
770	TrdRegTimestampType	C	Int	'2' Time In	Traded / Regulatory timestamp type. Required if NoTrdRegTimestamps (768) >0.

2.9 Financing Details

The <Financing Details> component block is used to define the financing details of a deal.

The <Financing Details> component block format is as follows:

Tag	Field name	Req'd	Type	Valid values	Comments
916	StartDate	N	LocalMktDate		Start date of a financing deal, i.e. the date the buyer pays the seller cash and takes control of the collateral. equities
917	EndDate	N	LocalMktDate		End date of a financing deal, i.e. the date the seller reimburses the buyer and takes back control of the collateral. (Used only for equities market)
919	DeliveryType	N	int	'3' (Hold In Custody)	Identifies type of settlement. equities

2.10 Underlying Instrument

The <Underlying Instrument> component block is used to provide additional information about security.

The <Underlying Instrument> component block format is as follows:

Tag	Field name	Req'd	Type	Valid values	Comments
311	UnderlyingSymbol	N	String		Underlying security's Symbol. (Used only for equities market)
	<Underlying Stipulations>	N			Underlying stipulations entries. (Used only for equities market)

2.11 Underlying Stipulations

The <Underlying Stipulations> component block is used to provide additional information about discounts.
The <Underlying Stipulations> component block format is as follows:

Tag	Field name	Req'd	Type	Valid values	Comments
887	NoUnderlyingStips	N	NumInGroup		Number of underlying stipulation entries. (Used only for Equities market)
888	UnderlyingStipType	C	String		Type of stipulation. Required if NoUnderlyingStips (887) >0. (Used only for Equities market)
889	UnderlyingStipValue	N	String		Value of stipulation. (Used only for equities market)

3   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.

3.1   Logon (MsgType = '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. Note that incorrect Logon message or a Logon with not authorized parameters will not be answered by server. The Logon reply normally can take up to 5 seconds to receive. Please allow 5 seconds waiting time for reply.

Note: SenderCompID cannot be used for more than single connection to any service. Cancel On Disconnect service is provided in two modes: mandatory order cancellation on FIX session drop or disconnect, or Cancel On Disconnect activation at FIX session logon. **Currently, only mandatory activation is supported at Trading System level.**

The logon format is as follows:

Tag	Field name	Req'd	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).
141	ResetSeqNumFlag	N	Boolean	'Y' (Yes) 'N' (No)	Indicates if the both sides of the FIX session should reset sequence numbers.
554	Password	Y*	String		User password. Note – The maximum length is 10 characters. Required field at MOEX
925	NewPassword	N	String		The new password for the SenderCompID(49). Note – The maximum length is 10 characters. If this field is specified, then later Logon must use this new password value in tag 554.
1409	SessionStatus	N	int	'0' (Session Active); '3' (New password doesn't comply with the policy)	Status of the request to change the password. Required if the message is generated by the server. Note: if new password is not accepted (a server returns 1409=3), then you must use previous successful password at logon.
6867	CancelOnDisconnect (COD)	N	String	'A'	Cancel on Disconnect (COD) service, if enabled, automatically cancels all active orders in the event of logoff, disconnection on failure, forced disconnection, FIX gateway failure. COD may be set mandatory ON or OFF, or optional per member firm request. If COD service is set to mandatory ON or OFF: • Field 6867 does not alter COD behavior. Logon reply will have text (58) with an error message related to rejection to alter COD service state. If COD functionality is set to optional (not supported now): • If field 6867=A is present in logon message then COD is activated until the next logon • If the field 6867 is not present in logon message, the COD service is not activated •
6936	LanguageID	N	String	'R' (Russian) 'E' (English)	Language of Trading System messages. Logon reply will have text (58) with one of the messages : "(262) Language successfully changed" "(261) Language is not available" "(260) Invalid language code"
<Standard Message Trailer>		Y			

3.2 Logout (MsgType = '5')

The logout message initiates or confirms the termination of a FIX session.
The logout format is as follows:

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

3.3 Heartbeat (MsgType = '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.
The heartbeat format is as follows:

Tag	Field name	Req'd	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			

3.4 Test Request (MsgType = '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).

The test request format is as follows:

Tag	Field name	Req'd	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			

3.5 Resend Request (MsgType = '2')

The resend request is sent by the receiving application 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.

The resend request format is as follows:

Tag	Field name	Req'd	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			

3.6 Sequence Reset (MsgType = '4')

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

Gap Fill 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).

Reset mode is described at the Reset sequence numbers section.

The Sequence Reset format is as follows:

Tag	Field name	Req'd	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			

3.7 Reject (MsgType = '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. As a rule, messages should be forwarded to the trading application for business level rejections whenever possible.

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.

The reject format is as follows:

Tag	Field name	Req'd	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) '17' (Non "data" value includes field delimiter) '99' (Other)	Code to identify reason for reject.
58	Text	N	String		Message to explain reason for rejection.
	<Standard Message Trailer>	Y			

4   MFIX Trade

4.1   FIX session establishment and termination scenario

4.1.1   Establish connection

The FIX client (initiator) sends a Logon message with SenderCompID and Password for Trading System in order to establish connection with MFIX Transactional 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 not sending any 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 5 seconds to receive. Please allow 5 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.

Note:
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.

4.1.2 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 MFIX Transactional 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.

When the incoming sequence number does not match the expected number corrective processing is required.

If the incoming message has a sequence number less than expected and the PossDupFlag is not set, it indicates a serious error. It is strongly recommended that the session be terminated and manual intervention be initiated.

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).

4.1.3 Management of session status

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 Heartbeat Interval is declared by the session initiator using the HeartBtInt field in the Logon message. 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.

4.1.4 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's 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 MFIX Transactional 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.

4.1.5 Close (terminate) FIX 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.

4.1.6   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 MFIX Transactional 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 MFIX Transactional 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 MFIX Transactional 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 MFIX Transactional 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 MFIX Transactional 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.~~ **Disabled since March 24, 2016**

4. In case a primary MFIX Transactional server isn't available the following steps should be taken:
    a. Switch to backup MFIX Transactional server;
    b. Continue to work with the backup server according to point 1 of this section.

Notes:
•   MFIX Trade service is provided via multiple servers that are combined in mirrored pairs. Each mirrored pair does synchronize messages between its parts. You can reconnect to mirrored server not losing message history.
•   Server pairs are defined at MOEX web site. Please contact help@moex.com for more information about connectivity parameters.

- *In a case of one of MFIX Trade servers being unavailable for connection it is strongly recommended to re-establish connection to its mirrored counterpart. This ensures fastest reconnection scenario.*
- *To improve connection resilience you can establish multiple FIX sessions distributed over multiple FIX servers. Then you can use round-robin rules to send orders and balance load. In case of any FIX session failure the remaining sessions may be used to cancel orders of that session, which may take less time than for Cancel on Disconnect service.*
- MFIX Trade Capture and MFIX Drop copy services operate each on two separate servers, which are not synchronized by messages. **It is strongly recommended to establish sessions to each instance of MFIX Trade Capture and MFIX Drop Copy services to keep receiving data in case of single server failure.**

4.2 Messages from Client to Server

4.2.1 Order Status Request (MsgType = 'H')

The Order Status Request (H) message is used by FIX client to request current order status.

Order Status Request format is:

Tag	Field name	Req'd	Type	Valid values	Comments
	\<Standard Message Header\>	Y			MsgType = H
37	OrderID	Y*	String		The OrderID of the order whose status is being requested.
11	ClOrdID	N*	String		. Arbitrary string (the maximum length is 20 characters). Tag is required under the standard protocol FIX 4.4, but MOEX does not support getting information about the status of orders on the user identifier of the order.
	\<Standard Message Trailer\>	Y			-

4.2.2 New Order - Single (MsgType = 'D')

New Order – Single (D) message is used to submit new order.

New Order – Single format is:

Tag	Field name	Req'd	Type	Valid values	Comments
	<Standard Message Header>	Y			MsgType = D
11	ClOrdID	Y	String (20)		Unique identifier of the order as assigned by institution or by the intermediary (CIV term, not a hub/service bureau) with closest association with the investor. Note: this field's value is sent in the BROKERREF field of native MOEX Order message. Please consult your broker how to format this field according to broker preferences. Note: If you send New Order Single or Order Cancel Replace or Order Cancel Request messages via the same SenderCompID in short sequence, then these messages are processed synchronously on the path between the FIX server and matching engine: new request is not sent to matching engine until FIX server code receives reply to previous request. Typical roundtrip time is less than 250 microseconds, however, there is a gap between delivering each subsequent message into the matching engine queue. To accelerate multiple message delivery it is recommended to use multiple FIX sessions and avoid sending message through single FIX session with less than 300 microseconds gaps between them
	component block <Parties>	N			Parties of the order. Usually contains client code. Note: It's recommended to define client code only for broker's client accounts, otherwise client code will be ignored by trading system and won't be returned in execution Reports, except the first message for this order.
1	Account	Y*	String (12)		Account mnemonic as agreed between buy and sell sides, e.g. broker and institution or investor/intermediary and fund manager. Is used to represent trading account.
111	MaxFloor	N	Qty		Maximum quantity in lots within an order to be shown on the exchange floor at any given time (For iceberg orders).
526	SecondaryClOrdID	N	String (12)		This field's value is sent to the EXTREF field of native MOEX order message. Please consult your broker how to fill this field.
386	NoTradingSessions	Y*	NumInGroup	*1 (one element)*	Specifies the number of repeating TradingSessionIDs. TradingSessionIDs group should contain only one element of group. Note: tags 386 and 336 compose a group and should be placed exactly in the order 386, then 336, and not separated by other tags.
336	=> TradingSessionID	Y*	String (4)		Identifier for Trading Session which contains MOEX security board (SECBOARD).

	component block <Instrument>	Y			This block contains Symbol (55) field for a security of the order (SECCODE in native interface). Note: FIX gateway checks that tags 336 and 55 combination points to existing security. If there is no match, the order is rejected with 'Unknown Security' error message.
54	Side	Y	char	'1' (Buy); '2' (Sell).	Side of the order. In case of swap order (currency market) Side is set on the base of spot with the later settlement date.
60	TransactTime	Y	UTCTimestamp		Time this order request was initiated/released by the trader, trading system, or intermediary. Required by FIX protocol but not processed at MOEX.
	component block <OrderQtyData>	Y			Quantity data of the order. Always expressed in lots of security. Please make sure that you get correct lot size values from marketdata streams. For REPO with CCP orders, monetary amount in the settlement currency may be specified instead of quantity in lots.
40	OrdType	Y	char	'1' (Market); '2' (Limit); 'W'(Weighted-average price) (Used only for Securities market)	Order type. Notes: 1. Market orders in closing or opening auctions are activated at auction collection phase 2. Market orders placed during closing or opening auctions stay active until the auction ends. Unmatched market orders remain active at trading at closing auction price period. 3. Market orders placed at trading at closing auction price period are automatically assigned closing auction price. The price assigned is not reported in Execution Report / New reply message. 4. Market orders are currently rejected for FX swap instruments 5. Certain Order types may be prohibited for specific user via trading system permissions
423	PriceType	N	int	'1' (Percentage); '2' (Per unit); '9' (Yield)	Code to represent the price type (Used only for Securities market). Note: Percentage for REPO with CCP trading has a REPO rate meaning.
44	Price	C	Price (9)		Order price. Expressed in 'per unit of quantity' or 'percentage' (for bonds). Required for limit OrdTypes. Must be zero for market orders at MOEX. Note: for REPO with CCP trading, price value contains REPO rate for a security. Maximum allowed length of Price field is 10 characters, including decimal point. Orders with price that does not fit in minimal price steps levels will be rejected.

Tag	Field name	Req'd	Type	Valid values	Comments
5202	TradeThruTime	N	char	C	Code to indicate the order 'in closing auction'. Used in Securities market. Only limit or market orders are accepted.
59	TimeInForce	N	char	'0' (Day); '3' (Immediate or Cancel); '4' (Fill or Kill);	Specifies how long the order remains in effect. Absence of this field is interpreted as DAY. At that, if the order received in the opening or closing period, then it would be placed with corresponding type ("At the Opening" or "At the Close") with corresponding price type.
168	EffectiveTime	N	UTCTimestamp		Time specified on the order at which the order should be considered valid (Used only for Securities market).
528	OrderCapacity	N	char	'P' (Principal)	Designates the capacity of the firm placing the order. Pair of OrderCapacity (528) = 'P' and OrderRestrictions (529) = '5' fields is used to represent market maker's order (Equities market only). Note: trader must have market maker permissions in the trading system, otherwise the order with this flag set will be rejected.
529	OrderRestrictions	N	MultipleValueString	'5' (Acting as Market Maker or Specialist in the security)	Restrictions associated with an order. Can contain multiple instructions separated by space. Pair of OrderCapacity (528) = 'P' and OrderRestrictions (529) = '5' fields is used to represent market maker's order (only for equities market).
1090	MaxPriceLevels	N	int	1 (non-splitted price)	Maximum number of price levels to trade through. If not present in the message, the number of price levels is unlimited.
<Standard Message Trailer>		Y			

4.2.3 Order Cancel Request (MsgType = 'F')

Order Cancel Request (F) message is used to cancel the order.

Order Cancel Request format is:

Tag	Field name	Req'd	Type	Valid values	Comments
<Standard Message Header>		Y			MsgType = F

41	OrigClOrdID	C	String(20)		ClOrdID of the previous non-rejected order (NOT the initial order of the day) when canceling an order. Used only if OrderID is not specified.
37	OrderID	C	String(12)		Unique identifier of most recent order as assigned by MOEX. Conditionally required if OrigClOrdID is not specified . If OrderID is specified then OrigClOrdID is ignored. Note: Order Cancel or Order Cancel Replace requests processing may result in the following cases: 1. MFIX Trade server could find exchange OrderID (37) that matches OrigClOrdID of Order Cancel Request, or OrderID (37) value is received in request. The request to cancel the order by the exchange OrderID(37) is then sent to matching engine. Matching engine attempts to find an active order with an exchange number matching the requested number, and eligible for cancellation by this user (SenderCompID). If an order is not found, then the return error string has the following format: (219) No orders withdrawn, 0 rejection(s). If an order is already filled or canceled, then this field contains error message indicating the order status. 2. No exchange OrderID exists for a given OrigClOrdID in Order Cancel Request. The request is rejected with Text field 58=can't find order. 3. If you send New Order Single or Order Cancel Replace or Order Cancel Request message via the same SenderCompID in short sequence, then these messages are processed synchronously on the path between the FIX server and matching engine: new request is not sent to matching engine until FIX server code receives reply to previous request. If an order is successfully registered or replaced, then the next cancel or replace request with specified OrigClOrdID will always find a match to OrderID and processing will go per variant 1. If order or order replacement were rejected, then OrderID does not exist and procession will follow variant 2 above.
11	ClOrdID	Y	String(20)		Unique ID of cancel request as assigned by the institution. Note: this field is placed in the UPDORDERREF field of native ORDERS table. FIX messages of types 35=F (Order Cancel Request) and 35=G (Order Cancel Replace Request) will be rejected via the Reject message (35=3) if the ClOrdID field (11) of these messages starts with a hash (pound) symbol '#'. You can use '#' symbol in any position of this string, except the first one.

54	Side	Y	char	'1' (Buy); '2' (Sell).	Side of the order. Required by FIX specifications, but ignored in processing by MOEX.
60	TransactTime	Y	UTCTimestamp		Time this order request was initiated/released by the trader or trading system. Required by FIX specifications, but ignored in processing by MOEX.
<Standard Message Trailer>		Y			

4.2.4 Order Cancel/Replace Request (G)

Order Cancel/Replace Request (G) message is used to replace the order.

Order Cancel Replace request is processed as an atomic action: no other request can be processed in between canceling original order and registering replaced order. Order Cancel Replace processing time (message roundtrip time) is longer than for New order Single message, but it is less than total roundtrip time for Order Cancel Request + New Order single pair. Average delivery time of Order Cancel Replace Request from client to the matching engine input queue does not differ from message delivery time for other messages ((New Order Single or Order Cancel Request).

This request has the following limitations, subject to change in future releases:

- Partially filled orders are not eligible for replacement. An attempt to replace partially filled order will be rejected with the error code in Text (58) field of Cancel Reject message being equal to 900. Field 58 string formats for each matching engine response types are fixed, but they have two different forms in Russian and English languages.
    - o The original order will be canceled if the CancelOrigOnReject (9619) flag is set to Y and not canceled for other values of this field or if the field is not in the message.
    - o For CancelOrigOnReject (9619)=Y, the order originator's SenderCompID will receive Execution Report with order cancellation and ClOrdID(11) being equal to the ClOrdID(11) of a message that created an order.
    .
- Iceberg orders are not eligible for replacement. Use separate Order Cancel Request and New Order single messages
- Orders in closing auction and REPO with CCP orders are not eligible for replacement. Use separate Order Cancel Request and New Order single messages
- You can change order price (44), quantity (38), or SecondaryClOrdID. Other order parameters must match parameters of an order to be amended
- Amended order always loses its time priority
- Amended order is assigned a new OrderID (37) and new MDEntryID (278)

- If an order is amended not by the SenderCompID, who has created or modified an order, then original order originator's SenderCompID will receive Execution Report with order cancellation and ClOrdID(11) being equal to the ClOrdID(11) of a message that created an order.
- Order Cancel Request messages that contain fields not listed in a table below will be rejected

Order Cancel/Replace Request (G) format is:

Tag	Поле	Наличие	Тип	Допустимые значения	Примечание
	<Standard Message Header>	Y			MsgType = 'G'
11	ClOrdID	Y	String(20)		Unique ID of replace request as assigned by the institution. FIX messages of types 35=F (Order Cancel Request) and 35=G (Order Cancel Replace Request) will be rejected via the Reject message (35=3) if the ClOrdID field (11) of these messages starts with a hash (pound) symbol '#'. You can use '#' symbol in any position of this string, except the first one.
41	OrigClOrdID	C	String (20)		ClOrdID of the previous non-rejected order (NOT the initial order of the day) when replacing an order. Conditionally required if OrderID is not specified. Note: Order Cancel or Order Cancel Replace requests processing may result in the following cases: 1. MFIX Trade server could find exchange OrderID (37) that matches OrigClOrdID of Order Cancel Request, or OrderID (37) value is received in request. The request to cancel the order by the exchange OrderID(37) is then sent to matching engine. Matching engine attempts to find an active order with an exchange number matching the requested number, and eligible for cancellation by this user (SenderCompID). If an order is not found, then the return error string has the following format: (219) No orders withdrawn, 0 rejection(s). If an order is already filled or canceled, then this field contains error message indicating the order status. 2. No exchange OrderID exists for a given OrigClOrdID in Order Cancel Request. The request is rejected with Text field 58=can't find order. 3. If you send New Order Single or Order Cancel Replace or Order

					Cancel Request messages via the same SenderCompID in short sequence, then these messages are processed synchronously on the path between the FIX server and matching engine: new request is not sent to matching engine until FIX server code receives reply to previous request. If an order is successfully registered or replaced, then the next cancel or replace request with specified OrigClOrdID will always find a match to OrderID and processing will go per variant 1. If order or order replacement were rejected, then OrderID does not exist and procession will follow variant 2 above.
37	OrderID	C	String (12)		Unique identifier of most recent order as assigned by MOEX. Conditionally required if OrigClOrdID is not specified. If OrderID is specified then OrigClOrdID is ignored.
1	Account	Y*	String (12)		Must match the Account field value of an order to be amended.
	component block <Parties>	N			Parties of the order. Client code (tag 448) value must match the value of an order to be amended. Note: if Client Code (448) field is specified for an order placed for the firm's own account, then the client code is ignored by the trading system and CLIENTCODE fields of ORDERS table remains empty. This order can be amended only if the Order Cancel Replace request has empty tag 448.
< «Instrument» component block>		Y			Must match the value of an order to be amended.
44	Price	C	Price		Order price. Expressed in 'per unit of quantity' or 'percentage' (for bonds). Required for limit OrdTypes. Maximum allowed length of Price field is 10 characters, including decimal point. Orders with price that does not fit in minimal price steps levels will be rejected.
38	OrderQty	C	Qty (10)		Quantity ordered, expressed in number of lots.
526	SecondaryClOrdID	N	String(12)		This field's value is sent to the EXTREF field of native MOEX order to be amended. Please consult your broker how to fill this field.
9619	CancelOrigOnReject	N	String (1)	Y – cancel original order if new order cannot be placed N – do not cancel original order if new order cannot be placed	Notes: • If the flag not defined, then the default behavior is to not cancel original order • The flag only affects accepted Cancel Replace message. If it is rejected due to not supported order type or not matching the original order's

386	NoTradingSessions	Y*	NumInGroup	1 (one element)	Account, Side, Parties, Instrument, TradingSessionId fields, then the original order remains intact. Specifies the number of repeating TradingSessionIDs. TradingSessionIDs group should contain only one element of group. Note: tags 386 and 336 compose a group and should be placed exactly in the order 386, then 336, and not separated by other tags.
336	=> TradingSessionID	Y*	String (4)		Identifier for Trading Session which contains MOEX security board (SECBOARD). Must match the value of an order to be amended
40	OrdType	Y	char		Type the order for replace. Must match the value of an order to be amended.
54	Side	Y	char	'1' (Buy); '2' (Sell).	Side of the order. Must match the value of an order to be amended.
60	TransactTime	Y	UTCTimestamp		Time this request was initiated/replaced by the trader, trading system, or intermediary. Required by FIX protocol but not processed at MOEX.
<Группа «Trailer»>		Y			

4.2.5 Order Mass Cancel Request (MsgType = 'q')

The Order Mass Cancel Request (q) message requests the cancelation of all of the remaining quantity of a group of orders matching criteria specified within the request. NOTE: This message can only be used to cancel order messages (reduce the full quantity).

- For canceling orders by particular security the following parameters should be defined:
    MassCancelRequestType (530) = '1', TradingSessionID (336) = <SECBOARD>, Symbol (55) = <SECCODE>;
- For canceling buy orders the following parameters should be defined:
    MassCancelRequestType (530) = '7', Side (54) = '1';
- For canceling sell orders the following parameters should be defined:
    MassCancelRequestType (530) = '7', Side (54) = '2';
- For canceling orders by particular trading account the following parameters should be defined:
    MassCancelRequestType (530) = '7', Account (1) = <trading account>;
- For canceling orders by particular user (trader, broker) the following parameters should be defined:
    MassCancelRequestType (530) = '7', NoPartyID (453) = 1, PartyID (448) = <user identifier>, PartyIDSource (447) = 'D', PartyRole (452) = '12';
- For canceling orders by particular firm the following parameters should be defined:
    MassCancelRequestType (530) = '7', NoPartyID (453) = 1, PartyID (448) = <firm identifier>, PartyIDSource (447) = 'D', PartyRole (452) = '1';

- For canceling all orders the following parameter should be defined:

  MassCancelRequestType (530) = '7'. Note: MOEX will cancel all orders according to trader's permissions to cancel orders in the trading system.

ClOrdID (11) and TransactTime (60) fields always should be present in Order Mass Cancel Request (q).

Order Mass Cancel Request format is:

Tag	Field name	Req'd	Type	Valid values	Comments
	<Standard Message Header>	Y			MsgType = q (lowercase Q)
11	ClOrdID	Y	String		Unique ID of Order Mass Cancel Request as assigned by the institution.
530	MassCancelRequestType	Y	char	'1' (Cancel orders for a security); '7' (Cancel all orders)	Specifies the type of cancellation requested. '99' (Cancel orders by multiply qualifier) is used to cancel orders by more restricted conditions (i.e. by trader and by side).
336	TradingSessionID	C*	String		Trading Session contains security board in which orders are to be canceled. Required if MassCancelRequestType (530) = '1' (Cancel orders for a security).
	component block <Instrument>	C*			Ticker symbol of the security of the orders are going to be cancelled. Required if MassCancelRequestType (530) = '1' (Cancel orders for a security).
54	Side	N	char	'1' (Buy); '2' (Sell).	Optional qualifier used to indicate the side of the market for which orders are to be canceled. Absence of this field indicates that orders are to be canceled regardless of side.
60	TransactTime	Y	UTCTimestamp		Time this request was initiated/released by the trader or trading system. Required by FIX protocol but not processed at MOEX.
1	Account	N	String(12)		Account mnemonic as agreed between buy and sell sides. It used to indicate the account for which orders are to be canceled.
526	SecondaryClOrdID	N	String		This field's value is sent to the EXTREF field of native MOEX WD_ORDERS message. If specified, only orders that have EXTREF field equal to this value will be canceled.

component block <Parties>	N	*Parties of the orders are going to be cancelled. Is used to represent trader and/or firm whose orders should be cancelled.*
<Standard Message Trailer>	Y	

4.3 Messages from Server to Client

4.3.1 Trading Session Status (MsgType = 'h')

Trading Session Status (h) message is used to represent connection status with appropriate MOEX market. This message is sent to FIX client unsolicited in the following cases:

- o MOEX Trading system was restarted;
- o Connection between FIX server and MOEX Trading System was established;
- o Connection between FIX server and MOEX Trading System was terminated correctly;
- o Connection between FIX server and MOEX Trading System was terminated non-gracefully;
- o Reconnecting between FIX server and MOEX Trading System is performed.

Note: It is not required to wait for 340=101 after successful logon reply. Trading session status message should be used only in negative scenarios:

- Receiving 340=103 message indicates that you will not receive Execution Reports until you receive 340=101.

- All order/order cancel/order replace/order status request messages will be rejected between receiving 340=103 and receiving 340=101 messages.

- Receiving 340=101 after 340=103 indicates that normal operation of FIX gateway is restored

Trading Session Status format is:

Tag	Field name	Req'd	Type	Valid values	Comments
	<Standard Message Header>	Y			MsgType = 'h'
336	TradingSessionID	Y	String		Identifier of trading session which contains code of super market.

325	UnsolicitedIndicator	N	Boolean	'Y' (Yes)	Indicates whether or not message is being sent as a result of a subscription request or not.
340	TradSesStatus	Y	int	*'100' (Trading system restarted)* *'101' (Connection to MOEX market established)* *'102' (Connection to MOEX market terminated correctly)* *'103' (Connection to MOEX market terminated non-gracefully)* *'104' (Reconnecting to MOEX market)*	State of the trading session.
58	Text	N	String		Free format text string.
<Standard Message Trailer>		Y			

4.3.2   Execution Report (MsgType = '8')

The Execution Report (8) message is used to:

1. Confirm the receipt of an order;

2. Confirm changes to an existing order (i.e. accept cancel request);

3. Relay order status information;

4. Relay fill information on working orders;

5. Reject orders or incorrect order status request or order cancel request.

Execution Report message may contain Text(58) field with the matching engine reply string and return code for the events of new order registration, cancellation, replacement, or rejection of these actions. It is recommended to parse this string for return code in parenthesis and addidional information, for example:

```
58=(210) 1 order(s) with total balance 1 withdrawn, 0 order(s) not withdrawn
```

The string format is fixed, however, it has two different versions for English and Russian reply languages. Return codes are identical for both versions.

Execution Report (8) messages with the same OrderID (37) are always generated on the base of swap orders at currency market and REPO with CCP orders at equities market:

- 1st Execution Report (8) with Symbol (55) set to SECCODE* of original swap or REPO trade and all correspondent price and quantity fields.

- 2nd and 3rd Execution Report (8) messages refer to spot instruments with correspondent settlement dates of original swap or REPO trade. Symbol (55) is equal to correspondent spot, lot size matches spot instrument as well. LeavesQty (151) is always set to "0" and CumQty (14) = LastQty (6) for such spot Execution Reports.

Execution Report format is:

Tag	Field name	Req'd	Type	Valid values	Comments
	<Standard Message Header>	Y			MsgType = '8'
37	OrderID	Y	String		OrderID assigned by MOEX. Unique across the given market history.
526	SecondaryClOrdID	N	String		This field's value is sent from the EXTREF field of native MOEX ORDERS table. The value follows the value of this tag in the originating New Order Single message.
11	ClOrdID	Y	String		Unique identifier for Order within a day for a given SenderCompID as assigned by the buy-side (institution, broker, intermediary etc.). This field contains value of BROKERREF field of MOEX trading system ORDERS table. The value follows the value of this tag in the originating New Order Single message Note: ClOrdID can be nonunique in the case when order wasn't placed via MFIX Transactional.
41	OrigClOrdID	N	String		ClOrdID (11) of the previous order (NOT the initial order of the day) when canceling or replacing an order. Conditionally required for response to an electronic Cancel request originated by order's creator.
	<Parties>	N			Parties of order. Required for ExecType (150) not in ('0' (New), '6' (Pending Cancel), '8' (Rejected)). Usually is used to represent firm and client (if applicable).

17	ExecID	Y	String		Unique identifier of execution message as assigned by sell-side (broker, exchange, ECN) (will be 0 (zero) for ExecType= I (Order Status)). Note: for trades at MOEX, this field's value is <MOEX TRADENO>|<B/S direction>|<Trade Time in HHMMSS format at Moscow time>. TRADENO, Buy/Sell and Trade Time values match the end of day MOEX trading reports.
150	ExecType	Y	char	'0' (New); '4' (Canceled) '5' (Replace) '6' (Pending Cancel); '8' (Rejected); 'F' (Trade); 'I' (Order Status)	Describes the purpose of the execution report.
39	OrdStatus	Y	char	'0' (New); '1' (Partially filled); '2' (Filled); '4' (Canceled); '6' (Pending Cancel); '8' (Rejected); '9' (Suspended) 'E'(Pending Replace);	Describes the current state of a CHAIN of orders, same scope as OrderQty, CumQty, LeavesQty, and AvgPx.
636	WorkingIndicator	N	Boolean	'N' (No) 'Y' (Yes)	Indicates if the order is currently being worked. Applicable only for OrdStatus = "New". This field is used for orders with defined EffectiveTime (168) field.
5202	TradeThruTime	N	char	'C' (Order in Closing Auction)	Type of activation order's event. Used only for Order in Closing Auction.
103	OrdRejReason	N	int	'1' (Unknown symbol); '3' (Order exceeds limit); '5' (Unknown Order); '6' (Duplicate Order (e.g. dupe ClOrdID (11))); '11' (Unsupported order characteristic); '13' (Incorrect quantity); '15' (Unknown account(s)); '99' (Other)	Code to identify reason for order rejection. Usually is defined for ExecType = '8'.
378	ExecRestatementReason	N	int	'97' (Cancel when can't be filled immediately); '98' (Cancel at the end of trading session or cancel when attempt to make cross trade);	Reason of unsolicited cancel.

				'100' (Cancel on disconnect)	
1	Account	Y*	String		Account mnemonic as agreed between buy and sell sides, e.g. broker and institution or investor/intermediary and fund manager. Is used to represent trading account. Corresponds to the ACCOUNT field of MICEX Bridge API.
64	SettlDate	N*	LocalMktDate		Specific date of trade settlement (SettlementDate) in YYYYMMDD format
5020	OptionSettlDate	N	LocalMktDate		Specific date of settlement as assigned to an order (SettlementDate) in YYYYMMDD format. The field is present only for execution reports with a trade. Note: order's settlement date may not match trade settlement date for linked trades of currency swaps and REPO with CCP trades.
5459	OptionSettlType	N	String		MOEX settlement code for a trade
<Instrument>		Y			Ticker symbol of the security of the order (SECCODE).
54	Side	Y	char	'1' (Buy); '2' (Sell)	Side of order. In case of Execution Report for swap order Side is set on the base of spot with the later settlement date.
<Order Qty Data>		N			Quantity data of the order.
40	OrdType	N	char	'1' (Market); '2' (Limit); 'W'(Weighted-average price)	Order type.
423	PriceType	N	int	'1' (Percentage) '2' (Per unit); '9' (Yield) (Used only for Equities market)	Code to represent the price type. Note: Percentage for REPO with CCP trading has a REPO rate meaning.
44	Price	C	Price		Order price expressed in 'per unit of quantity'. Required if specified on the order.
59	TimeInForce	N	char	'0' (Day); '3' (Immediate or Cancel); '4' (Fill or Kill); '7' (At the Close)	Specifies how long the order remains in effect. Absence of this field is interpreted as DAY.

168	EffectiveTime	N	UTCTimestamp		Time specified on the order at which the order should be considered valid (Used only for equities market).
528	OrderCapacity	N	char	'P' (Principal)	Designates the capacity of the firm placing the order. Pair of OrderCapacity (528) = 'P' and OrderRestrictions (529) = '5' fields is used to represent market maker's order.
529	OrderRestrictions	N	MultipleValueString	'5' (Acting as Market Maker or Specialist in the security)	Restrictions associated with an order. Can contain multiple instructions separated by space. Pair of OrderCapacity (528) = 'P' and OrderRestrictions (529) = '5' fields is used to represent market maker's order.
32	LastQty	C	Qty		Quantity (e.g. shares) bought/sold on this (last) fill. Required if ExecType (150) = Trade.
31	LastPx	C	Price		Price of this (last) fill. Required if ExecType (150) = Trade. Note: Percentage for REPO with CCP trades has a REPO rate meaning.
336	TradingSessionID	Y*	String		Identifier for Trading Session is used to represent SECBOARD.
625	TradingSessionSubID	N	String	'OPEN' – opening period 'S' – opening auction period 'N' – normal trading period 'CLOS' – closing period 'CA' - closing auction 'D' -dark pool auction period (only Equities) 'I' -discrete auction period 'E'- Trading at the closing auction price period	Period of trading session. I.e. "OPEN", "CLOS".
151	LeavesQty	Y	Qty		Quantity open for further execution (order balance). LeavesQty is always "0" in spot Execution Reports of swap orders. LeavesQty is always "0" if OrdStatus (39) in ('4', '6', '8')'. Else LeavesQty = OrderQty – CumQty.
14	CumQty	Y	Qty		Currently executed shares for chain of orders. CumQty is always "0" if OrdStatus (39) = '6' (Pending Cancel).
6	AvgPx	Y	Price		Calculated average price of all fills on this order. Always '0' for MOEX.

60	TransactTime	C*	UTCTimestamp		The time at which the Order, Order Cancel, Order Cancel Replace message was registered, or time of trade or unsolicited order modification . Required if MsgType (35) in ('8', '9'). Note: Tags 60 and 9412 form microsecond precision timestamp for a time when an order execution started at matching engine. If there is an execution queue at the matching engine entry, then the order registration timestamp reflects the queue waiting time.
9412	OrigTime	N	int		Indicates microseconds fraction of time at which the Order was registered, replaced, withdrawn, rejected, or time of trade.. Microseconds fraction should be added to the tag 60 value to enhance its precision.
6636	StipulationValue	N	char	'Y' (Yes) 'N' (No)	Set to "Y" if Trade occurred for own iceberg-order
6029	CurrencyCode	N	String		Settlement currency code for a trade
5155	InstitutionID	N	String		Settlement organization ID. BANKID field in TRADES table of native API
7695	CoverID	N	char	'Y' (Yes) 'N' (No)	Block collaterals on a special account for REPO trades
7693	ClientAccID	N	String		Settlement account/code ID in the settlement house. BANKACCID field in TRADES table of native API.
9173	TradeIssueDate	N	Date		Trade date
9945	OrigOrderID	N	String		Exchange ID of an order that has been replaced (tag 37value of Order Cancel Replace request)
9580	ParentID	N	String		Indicates the showcase trade number of REPO or currency swap trade for linked spot trades..
<Commission Data>		N			Total trade commission. Available in MFIX Trade Capture and MFIX Drop Copy according to user permissions. GROUP IS REMOVED from MFIX Trade since January 2015
<Yield Data>		N			Yield calculated at the trade price.
159	AccruedInterestAmt	N	Amt		Amount of Accrued Interest for trade (Used only for Equities market).
58	Text	N	String		Free format text string.
136	NoMiscFees	N	NumInGroup		**Number of repeating MiscFees group entries.**

				Available in MFIX Trade Capture and MFIX Drop Copy according to user permissions. **REMOVED in MFIX Trade since January 2015.**
137	=>MiscFeeAmt	C	Amt		**Miscellaneous fee value is used to represent the amount of trade commission. Required if NoMiscFees (136) > 0 (Used only for Equities market).** **Available in MFIX Trade Capture and MFIX Drop Copy according to user permissions.** **REMOVED in MFIX Trade since January 2015.**
139	=>MiscFeeType	C	int	'4' (Exchange fees); '98' (Clearing fees); '99' (Technical access fees)	**Miscellaneous fee value is used to represent type of trade commission. Required if NoMiscFees (136) > 0.** **Available in MFIX Trade Capture and MFIX Drop Copy according to user permissions.** **REMOVED in MFIX Trade since January 2015.**
	<TrdRegTimestamps>	N			*Additional time stamps of the order/execution is used to represent time when order was registered at MOEX.*
1090	*MaxPriceLevels*	N	*int*	*1 (non-splitted price)*	*Maximum number of price levels to trade through.*

278	*MDEntryID*	N	*string*		*MDEntryID (278) of an order in the FAST udp multicast marketdata Order List Refresh / Snapshot feed. This field helps to identify own orders in the marketdata feed.*
					 Public MDEntryID is assigned for:
					   - *new limit order*
					   - *IOC, FOK, market orders which are filled or partially filled*
					   - *Iceberg orders are assigned new MDEntryID for each replenishment of visible quantity*
					   - *Replaced orders are assigned new MDEntryID and new OrderID*
					    Orders that do not stay active in the book (filled at registration, market, IOC, FOK) are not published in the FAST marketdata feed, but may have assigned MDEntryID in Execution report.
					    If a limit order is filled, replaced, or canceled within a very short (less than 200 microseconds) time interval it may be not published in FAST feed, but have MDEntryID assigned in Execution report.

	<Standard Message Trailer>	Y			

4.3.3   Order Mass Cancel Report (MsgType = 'r')

The Order Mass Cancel Report (r) is used to acknowledge an Order Mass Cancel Request (q) message or reject incorrect ones. Note that each affected order that is canceled is acknowledged with a separate Execution Report (8) message.

Note: It is not guaranteed that client will receive this message if client disconnects immediately after Order Mass Cancel Request is sent. In such case only Execution Reports for each affected order will be sent.

Order Mass Cancel Report format is follows:

Tag	Field name	Req'd	Type	Valid values	Comments
<Standard Message Header>		Y			MsgType = r (lowercase R)
11	ClOrdID	N	String		ClOrdID provided on the Order Mass Cancel Request (q).
526	SecondaryClOrdID	N	String		This field's value is equal to the value of this tag in the originating Order Mass Cancel Request
37	OrderID	Y	String		Unique Identifier for the Order Mass Cancel Request assigned by the recipient of the Order Mass Cancel Request
530	MassCancelRequestType	Y	char	'0' (Cancel Request Rejected); '1' (Cancel orders for a security); '7' (Cancel all orders)	Order Mass Cancel Request Type accepted by the system
531	MassCancelResponse	Y	char	'0' (Cancel Request Rejected); '1' (Cancel orders for a security); '7' (Cancel all orders)	Indicates the action taken by the counterparty order handling system as a result of the Cancel Request. 0 – Indicates Order Mass Cancel Request was rejected.

Tag	Field name	Req'd	Type	Valid values	Comments
532	MassCancelRejectReason	C	int	'0' (Mass Cancel Not Supported); '1' (Invalid or unknown Security); '99' (Other)	Indicates why Order Mass Cancel Request was rejected. Required if MassCancelResponse = 0
336	TradingSessionID	C*	String		Trading Session contains security board in which orders were requested to be canceled. Required if MassCancelRequestType (530) = '1' (Cancel orders for a security).
	component block <Instrument>	C*			Ticker symbol of the security of the orders which were requested to be cancelled. Required if MassCancelRequestType (530) = '1' (Cancel orders for a security).
54	Side	N	char	'1' (Buy); '2' (Sell).	Optional qualifier used to indicate the order side for which were requested to be cancelled.
1	*Account*	*N*	*String(12)*		*Account of which orders were requested to be cancelled.*
	component block <Parties>	*N*			*Parties for which orders were requested to be cancelled.* *Is used to represent trader and/or firm whose orders should be cancelled.*
58	Text	N	String		Free format text string.
60	TransactTime	*N*	UTCTimestamp		*The time at which the Order Mass Cancel Request was registered*
9412	*OrigTime*	*N*	*int*		*Indicates microseconds fraction of time at which Order Mass Cancel Request was registered. Microseconds fraction should be added to the tag 60 value to enhance its precision.*
	<Standard Message Trailer>	Y			

4.3.4   Order Cancel Reject (MsgType = '9')

The Order Cancel Reject (9) message is issued by the MFIX Transactional server upon receipt of a cancel request or cancel replace message which cannot be honored (i.e. when order is already in final state).

Order Cancel Reject format is follows:

Tag	Field name	Req'd	Type	Valid values	Comments

	<Standard Message Header>	Y			MsgType = 9
37	OrderID	Y	String		OrdrID of the order which can't be canceled. If CxlRejReason="Unknown order", specify "NONE".
11	ClOrdID	Y	String		Unique identifier of cancel request.
41	OrigClOrdID	Y	String		ClOrdID which could not be canceled. ClOrdID of the previous accepted order (NOT the initial order of the day) when canceling or replacing an order.
39	OrdStatus	Y	char	'8' (Rejected) '4' (Canceled) '2' (Filled)	*OrdStatus indicates order status at a time when corresponding Order Cancel Request was received. Order status is not modified in the case of order cancel reject is sent.* *Note: if you receive OrdStatus as canceled or filled in Order Cancel Reject message, then this ensures that an order is no more active at matching engine. You may receive Execution Reports indicating final order status later then this rejection, however you can safely use order status value in Order Cancel Reject to accelerate processing at your side.*
434	CxlRejResponseTo	Y	char	'1' (Order Cancel Request) '2' (Order Cancel/Replace Request)	Identifies the type of request that an Order Cancel Reject (9) is in response to.
102	CxlRejReason	N	int	'0' (Too late to cancel) '1' (Unknown order); '3' (Order already in Pending Cancel or Pending Replace status); '6' (Duplicate ClOrdID (11) received); '11'(Value ClOrdID more than 20 symbols) '99' (Other)	Code to identify reason for cancel rejection.
58	Text	N	String		Free format text string is used to represent detail reject reason. Note: Order Cancel or Order Cancel Replace requests processing may result in the following cases: 1. MFIX Trade server could find exchange OrderID (37) that matches OrigClOrdID of Order Cancel Request, or OrderID (37) value is received in request. The request to cancel the order by the exchange OrderID(37) is then sent to matching engine. Matching engine attempts

					to find an active order with an exchange number matching the requested number, and eligible for cancellation by this user (SenderCompID). If an order is not found, then the return error string has the following format: (219) No orders withdrawn, 0 rejection(s). If an order is already filled or canceled, then this field contains error message indicating the order status. 2. No exchange OrderID exists for a given OrigClOrdID in Order Cancel Request. The request is rejected with Text field 58=can't find order. If you send New Order Single or Order Cancel Replace or Order Cancel Request message via the same SenderCompID in short sequence, then these messages are processed synchronously on the path between the FIX server and matching engine: new request is not sent to matching engine until FIX server code receives reply to previous request. If an order is successfully registered or replaced, then the next cancel or replace request with specified OrigClOrdID will always find a match to OrderID and processing will go per variant 1. If order or order replacement were rejected, then OrderID does not exist and processing will follow variant 2 above.
60	TransactTime	N	UTCTimestamp		*The time at which the rejected Order Cancel Request or Order Cancel Replace request was registered*
9412	*OrigTime*	N	*int*		*Indicates microseconds fraction of time at which the rejected Order Cancel Request or Order Cancel Replace request was registered. Microseconds fraction should be added to the tag 60 value to enhance its precision.*
<Standard Message Trailer>		Y			

5 MFIX Trade Capture

MFIX Trade capture service is used to receive Trade Capture Reports (AE) for all trades that are visible to a given UserID in the MOEX trading system. Depending on user permissions, the trades received may cover entire firm, a trader, or a limited set of trades that correspond to narrower permissions. The visibility permissions are set by MOEX trading system administrators.

To use MFIX Trade capture service, logon to the specified address+port+TargetCompId. The server then start sending Trade Capture reports for each eligible trade with minimal delays.

Note: you do not need to send Trade Capture Report Request in this updated version of the service.

5.1 Messages from Server to Client

5.1.1 Trading Session Status (MsgType = 'h')

Trading Session Status (h) message is used to represent connection status with appropriate MOEX market. This message is sent to FIX client unsolicited in the following cases:

- o MOEX Trading system was restarted;
- o Connection between FIX server and MOEX Trading System was established;
- o Connection between FIX server and MOEX Trading System was terminated correctly;
- o Connection between FIX server and MOEX Trading System was terminated non-gracefully;
- o Reconnecting between FIX server and MOEX Trading System is performed.

Note: It is not required to wait for 340=101 after successful logon reply. Trading session status message should be used only in negative scenarios:

- Receiving 340=103 message indicates that you will not receive Execution Reports until you receive 340=101.

- All order/order cancel/order replace/order status request messages will be rejected between receiving 340=103 and receiving 340=101 messages.

- Receiving 340=101 after 340=103 indicates that normal operation of FIX gateway is restored

Trading Session Status format is:

Tag	Field name	Req'd	Type	Valid values	Comments
<Standard Message Header>		Y			MsgType = 'h'
336	TradingSessionID	Y	String		Identifier of trading session which contains code of super market.
325	UnsolicitedIndicator	N	Boolean	'Y' (Yes)	Indicates whether or not message is being sent as a result of a subscription request or not.
340	TradSesStatus	Y	int	*'100' (Trading system restarted)* *'101' (Connection to MOEX market established)*	State of the trading session.

Tag	Field name		Type	Valid values	Comments
				'102' (Connection to MOEX market terminated correctly) '103' (Connection to MOEX market terminated non-gracefully) '104' (Reconnecting to MOEX market)	
58	Text	N	String		Free format text string.
<Standard Message Trailer>		Y			

5.1.2   Trade Capture Report (MsgType = 'AE')

This message represents trade data. Trade Capture Reports are sent automatically and do not require sending the Trade Capture Report Request to initiate their publishing.

Trade Capture Report format is as follows.

Tag	Field name	Req'd	Type	Valid values	Comments
<Standard Message Header>		Y			MsgType = 'AE'
571	TradeReportID	Y	String		Unique identifier for the Trade Capture Report (AE) message.
828	TrdType	Y*	int	'0' (Regular Trade) '2' (REPO with CCP); '22'(Privately Negotiated Trades); '53' (Repurchase agreement); '101' (Margin call); '102' (REPO with CCP assets refund technical deal)	Type of Trade.
829	TrdSubType	C*	int	'1' (1st REPO leg); '2' (2nd REPO leg); '3' (Settlement trade for SWAP); '4' (Settlement trade for currency basket)	Further qualification to the trade type. Required for TrdType (828) = '53' (Repurchase agreement) and for settlement trades for SWAPs and currency baskets. (Used for equities and Currency markets)

				'5 '(REPO with CCP); '6' (1st REPO with CCP leg) '7' (2nd REPO with CCP leg); '8' (Negotiated REPO with CCP deal); '9' (1st negotiated REPO with CCP deal leg); '10' (2nd negotiated REPO with CCP deal leg).	
818	SecondaryTradeReportID	N	String		Secondary trade report identifier - used as an identifier of original REPO trade. (Used for only equities market)
150	ExecType	Y*	char	'F' (Trade); 'H' (Trade Cancel).	Type of Execution being reported: Uses subset of ExecType for Trade Capture Reports.
17	ExecID	Y*	String		Exchanged assigned Execution ID (Trade Identifier).
570	PreviouslyReported	Y	Boolean	'N' (not reported to counterparty)	Indicates if the trade capture report was previously reported to the counterparty.
<Instrument>		Y		Ticker symbol of the security of the trade.	
<Financing Details>		N			Financing details of transaction.
<Yield Data>		N			Security yield.
711	NoUnderlyings	N	NumInGroup		Number of underlying legs that make up the security. (Used only for equities market)
<Underlying Instrument>		N			Additional information about security.
32	LastQty	Y	Qty		Trade Quantity.
31	LastPx	Y	Price		Trade Price.
1056	CalculatedCcyLastQty	N	Amt		Trade quantity expressed in the units of security. Calculated as , LastQty (32)* LotSize (the number of securities in one lot).
75	TradeDate	Y	LocalMktDate		Indicates date of trade referenced in this message in YYYYMMDD format. Absence of this field indicates current day (expressed in local time at place of trade).

				'5 '(REPO with CCP); '6' (1st REPO with CCP leg) '7' (2nd REPO with CCP leg); '8' (Negotiated REPO with CCP deal); '9' (1st negotiated REPO with CCP deal leg); '10' (2nd negotiated REPO with CCP deal leg).	
818	SecondaryTradeReportID	N	String		Secondary trade report identifier - used as an identifier of original REPO trade. (Used for only equities market)
150	ExecType	Y*	char	'F' (Trade); 'H' (Trade Cancel).	Type of Execution being reported: Uses subset of ExecType for Trade Capture Reports.
17	ExecID	Y*	String		Exchanged assigned Execution ID (Trade Identifier).
570	PreviouslyReported	Y	Boolean	'N' (not reported to counterparty)	Indicates if the trade capture report was previously reported to the counterparty.
<Instrument>		Y		Ticker symbol of the security of the trade.	
<Financing Details>		N			Financing details of transaction.
<Yield Data>		N			Security yield.
711	NoUnderlyings	N	NumInGroup		Number of underlying legs that make up the security. (Used only for equities market)
<Underlying Instrument>		N			Additional information about security.
32	LastQty	Y	Qty		Trade Quantity.
31	LastPx	Y	Price		Trade Price.
1056	CalculatedCcyLastQty	N	Amt		Trade quantity expressed in the units of security. Calculated as , LastQty (32)* LotSize (the number of securities in one lot).
75	TradeDate	Y	LocalMktDate		Indicates date of trade referenced in this message in YYYYMMDD format. Absence of this field indicates current day (expressed in local time at place of trade).

60	TransactTime	Y	UTCTimestamp		Time the transaction represented by this Trade Capture Report (AE) occurred. Whole seconds only timestamp is available.
63	SettlType	N	String	'1' (Cash); '2' (Next Day (T+1)) (Used only for Currency market); '3'(T+2);'4' (T+3); '4' (T+3); '5' (T+4); '6'(Future); 'Dx' (The amount of days(x)); 'Wx'(The amount of weeks(x)); 'Mx' (The amount of months(x)); 'Yx' (The amount of years(x))	Indicates order settlement period.
64	SettlDate	N	LocalMktDate		Specific date of trade settlement (SettlementDate) in YYYYMMDD format.
5459	OptionSettlType	N	String		MOEX settlement code for a trade
552	NoSides	Y	NumInGroup		Number of repeating Sides group entries.
54	=> Side	Y	char	'1' (Buy); '2' (Sell).	Side of order.
37	=> OrderID	N	String		MOEX assigned Order number that resulted in a trade.
11	=> ClOrdID	N	String		Unique identifier of the order as assigned by institution or by the intermediary (CIV term, not a hub/service bureau) with closest association with the investor. This field contains the value of BROKERREF field of an order in the native MOEX trading system ORDERS table. Note: ClOrdID can be nonunique in the case when order wasn't placed via MFIX Transactional.
526	=>SecondaryClOrdID	N	String		EXTREF field of executed order in the MOEX trading system Orders table.
=><Parties>		N			Parties of the trade. Usually contains firm identifier and client code (if applicable).
1	=> Account	N	String		Account mnemonic as agreed between buy and sell sides, e.g. broker and institution or investor/intermediary and fund manager. This field is used to represent trading account.

336	=> TradingSessionID	Y*	String		Identifier for Trading Session is used to represent SECBOARD.
625	=>TradingSessionSubID	N	String	'OPEN' – opening period 'S' – opening auction period 'N' – normal trading period 'CLOS' – closing period 'L' - closing auction 'D' -dark pool auction period (only FOND) 'I' -discrete auction period 'E'- Trading at the closing auction price period 'IPO' - primary distribution	Period of trading session. I.e. "OPEN", "CLOS". Note: If new trading period is added to the MOEX trading system, this field will have additional values.
=><Commission Data>		N			Total trade commission.
381	=>GrossTradeAmt	N	Amt		Amount of this trade expressed in units of currency.
159	=>AccruedInterestAmt	N	Amt		Amount of Accrued Interest for convertible bonds and fixed income (Used only for Equitiesequities market).
920	=>EndAccruedInterestAmt	N	Amt		Accrued Interest Amount applicable to a financing transaction on the End Date (Used only for Equitiesequities market).
921	=>StartCash	N	Amt		Starting dirty cash consideration of a financing deal, i.e. paid to the seller on the Start Date (Used only for Equities market).
922	=>EndCash	N	Amt		Ending dirty cash consideration of a financing deal. i.e. reimbursed to the buyer on the End Date (Used only for Equities market).
136	=>NoMiscFees	N	NumInGroup		Number of repeating MiscFees group entries.
137	=>=>MiscFeeAmt	C	Amt		Miscellaneous fee value is used to represent the amount of trade commission. Required if NoMiscFees (136) > 0.
139	=>=>MiscFeeType	C	int	'4' (Exchange fees); '98' (Clearing fees); '99' (Technical support fees)	Miscellaneous fee value is used to represent type of trade commission. Required if NoMiscFees (136) > 0.
162	=>SettlInstID	N	String		Unique identifier for Settlement Instruction.

640	Price2	N	Price		Price of the future part. (Used only for equities market)
44	Price	N	Price		Price per unit of quantity.
423	PriceType	N	int		Code to represent the price type.
5155	InstitutionID	N	String		Settlement organization ID. BANKID field in TRADES table of native API
5202	TradeThruTime	N	char	'C' (Order in Closing Auction)	Type of order's activation event. Used only for Order in Closing Auction.
5440	ClearingStatus	N	String		Clearing status. (Used only for equities market)
5934	ClearingRefNo	N	String		A unique reference number assigned by the clearing system. (Used only for equities market)
6029	CurrencyCode	N	String		Settlement currency code for a trade
7693	ClientAccID	N	String		Settlement account/code ID in the settlement house. BANKACCID field in TRADES table of native API.
9580	ParentID	N	String		Indicates the showcase trade number of REPO or currency swap trade for linked spot trades..
9938	ClearingHandlingType	N	String		Type of clearing.
10003	OperationType	N	int		Operation type.
10104	Price1	N	Price		Price of the first REPO leg. (Used only for equities market)
10105	CurrentRepoValue	N	Amt		REPO value calculated for the current date. (Used only for equities market)
10106	CurrentRepoBackValue	N	Amt		Current value of the 2nd (buy-back) REPO leg calculated for the settlement date. (Used only for equities market)
10110	TradeBalance	N	Qty		Volume of unmatched part of an order. (Used only for equities market)
10111	TotalAmount	N	Amt		Total value of the trade.

10117	LastCouponPaymentValue	N	Amt		Value of coupon payment since the last security trading day. (Used only for equities market)
10118	LastCouponPaymentDate	N	LocalMktDate		Date of the last coupon payment. (Used only for equities market)
10119	LastPrincipalPaymentValue	N	Amt		Principal debt payment. (Used only for equities market)
10120	LastPrincipalPaymentDate	N	LocalMktDate		Date of principal debt payment. (Used only for equities market)
10121	ExpectedDiscount	N	Price		Discount after deposit. (Used only for equities market)
10122	ExpectedQty	N	Qty		Quantity after deposit. (Used only for equities market)
10123	ExpectedRepoValue	N	Amt		Value after deposit. (Used only for equities market)
10124	ExpectedRepoBackValue	N	Amt		Repurchase value after deposit. (Used only for equities market)
10125	ExpectedReturnValue	N	Amt		Return value after deposit. (Used only for equities market)
10126	PreciseBalance	N	Qty		Precise shares quantity. (Used only for equities market)
<Standard Message Trailer>		Y			

6   MFIX Drop Copy

6.1   Execution Report (MsgType = '8')

MFIX Drop Copy uses Execution Report message (8) to transmit the information. The information is provided for firms and contains orders and trades for the connected user's scope of visibility. Typically, you should use this service on a firm level, which requires logging on by a user with Firm Manager role in MOEX trading system. For specialized use, the scope of visibility may be limited to a user group.

Notes:

- MFIX Drop Copy message traffic is bandwidth limited to avoid massive overloading of network lines by large volume of reports. The Execution Report message delivery latency is not expected to be very low. One should expect latencies of below 100 milliseconds in normal operation.

- Drop Copy does not deliver reports for rejected messages that do not alter the list of orders within the scope of visibility

Execution Report format is:

Tag	Field name	Req'd	Type	Valid values	Comments
	\<Standard Message Header\>	Y			MsgType = '8'
37	OrderID	Y	String		OrderID assigned by MOEX. Must be unique across the given market history.
526	SecondaryClOrdID	N	String		This field's value is sent from the EXTREF field of native MOEX ORDERS table. The value follows the value of this tag in the originating New Order Single message.
11	ClOrdID	Y	String		Unique identifier for Order within a day for a given SenderCompID as assigned by the buy-side (institution, broker, intermediary etc.). This field contains value of BROKERREF field of MOEX trading system ORDERS table. The value follows the value of this tag in the originating New Order Single message

Note: ClOrdID can be nonunique in the case when order wasn't placed via MFIX Transactional. |
| 41 | OrigClOrdID | N | String | | ClOrdID (11) of the previous order (NOT the initial order of the day) when canceling an order. Conditionally required for response to an electronic Cancel request (ExecType (150) = PendingCancel or Cancelled (only for cancel by user request)). |
| | \<Parties\> | N | | | Parties of order. Required for ExecType (150) not in ('0' (New), '6' (Pending Cancel), '8' (Rejected)).
Usually is used to represent firm and client (if applicable). |

17	ExecID	Y	String		Unique identifier of execution message as assigned by sell-side (broker, exchange, ECN) (will be 0 (zero) for ExecType= I (Order Status)). Note: for trades at MOEX, this field's value is <MOEX ORDERNO>|<B/S direction>|<Trade Time in HHMMSS format at Moscow time>. ORDERNO, Buy/Sell and Trade Time values match the end of day MOEX trading reports.
150	ExecType	Y	char	'0' (New); '4' (Canceled) '5' (Replace) '6' (Pending Cancel); '8' (Rejected); 'F' (Trade); 'I' (Order Status)	Describes the purpose of the execution report.
39	OrdStatus	Y	char	'0' (New); '1' (Partially filled); '2' (Filled); '4' (Canceled); '6' (Pending Cancel); '8' (Rejected); '9' (Suspended) 'E'(Pending Replace);	Describes the current state of a CHAIN of orders, same scope as OrderQty, CumQty, LeavesQty, and AvgPx.
636	WorkingIndicator	N	Boolean	'N' (No) 'Y' (Yes)	Indicates if the order is currently being worked. Applicable only for OrdStatus = "New". This field is used for orders with defined EffectiveTime (168) field.
5202	TradeThruTime	N	char	'C' (Order in Closing Auction)	Type of activation order's event. Used only for Order in Closing Auction.
378	ExecRestatementReason	N	int	'97' (Cancel when can't be filled immediately); '98' (Cancel at the end of trading session or cancel when attempt to make cross trade); '100' (Cancel on disconnect)	Reason of unsolicited cancel.
1	Account	Y*	String		Account mnemonic as agreed between buy and sell sides, e.g. broker and institution or investor/intermediary and fund manager. Is used to represent trading account.
64	SettlDate	N*	LocalMktDate		Specific date of trade settlement (SettlementDate) in YYYYMMDD format (Used only for Equities and Currency markets).
<Instrument>		Y			Ticker symbol of the security of the order (SECCODE).
54	Side	Y	char	'1' (Buy); '2' (Sell)	Side of order. In case of Execution Report for swap order Side is set on the base of spot with

					the later settlement date.
<Order Qty Data>		N			Quantity data of the order.
40	OrdType	N	char	'1' (Market); '2' (Limit); 'W'(Weighted-average price) (Used only for Equities and Currency markets)	Order type.
423	PriceType	N	int	'1' (Percentage) (Used only for Equities market); '2' (Per unit); '9' (Yield) (Used only for Equities market)	Code to represent the price type.
44	Price	C	Price		Order price expressed in 'per unit of quantity'. Required if specified on the order.
59	TimeInForce	N	char	'0' (Day); '3' (Immediate or Cancel); '4' (Fill or Kill); '7' (At the Close) (Used only for Equities market)	Specifies how long the order remains in effect. Absence of this field is interpreted as DAY.
168	EffectiveTime	N	UTCTimestamp		Time specified on the order at which the order should be considered valid (Used only for Equities market).
528	OrderCapacity	N	char	'P' (Principal)	Designates the capacity of the firm placing the order. Pair of OrderCapacity (528) = 'P' and OrderRestrictions (529) = '5' fields is used to represent market maker's order (Used only for Equities market).
529	OrderRestrictions	N	MultipleValueString	'5' (Acting as Market Maker or Specialist in the security)	Restrictions associated with an order. Can contain multiple instructions separated by space. Pair of OrderCapacity (528) = 'P' and OrderRestrictions (529) = '5' fields is used to represent market maker's order (Used only for Equities market).
32	LastQty	C	Qty		Quantity (e.g. shares) bought/sold on this (last) fill. Required if ExecType (150) = Trade.
31	LastPx	C	Price		Price of this (last) fill. Required if ExecType (150) = Trade.
336	TradingSessionID	Y*	String		Identifier for Trading Session is used to represent SECBOARD.

625	TradingSessionSubID	N	String	'OPEN' – opening period 'S' – opening auction period 'N' – normal trading period 'CLOS' – closing period 'FCLOS'-final closing period 'CA' - closing auction 'D' - dark pool auction period (only FOND) 'I' -discrete auction period 'E'- Trading at the closing auction price period 'IPO' - primary distribution	Period of trading session. I.e. "OPEN", "CLOS".
151	LeavesQty	Y	Qty		Quantity open for further execution (order balance). LeavesQty is always "0" in spot Execution Reports of swap orders. LeavesQty is always "0" if OrdStatus (39) in ('4', '6', '8')'. Else LeavesQty = OrderQty – CumQty.
14	CumQty	Y	Qty		Currently executed lots for chain of orders. CumQty is always "0" if OrdStatus (39) = '6' (Pending Cancel).
6	AvgPx	Y	Price		Calculated average price of all fills on this order. Always '0' for MOEX.
60	TransactTime	C*	UTCTimestamp		*The time at which the Order was registered, withdrawn, replaced, or time of trade. Required if ExecType (150) in ('F', '4').*
9412	OrigTime	N	int		*Indicates the time of the transaction in microseconds as indicated by the Originating system. Microseconds fraction should be added to the tag 60 value to enhance its precision.*
6636	StipulationValue	N	char	*'Y' (Yes)* *'N' (No)*	*Set to "Y" if Trade occurred for own iceberg-order*
<Commission Data>		N			Total trade commission.
<Yield Data>		N			Yield calculated at the trade price.
6029	CurrencyCode	N	String		*Settlement currency code for a trade*
5155	InstitutionID	N	String		*Settlement organization ID. BANKID field in TRADES table of native API*
7695	CoverID	N	char	*'Y' (Yes)* *'N' (No)*	*Block collaterals on a special account for REPO trades*

7693	ClientAccID	N	String		Settlement account/code ID in the settlement house. BANKACCID field in TRADES table of native API.
9173	TradeIssueDate	N	Date		Trade date
9945	OrigOrderID	N	String		Exchange ID of an order that has been replaced (tag 37value of Order Cancel Replace request)
9580	ParentID	N	String		Indicates the showcase trade number of REPO or currency swap trade for linked spot trades.
159	AccruedInterestAmt	N	Amt		Amount of Accrued Interest for trade (Used only for Equities market).
58	Text	N	String		Free format text string.
136	NoMiscFees	N	NumInGroup		Number of repeating MiscFees group entries.
137	=>MiscFeeAmt	C	Amt		Miscellaneous fee value is used to represent the amount of trade commission. Required if NoMiscFees (136) > 0 (Used only for Equities market).
139	=>MiscFeeType	C	int	'4' (Exchange fees); '98' (Clearing fees); '99' (Technical access fees)	Miscellaneous fee value is used to represent type of trade commission. Required if NoMiscFees (136) > 0.
<TrdRegTimestamps>		N			Additional time stamps of the order/execution is used to represent time when order was registered at MOEX.
1090	MaxPriceLevels	N	int	1 (non-splitted price)	Maximum number of price levels to trade through.
<Standard Message Trailer>		Y			