MICEX

USER MANUAL

PUBLIC FIX INTERFACE

Of MICEX-FIX MARKET DATA

Version 1.0

Table of Contents

Introduction

Financial Information eXchange (FIX™) protocol protocol is a technical specification (language) for communicating financial information electronically.

The FIX Protocol is built and maintained by FIX Protocol Ltd; an independent committee of industry participants from across the globe. For the latest FIX specifications or more information on FIX Protocol Ltd and the FIX Protocol, please visit http://www.fixprotocol.org.

You can download the FIX standard Version 4.4 or Version4.2 specifications from the following link: http://www.fixprotocol.org/specifications. See Version 4.4 (with Errata 20030618) or Version 4.2 (with Errata 20010501) accordingly. Refer to Volume 3 for the section on market data.

The “Public FIX interface and User Guide” document describes the FIX standard protocol implementation in the MICEX-FIX MARKET DATA. It provides clients with understanding how to connect and use MICEX-FIX Bridge for requesting and receiveing market data. This document details the message specifications of all of the FIX messages that are being supported by MICEX.

Public FIX Interface assumes the default definition of the FIX standard. Any deviations from this standard will be clearly stated in this document. Only the messages and fields which are supported by MICEX-FIX MARKET DATA are included in this document.

User Guide

System description

MICEX-FIX MARKET DATA system is a server application that publishes MICEX market data for clients according to the standard FIX protocol.

This document describes the format of FIX messages and rules of communication between MICEX-FIX MARKET DATA server and client application – the consumer of the market data.

Version support

MICEX-FIX MARKET DATA currently supports FIX versions 4.2 и 4.4.

FIX message syntax

Messages consist of a series of "Tag=Value" fields separated by a field delimiter. The delimiter is ASCII 1 (SOH) symbol. All messages begin with a standard header and are terminated with a standard trailer.

The detailed definition of FIX messages is described in "Public FIX interface" section.

MICEX-FIX MARKET DATA Functionality

1. Connecting to MICEX-FIX MARKET DATA

For each registered user (FIX-Client) the following set of connection parameters is allocated:

- SenderCompID (client identofier, string)
- TargetCompID (server identifier, TargetCompID = "MicexFixBridge")
- IP address and port to connect to MICEX-FIX MARKET DATA server
- Username
- Password
- Allowed IP addresses range for client computer
- Communication approach (Internet, VPN, other)

The SenderCompID and TargetCompID pair is the unique session identifier.

Also user is provided with the list of available market data he is allowed to access.

System works as acceptor (FIX session acceptor), whereas client works as initiator (FIX session initiator).

While connecting to MICEX-FIX MARKET DATA user passes through the following authentication procedure:

1. On the first step MICEX-FIX MARKET DATA checks user IP address and SenderCompID (49). Client should inform about chanes in IP addresses in advance.
2. The second step is the authorization on the FIX level (FIX Logon). User sets its unique name and password in FIX Logon message using tags Username(553) и Password(554), same for moth versions of FIX - 4.2 or 4.4.

2. Session management and sequence numbers

MICEX-FIX MARKET DATA server and clients use FIX messages to communication each other.

All messages in sessions are numbered. Each consequent message has number increased by 1. Numbers for incoming and outgoing messages (incoming and outgoing sequence numbers) are controlled on both server and client sides. Client can request sequence reset by sending Logon message with tag ResetSeqNumFlag (141) set to "Y". MICEX-FIX MARKET DATA starts seqeuence numbers from 1 when restores session after terminatrion or when the new session is created.

Incoming and outgoing sequence numbers are to be controlled by client independently. Sequence numbers are used to identify messages gaps. After authentication server and client synchronize their sequence numbers and then continue sending messages.

If number of received message is greater than expected then client must send Resend Request message specifying sequence gap range in tags BeginSeqNo(7) and EndSeqNo(16). MICEX-FIX MARKET DATA supports 0 (unlimited) in tag 16, which is treated as "maximum available number".

If number of received message is less than expected then sequence numbers are changed accordingly and continue from this number as if it was correct.

MICEX-FIX MARKET DATA resets sequence numbers every day on start at 7:55am.

3. Getting missing or lost messages

Messages gaps can be discovered during authorization procedure or in the middle of FIX session when sequence numbers are broken. These gaps must be corrected using message recovery procedure. Use can send Resend Request asking for resending missing message. For all missed administration messages Sequence Reset message is sent, tag NewSeqNo(36) contains sequence number, which will be used for next message. All business messages are resent. The gap fill process ends when all business message are resent. Administrative messages (Logon, Logout, Resend Request, Heartbeat, Test Request, Sequence Reset, Reject) are not resent.

4. Supported market data

FIX client can request all market data or certain parts of information: Order Book, Trade List, Market Statistics, Index Information. Market statistics включает Best Bid, Best Offer, Last Trade, Weighted Average, Settlement, Open Interest, Open Price, Close Price, High Price, Low Price, High Bid Price, Low Offer Price, Volume Today, Admitted Quote, Bid Total, Offer Total, Market, Closing, Official Close, Negotiated Deals Today, Last Bid Price, Last Offer Price, Market Price 2, Opening Price, Last Negotiated Deal, Official Open Price, Official Current Price, Duration.

5. Subscription

Marker data is sent in response to Market Data Request. FIX client connects to the source MICEX-FIX MARKET DATA, passes authorization and sends one or more messages specifying instruments (securities) he wants to receive information for. In the result MICEX-FIX MARKET DATA starts sending FIX market data messages for specified instruments. To cancel subscription client must send message specifying instrument to unsubscribe from.

On disconnect sunscription is discarded. On reconnect client must resend all subscription messages.

FIX Client and MFIX Market Data communicate using the following messages:

- From client:
  - Market Data Request (MsgType = V) – subscribe for (unsubscribe from) market data
  - Security Definition request
  - Security Status request
- From server:

- Market Data Request Reject (MsgType = Y) – response to Market Data Request, which cannot be satisfied for some reason
- Market Data - Snapshot (MsgType = W) - Market data snapshot message
- Market Data - Incremental Refresh (MsgType = X) - Market data update messages
- Security Definition
- Security Status

6. Unsupported messages and fields

If FIX client sends message, which is not supported he receives Business Message Reject (MsgType = 'j').

If FIX client sends supported messages:

- When unsupported field is sent the message is blocked by FIX server. Subscription is not created.
- When field with incorrect value is sent FIX client receives reject message with error description. Subscription is not created.
- Message with incorrect checksum or body length is ignored.
- FIX Logon message sent from not allowed IP address (for corresponding SenderCompID (49)) is ignored.
- Logon message with incorrect name and password is rejected. Server sends Logout message with error description.

Public FIX interface

This section describes messages, blocks and fields of FIX 4.2 и FIX 4.4. Message (block) is described by table, where strings correspond to fields (blocks). Для каждого field tag, field name, req'd, type присутствуют правильные значения и комментарии, где:

- Tag – number used in FIX message
- Field name – short description of tag, not used in message
- Req'd – shows if field’s presence in message is required
- Type – field type
- Valid values – additional restriction on field value
- Comments – detailed field description

FIX4.4 messages format

FIX component blocks

Each message starts from standard header and ends with standard trailer that contains checksum.

Standard Message Header

Each standard FIX message contains header. Header contains protocol version, message body length, sequence number, message type and other technical information. Comment for tag MsgType (35) contains the list of all supported message types. Unsupported message types and fields values are rejected by Business Message Reject (j) with error description.

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		Assigned value used to identify firm sending message. Always unencrypted.
56	TargetCompID	Y	String		Assigned value used to identify receiving firm. Always unencrypted.
90	SecureDataLen	N	Length		Byte length of SecureData (91) field. Always unencrypted. Required when encryption is used.
91	SecureData	N	data		Actual encrypted data stream. Required when encryption is used.
34	MsgSeqNum	Y	SeqNum		Integer message sequence number. Can be embedded within encrypted data section.
43	PossDupFlag	N	Boolean	'Y' (Possible duplicate) 'N' (Original transmission)	Indicates possible retransmission of message with this sequence number. Can be embedded within encrypted data section. Required for retransmitted messages.
52	SendingTime	Y	UTCTimestamp		Time of message transmission (expressed in UTC). Can be embedded within encrypted data section.

122	OrigSendingTime	N	UTCTimestamp		Original time of message transmission when transmitting messages as the result of resend request (expressed in UTC). Can be embedded within encrypted data section. Required for message resent as a result of a resend request.
347	MessageEncoding	N	String(11)	'UTF-8' (Unicode)	Type of message encoding (non-ASCII characters). Required if any "Encoding" fields are used.

Back to the Table of Contents

Standard Message Trailer

Standard trailer ends each message. Refer to the Volume 2 of FIX 4.4 (FIX 4.4 Specification) for more information about how to calculate checksum.

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.

Back to the Table of Contents

FIX session-level messages

Logon (MsgType (35) = 'A')

Logon (MsgType (35) = 'A') with username and password is sent to create session with MICEX-FIX MARKET DATA.

MICEX-FIX MARKET DATA resets sequence number to 1 every day at 07:40 MSK. By that time client must disconnect from system otherwise system will terminate session by itself. New session can be created by client in 20 minutes.

Client may connect and disconnect unlimited number of times during a day.

Tag	Field name	Req'd	Type	Valid values	Comments
<Standard Message Header>		Y			MsgType = 'A'
98	EncryptMethod	Y	int	'0' (None) '5' (PGP/DES-MD5)	Method of encryption. Always unencrypted.
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.
553	Username	Y*	String		Username.
554	Password	Y*	String		Password.
<Standard Message Trailer>		Y			

Logout (MsgType (35) = '5')

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			

Heartbeat (MsgType = '0')

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			

Test Request (MsgType = '1')

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			

Resend Request (MsgType = '2')

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			

Sequence Reset (MsgType = '4')

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,	Indicates that the Sequence Reset (4) message is replacing administrative or application messages which will not be

				ignore MsgSeqNum)	resent.
36	NewSeqNo	Y	SeqNum		New sequence number.
<Standard Message Trailer>		Y			

Reject (MsgType = '3')

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	Code to identify reason for reject.

				of required order) '15' (Repeating group fields out of order) '16' (Incorrect NumInGroup coun for repeating group) '17' (Non "data" value includes field delimiter) '99' (Other)	
58	Text	N	String		Message to explain reason for rejection.
<Standard Message Trailer>		Y			

Back to the Table of Contents

FIX application-level messages

Security Definition Request (MsgType = 'c')

Security Definition Request (MsgType = 'c') message is used for requesting information about instrument. Requested instrument can be defined as multi-leg instrument i.e. containing more than one instrument. Subscription for security status can be optionally defined by setting fields SubscriptionRequestType (263) in the message.

Tag	Field name	Req'd	Type	Valid values	Comments
<Standard Message Header>		Y			MsgType = 'c'
320	SecurityReqID	Y	String		Unique identifier of Security Definition Request message.
321	SecurityRequestType	Y	int	'3' (Request list of securities)	Type of Security Definition Request message.
336	TradingSessionID	Y*	String		Optional identifier of trading session to specify a particular trading session for which you want to obtain a list of securities. The format is “<Market>|<Board>” or “<Market>”. The list of supported Super Markets: • FOND • CURR • GKOM • FOPT List of boards are distributed for each market off-line. Also it is not required to mention board, it will be sent in security definition.
<Standard Message Trailer>		Y			

Security Status Request (MsgType = 'e')

Security Status Request (MsgType ='e') is used to get instrument status in snapshot or in snapshot with updates. Only one subscription per instrument is allowed.

In case of SubscriptionRequestType (263) = 0 (snapshot request), Security Status (MsgType = 'f') message with snapshot will be sent to client. If request cannot be processed it will be rejected.

In case of SubscriptionRequestType (263) = 1 (snapshot request with updates), initial Security Status (MsgType = 'f') snapshot will be sent to client followed by update messages with instrument statuses. If request cannot be processed it will be rejected.

In case of SubscriptionRequestType (263) = 2 (unsubscription requets), Security Status (MsgType ='f') with no update information will be sent. No more updates will be sent after request is processed. If request cannot be processed it will be rejected.

All other SubscriptionRequestType values will be rejected.

Tag	Field name	Req'd	Type	Valid values	Comments
<Standard Message Header>		Y			MsgType = 'e'
324	SecurityStatusReqID	Y	String		Unique identifier of Security Status Request message.
<Instrument>					
55	Symbol	Y	String		Ticker symbol. Common, "human understood" representation of the security. "[N/A]" is used for products which do not have a symbol.
48	SecurityID	N	String		Security identifier value (e.g. CUSIP, SEDOL, ISIN, etc).
22	SecurityIDSource	N	String	'4' (ISIN number)	Identifies class or source of the security identifier value. Required if SecurityID (48) is specified.
</Instrument>					
263	SubscriptionRequestType	Y	char	'0' (Snapshot) '1' (Snapshot + Updates (Subscribe)) '2' (Unsubscribe)	Type of subscription request.
336	TradingSessionID	Y*	String		Optional identifier of trading session to specify a particular trading session for which you want to obtain a list of securities.
<Standard Message Trailer>		Y			

Security Definition (MsgType = 'd')

Tag	Field name	Req'd	Type	Valid values	Comments
<Standard Message Header>		Y			MsgType = 'd'
320	SecurityReqID	Y	String		Identifier of Security Definition Request message which this message is sent in response to.
322	SecurityResponseID	Y	String		Unique identifier of Security Definition message.
323	SecurityResponseType	Y	int	'4' (List of securities returned per request) '6' (Can not match selection criteria) *'100' (Initial download from corresponding MICEX market has not been performed yet)*	Type of Security Definition message.
<Instrument>					
55	Symbol	Y	String		Ticker symbol. Common, "human understood" representation of the security. "[N/A]" is used for products which do not have a symbol.
48	SecurityID	N	String		Security identifier value (e.g. CUSIP, SEDOL, ISIN, etc).
22	SecurityIDSource	N	String	'4' (ISIN number)	Identifies class or source of the security identifier value. Required if SecurityID (48) is specified.
460	Product	N	int	'3' (CORPORATE) '4' (CURRENCY) '5' (EQUITY) '6' (GOVERNMENT) '7' (INDEX)	Indicates the type of product the security is associated with.

				'11' (MUNICIPAL) '13' (FINANCING)	
461	CFICode	N	String		Indicates the type of security using ISO 10962 standard, Classification of Financial Instruments (CFI code) values.
167	SecurityType	N	String	'CORP' (Corporate bond) 'FOR' (Foreign exchange contract) 'CS' (Common stock) 'PS' (Preferred stock) 'EUSOV' (Euro sovereigns) 'REPO' (Repurchase) 'BN' (Bank notes) 'MF' (Mutual Fund) *'MUNI' (Municipal bond)*	Indicates type of security.
200	MaturityMonthYear	N	month-year		Specifies the month and year of maturity.
541	MaturityDate	N	LocalMktDate		Specifies date of maturity (a full date).
224	CouponPaymentDate	N	LocalMktDate		Date interest is to be paid.
228	Factor	N	float		For Fixed Income: amorization factor for deriving Current face from Original face for ABS or MBS securities. In TIPS securities this is the Inflation index. Qty * Factor * Price = Gross Trade Amount. For Derivatives: Contract Value Factor by which price must be adjusted to determine the true nominal value of one

					futures/options contract. (Qty * Price) * Factor = Nominal Value.
202	StrikePrice	N	Price		Strike price for an option.
223	CouponRate	N	Percentage		The rate of interest that, when multiplied by the principal, par value, or face value of a bond, provides the currency amount of the periodic interest payment.
107	SecurityDesc	N	String		Security description.
350	EncodedSecurityDescLen	N	Length		Byte length of encoded (non-ASCII characters) EncodedSecurityDesc (351) field.
351	EncodedSecurityDesc	N	data		Encoded (non-ASCII characters) representation of the SecurityDesc (107) field in the encoded format specified via the MessageEncoding (347) field.
864	NoEvents	N	NumInGroup		Number of repeating events following.
=> 865	EventType	N	int	'7' (Last eligible trade date) '100' (First eligible trade date)	Code to represent the type of event. Required if NoEvents (864) > 0.
=> 866	EventDate	N	LocalMktDate		Date of event.
5217	*StateSecurityID*	*N*	*String*		*State security identification number.*
5382	*EncodedShortSecurityDescLen*	*N*	*Length*		*Byte length of encoded (non-ASCII characters) EncodedShortSecurityDesc (5383) field.*
5383	*EncodedShortSecurityDesc*	*N*	*data*		*Encoded (non-ASCII characters) representation of the short security description in the encoded format specified via the MessageEncoding (347) field.*
5556	*BaseSwapPx*	*N*	*Price*		*Base SWAP price.*

5558	*BuyBackPx*	*N*	*Price*		*Buy back price.*
5559	*BuyBackDate*	*N*	*LocalMktDate*		*Buy back date.*
5677	*Repo2Px*	*N*	*Price*		*Price of the second part of REPO.*
</Instrument>					
<InstrumentExtension>					
870	NoInstrAttrib	N	NumInGroup		Number of repeating InstrAttrib group entries.
=> 871	InstrAttribType	N	int	'8' (Coupon period) '27' (Instrument price precision)	Type of instrument attribute. Required if NoInstrAttrib (870) > 0.
=> 872	InstrAttribValue	N	String		Value of instrument attribute (if applicable).
</InstrumentExtension>					
<UnderlyingInstrument>					
711	NoUnderlyings	N	NumInGroup		Number of UnderlyingInstrument repeating group instances.
=> 311	UnderlyingSymbol	N	String		Underlying security's Symbol. Required if NoUnderlyings (711) > 0.
</UnderlyingInstrument>					
15	Currency	N	Currency		Instrument currency.
336	TradingSessionID	N	String		Identifier of trading session. The format is “<Market>|<Board>”.
58	Text	N	String		Comment, instructions, or other identifying information.
561	RoundLot	N	Qty		The trading lot size of a security.
120	*SettlCurrency*	*N*	*Currency*		*Currency code of settlement denomination.*

423	*PriceType*	*N*	*int*	*'1' (Percentage (e.g. percent of par))* *'2' (Per unit (i.e. per share or contract))*	*Code to represent the price type.*
5385	*MarketCode*	*N*	*String*		*Code of market where instrument is traded.*
5386	*MinPriceIncrement*	*N*	*Price*		*Minimum price increase for security.*
5387	*MktShareLimit*	*N*	*Percentage*		*Market share limit.*
5388	*MktShareThreshold*	*N*	*Qty*		*Market share limit threshold.*
5389	*MaxOrdersVolume*	*N*	*Qty*		*Maximum summary volume of active buy and sell orders.*
5470	*PriceMvmLimit*	*N*	*Price*		*Maximum deviation of prices from settlement price.*
5472	*PriceMvmLimitT1*	*N*	*Price*		*Maximum deviation of prices from settlement price at T+1.*
5508	*FaceValue*	*N*	*Amt*		*Face value of security.*
7595	*NoSharesIssued*	*N*	*Qty*		*Number of shares issued.*
9199	*HighLimit*	*N*	*Price*		*Allowable high limit price for the trading day.*
9200	*LowLimit*	*N*	*Price*		*Allowable low limit price for the trading day.*
<Standard Message Trailer>		Y			

Security Status (MsgType = 'f')

Message of type Security Status (MsgType ='f') contains information about instrument in trading system.

Tag	Field name	Req'd	Type	Valid values	Comments

[<Standard Message Header>]		Y			MsgType = 'f'
324	SecurityStatusReqID	N	String		Identifier of Security Status Request message which this message is sent in response to.
<Instrument>					
55	Symbol	Y	String		Ticker symbol. Common, "human understood" representation of the security. "[N/A]" is used for products which do not have a symbol.
</Instrument>					
336	TradingSessionID	N	String		Optional identifier of trading session. The format is “<Market>|<Board>” or “<Market>”.
326	SecurityTradingStatus	N	int	'2' (Trading halt) '17' (Ready to trade (start of session)) '18' (Not available for trading (end of session)) '20' (Unknown or invalid) '100' (Initial download from corresponding MICEX market has not been performed yet)	Identifies the trading status applicable to the transaction.
58	Text	N	String		Comment, instructions, or other identifying information.
5509	*AuctionIndicator*	*N*	*Boolean*	*'Y' (Yes)* *'N' (No)*	*Indicates whether or not the auction is being held for the security.*
[<Standard Message Trailer>]		Y			

Trading Session Status (MsgType = 'h')

Message of type Trading Session Status (MsgType = ‘h’) gives information about market status and is sent in the following cases:

- Connection with MICEX is established
- Connection with MICEX is not established or non-gracefully terminated, MICEX-FIX MARKET DATA tries to reconnect
- At the end-of-day MICEX-FIX MARKET DATA terminates connection with MICEX
- On reconnect MICEX-FIX MARKET DATA determines that session's TradingSessionId was changed
- On disconnect MICEX-FIX MARKET DATA cannot restore connection

Tag	Field name	Req'd	Type	Valid values	Comments
<Standard Message Header>		Y			MsgType = 'h'
336	TradingSessionID	Y	String		Optional identifier of trading session.
325	UnsolicitedIndicator	N	Boolean	'Y' (Yes) 'N' (No)	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 MICEX market established)* *'102' (Connection to MICEX market terminated correctly)* *'103' (Connection to MICEX market terminated non-gracefully)* *'104' (Reconnecting to MICEX market)*	State of the trading session.
58	Text	N	String		Free format text string.
<Standard Message Trailer>		Y			

Market Data Request(MsgType = 'V')

Message of type Market Data Request is used for requesting market data for certain instrument. Field SubscriptionRequestType (263) contains type of requested information:

- In case of SubscriptionRequestType (263) = 0 (snapshot request), Market Data Snapshot/Full Refresh (MsgType ='W') with snapshot will be sent. If request cannot be processed Market Data - Request Reject (MsgType ='Y') will be sent.

- In case of SubscriptionRequestType (263) = 1 (snapshot with updates request), initial Market Data Snapshot / Full Refresh (MsgType ='W') will be sent with snapshot followed by updates of type Market Data Incremental Refresh (MsgType ='X') any type new information will appear. If request cannot be processed Market Data - Request Reject (MsgType ='Y') will be sent.
- In case of SubscriptionRequestType (263) = 2 (subscription reject), Market Data – Incremental Refresh (MsgType ='X') will be sent without updates. No updates will be sent later on. If request cannot be processed Market Data - Request Reject (MsgType ='Y') will be sent.

For all other values of SubscriptionRequestType Market Data - Request Reject (MsgType ='Y') will be sent.

Tag	Field name	Req'd	Type	Valid values	Comments
<Standard Message Header>		Y			MsgType = 'V'
262	MDReqID	Y	String		Unique identifier of Market Data Request message when new subscription/request is sent, or identifier of previous Market Data Request message in case of unsubscribe.
263	SubscriptionRequestType	Y	char	'0' (Snapshot) '1' (Snapshot + Updates (Subscribe)) '2' (Unsubscribe)	Type of subscription request.
264	MarketDepth	Y	int	'0' (Full Book) '1' (Top of Book)	Depth of market.
265	MDUpdateType	N	int	'1' (Incremental Refresh)	Specifies the type of market data update. Required if SubscriptionRequestType (263) = '1'.
267	NoMDEntryTypes	Y	NumInGroup		Number of MDEntryType fields requested.
=> 269	MDEntryType	Y	char	'0' (Bid) '1' (Offer) '2' (Trade) '3' (Index value) '4' (Opening price) '5' (Closing price) '6' (Settlement price) '7' (Trading session high price) '8' (Trading session low price)	Type of market data entry.

				'9' (Trading session VWAP price) *'B' (Trade volume)* *'C' (Open interest)* *'N' (Session high bid)* *'O' (Session low offer)* *'h' (Open period price)* *'i' (Last bid price)* *'j' (Last offer price)* *'k' (Close period price)* *'l' (Market price 2)* *'m' (Market price)* *'n' (Last negotiated deal)* *'o' (Official open price)* *'p' (Official current price)* *'q' (Admitted quote)* *'r' (Official close price)* *'v' (Total bid volume)* *'w' (Total offer volume)* *'t' (Negotiated deals value)* *'u' (Duration)* *'y' (Market statistics)* *'z' (Trade list)* *'*' (All market data)*	
146	NoRelatedSym	Y	NumInGroup		Number of instruments requested.
<Instrument>					
=> 55	Symbol	Y	String		Ticker symbol. Common, "human understood" representation of the security. "[N/A]" is used for products which do not have a symbol. Use * to request all

					instruments for a market.к
=> 48	SecurityID	N	String		Security identifier value (e.g. CUSIP, SEDOL, ISIN, etc).
=> 22	SecurityIDSource	N	String	'4' (ISIN number)	Identifies class or source of the security identifier value. Required if SecurityID (48) is specified.
</Instrument>					
386	NoTradingSessions	Y*	NumInGroup		Number of trading sessions for which the request is valid.
=> 336	TradingSessionID	Y*	String		Identifier of trading session received in security definition. Required if NoTradingSessions (386) > 0.
<Standard Message Trailer>		Y			

Market Data Request Reject (MsgType = 'Y')

Message of type Market Data Request Reject (MsgType = 'Y') is sent when Market Data Request cannot be satisfied.

Tag	Field name	Req'd	Type	Valid values	Comments
<Standard Message Header>		Y			MsgType = 'Y'
262	MDReqID	Y	String		Identifier of Market Data Request message which is rejected.
281	MDReqRejReason	N	char	'0' (Unknown symbol) '1' (Duplicate MDReqID) '3' (Insufficient permissions) '4' (Unsupported SubscriptionRequestType) '8' (Unsupported MDEntryType) '9' (Unsupported TradingSessionID) *'x' (Instrument is not*	Reason for the rejection of Market Data Request message.

				traded at board) *'y' (Unknown MDReqID)* *'z' (Initial download from corresponding MICEX market has not been performed yet)*	
58	Text	N	String		Message to explain reason for rejection.
[<Standard Message Trailer>]()		Y			

Market Data - Snapshot/Full Refresh (MsgType = 'W')

Message of type The Market Data – Snapshot/Full Refresh (MsgType = 'W') is sent in response to Market Data Request (MsgType = 'V') with SubscriptionRequestType = '0' (snapshot reuquest).

Each Market Data – Snapshot/Full Refresh (MsgType = 'W') is linked to some Market Data Request (MsgType = 'V'). Value of tag MDReqID(262) in snapshot message is equal to the value of this tag in Market Data Request (MsgType = 'V').

Tag	Field name	Req'd	Type	Valid values	Comments
[<Standard Message Header>]()		Y			MsgType = 'W'
262	MDReqID	N	String		Identifier of Market Data Request message which this message is sent in response to.
<Instrument>					
55	Symbol	Y	String		Ticker symbol. Common, "human understood" representation of the security. "[N/A]" is used for products which do not have a symbol.
</Instrument>					
451	NetChgPrevDay	N	PriceOffset		Net change from previous day's closing price vs. last traded price.
268	NoMDEntries	Y	NumInGroup		Number of market data entries following.

=> 269	MDEntryType	Y	char	'0' (Bid) '1' (Offer) '2' (Trade) '3' (Index value) '4' (Opening price) '5' (Closing price) '6' (Settlement price) '7' (Trading session high price) '8' (Trading session low price) '9' (Trading session VWAP price) *'B' (Trade volume)* *'C' (Open interest)* *'J' (Empty book)* *'N' (Session high bid)* *'O' (Session low offer)* *'h' (Open period price)* *'i' (Last bid price)* *'j' (Last offer price)* *'k' (Close period price)* *'l' (Market price 2)* *'m' (Market price)* *'n' (Last negotiated deal)* *'o' (Official open price)* *'p' (Official current price)* *'q' (Admitted quote)* *'r' (Official close price)* *'v' (Total bid volume)*	Type of market data entry.

				'w' (Total offer volume) *'t' (Negotiated deals value)* *'u' (Duration)* *'z' (Trade list)*	
=> 278	MDEntryID	N	String		Unique market data entry identifier.
=> 270	MDEntryPx	N	Price		Price of the market data entry. Required if MDEntryType (269) != 'A', 'B', 'C', 'J'.
=> 271	MDEntrySize	N	Qty		Quantity or volume represented by the market data entry. Required if MDEntryType (269) = '0', '1', '2', 'B', 'C'.
=> 272	MDEntryDate	N	UTCDateOnly		Date of market data entry.
=> 273	MDEntryTime	N	UTCTimeOnly		Time of market data entry.
=> 336	TradingSessionID	N	String		Identifier of trading session. The format is “<Market>|<Board>” or “<Market>|<Board>|<State>”, where state can be: • OPEN • CLOS
=> 276	QuoteCondition	N	MultipleValueString	'C' (Exchange best)	Space-delimited list of conditions describing a quote.
=> 277	TradeCondition	N	MultipleValueString	'J' (Next day trade) 'R' (Opening price) *'AJ' (Official closing price)*	Space-delimited list of conditions describing a trade.
=> 286	OpenCloseSettlFlag	N	MultipleValueString	'4' (Entry from previous business day)	Flag that identifies a market data entry.

=> 5384	*AccruedInterestAmt*	*N*	*Amt*		*Amount of accrued interest.*
=> 5510	*ChgFromWAPrice*	*N*	*PriceOffset*		*Indicates change from previous day's weighted average price vs. last traded price.*
=> 5511	*ChgOpenInterest*	*N*	*Qty*		*Indicates change from previous day's open interest.*
=> 6139	*TotalNumOfTrades*	*N*	*int*		*Total number of trades.*
=> 6143	*TradeValue*	*N*	*Amt*		*Trade value.*
=> 6608	*Yield*	*N*	*Percentage*		*Yield calculated at MDEntryPx (270).*
=> 9168	*OfferNbOr*	*N*	*int*		*Total number of sell orders.*
=> 9169	*BidNbOr*	*N*	*int*		*Total number of buy orders.*
=> 9750	*ChgFromSettlmnt*	*N*	*PriceOffset*		*Indicates the change from previous day's settlement price vs. last traded price.*
<Standard Message Trailer>		Y			

Market Data - Incremental Refresh (MsgType = 'X')

Message of type Market Data – Incremental Refresh (MsgType ='X') is sent by MICEX-FIX MARKET DATA when market data for subscribed instrument (Market Data Request (MsgType ='V') was previously sent with SubscriptionRequestType = '1' (snapshot with updates request)) is changed. This message is also sent in response to unsubscribe request SubscriptionRequestType = '2'.

Tag	Field name	Req'd	Type	Valid values	Comments
<Standard Message Header>		Y			MsgType = 'X'
262	MDReqID	N	String		Identifier of Market Data Request message

					which this message is sent in response to.
268	NoMDEntries	Y	NumInGroup		Number of market data entries following.
=> 279	MDUpdateAction	Y	char	'0' (New) '1' (Change) '2' (Delete)	Type of market data update action.
=> 269	MDEntryType	N	char	'0' (Bid) '1' (Offer) '2' (Trade) '3' (Index value) '4' (Opening price) '5' (Closing price) '6' (Settlement price) '7' (Trading session high price) '8' (Trading session low price) '9' (Trading session VWAP price) *'B' (Trade volume)* *'C' (Open interest)* *'N' (Session high bid)* *'O' (Session low offer)* *'h' (Open period price)* *'i' (Last bid price)* *'j' (Last offer price)* *'k' (Close period price)* *'l' (Market price 2)* *'m' (Market price)* *'n' (Last negotiated deal)*	Type of market data entry. Required if MDUpdateAction (279) = '0'.

				'o' (Official open price) *'p' (Official current price)* *'q' (Admitted quote)* *'r' (Official close price)* *'v' (Total bid volume)* *'w' (Total offer volume)* *'t' (Negotiated deals value)* *'u' (Duration)* *'z' (Trade list)*	
=> 278	MDEntryID	N	String		Unique market data entry identifier.
<Instrument>					
=> 55	Symbol	Y	String		Ticker symbol. Common, "human understood" representation of the security. "[N/A]" is used for products which do not have a symbol.
</Instrument>					
=> 270	MDEntryPx	N	Price		Price of the market data entry. Required if MDUpdateAction (279) = '0' and MDEntryType (269) != 'A', 'B', 'C', 'J'.
=> 271	MDEntrySize	N	Qty		Quantity or volume represented by the market data entry. Required if MDUpdateAction (279) = '0' and MDEntryType (269) = '0', '1', '2', 'B', 'C'.
=> 272	MDEntryDate	N	UTCDateOnly		Date of market data entry.
=> 273	MDEntryTime	N	UTCTimeOnly		Time of market data entry.
=> 336	TradingSessionID	N	String		Identifier of trading session.

=> 276	QuoteCondition	N	MultipleValueString	'C' (Exchange best)	Space-delimited list of conditions describing a quote.
=> 277	TradeCondition	N	MultipleValueString	'J' (Next day trade) 'R' (Opening price) *'AJ' (Official closing price)*	Space-delimited list of conditions describing a trade.
=> 286	OpenCloseSettlFlag	N	MultipleValueString	'4' (Entry from previous business day)	Flag that identifies a market data entry.
=> 451	NetChgPrevDay	N	PriceOffset		Net change from previous day's closing price vs. last traded price.
=> *5384*	*AccruedInterestAmt*	*N*	*Amt*		*Amount of accrued interest.*
=> *5510*	*ChgFromWAPrice*	*N*	*PriceOffset*		*Indicates change from previous day's weighted average price vs. last traded price.*
=> *5511*	*ChgOpenInterest*	*N*	*Qty*		*Indicates change from previous day's open interest.*
=> *6139*	*TotalNumOfTrades*	*N*	*int*		*Total number of trades.*
=> *6143*	*TradeValue*	*N*	*Amt*		*Trade value.*
=> *6608*	*Yield*	*N*	*Percentage*		*Yield calculated at MDEntryPx (270).*
=> *9168*	*OfferNbOr*	*N*	*int*		*Total number of sell orders.*
=> *9169*	*BidNbOr*	*N*	*int*		*Total number of buy orders.*
=> *9750*	*ChgFromSettlmnt*	*N*	*PriceOffset*		*Indicates the change from previous day's settlement price vs. last traded price.*
<Standard Message Trailer>		Y			

Email (MsgType = 'C')

FIX Email message (MsgType = ‘C’) is used to notify clients about errors, which cannot be described using standard FIX reject messages. This can be error during message conversion, failure during state restore, etc.

Tag	Field name	Req'd	Type	Valid values	Comments
<Standard Message Header>		Y			MsgType = 'C'
164	EmailThreadID	Y	String		Unique identifier for the email message thread.
94	EmailType	Y	char	'0' (New)	Email message type.
42	OrigTime	N	UTCTimestamp		Time of message origination.
147	Subject	Y	String		Subject of the email message.
33	LinesOfText	Y	NumInGroup		Specifies the number of repeating lines of text specified.
=> 58	Text	Y	String		Free format text string.
95	RawDataLength	N	Length		Byte length of RawData (96) field.
96	RawData	N	data		Unformatted raw data, can include bitmaps, word processor documents, etc.
<Standard Message Trailer>		Y			

Business Message Reject (MsgType ='j')

Message of type Business Message Reject is used to reject business level messages, wich cannot be rejectd using specialized reject messages. Please note that if message has error on session level (for example reuquired tag is missing) than administration reject message is sent Reject (MsgType = ‘3’).

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

45	RefSeqNum	N	SeqNum		MsgSeqNum (34) of rejected message.
372	RefMsgType	Y	String(10)		The MsgType (35) of the FIX message being referenced.
380	BusinessRejectReason	Y	int	'3' (Unsupported message type)	Code to identify reason for a Business Message Reject message.
58	Text	N	String		Free format text string.
<Standard Message Trailer>		Y			

FIX4.2 Message formats

FIX component blocks

Each message starts from standard header and ends with standard trailer that contains checksum.

Standard Message Header

Each standard FIX message contains header. Header contains protocol version, message body length, sequence number, message type and other technical information. Comment for tag MsgType (35) contains the list of all supported message types. Unsupported message types and fields values are rejected by Business Message Reject (j) with error description.

Tag	Field name	Req'd	Type	Valid values	Comments
8	BeginString	Y	String(7)	'FIX.4.2'	Identifies beginning of new message and protocol version. Always unencrypted, must be first field in message.
9	BodyLength	Y	int		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		Assigned value used to identify firm sending message. Always unencrypted.

56	TargetCompID	Y	String		Assigned value used to identify receiving firm. Always unencrypted.
90	SecureDataLen	N	int		Byte length of SecureData (91) field. Always unencrypted. Required when encryption is used.
91	SecureData	N	data		Actual encrypted data stream. Required when encryption is used.
34	MsgSeqNum	Y	int		Integer message sequence number. Can be embedded within encrypted data section.
43	PossDupFlag	N	Boolean	'Y' (Possible duplicate) 'N' (Original transmission)	Indicates possible retransmission of message with this sequence number. Can be embedded within encrypted data section. Required for retransmitted messages.
52	SendingTime	Y	UTCTimestamp		Time of message transmission (expressed in UTC). Can be embedded within encrypted data section.
122	OrigSendingTime	N	UTCTimestamp		Original time of message transmission when transmitting messages as the result of resend request(expressed in UTC). Can be embedded within encrypted data section. Required for message resent as a result of a resend request.
347	MessageEncoding	N	String(11)	'UTF-8' (Unicode)	Type of message encoding (non-ASCII characters). Required if any "Encoding" fields are used.

Standard Message Trailer

Standard trailer ends each message. Refer to the Volume 2 of FIX 4.4 (FIX 4.4 Specification) for more information about how to calculate checksum.

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.

FIX session-level messages

Logon (MsgType = 'A')

Logon (MsgType (35) = 'A') with username and password is sent to create session with MICEX-FIX MARKET DATA.

MICEX-FIX MARKET DATA resets sequence number to 1 every day at 07:40 MSK. By that time client must disconnect from system otherwise system will terminate session by itself. New session can be created by client in 20 minutes.

Client may connect and disconnect unlimited number of times during a day.

Tag	Field name	Req'd	Type	Valid values	Comments
<Standard Message Header>		Y			MsgType = 'A'
98	EncryptMethod	Y	int	'0' (None) '5' (PGP/DES-MD5)	Method of encryption. Always unencrypted.
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.
553	Username	Y	String		Username.
554	Password	Y	String		Password.
<Standard Message Trailer>		Y			

Logout (MsgType = '5')

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			

Heartbeat (MsgType = '0')

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			

Test Request (MsgType = '1')

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			

Resend Request (MsgType = '2')

Tag	Field name	Req'd	Type	Valid values	Comments
<Standard Message Header>		Y			MsgType = '2'
7	BeginSeqNo	Y	int		Message sequence number of first message in range to be resent.
16	EndSeqNo	Y	int		Message sequence number of last message in range to be resent. If request is for a single message BeginSeqNo (7) = EndSeqNo (16). If request is for all messages subsequent to a particular message, EndSeqNo (16) = '0' (representing infinity).
<Standard Message Trailer>		Y			

Sequence Reset (MsgType = '4')

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	int		New sequence number.
<Standard Message Trailer>		Y			

Reject (MsgType = '3')

Tag	Field name	Req'd	Type	Valid values	Comments
<Standard Message Header>		Y			MsgType = '3'
45	RefSeqNum	Y	int		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)	Code to identify reason for reject.

				'7' (Decryption problem) '8' (Signature problem) '9' (CompID problem) '10' (SendingTime accuracy problem) '11' (Invalid MsgType)	
58	Text	N	String		Message to explain reason for rejection.
<Standard Message Trailer>		Y			

FIX application-level messages

Security Definition Request (MsgType = 'c')

Security Definition Request (MsgType = 'c') message is used for requesting information about instrument. Requested instrument can be defined as multi-leg instrument i.e. containing more than one instrument. Subscription for security status can be optionally defined by setting fields SubscriptionRequestType (263) in the message.

Tag	Field name	Req'd	Type	Valid values	Comments
<Standard Message Header>		Y			MsgType = 'c'
320	SecurityReqID	Y	String		Unique identifier of Security Definition Request message.
321	SecurityRequestType	Y	int	'3' (Request list of securities)	Type of Security Definition Request message.
336	TradingSessionID	Y*	String		Optional identifier of trading session to specify a particular trading session for which you want to obtain a list of securities.
<Standard Message Trailer>		Y			

Security Status Request (MsgType = 'e')

Security Status Request (MsgType ='e') is used to get instrument status in snapshot or in snapshot with updates. Only one subscription per instrument is allowed.

In case of SubscriptionRequestType (263) = 0 (snapshot request), Security Status (MsgType = 'f') message with snapshot will be sent to client. If request cannot be processed it will be rejected.

In case of SubscriptionRequestType (263) = 1 (snapshot request with updates), initial Security Status (MsgType = 'f') snapshot will be sent to client followed by update messages with instrument statuses. If request cannot be processed it will be rejected.

In case of SubscriptionRequestType (263) = 2 (unsubscription requets), Security Status (MsgType ='f') with no update information will be sent. No more updates will be sent after request is processed. If request cannot be processed it will be rejected.

All other SubscriptionRequestType values will be rejected.

Tag	Field name	Req'd	Type	Valid values	Comments
<Standard Message Header>		Y			MsgType = 'e'
324	SecurityStatusReqID	Y	String		Unique identifier of Security Status Request message.
55	Symbol	Y	String		Ticker symbol of the security for which status is reported. Common, "human understood" representation of the security. "[N/A]" is used for products which do not have a symbol.
48	SecurityID	N	String		Security identifier value (e.g. CUSIP, SEDOL, ISIN, etc).
22	IDSource	N	String	'4' (ISIN number)	Identifies class or source of the security identifier value. Required if SecurityID (48) is specified.
263	SubscriptionRequestType	Y	char	'0' (Snapshot) '1' (Snapshot + Updates (Subscribe)) '2' (Unsubscribe)	Type of subscription request.
336	TradingSessionID	Y*	String		Optional identifier of trading session.
<Standard Message Trailer>		Y			

Security Definition (MsgType = 'd')

Tag	Field name	Req'd	Type	Valid values	Comments

<Standard Message Header>		Y			MsgType = 'd'
320	SecurityReqID	Y	String		Identifier of Security Definition Request message which this message is sent in response to.
322	SecurityResponseID	Y	String		Unique identifier of Security Definition message.
323	SecurityResponseType	Y	int	'4' (List of securities returned per request) '6' (Can not match selection criteria) *'100' (Initial download from corresponding MICEX market has not been performed yet)*	Type of Security Definition message.
393	TotalNumSecurities	Y	int		Total number of securities.
55	Symbol	N	String		Ticker symbol of the requested security. Common, "human understood" representation of the security. "[N/A]" is used for products which do not have a symbol.
48	SecurityID	N	String		Security identifier value (e.g. CUSIP, SEDOL, ISIN, etc).
22	IDSource	N	String	'4' (ISIN number)	Identifies class or source of the security identifier value. Required if SecurityID (48) is specified.
167	SecurityType	N	String	'CORP' (Corporate bond) 'CS' (Common stock) 'FOR' (Foreign exchange contract) 'FUT' (Future)	Indicates type of security.

				'MF' (Mutual Fund) 'MUNI' (Municipal bond) 'OPT' (Option) 'PS' (Preferred stock) 'RP' (Repurchase agreement) *'EUSOV' (Euro sovereigns)* *'BN' (Bank notes)* *'INDEX' (Index)*	
200	MaturityMonthYear	N	month-year		Specifies the month and year of maturity.
205	MaturityDay	N	day-of-month		Day of month, can be used in conjunction with MaturityMonthYear (200) to specify the maturity date.
201	PutOrCall	N	int	'0' (Put) '1' (Call)	Indicates whether an option is for a put or call.
202	StrikePrice	N	Price		Strike price for an option.
206	OptAttribute	N	char	'L' (Long (a.k.a. 'American'))	Used for derivatives, such as options and covered warrants to indicate a versioning of the contract when required due to corporate actions to the underlying.
223	CouponRate	N	float		The rate of interest that, when multiplied by the principal, par value, or face value of a bond, provides the currency amount of the periodic interest payment.
107	SecurityDesc	N	String		Security description.
350	EncodedSecurityDescLen	N	int		Byte length of encoded (non-ASCII characters) EncodedSecurityDesc (351) field.
351	EncodedSecurityDesc	N	data		Encoded (non-ASCII characters) representation of the SecurityDesc (107) field in the encoded format specified via

					the MessageEncoding (347) field.
146	NoRelatedSym	N	int		Number of underlying instruments following.
=> 311	UnderlyingSymbol	N	String		Underlying security's Symbol. See Symbol (55) field for description. Required if NoRelatedSym (146) > 0.
15	Currency	N	Currency		Instrument currency.
336	TradingSessionID	N	String		Identifier of trading session.
58	Text	N	String		Comment, instructions, or other identifying information.
120	*SettlCurrency*	*N*	*Currency*		*Currency code of settlement denomination.*
224	*CouponPaymentDate*	*N*	*LocalMktDate*		*Date interest is to be paid.*
228	*Factor*	*N*	*float*		*For Fixed Income: amorization factor for deriving Current face from Original face for ABS or MBS securities.* *In TIPS securities this is the Inflation index.* *Qty * Factor * Price = Gross Trade Amount.* *For Derivatives: Contract Value Factor by which price must be adjusted to determine the true nominal value of one futures/options contract. (Qty * Price) * Factor = Nominal Value.*
423	*PriceType*	*N*	*int*	*'1' (Percentage (e.g. percent of par))* *'2' (Per unit (i.e. per share or contract))*	*Code to represent the price type.*
561	*RoundLot*	*N*	*Qty*		*The trading lot size of a security.*
5217	*StateSecurityID*	*N*	*String*		*State security identification number.*
5382	*EncodedShortSecurityDescLen*	*N*	*int*		*Byte length of encoded (non-ASCII characters)*

					EncodedShortSecurityDesc (5383) field.
5383	*EncodedShortSecurityDesc*	*N*	*data*		*Encoded (non-ASCII characters) representation of the short security description in the encoded format specified via the MessageEncoding (347) field.*
5385	*MarketCode*	*N*	*String*		*Code of market where instrument is traded.*
5386	*MinPriceIncrement*	*N*	*Price*		*Minimum price increase for security.*
5387	*MktShareLimit*	*N*	*float*		*Market share limit.*
5388	*MktShareThreshold*	*N*	*Qty*		*Market share limit threshold.*
5389	*MaxOrdersVolume*	*N*	*Qty*		*Maximum summary volume of active buy and sell orders.*
5470	*PriceMvmLimit*	*N*	*Price*		*Maximum deviation of prices from settlement price.*
5472	*PriceMvmLimitT1*	*N*	*Price*		*Maximum deviation of prices from settlement price at T+1.*
5508	*FaceValue*	*N*	*Amt*		*Face value of security.*
5512	*FirstEligibleTradeDate*	*N*	*LocalMktDate*		*First eligible trade date.*
5513	*LastEligibleTradeDate*	*N*	*LocalMktDate*		*Last eligible trade date.*
5514	*InstrumentPricePrecision*	*N*	*int*		*Number of decimals in prices.*
5556	*BaseSwapPx*	*N*	*Price*		*Base SWAP price.*
5558	*BuyBackPx*	*N*	*Price*		*Buy back price.*
5559	*BuyBackDate*	*N*	*LocalMktDate*		*Buy back date.*
5677	*Repo2Px*	*N*	*Price*		*Price of the second part of REPO.*
7595	*NoSharesIssued*	*N*	*Qty*		*Number of shares issued.*
7910	*DayCountFraction*	*N*	*String*		*Describes the method used for calculating accrued interest for a bond.*

9127	*DeliveryType*	*N*	*char*	*'1' (Cash settled)* *'2' (Physical delivery)*	*Specifies type of delivery for futures and options.*
9199	*HighLimit*	*N*	*Price*		*Allowable high limit price for the trading day.*
9200	*LowLimit*	*N*	*Price*		*Allowable low limit price for the trading day.*
<Standard Message Trailer>		Y			

Security Status (MsgType = 'f')

Message of type Security Status (MsgType ='f') contains information about instrument in trading system.

Tag	Field name	Req'd	Type	Valid values	Comments
<Standard Message Header>		Y			MsgType = 'f'
324	SecurityStatusReqID	N	String		Identifier of Security Status Request message which this message is sent in response to.
55	Symbol	Y	String		Ticker symbol of the security for which status is reported. Common, "human understood" representation of the security. "[N/A]" is used for products which do not have a symbol.
336	TradingSessionID	N	String		Optional identifier of trading session.
326	SecurityTradingStatus	N	int	'2' (Trading halt) '17' (Ready to trade (start of session)) '18' (Not available for trading (end of session)) '20' (Unknown or invalid) '100' (Initial download from corresponding MICEX market has not been	Identifies the trading status applicable to the transaction.

				performed yet)	
58	Text	N	String		Comment, instructions, or other identifying information.
5509	*AuctionIndicator*	*N*	*Boolean*	*'Y' (Yes)* *'N' (No)*	*Indicates whether or not the auction is being held for the security.*
<Standard Message Trailer>		Y			

Trading Session Status (MsgType = 'h')

Message of type Trading Session Status (MsgType = ‘h’) gives information about market status and is sent in the following cases:

- Connection with MICEX is established
- Connection with MICEX is not established or non-gracefully terminated, MICEX-FIX MARKET DATA tries to reconnect
- At the end-of-day MICEX-FIX MARKET DATA terminates connection with MICEX
- On reconnect MICEX-FIX MARKET DATA determines that session’s TradingSessionId was changed
- On disconnect MICEX-FIX MARKET DATA cannot restore connection

Tag	Field name	Req'd	Type	Valid values	Comments
<Standard Message Header>		Y			MsgType = 'h'
336	TradingSessionID	Y	String		Optional identifier of trading session.
325	UnsolicitedIndicator	N	Boolean	'Y' (Yes) 'N' (No)	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 MICEX market established)* *'102' (Connection to MICEX market terminated correctly)* *'103' (Connection to MICEX market terminated non-gracefully)*	State of the trading session.

				'104' (Reconnecting to MICEX market)	
58	Text	N	String		Free format text string.
<Standard Message Trailer>		Y			

Market Data Request (MsgType = 'V')

Message of type Market Data Request is used for requesting market data for certain instrument. Field SubscriptionRequestType (263) contains type of requested information:

- In case of SubscriptionRequestType (263) = 0 (snapshot request), Market Data Snapshot/Full Refresh (MsgType ='W') with snapshot will be sent. If request cannot be processed Market Data - Request Reject (MsgType ='Y') will be sent.
- In case of SubscriptionRequestType (263) = 1 (snapshot with updates request), initial Market Data Snapshot / Full Refresh (MsgType ='W') will be sent with snapshot followed by updates of type Market Data Incremental Refresh (MsgType ='X') any type new information will appear. If request cannot be processed Market Data - Request Reject (MsgType ='Y') will be sent.
- In case of SubscriptionRequestType (263) = 2 (subscription reject), Market Data – Incremental Refresh (MsgType ='X') will be sent without updates. No updates will be sent later on. If request cannot be processed Market Data - Request Reject (MsgType ='Y') will be sent.

For all other values of SubscriptionRequestType Market Data - Request Reject (MsgType ='Y') will be sent.

Tag	Field name	Req'd	Type	Valid values	Comments
<Standard Message Header>		Y			MsgType = 'V'
262	MDReqID	Y	String		Unique identifier of Market Data Request message when new subscription/request is sent, or identifier of previous Market Data Request message in case of unsubscribe.
263	SubscriptionRequestType	Y	char	'0' (Snapshot) '1' (Snapshot + Updates (Subscribe)) '2' (Unsubscribe)	Type of subscription request.
264	MarketDepth	Y	int	'0' (Full Book) '1' (Top of Book)	Depth of market.

265	MDUpdateType	N	int	'1' (Incremental Refresh)	Specifies the type of market data update. Required if SubscriptionRequestType (263) = '1'.
267	NoMDEntryTypes	Y	int		Number of MDEntryType fields requested.
=> 269	MDEntryType	Y	char	'0' (Bid) '1' (Offer) '2' (Trade) '3' (Index value) '4' (Opening price) '5' (Closing price) '6' (Settlement price) '7' (Trading session high price) '8' (Trading session low price) '9' (Trading session VWAP price) *'B' (Trade volume)* *'C' (Open interest)* *'N' (Session high bid)* *'O' (Session low offer)* *'h' (Open period price)* *'i' (Last bid price)* *'j' (Last offer price)* *'k' (Close period price)* *'l' (Market price 2)* *'m' (Market price)* *'n' (Last negotiated deal)* *'o' (Official open price)* *'p' (Official current price)* *'q' (Admitted quote)* *'r' (Official close price)* *'v' (Total bid volume)* *'w' (Total offer volume)* *'t' (Negotiated deals value)* *'u' (Duration)*	Type of market data entry.

				'y' (Market statistics) *'z' (Trade list)* *'*' (All market data)*	
146	NoRelatedSym	Y	int		Number of instruments requested.
=> 55	Symbol	Y	String		Ticker symbol of requested instrument. Common, "human understood" representation of the security. "[N/A]" is used for products which do not have a symbol.
=> 48	SecurityID	N	String		Security identifier value (e.g. CUSIP, SEDOL, ISIN, etc).
=> 22	IDSource	N	String	'4' (ISIN number)	Identifies class or source of the security identifier value. Required if SecurityID (48) is specified.
=> 336	TradingSessionID	Y*	String		Identifier of trading session.
<Standard Message Trailer>		Y			

Market Data Request Reject (MsgType = 'Y')

Сообщение типа Market Data Request Reject (MsgType = 'Y') отправляется в том случае, когда запрос Market Data Request не может быть выполнен.

Tag	Field name	Req'd	Type	Valid values	Comments
<Standard Message Header>		Y			MsgType = 'Y'
262	MDReqID	Y	String		Identifier of Market Data Request message which is rejected.
281	MDReqRejReason	N	char	'0' (Unknown symbol) '1' (Duplicate MDReqID) '3' (Insufficient permissions) '4' (Unsupported SubscriptionRequestType) '8' (Unsupported	Reason for the rejection of Market Data Request message.

				MDEntryType) '9' (Unsupported TradingSessionID) *'x' (Instrument is not traded at board)* *'y' (Unknown MDReqID)* *'z' (Initial download from corresponding MICEX market has not been performed yet)*	
58	Text	N	String		Message to explain reason for rejection.
<Standard Message Trailer>		Y			

Market Data - Snapshot/Full Refresh (MsgType = 'W')

Message of type The Market Data – Snapshot/Full Refresh (MsgType = 'W') is sent in response to Market Data Request (MsgType = 'V') with SubscriptionRequestType = '0' (snapshot reuquest).

Each Market Data – Snapshot/Full Refresh (MsgType = 'W') is linked to some Market Data Request (MsgType = 'V'). Value of tag MDReqID(262) in snapshot message is equal to the value of this tag in Market Data Request (MsgType = 'V').

Tag	Field name	Req'd	Type	Valid values	Comments
<Standard Message Header>		Y			MsgType = 'W'
262	MDReqID	N	String		Identifier of Market Data Request message which this message is sent in response to.
55	Symbol	Y	String		Ticker symbol of the requested security. Common, "human understood" representation of the security. "[N/A]" is used for products which do not have a symbol.
451	*NetChgPrevDay*	*N*	*PriceOffset*		*Net change from previous day's closing price vs. last traded price.*
268	NoMDEntries	Y	int		Number of market data entries following.

=> 269	MDEntryType	Y	char	'0' (Bid) '1' (Offer) '2' (Trade) '3' (Index value) '4' (Opening price) '5' (Closing price) '6' (Settlement price) '7' (Trading session high price) '8' (Trading session low price) '9' (Trading session VWAP price) *'B' (Trade volume)* *'C' (Open interest)* *'J' (Empty book)* *'N' (Session high bid)* *'O' (Session low offer)* *'h' (Open period price)* *'i' (Last bid price)* *'j' (Last offer price)* *'k' (Close period price)* *'l' (Market price 2)* *'m' (Market price)* *'n' (Last negotiated deal)* *'o' (Official open price)* *'p' (Official current price)* *'q' (Admitted quote)* *'r' (Official close price)* *'v' (Total bid*	Type of market data entry.

				volume) *'w' (Total offer volume)* *'t' (Negotiated deals value)* *'u' (Duration)* *'z' (Trade list)*	
=> 278	MDEntryID	N	String		Unique market data entry identifier.
=> 270	MDEntryPx	Y	Price		Price of the market data entry.
=> 271	MDEntrySize	N	Qty		Quantity or volume represented by the market data entry. Required if MDEntryType (269) = '0', '1', '2'.
=> 272	MDEntryDate	N	UTCDate		Date of market data entry.
=> 273	MDEntryTime	N	UTCTimeOnly		Time of market data entry.
=> 336	TradingSessionID	N	String		Identifier of trading session.
=> 276	QuoteCondition	N	MultipleValueString	'C' (Exchange best)	Space-delimited list of conditions describing a quote.
=> 277	TradeCondition	N	MultipleValueString	'J' (Next day trade) *'R' (Opening price)* *'AJ' (Official closing price)*	Space-delimited list of conditions describing a trade.
=> 286	OpenCloseSettlFlag	N	char	'4' (Entry from previous business day)	Flag that identifies a market data entry.
=> 5384	*AccruedInterestAmt*	*N*	*Amt*		*Amount of accrued interest.*
=> 5510	*ChgFromWAPrice*	*N*	*PriceOffset*		*Indicates change from previous day's weighted average price vs. last traded*

					price.
=> 5511	*ChgOpenInterest*	*N*	*Qty*		*Indicates change from previous day's open interest.*
=> 6139	*TotalNumOfTrades*	*N*	*int*		*Total number of trades.*
=> 6143	*TradeValue*	*N*	*Amt*		*Trade value.*
=> 6608	*Yield*	*N*	*float*		*Yield calculated at MDEntryPx (270).*
=> 9168	*OfferNbOr*	*N*	*int*		*Total number of sell orders.*
=> 9169	*BidNbOr*	*N*	*int*		*Total number of buy orders.*
=> 9750	*ChgFromSettlmnt*	*N*	*PriceOffset*		*Indicates the change from previous day's settlement price vs. last traded price.*
<Standard Message Trailer>		Y			

Market Data — Incremental Refresh (MsgType = 'X')

Message of type Market Data – Incremental Refresh (MsgType ='X') is sent by MICEX-FIX MARKET DATA when market data for subscribed instrument (Market Data Request (MsgType ='V') was previously sent with SubscriptionRequestType = '1' (snapshot with updates request)) is changed. This message is also sent in response to unsubscribe request SubscriptionRequestType = ‘2’.

Tag	Field name	Req'd	Type	Valid values	Comments
<Standard Message Header>		Y			MsgType = 'X'
262	MDReqID	N	String		Identifier of Market Data Request message which this message is sent in response to.
268	NoMDEntries	Y	int		Number of market data entries following.

=> 279	MDUpdateAction	Y	char	'0' (New) '1' (Change) '2' (Delete)	Type of market data update action.
=> 269	MDEntryType	N	char	'0' (Bid) '1' (Offer) '2' (Trade) '3' (Index value) '4' (Opening price) '5' (Closing price) '6' (Settlement price) '7' (Trading session high price) '8' (Trading session low price) '9' (Trading session VWAP price) *'B' (Trade volume)* *'C' (Open interest)* *'N' (Session high bid)* *'O' (Session low offer)* *'h' (Open period price)* *'i' (Last bid price)* *'j' (Last offer price)* *'k' (Close period price)* *'l' (Market price 2)* *'m' (Market price)* *'n' (Last negotiated deal)* *'o' (Official open price)* *'p' (Official current price)* *'q' (Admitted*	Type of market data entry. Required if MDUpdateAction (279) = '0'.

				quote) *'r' (Official close price)* *'v' (Total bid volume)* *'w' (Total offer volume)* *'t' (Negotiated deals value)* *'u' (Duration)* *'z' (Trade list)*	
=> 278	MDEntryID	N	String		Unique market data entry identifier.
=> 55	Symbol	N	String		Ticker symbol of the requested security. Common, "human understood" representation of the security. "[N/A]" is used for products which do not have a symbol.
=> 270	MDEntryPx	N	Price		Price of the market data entry. Required if MDUpdateAction (279) = '0'.
=> 271	MDEntrySize	N	Qty		Quantity or volume represented by the market data entry. Required if MDUpdateAction (279) = '0' and MDEntryType (269) = '0', '1', '2'.
=> 272	MDEntryDate	N	UTCDate		Date of market data entry.
=> 273	MDEntryTime	N	UTCTimeOnly		Time of market data entry.
=> 336	TradingSessionID	N	String		Identifier of trading session.
=> 276	QuoteCondition	N	MultipleValueString	'C' (Exchange best)	Space-delimited list of conditions describing a quote.
=> 277	TradeCondition	N	MultipleValueString	'J' (Next day trade) *'R' (Opening price)* *'AJ' (Official*	Space-delimited list of conditions describing a trade.

				closing price)	
=> 286	OpenCloseSettlFlag	N	char	'4' (Entry from previous business day)	Flag that identifies a market data entry.
=> 451	*NetChgPrevDay*	*N*	*PriceOffset*		*Net change from previous day's closing price vs. last traded price.*
=> 5384	*AccruedInterestAmt*	*N*	*Amt*		*Amount of accrued interest.*
=> 5510	*ChgFromWAPrice*	*N*	*PriceOffset*		*Indicates change from previous day's weighted average price vs. last traded price.*
=> 5511	*ChgOpenInterest*	*N*	*Qty*		*Indicates change from previous day's open interest.*
=> 6139	*TotalNumOfTrades*	*N*	*int*		*Total number of trades.*
=> 6143	*TradeValue*	*N*	*Amt*		*Trade value.*
=> 6608	*Yield*	*N*	*float*		*Yield calculated at MDEntryPx (270).*
=> 9168	*OfferNbOr*	*N*	*int*		*Total number of sell orders.*
=> 9169	*BidNbOr*	*N*	*int*		*Total number of buy orders.*
=> 9750	*ChgFromSettlmnt*	*N*	*PriceOffset*		*Indicates the change from previous day's settlement price vs. last traded price.*
<Standard Message Trailer>		Y			

Email (MsgType = 'C')

FIX Email message (MsgType = ‘C’) is used to notify clients about errors, which cannot be described using standard FIX reject messages. This can be error during message conversion, failure during state restore, etc.

Tag	Field name	Req'd	Type	Valid values	Comments
<Standard Message Header>		Y			MsgType = 'C'
164	EmailThreadID	Y	String		Unique identifier for the email message thread.
94	EmailType	Y	Char	'0' (New)	Email message type.
42	OrigTime	N	UTCTimestamp		Time of message origination.
147	Subject	Y	String		Subject of the email message.
33	LinesOfText	Y	int		Specifies the number of repeating lines of text specified.
=> 58	Text	Y	String		Free format text string.
95	RawDataLength	N	Int		Byte length of RawData (96) field.
96	RawData	N	Data		Unformatted raw data, can include bitmaps, word processor documents, etc.
<Standard Message Trailer>		Y			

Business Message Reject (MsgType = 'j')

Message of type Business Message Reject is used to reject business level messages, wich cannot be rejectd using specialized reject messages. Please note that if message has error on session level (for example reuquired tag is missing) than administration reject message is sent Reject (MsgType = '3').

Tag	Field name	Req'd	Type	Valid values	Comments
<Standard Message Header>		Y			MsgType = 'j'
45	RefSeqNum	N	int		MsgSeqNum(34) of rejected message.
372	RefMsgType	Y	char		MsgType(35) of rejected message.
380	BusinessRejectReason	Y	int	'3' (Unsupported Message Type)	Code to identify reason for reject.

58	Text	N	String		Message to explain reason for rejection.
<Standard Message Trailer>		Y			

Usage of Market Data messages

Market Data Request

Market Data Request (MsgType = 'V') is used to subscribe for- or unsubscribe from- market data. Tags and values to be used are described below.

Request Snapshot/Subscribe for MarketData

Market Data Request snapshot/subscribe for Market Data:

Tag	Field name	Value	Comment
262	MDReqID	<NewReqID>	New unique identifier of Market Data Request message should be generated.
263	SubscriptionRequestType	'0' (Snapshot) '1' (Snapshot + Updates (Subscribe))	
264	MarketDepth	'0' (Full Book)	
265	MDUpdateType	'1' (Incremental Refresh)	This tag should be present only if SubscriptionRequestType (263) = '1'.
267	NoMDEntryTypes	'1'	
=> 269[1]	MDEntryType	'*'[*] (Market Data)	
Tags related to instrument identification.			
Tags related to trading session identification.			

Request Snapshot/Subscribe for OrderBook

Market Data Request to request snapshot/subscribe for Order Book:

Tag	Field name	Value	Comment
262	MDReqID	<NewReqID>	New unique identifier of Market Data Request message should be generated.
263	SubscriptionRequestType	'0' (Snapshot) '1' (Snapshot + Updates (Subscribe))	
264	MarketDepth	'0' (Full Book)	
265	MDUpdateType	'1' (Incremental Refresh)	This tag should be present only if SubscriptionRequestType (263) = '1'.
267	NoMDEntryTypes	'2'	
=> 269[1]	MDEntryType	'0' (Bid)	
=> 269[2]	MDEntryType	'1' (Offer)	
Tags related to instrument identification.			
Tags related to trading session identification.			

Request Snapshot/Subscribe for TradeList

Market Data Request to request snapshot/subscribe for Trade List:

Tag	Field name	Value	Comment
262	MDReqID	<NewReqID>	New unique identifier of Market Data Request message should be generated.
263	SubscriptionRequestType	'0' (Snapshot) '1' (Snapshot + Updates (Subscribe))	
264	MarketDepth	'0' (Full Book)	
265	MDUpdateType	'1' (Incremental Refresh)	This tag should be present only if SubscriptionRequestType (263) = '1'.

267	NoMDEntryTypes	'1'	
=> 269[1]	MDEntryType	'z'* (Trade list)	
Tags related to instrument identification.			
Tags related to trading session identification.			

Request Snapshot/Subscribe for MarketStatistics

Market Data Request to request snapshot/subscribe for Market Statistics:

Tag	Field name	Value	Comment
262	MDReqID	<NewReqID>	New unique identifier of Market Data Request message should be generated.
263	SubscriptionRequestType	'0' (Snapshot) '1' (Snapshot + Updates (Subscribe))	
264	MarketDepth	'0' (Full Book)	
265	MDUpdateType	'1' (Incremental Refresh)	This tag should be present only if SubscriptionRequestType (263) = '1'.
267	NoMDEntryTypes	'1'	
=> 269[1]	MDEntryType	'y'* (Market Statistics)	
Tags related to instrument identification.			
Tags related to trading session identification.			

Market Data Request to request snapshot/subscribe for BestBid/BestOffer:

Tag	Field name	Value	Comment
262	MDReqID	<NewReqID>	New unique identifier of Market Data Request message should be generated.

263	SubscriptionRequestType	'0' (Snapshot) '1' (Snapshot + Updates (Subscribe))	
264	MarketDepth	'1' (Top of Book)	
265	MDUpdateType	'1' (Incremental Refresh)	This tag should be present only if SubscriptionRequestType (263) = '1'.
267	NoMDEntryTypes	'2'	
=> $269_{[1]}$	MDEntryType	'0' (Bid)	
=> $269_{[2]}$	MDEntryType	'1' (Offer)	
Tags related to instrument identification.			
Tags related to trading session identification.			

Market Data Request to request snapshot/subscribe for Last Trade, Weighted Average, Settlement, Open Interest, Open Price, Close Price, High Price, Low Price, High Bid Price, Low Offer Price, Volume Today, Admitted Quote, Bid Total, Offer Total, Market, Closing, Official Close, Negotiated Deals Today, Last Bid Price, Last Offer Price, Market Price 2, Opening Price, Last Negotiated Deal, Official Open Price, Official Current Price, Duration:

Tag	Field name	Value	Comment
262	MDReqID	<NewReqID>	New unique identifier of Market Data Request message should be generated.
263	SubscriptionRequestType	'0' (Snapshot) '1' (Snapshot + Updates (Subscribe))	
264	MarketDepth	'0' (Full Book)	
265	MDUpdateType	'1' (Incremental Refresh)	This tag should be present only if SubscriptionRequestType (263) = '1'.
267	NoMDEntryTypes	'26'	
=> $269_{[1]}$	MDEntryType	'2' (Trade)	

=> $269_{[2]}$	MDEntryType	'4' (Opening price)	
=> $269_{[3]}$	MDEntryType	'5' (Closing price)	
=> $269_{[4]}$	MDEntryType	'6' (Settlement price)	
=> $269_{[5]}$	MDEntryType	'7' (Trading session high price)	
=> $269_{[6]}$	MDEntryType	'8' (Trading session low price)	
=> $269_{[7]}$	MDEntryType	'9' (Trading session VWAP price)	
=> $269_{[8]}$	MDEntryType	'B' (Trade volume)	
=> $269_{[9]}$	MDEntryType	'C' (Open Interest)	
=> $269_{[10]}$	MDEntryType	'N'*(Session high bid)	
=> $269_{[11]}$	MDEntryType	'O'* (Session low offer)	
=> $269_{[12]}$	MDEntryType	'h'* (Open period price)	
=> $269_{[13]}$	MDEntryType	'i'* (Last bid price)	
=> $269_{[14]}$	MDEntryType	'j'* (Last offer price)	
=> $269_{[15]}$	MDEntryType	'k'* (Close period price)	
=> $269_{[16]}$	MDEntryType	'l'*(Market price 2)	
=>	MDEntryType	'm'* (Market price)	

269[17]			
=> 269[18]	MDEntryType	'n'* (Last negotiated deal)	
=> 269[19]	MDEntryType	'o'* (Official open price)	
=> 269[20]	MDEntryType	'p'* (Official current price)	
=> 269[21]	MDEntryType	'q'* (Admitted quote)	
=> 269[22]	MDEntryType	'r'* (Official close price)	
=> 269[23]	MDEntryType	'v'* (Total bid volume)	
=> 269[24]	MDEntryType	'w'* (Total offer volume)	
=> 269[25]	MDEntryType	't'* (Negotiated deals value)	
=> 269[26]	MDEntryType	'u'* (Duration)	
Tags related to instrument identification.			
Tags related to trading session identification.			

Attention: It is possible to subscribe for any combination of these elements instead of subscription for everything. In this case only required elements (with corresponding MDEntryType (269)) must be included into Market Data Request.

Subscribe for Index information

Market Data Request to request snapshot/subscribe for snapshot/subscribe for index:

Tag	Field name	Value	Comment
262	MDReqID	<NewReqID>	New unique identifier of Market Data Request message should

			be generated.
263	SubscriptionRequestType	'0' (Snapshot) '1' (Snapshot + Updates (Subscribe))	
264	MarketDepth	'0' (Full Book)	
265	MDUpdateType	'1' (Incremental Refresh)	This tag should be present only if SubscriptionRequestType (263) = '1'.
267	NoMDEntryTypes	'1'	
=> $269_{[1]}$	MDEntryType	'3' (Index value)	
Tags related to instrument identification.			
Tags related to trading session identification.			

Unsubscribe

Tag	Field name	Value	Comment
262	MDReqID	<ReqIDFromRequest>	Identifier of corresponding Market Data Request (Subscribe) message.
263	SubscriptionRequestType	'2' (Unsubscribe)	
264	MarketDepth	'0' (Full Book) '1' (Top of Book)	Should match value which was sent in corresponding Market Data Request (Subscribe) message.
267	NoMDEntryTypes	<RecordsNo>	Should match value which was sent in corresponding Market Data Request (Subscribe) message.
=> $269_{[i]}$	MDEntryType	'0' (Bid) '1' (Offer) '2' (Trade) '3' (Index value) '4' (Opening price) '5' (Closing price) '6' (Settlement price) '7' (Trading session high price) '8' (Trading session	Should match value which was sent in corresponding Market Data Request (Subscribe) message.

		low price) '9' (Trading session VWAP price) 'B' (Trade volume) 'C' (Open interest) 'N'* (Session high bid) 'O'* (Session low offer) 'h'* (Open period price) 'i'* (Last bid price) 'j'* (Last offer price) 'k'* (Close period price) 'l'* (Market price 2) 'm'* (Market price) 'n'* (Last negotiated deal) 'o'* (Official open price) 'p'* (Official current price) 'q'* (Admitted quote) 'r'* (Official close price) 'v'* (Total bid volume) 'w'* (Total offer volume) 't'* (Negotiated deals value) 'u'* (Duration) 'y'* (Market statistics) 'z'* (Trade list) '*' (All market data)	
Tags related to instrument identification.			
Tags related to trading session identification.			

Market Data — Snapshot/Full Refresh

Market Data — Snapshot/Full Refresh (MsgType = 'W') is used to send initial snapshot of market data.

Attention: Tags and fields marked with * are used defined i.e. they are not defined in standard protocol.

Snapshot for OrderBook

Market Data — Snapshot/Full Refresh as shown blow is sent as empty snapshot for OrderBook:

Tag	Field name	Value	Comment
262	MDReqID	<ReqIDFromRequest>	Identifier of corresponding Market Data Request message is returned.
Tags related to instrument identification.			
268	NoMDEntries	'1'	
=> $269_{[1]}$	MDEntryType	'J'* (Empty book)	
=> Tags related to trading session identification.			

Market Data — Snapshot/Full Refresh as shown below is sent as non-empty snapshot for OrderBook:

Tag	Field name	Value	Comment
262	MDReqID		Identifier of corresponding Market Data Request message is returned.
Tags related to instrument identification.			
268	NoMDEntries		Number of entries depends on number of records in Order Book.
=> $269_{[i]}$	MDEntryType	'0' (Bid)	
=> $278_{[i]}$	MDEntryID	<TradeId>	Unique identifier of OrderBook entry.
=> $270_{[i]}$	MDEntryPx	Bid price	Price of OrderBook bid entry.
=> $271_{[i]}$	MDEntrySize	Bid quantity	Quantity of OrderBook bid entry.
=>	Yield	Bid yield	Yield at bid price of OrderBook bid entry.

6608$^{*}_{[i]}$			
=> 269$_{[j]}$	MDEntryType	'1' (Offer)	
=> 270$_{[j]}$	MDEntryPx	Offer price	Price of OrderBook offer entry.
=> 271$_{[j]}$	MDEntrySize	Offer quantity	Quantity of OrderBook offer entry.
=> 6608$^{*}_{[j]}$	Yield	Offer yield	Yield at offer price of OrderBook offer entry.
=> Tags related to trading session identification.			

Snapshot for TradeList

Market Data — Snapshot/Full Refresh as shown below is sent as empty snapshot for TradeList:

Tag	Field name	Value	Comment
262	MDReqID	<ReqIDFromRequest>	Identifier of corresponding Market Data Request message is returned.
Tags related to instrument identification.			
268	NoMDEntries	'1'	
=> 269$_{[1]}$	MDEntryType	'J'* (Empty book)	
=> Tags related to trading session identification.			

Market Data — Snapshot/Full Refresh as shown below is sent as non-empty snapshot for TradeList:

Tag	Field name	Value	Comment
262	MDReqID		Identifier of corresponding Market Data Request message is returned.

Tags related to instrument identification.			
268	NoMDEntries		Number of entries depends on number of records in TradeList.
=> 269[i]	MDEntryType	'z'* (Trade list)	
=> 270[i]	MDEntryPx	Trade price	Price of TradeList entry.
=> 271[i]	MDEntrySize	Trade quantity	Quantity of TradeList entry.
=> 273[i]	MDEntryTime	Trade time	Time of TradeList entry.
=> 277[i]	TradeCondition	'J' (Next day trade)	Trade condition of TradeList entry.
=> 278[i]	MDEntryID	Trade identifier	Unique identifier of TradeList entry.
=> 6143*[i]	TradeValue	Trade value	Value of TradeList entry.
=> 6608*[i]	Yield	Yield at trade price	Yield at trade price of TradeList entry.
=> 5384*[i]	AccruedInterestAmt	Accrued interest	Amount of accrued interest of TradeList entry.
=> Tags related to trading session identification.			

Snapshot for MarketStatistics

Market Data — Snapshot/Full Refresh as shown below is sent as empty snapshot for BestBid/BestOffer:

Tag	Field name	Value	Comment
262	**MDReqID**		**Identifier of corresponding Market Data Request message is returned.**

Tags related to instrument identification.			
268	NoMDEntries	'1'	
=> 269[1]	MDEntryType	'J'* (Empty book)	
=> Tags related to trading session identification.			

Market Data — Snapshot/Full Refresh as shown below is sent as non-empty snapshot for BestBid/BestOffer:

Tag	Field name	Value	Comment
262	**MDReqID**		**Identifier of corresponding Market Data Request message is returned.**
Tags related to instrument identification.			
268	NoMDEntries	'2'	
=> 269[1]	MDEntryType	'0' (Bid)	
=> 270[1]	MDEntryPx	Bid price	Price of best bid entry.
=> 271[1]	MDEntrySize	Bid quantity	Quantity of best bid entry.
=> 276[1]	QuoteCondition	'C' (Exchange best)	
=> 269[2]	MDEntryType	'1' (Offer)	
=> 270[2]	MDEntryPx	Offer price	Price of best offer entry.
=> 271[2]	MDEntrySize	Offer quantity	Quantity of best offer entry.
=>	QuoteCondition	'C' (Exchange best)	

276[2]			
=> Tags related to trading session identification.			

Market Data — Snapshot/Full Refresh message as shown below is sent as empty for Last, High, Low, Open, Close, today's TradeVolume, HighBid, LowOffer, WeightedAveragePrice, SettlementPrice:

Tag	Field name	Value	Comment
262	MDReqID	<ReqIDFromRequest>	Identifier of corresponding Market Data Request message is returned.
Tags related to instrument identification.			
268	NoMDEntries	'1'	
=> 269[1]	MDEntryType	'J'* (Empty book)	
=> Tags related to trading session identification.			

Market Data — Snapshot/Full Refresh as shown below is sent as non-empty snapshot for Last Trade, Weighted Average, Settlement, Open Interest, Open Price, Close Price, High Price, Low Price, High Bid Price, Low Offer Price, Volume Today, Admitted Quote, Bid Total, Offer Total, Market, Closing, Official Close, Negotiated Deals Today, Last Bid Price, Last Offer Price, Market Price 2, Opening Price, Last Negotiated Deal, Official Open Price, Official Current Price, Duration:

Tag	Field name	Value	Comment
262	MDReqID		Identifier of corresponding Market Data Request message is returned.
Tags related to instrument identification.			
451	NetChgPrevDay	Net change	Net change from previous day's closing price vs. last traded price.
268	NoMDEntries	'31'	
=> 269[1]	MDEntryType	'2' (Trade)	
=>	TradeCondition	'J' (Next day trade)	Condition of trade.

277[1]			
=> 270[1]	MDEntryPx		Last Trade price.
=> 271[1]	MDEntrySize		Last Trade quantity.
=> 273[1]	MDEntryTime		Last Trade time.
=> 5510*[1]	ChgFromWAPrice		Change from previous day's weighted average price.
=> 9750*[1]	ChgFromSettlmnt		Change from previous day's settlement price.
=> 6143*[1]	TradeValue		Last Trade value.
=> 6608*[1]	Yield		Yield at Last Trade price.
=> 5384*[1]	AccruedInterestAmt	Accrued interest	Amount of accrued interest.
=> 269[2]	MDEntryType	'4' (Opening price)	
=> 270[2]	MDEntryPx		Open price.
=> 269[3]	MDEntryType	'5' (Closing price)	
=> 270[3]	MDEntryPx		Close price.
=> 269[4]	MDEntryType	'6' (Settlement price)	
=> 270[4]	MDEntryPx		Settlement price.
=> 269[5]	MDEntryType	'6' (Settlement price)	

=> 272[5]	MDEntryDate	TradeDate	Last Trade date.
=> 286[5]	OpenCloseSettlFlag	'4' (Entry from previous business day)	
=> 270[5]	MDEntryPx		Previous day settlement price.
=> 269[6]	MDEntryType	'7' (Trading session high price)	
=> 270[6]	MDEntryPx		High price.
=> 269[7]	MDEntryType	'8' (Trading session low price)	
=> 270[7]	MDEntryPx		Low price.
=> 269[8]	MDEntryType	'9' (Trading session VWAP price)	
=> 270[8]	MDEntryPx		Weighted average price.
=> 6608*[8]	Yield		Yield at weighted average price.
=> 269[9]	MDEntryType	'9' (Trading session VWAP price)	
=> 286[9]	OpenCloseSettlFlag	'4' (Entry from previous business day)	
=> 270[9]	MDEntryPx		Previous day weighted average price.
=> 272[9]	MDEntryDate	Trade Date	Last Trade date.
=> 6608*[9]	Yield		Yield at previous day weighted average price.
=>	MDEntryType	'B' (Trade volume)	

269[10]			
=> 271[10]	MDEntrySize	Trade volume	Today's volume.
=> 6143*[10]	Trade value		Today's trade value.
=> 6139*[10]	TotalNumOfTrades		Today's total number of trades
=> 269[11]	MDEntryType	'C' (Open Interest)	
=> 271[11]	MDEntrySize		Open Interest.
=> 5511*[11]	ChgOpenInterest	Open interest change	Change from previous day's open interest.
=> 269[12]	MDEntryType	'N'* (Session high bid)	
=> 270[12]	MDEntryPx		High bid price.
=> 269[13]	MDEntryType	'O'* (Session low offer)	
=> 270[13]	MDEntryPx		Low offer price.
=> 269[14]	MDEntryType	'h'* (Open period price)	
=> 270[14]	MDEntryPx	Opening price	Open period price.
=> 269[15]	MDEntryType	'i'* (Last bid price)	
=> 270[15]	LastBidPx		Last bid price.
=> 269[16]	MDEntryType	'j'* (Last offer price)	

=> 270[16]	MDEntryPx		Last offer price.
=> 269[17]	MDEntryType	'k'* (Close period price)	
=> 270[17]	MDEntryPx	Closing price	Close period price.
=> 6608*[17]	Yield		Yield at close period price.
=> 269[18]	MDEntryType	'l'* (Market price 2)	
=> 270[18]	MDEntryPx		Market price 2.
=> 269[19]	MDEntryType	'm'* (Market price)	
=> 270[19]	MDEntryPx		Market price.
=> 269[20]	MDEntryType	'm'* (Market price)	
=> 286[20]	OpenCloseSettlFlag	'4' (Entry from previous business day)	
=> 270[20]	MDEntryPx		Previous day market price.
=> 272[20]	MDEntryDate	Entry date	Previous business date.
=> 269[21]	MDEntryType	'n'* (Last negotiated deal)	
=> 270[21]	MDEntryPx		Last negotiated deal.
=> 269[22]	MDEntryType	'o'* (Official open price)	
=>	MDEntryPx		Official open price.

270[22]			
=> 269[23]	MDEntryType	'p'* (Official current price)	
=> 270[23]	MDEntryPx		Official current price.
=> 269[24]	MDEntryType	'q'* (Admitted quote)	
=> 270[24]	MDEntryPx		Admitted quote.
=> 269[25]	MDEntryType	'q'* (Admitted quote)	
=> 270[25]	MDEntryPx		Previous day admitted quote.
=> 286[25]	OpenCloseSettlFlag	'4' (Entry from previous business day)	
=> 272[25]	MDEntryDate		Previous business date.
=> 269[26]	MDEntryType	'r'* (Official close price)	
=> 270[26]	MDEntryPx		Official close price.
=> 269[27]	MDEntryType	'r'* (Official close price)	
=> 270[27]	MDEntryPx		Previous day official close price.
=> 286[27]	OpenCloseSettlFlag	'4' (Entry from previous business day)	
=> 272[27]	MDEntryDate		Previous business date.
=> 269[28]	MDEntryType	'v'* (Total bid volume)	

=> 9169[28]	BidNbOr	Bid depth total	Total bid volume.
=> 269[29]	MDEntryType	'w'* (Total offer volume)	
=> 9168[29]	OfferNbOr	Offer depth total	Total offer volume.
=> 269[30]	MDEntryType	't'* (Negotiated deals value)	
=> 6143*[30]	TradeValue		Today's negotiated deals value
=> 6139*[30]	TotalNumOfTrades		Today's total number of negotiated deals
=> 269[31]	MDEntryType	'u'* (Duration)	
=> 273[31]	MDEntryTime		Duration.
=> Tags related to trading session identification.			

Snapshot for Index information

Market Data — Snapshot/Full Refresh as shown below is sent as wmpty snapshot for index:

Tag	Field name	Value	Comment
262	MDReqID	<ReqIDFromRequest>	Identifier of corresponding Market Data Request message is returned.
Tags related to instrument identification.			
268	NoMDEntries	'1'	
=> 269[1]	MDEntryType	'J'* (Empty book)	
=> Tags related to trading session identification.			

Market Data — Snapshot/Full Refresh as shown below is sent as non-empty snapshot for index:

Tag	Field name	Value	Comment
262	MDReqID		Identifier of corresponding Market Data Request message is returned.
Tags related to instrument identification.			
268	NoMDEntries	'3'	
=> $269_{[1]}$	MDEntryType	'3' (Index value)	
=> $270_{[1]}$	MDEntryPx		Last index value.
=> $273_{[1]}$	MDEntryTime	Last index time	Time of last update of last index value.
=> $6143^{*}_{[1]}$	TradeValue		Today's index value.
=> $269_{[2]}$	MDEntryType	'3' (Index value)	
=> $277_{[2]}$	TradeCondition	'AJ'* (Official closing price)	
=> $270_{[2]}$	MDEntryPx		Close index value.
=> $269_{[3]}$	MDEntryType	'3' (Index value)	
=> $277_{[3]}$	TradeCondition	'R' (Opening price)	
=> $270_{[3]}$	MDEntryPx		Open index value.
=> Tags related to trading session identification.			

If Market Data — Snapshot/Full Refresh для MarketStatistics ((269) = 'y') is request then the following sequence of messages will be sent:

- Snapshot for Best Bid/Best Offer
- Snapshot for other Market Statistics

If Market Data — Snapshot/Full Refresh for Market Data ((269) = '*') is requested then the following sequence of messages will be sent:

If instrument is not index then:

- Snapshot for Trade List
- Snapshot for Order Book
- Snapshot for Best Bid/Best Offer
- Snapshot for other Market Statistics

If instrument is index then:

- Snapshot for Index Statistics.

MICEX-FIX MARKET DATA custom tags and values

FIX4.4 custom tags and values

Custom values

Tag	Field name	Value	Description
167	SecurityType	'MUNI'	Municipal Bond
269	MDEntryType	'J' 'N' 'O' 'h' 'i' 'j' 'k' 'l' 'm' 'n' 'o' 'p' 'q' 'r' 'v' 'w' 't' 'u' 'y' 'z' '*'	Empty book Session high bid Session low offer Open period price Last bid price Last offer price Close period price Market price 2 Market price Last negotiated deal Official open price Official current price Admitted quote Official close price Total bid volume Total offer volume Negotiated deals value Duration Market statistics Trade list All market data
277	TradeCondition	'AJ'	Official close price
281	MDReqRejReason	'x' 'y' 'z'	Instrument is not traded on board Unknown MDReqId Initial download from corresponding MICEX market has not been performed yet

323	SecurityResponseType	'100'	Initial download from corresponding MICEX market has not been performed yet.
326	SecurityTradingStatus	'100'	Initial download from corresponding MICEX market has not been performed yet.
340	TradSesStatus	'100' '101' '102' '103' '104'	Trading system restarted Connection to MICEX market established Connection to MICEX market terminated correctly Connection to MICEX market terminated non-gracefully Reconnecting to MICEX market
865	EventType	'7' '100'	Last eligible trade date First eligible trade date
871	InstrAttribType	'27'	Instrument price precision

Custom tags

Tag	Field name	Data type	Valid values	Description	Used in
5217	StateSecurityId	String		State security identification number.	Security Definition (MsgType = 'd')
5382	EncodedShortSecurityDescLen	Length		Byte length of encoded (non-ASCII characters) EncodedShortSecurityDesc (5383) field.	Security Definition (MsgType = 'd')
5383	EncodedShortSecurityDesc	data		Encoded (non-ASCII characters) representation of the short security description in the encoded format specified via the MessageEncoding (347) field.	Security Definition (MsgType = 'd')
5384	AccruedInterestAmt	Amt		Amount of accrued interest.	Market Data - Snapshot/Full Refresh (MsgType = 'W'), Market Data - Incremental Refresh (Msgtype = 'X')
5385	MarketCode	String		Code of market where	Security Definition (MsgType

				instrument is traded.	= 'd')
5386	MinPriceIncrement	Price		Minimum price increase for security.	Security Definition (MsgType = 'd')
5387	MktShareLimit	Percentage		Market share limit.	Security Definition (MsgType = 'd')
5388	MktShareThreshold	Qty		Market share limit threshold.	Security Definition (MsgType = 'd')
5389	MaxOrdersVolume	Qty		Maximum summary volume of active buy and sell orders.	Security Definition (MsgType = 'd')
5470	PriceMvmLimit	Price		Maximum deviation of prices from settlement price.	Security Definition (MsgType = 'd')
5472	PriceMvmLimitT1	Price		Maximum deviation of prices from settlement price at T+1.	Security Definition (MsgType = 'd')
5508	FaceValue	Amt		Face value of security.	Security Definition (MsgType = 'd')
5509	AuctionIndicator	Boolean	'Y' - Yes 'N' - No	Indicates whether or not the auction is being held for the security.	Security Status (MsgType = 'f')
5510	ChgFromWAPrice	PriceOffset		Indicates change from previous day's weighted average price vs. last traded price.	Market Data - Snapshot/Full Refresh (MsgType = 'W'), Market Data - Incremental Refresh (Msgtype = 'X')
5511	ChgOpenInterest	Qty		Indicates change from previous day's open interest.	Market Data - Snapshot/Full Refresh (MsgType = 'W'), Market Data - Incremental Refresh (Msgtype = 'X')
5556	BaseSwapPx	Price		Base SWAP price.	Security Definition (MsgType = 'd')
5558	BuyBackPx	Price		Buy back price.	Security Definition (MsgType = 'd')

5559	BuyBackDate	LocakMktDate		Buy back date.	Security Definition (MsgType = 'd')
5677	Repo2Px	Price		Price of the second part of REPO.	Security Definition (MsgType = 'd')
6139	TotalNumOfTrades	int		Total number of trades.	Market Data - Snapshot/Full Refresh (MsgType = 'W'), Market data - Incremental Refresh (Msgtype = 'X')
6143	TradeValue	Amt		Trade value.	Market Data - Snapshot/Full Refresh (MsgType = 'W'), Market data - Incremental Refresh (Msgtype = 'X')
6608	Yield	Percentage		Yield calculated at MDEntryPx (270).	Market Data - Snapshot/Full Refresh (MsgType = 'W'), Market data - Incremental Refresh (Msgtype = 'X')
7595	NoSharesIssued	Qty		Number of shares issued.	Security Definition (MsgType = 'd')
9168	OfferNbOr	int		Total number of sell orders.	Market Data - Snapshot/Full Refresh (MsgType = 'W'), Market data - Incremental Refresh (Msgtype = 'X')
9169	BidNbOr	int		Total number of buy orders.	Market Data - Snapshot/Full Refresh (MsgType = 'W'), Market data - Incremental Refresh (Msgtype = 'X')
9199	HighLimit	Price		Allowable high limit price for the trading day.	Security Definition (MsgType = 'd')
9200	LowLimit	Price		Allowable low limit price for the trading day.	Security Definition (MsgType = 'd')
9750	ChgFromSettlmnt	PriceOffset		Indicates the change from previous day's settlement price vs. last traded price.	Market Data - Snapshot/Full Refresh (MsgType = 'W'), Market data - Incremental Refresh (Msgtype = 'X'

FIX4.2 custom tags and values

Custom values

Tag	Field name	Value	Description
167	SecurityType	'EUSOV' 'BN' 'INDEX'	Euro sovereigns Bank motes Index
269	MDEntryType	'B' 'C' 'J' 'N' 'O' 'h' 'i' 'j' 'k' 'l' 'm' 'n' 'o' 'p' 'q' 'r' 'v' 'w' 't' 'u' 'y' 'z' '*'	Trade volume Open interest Empty book Session high bid Session low offer Open period price Last bid price Last offer price Close period price Market price 2 Market price Last negotiated deal Official open price Official current price Admitted quote Official close price Total bid volume Total offer volume Negotiated deals value Duration Market statistics Trade list All market data
277	TradeCondition	'R' 'AJ'	Opening price Official close price
281	MDReqRejReason	'9' 'x' 'y' 'z'	Unsupported TradingSessionId Instrument is not traded on board Unknown MDReqId Initial download from corresponding MICEX market has not been performed yet
286	OpenCloseSettleFlag	'4'	Entry from previous business day
323	SecurityResponseType	'100'	Initial download from corresponding MICEX market has not been performed yet.

326	SecurityTradingStatus	'100'	Initial download from corresponding MICEX market has not been performed yet.
340	TradSesStatus	'100' '101' '102' '103' '104'	Trading system restarted Connection to MICEX market established Connection to MICEX market terminated correctly Connection to MICEX market terminated non-gracefully Reconnecting to MICEX market
865	EventType	'7' '100'	Last eligible trade date First eligible trade date
871	InstrAttribType	'27'	Instrument price precision

Custom tags

Tag	Field name	Data type	Valid values	Description	Used in
451	NetChgPrevDay	PriceOffset		Net change from previous day's closing price vs. last traded price.	Market Data - Snapshot/Full Refresh (MsgType = 'W')
553	Username	String		Username.	Logon (MsgType = 'A')
554	Password	String		Password	Logon (MsgType = 'A')
561	RoundLot	Qty		The trading lot size of a security.	Security Definition (MsgType = 'd')
5217	StateSecurityId	String		State security identification number.	Market Data - Snapshot/Full Refresh (MsgType = 'W'), Market data - Incremental Refresh (Msgtype = 'X'), Security Definition (MsgType = 'd'), Security Status (MsgType = 'f')
5382	EncodedShortSecurityDescLen	Length		Byte length of encoded (non-ASCII characters) EncodedShortSecurityDesc (5383) field.	Security Definition (MsgType = 'd')

5383	EncodedShortSecurityDesc	data		Encoded (non-ASCII characters) representation of the short security description in the encoded format specified via the MessageEncoding (347) field.	Security Definition (MsgType = 'd')
5384	AccruedInterestAmt	Amt		Amount of accrued interest.	Market Data - Snapshot/Full Refresh (MsgType = 'W'), Market Data - Incremental Refresh (Msgtype = 'X')
5385	MarketCode	String		Code of market where instrument is traded.	Security Definition (MsgType = 'd')
5386	MinPriceIncrement	Price		Minimum price increase for security.	Security Definition (MsgType = 'd')
5387	MktShareLimit	Percentage		Market share limit.	Security Definition (MsgType = 'd')
5388	MktShareThreshold	Qty		Market share limit threshold.	Security Definition (MsgType = 'd')
5389	MaxOrdersVolume	Qty		Maximum summary volume of active buy and sell orders.	Security Definition (MsgType = 'd')
5470	PriceMvmLimit	Price		Maximum deviation of prices from settlement price.	Security Definition (MsgType = 'd')
5472	PriceMvmLimitT1	Price		Maximum deviation of prices from settlement price at T+1.	Security Definition (MsgType = 'd')
5508	FaceValue	Amt		Face value of security.	Security Definition (MsgType = 'd')
5509	AuctionIndicator	Boolean	'Y' - Yes 'N' - No	No	Security Status (MsgType = 'f')
5510	ChgFromWAPrice	PriceOffset		Indicates change from previous day's weighted average price vs. last	Market Data - Snapshot/Full Refresh (MsgType = 'W'),

				traded price.	Market Data - Incremental Refresh (Msgtype = 'X')
5511	ChgOpenInterest	Qty		Indicates change from previous day's open interest.	Market Data - Snapshot/Full Refresh (MsgType = 'W'), Market Data - Incremental Refresh (Msgtype = 'X')
5512	FirstEligibleTradeDate	LocalMktDate		First eligible trade date.	Security Definition (MsgType = 'd')
5513	LastEligibleTradeDate	LocalMktDate		Last eligible trade date.	Security Definition (MsgType = 'd')
5514	InstrumentPricePrecision	int		Number of decimals in prices.	Security Definition (MsgType = 'd')
5556	BaseSwapPx	Price		Base SWAP price.	Security Definition (MsgType = 'd')
5558	BuyBackPx	Price		Buy back price.	Security Definition (MsgType = 'd')
5559	BuyBackDate	LocakMktDate		Buy back date.	Security Definition (MsgType = 'd')
5677	Repo2Px	Price		Price of the second part of REPO.	Security Definition (MsgType = 'd')
6139	TotalNumOfTrades	int		Total number of trades.	Market Data - Snapshot/Full Refresh (MsgType = 'W'), Market data - Incremental Refresh (Msgtype = 'X')
6143	TradeValue	Amt		Trade value.	Market Data - Snapshot/Full bRefresh (MsgType = 'W'), Market data - Incremental Refresh (Msgtype = 'X')
6608	Yield	Percentage		Yield calculated at MDEntryPx (270).	Market Data - Snapshot/Full Refresh (MsgType = 'W'), Market data - Incremental

					Refresh (Msgtype = 'X')
7595	NoSharesIssued	Qty		Number of shares issued.	Security Definition (MsgType = 'd')
7910	DayCountFraction	String		Describes the method used for calculating accrued interest for a bond.	Security Definition (MsgType = 'd')
9168	OfferNbOr	int		Total number of sell orders.	Market Data - Snapshot/Full Refresh (MsgType = 'W'), Market data - Incremental Refresh (Msgtype = 'X')
9169	BidNbOr	int		Total number of buy orders.	Market Data - Snapshot/Full Refresh (MsgType = 'W'), Market data - Incremental Refresh (Msgtype = 'X')
9127	DeliveryType	char	'1' (Cash settled) '2' (Physical delivery)	Specifies type of delivery for futures and options.	Security Definition (MsgType = 'd')
9199	HighLimit	Price		Allowable high limit price for the trading day.	Security Definition (MsgType = 'd')
9200	LowLimit	Price		Allowable low limit price for the trading day.	Security Definition (MsgType = 'd')
9750	ChgFromSettlmnt	PriceOffset		Indicates the change from previous day's settlement price vs. last traded price.	Market Data - Snapshot/Full Refresh (MsgType = 'W'), Market data - Incremental Refresh (Msgtype = 'X'