



FairX

FIX Order & Market Data API Specification

Version 0.21



Table of Contents

Contents

Revision History	5
Overview	6
The FairX Hours of Operation	6
Certification	6
Firm Identifiers	6
Order Identifiers	7
Trade Busts and Corrects	7
Done for Day and Expiration Reports	7
Self-Match Prevention	7
Failover and Delivery Guarantees	8
Cancel On Disconnect	8
Session Protocol	9
Messages Format	10
Standard Header Component	10
Standard Trailer Component	11
Administrative Messages	12
Logon (MsgType = A)	12
Logout (MsgType = 5)	12
Resend Request (MsgType = 2)	13
Sequence Reset (MsgType = 4)	13
Test Request (MsgType = 1)	13
Heartbeat (MsgType = 0)	13
Reject (MsgType = 3)	14
Order Management Messages	15



Overview	15
New Order Single (MsgType = D)	16
Order Cancel Request (MsgType = F)	18
Order Cancel/Replace Request (MsgType = G)	19
Execution Report: Accepted Order (MsgType = 8, ExecType = 0 or 5)	22
Execution Report: Order (MsgType = 8, ExecType = 4)	24
Execution Report: Stop Triggered (MsgType = 8, ExecType = L)	26
Execution Report: Trade (MsgType = 8, ExecType = F)	28
Execution Report: Order is Rejected (MsgType=8, ExecType = 8)	30
Execution Report: Done for Day (MsgType = 8, ExecType = 3)	31
Execution Report: Expired (MsgType = 8, ExecType = C)	32
Order Cancel Reject (MsgType = 9)	34
Last ExecId Request (MsgType = F1)	35
Last ExecId (MsgType = F2)	35
Event Resend Request (MsgType = F3)	35
Event Resend Complete (MsgType = F4)	35
Event Resend Reject (MsgType = F5)	36
Business Reject (MsgType = j)	36
Market Data	37
Security Definition Messages	37
Security Definition (MsgType = d)	37
SecurityListRequest	40
SecurityList	41
Market Data Messages	43
MarketDataRequest	43
MarketDataRequestReject	44
MarketDataSnapshotFullRefresh	44



MarketDataIncrementalRefresh	45
Tag Values	47
Order Status (tag 39)	47
Order Types (tag 40)	47
Order Sides (tag 54 and tag 624)	47
Order Time-in-force (tag 59)	47
Position Effect (tag 77)	48
Cancel Reject Reason (tag 102)	48
Order Reject Reason (tag 103)	48
Execution Types (tag 150)	48
Security Types (tag 167 and 609)	49
MD Entry Types (tag 269)	49
Session Reject Reason Codes (tag 373)	49
ExecRestatementReason (tag 378)	50
MultiLegReporting (tag 442)	50
OrderCapacity (tag 528)	50
CustOrderCapacity (tag 582)	50
SecuritySubType (tag 762)	51
CustomerOrderHandlingInst (tag 1031)	51
AggressorIndicator (tag 1057)	51
Manual or Automated flow (tag 1028)	51
SelfMatchPreventionStrategy (tag 8000)	51



1. Revision History

Version	Date	Author	Description
0.01	Dec-4-2019	ET	Initial draft
0.02	Jan-3-2020	MG, ET	Fixed Market Data Tags, SenderCompID improvements
0.03	Jan-24-2020	MG, FY	Market Data Tag updates and errata
0.04	Jan-28-2020	KW	Added MarketDataSnapshotFullRefresh message, Added 'change' value to MDUpdateAction, Dropped MarketDepth and MDUpdateType from incremental refresh message
0.05	Apr-22-2020	FY	Market Data Subscription updates
0.06	May-12-2020	JT	Spread Instrument support: Add InstrumentLegGroup to SecurityDefinition and SecurityList Add SecuritySubType tag and new MDEntryType values
0.07	Aug-4-2020	ET, FY	Execution Report ExecType updates, Add OnBehalfOfSenderSubID (116)
0.08	Oct-8-2020	MG	Added optional NoRelatedSymbol repeating group for MarketDataRequest
0.09	Nov-16-2020	FY	SenderCompID updates
0.10	Dec-1-2020	MG	Corrected MinQty description in NewOrderSingle, Removed Memo (5149) from all order message, Removed unsupported Order Status messages, Add ExecType=L for Stop Trigger, Update Fill ExecType=F instead of 1,2
0.11	Dec-2-2020	KW	Added LastExecIdRequest, LastExecId, EventResendRequest, EventResendAck and EventResendReject messages.
0.12	Dec-8-2020	ET	Removed Market Limit order type, updated Cancel/Replace MSG(G), update SelfMatchPreventionStrategy, Last ExecID Request msg updated, removed Good Till Date (59)
0.13	Dec-17-2020	KW	Added Execution Report ExecType=L for Stop Trigger Updated BusinessRejectRefID=Tag 379
0.14	Dec-28-2020	FY	Update Symbol (55) Length to 24
0.15	Dec-30-2020	KW	Added LastMsgSeqNumProcessed (369) to standard message header.
0.16	Jan-5-2021	ET	Updated Tag108 HeartBtInt range, Tag59 TimeInForce Removed Tag 110 MinQty from MsgType=G and MsgType=8, ExecType=4 / L, Removed Tag 432 ExpireDate MsgType=8, ExecType=C
0.17	Jan-25-2021	VD	Added Tag 286 (OpenCloseSettlFlag) & tag 6 (AvgPx), updated Tag 1682 (MDSecurityTradingStatus), Tag 269 (MDEntryType), Removed Tag 731 (SettlPriceType), Tag 1150 (TradingReferencePrice) and Tag 5796 (TradingReferenceDate), 5791 (ClearedVolume) & 5792 (OpenInterestQty) from 35=d/35=y
0.18	Mar-4-2021	FY	Update SelfMatchPreventionID to numeric
0.19	May-20-2021	VD FY	Addition of FIX tag 731 (SettlPriceType, value=2) for theoretical settlement price sent during initial seeding of a brand new contract Clarify Event Time 1145 Description and DataType on Security Definition and Security List Messages
0.20	June-21-2021	VD	Addition of FIX Tag 880 (TrdMatchID)
0.21	July-26-2021	VD, KW, FY	Adding Tag 5796 (TradingReferenceDate) to 35=y, 35=d, 35=W & 35=X Adding Tag 5797 (AggressorSide) to 35=X Adding 1682 (MDSecurityTradingStatus)=98 (Forbidden)



2. Overview

This document describes the FairX order management FIX API for brokers and market makers. The API allows connected firms to send, modify and cancel their orders. The expected latency for the FIX Order API is 400-500 microseconds roundtrip.

The API uses FIX protocol version 4.4. This document describes the messages supported by FairX. The document is not intended to serve as a full-fledged specification of the FIX protocol. Please refer to the official FIX specifications at <https://www.fixtrading.org/> website for additional details.

2.1. The FairX Hours of Operation

Contact the exchange for the current trading schedule. Orders entered outside trading hours will be rejected. Firms are encouraged to stay connected 15 minutes after the official close to receive execution reports that are generated due to trading session closing logic (e.g. Expired reports, Done for Day).

The exchange will reset its FIX sequences on a weekly basis. Reset schedule is configured during initial setup.

2.2. Certification

In order to connect to FairX, firms must be certified. FairX provides a separate environment for integration, acceptance testing and certification. Please contact the FairX team to obtain additional information.

2.3. Firm Identifiers

All messages sent from/to the exchange *must* contain both the SenderCompID (49) and TargetCompID (56) fields. The firm and the exchange agree on these values at the time of the firm onboarding.

Field	Firm to Exchange	Exchange to Firm
SenderCompID	Id of connection. A firm may have multiple connections, each connection will have its own CompID. This tag consists of 2 subcomponents: SubFirmID and Session ID. SubFirmID is the left-most 3 characters. The next 3 characters represent Session ID. Example: EBR123 SubFirmID: EBR Session ID: 123	Always "FAIRX"
TargetCompID	Always "FAIRX"	Id of connection. A firm may have multiple connections, each connection will have its own CompID. This tag consists of 2 subcomponents: SubFirmID and Session ID. SubFirmID is the left-most 3 characters. The next 3 characters represent Session ID. Example: EBR123 SubFirmID: EBR Session ID: 123



All application messages sent from or to the exchange *must* also contain both the SenderSubID (50) and TargetSubID (57) fields

Field	Firm to Exchange	Exchange to Firm
SenderSubID	Unique identifier for the end trader submitting orders. Tag 50 IDs are issued by an exchange clearing member firm (clearing member), their contracted vendors or assignees	Identifies the environment on the exchange side. This can be either “PROD” or “TEST”. The firm can use this value to guard itself from accidentally issuing UAT or test orders on the live market.
TargetSubID	Identifies the target environment on the exchange side. This can be either “PROD” or “TEST”. The firm can use this value to guard itself from accidentally issuing UAT or test orders on the live market. The exchange will reject any messages sent to unexpected environment (e.g. production environment will reject any TEST messages).	Unique identifier for the end trader submitting orders. Tag 50 IDs are issued by an exchange clearing member firm (clearing member), their contracted vendors or assignees

2.4. Order Identifiers

All orders submitted to FairX must have a unique ClOrdID (tag 11). The exchange only enforces the uniqueness of the identifier among working orders (for example, GTC and non-triggered stops). Non-unique ids can cause issues with reporting, clearing, and support.

2.5. Trade Busts and Corrects

The FairX FIX Order Management API does **not** distribute unsolicited reports about trade busts and corrects. The firms are expected to utilize the FIX Drop Copy connections if they need to receive these messages.

2.6. Done for Day and Expiration Reports

During initial setup a connection may be configured to distribute Done for Day execution reports for all the GTC/GTD orders that remain open after the trading day is closed. It is also possible to configure the API to distribute the Expired execution reports for the DAY orders that have been expired.

Firms are encouraged to stay connected 15 minutes after the official close to receive execution reports that are generated due to trading session closing logic (e.g. Expired reports, Done for Day).

2.7. Self-Match Prevention

Self-match prevention functionality allows market participants to prevent Buy and Sell orders for the same account, firm or group of accounts to match with each other. The functionality is optional and is controlled with a pair of FIX tags in incoming orders:

- **SelfMatchPreventionID** (7928) – orders with the same Self Match Prevention ID for the same executing firm will not match. Maximum length of the ID is 8 Digits.
- **SelfMatchPreventionStrategy** (8000) – this value defines the strategy of dealing with matching orders if self-match prevention is triggered. The exchange can either cancel the aggressor order, the resting order or both. The exchange uses the strategy from the *aggressor* order to deal with self-matched orders.



2.8. Failover and Delivery Guarantees

FairX will operate a pair of active gateways. Clients are free to connect to one or the other or both. All order events for a user will be sent to both gateways and be available for resend regardless of whether or not the client has an active session. Sessions on each gateway are distinct and thus have independent sequence numbers. However, an event `execId` is guaranteed to uniquely identify an event and can be used to de-duplicate events received from dual sessions.

Order requests may be sent via either gateway, but should not be sent via both simultaneously. FairX may reject duplicate requests, but they may also result in duplicate fills.

If the event of a disconnect with one or more pending/unacked requests, the client should connect to the other gateway (if not already connected) and send a Last `ExecId` Request to determine the `execId` of the last order event sent by the trading system to the client. If the client is missing events (the last seen `execId` is less than `lastExecId` returned in Last `ExecId` message), the client may either send an Event Resend Request or a session-level ResendRequest message to retrieve missed events. Having recovered all missed events, the client can safely assume that any requests for which there was no corresponding event was either not processed by the trading system or was rejected.

2.9. Cancel On Disconnect

A client may optionally be configured for cancel-on-disconnect orders. Cancel-on-disconnect applies to DAY orders only (not good-till-cancel orders).

If a client has cancel-on-disconnect enabled and is connected to both active gateways and disconnects from one of them, only the orders submitted (or last replaced/modified) by the disconnected session will be canceled. For example, if a client submits an order on gateway A and then updates that order on gateway B, the order will not be canceled if they disconnect from gateway A but will be canceled if they disconnect from gateway B.



3. Session Protocol

Session protocol assures client identification, sequential request processing, session state control and ability to restore the session after downtime. In the scope of a session, all FIX messages are identified by unique integer sequence numbers and are processed in that order. When an incoming sequence number does not match the expected one, the session must be recovered. If incoming sequence number is less than expected and PossDupFlag is not set to Y, then it is considered a fatal error, and the connection is dropped by the server. If the sequence number of incoming message is greater than the next expected number, then Resend Request is issued for missed messages.

The exchange resets the sequences weekly. Reset schedule is configured during initial setup. Resets may also be initiated by a firm sending a Logon message with ResetSeqNumFlag = Y.

Client and server use the following administrative messages:

- **Logon** – initiates (client) or approves (server) session opening.
- **Logout** – initiates or approves session closing.
- **Resend Request** – requests missed fix messages.
- **Sequence Reset – Gap Fill**, must be used instead of resend of administrative messages.
- **Test Request** – used to control the session state. Requires a reply **Heartbeat** message with properly filled TestReqID (112) field.
- **Heartbeat** – used to control session connection state.
- **Reject** – for administrative message reject.



4. Messages Format

All FIX messages, either administrative or business, require standard header and trailer components.

4.1. Standard Header Component

Tag	Name	FIX Type	Req	Description
8	BeginString	String(7)	Y	FIX Version for session, i.e. 'FIX.4.4'
9	BodyLength	Int(6)	Y	Number of bytes in the message body.
35	MsgType	String(2)	Y	Identifies FIX message type.
34	MsgSeqNum	Int(9)	Y	Sequence number
43	PossDupFlag	Boolean(1)	N	Indicates possible retransmission of message with this sequence number. Must be set to 'Y' for messages sent in response to a Resend Request from FairX
49	SenderCompID	String(6)	Y	To Exchange: Connection identifier assigned by FairX. This tag consists of 2 subcomponents: SubFirm ID and Session ID SubFirm ID is the left-most 3 characters. The next 3 characters represent Session ID. Example: EBR123 SubFirm ID: EBR Session ID: 123 From Exchange: Value from order entry tag 56-TargetCompID - "FAIRX"
50	SenderSubID	String(18)	Y	Unique identifier for the end trader submitting orders. Tag 50 IDs are issued by an exchange clearing member firm (clearing member), their contracted vendors or assignees
116	OnBehalfOfSubID	String(18)	N	Can be used to identify the end trader submitting orders if Tag 50 is populated by a third party connection provider.
52	SendingTime	UTCTimestamp(21)	Y	Timestamp of the message leaving the client system.
56	TargetCompID	String(7)	Y	To Exchange: Identifies entity receiving the message = "FAIRX" From Exchange Connection identifier assigned by FairX. This tag consists of 2 subcomponents: SubFirm ID and Session ID SubFirm ID is the left-most 3 characters. The next 3 characters represent Session ID. Example: EBR123 SubFirm ID: EBR

				Session ID: 123
57	TargetSubID	String(20)	Y	To Exchange: Identifies destination exchange system - "PROD" or "TEST" From Exchange: Echo back Tag57 sent by client system
122	OrigSendingTime	UTCTimestamp(21)	C	For resent messages only, contains timestamp from tag 52-SendingTime from original message.
369	LastMsgSeqNumProcessed	Int(9)	N	MsgSeqNum (34) of the last message from the client received and processed by the gateway.

4.2. Standard Trailer Component

Tag	Name	FIX Type	Req	Description
10	Checksum	String(3)	Y	Always last tag in message. Functions as end-of-message delimiter.



5. Administrative Messages

5.1. Logon (MsgType = A)

Initiates connection from client side and approves connection if sent by the exchange. ResetSeqNumFlag allows clients to start a new session (reset session sequence numbers).

The Heartbeat Interval is declared by the session initiator using the HeartBtInt field in the Logon message. The heartbeat interval timer should be reset after every message is transmitted (not just heartbeats). The HeartBtInt value should be agreed upon and specified by the Logon initiator (client) and echoed back by the Logon acceptor. The same HeartBtInt value is used by both sides, the Logon “initiator” and Logon “acceptor”.

Tag	Name	FIX Type	Req	Description
<Header>			Y	MsgType = “A”
98	EncryptMethod	Int	Y	“0” - None. Security must be guaranteed on transport level
108	HeartBtInt	Int	Y	Heartbeat interval in seconds – Allowed values are between 5s and 60s, 30s is recommended
141	ResetSeqNumFlag	Boolean	N	“N” - use previous sequences “Y” - reset sequences (start new session). If clients cannot recover the previous session they start new session with 1 and set this field to “Y”. The flag should be used with caution especially during trading session as this might lead to business data loss
553	Username		Y	Username
554	Password		Y	Password
<Trailer>			Y	

5.2. Logout (MsgType = 5)

The message initiates or confirms termination of a FIX session.

Tag	Name	FIX Type	Req	Description
<Header>			Y	MsgType = “5”
58	Text	String (200)	N	Logout reason
<Trailer>			Y	

5.3. Resend Request (MsgType = 2)

The message is used to recover an inbound session sequence if a message was missed.

Tag	Name	FIX Type	Req	Description
<Header>			Y	MsgType = "2"
7	BeginSeqNo	SeqNum	Y	Sequence number of the first message in range to be resent
16	EndSeqNo	SeqNum	Y	Sequence number of the last message in range to be resent
<Trailer>			Y	

5.4. Sequence Reset (MsgType = 4)

The message may be used in two modes:

- **Reset Mode** forces counterparty to adjust inbound message sequence, GapFillFlag = "N" or omitted.
- **Fill Gap Mode** is used during retransmission of messages missed by a client. Administrative messages and rejected business messages are not to be retransmitted. Instead a Sequence Reset message with GapFillFlag = "Y" must be used.

Tag	Name	FIX Type	Req	Description
<Header>			Y	MsgType = "4"
123	GapFillFlag	Boolean	N	"N" - sequence reset, the counterparty must adjust inbound sequence number. "Y" - indicates the message is used instead of administrative or business messages which are not to be resent
36	NewSeqNo	SeqNum	Y	New adjusted sequence number
<Trailer>			Y	

5.5. Test Request (MsgType = 1)

A Test message is useful for checking sequence numbers or verifying the communication line status. Connection participant received the message is required to reply with Heartbeat message referring to TestReqID of the initial message

Tag	Name	FIX Type	Req	Description
<Header>			Y	MsgType = "1"
112	TestReqID	String (20)	Y	Identifier to be returned in resulting Heartbeat message
<Trailer>			Y	

5.6. Heartbeat (MsgType = 0)

Used for replying to the Test request as well as checking the status of communication.

Tag	Name	FIX Type	Req	Description
<Header>			Y	MsgType = "0"
112	TestReqID	String (20)	Y	Identifier to be returned in resulting Heartbeat message
<Trailer>			Y	

5.7. Reject (MsgType = 3)

Issued by a party if an incoming FIX message is unsupported or not properly formed. Rejected messages must not be resent if a Resend Request is received; instead a SequenceReset with GapFillFlag = "Y" is expected.

Tag	Name	FIX Type	Req	Description
<Header>			Y	MsgType = "3"
45	RefSeqNum	SeqNum	Y	Sequence number of the rejected message
58	Text	String (200)	Y	Error message
373	SessionRejectReason	Int (2)	Y	Reject reason code. See Session Reject Reason Codes (tag 373).
<Trailer>			Y	



6. Order Management Messages

6.1. Overview

The FairX Order Management FIX API supports the following client-originated messages:

- NewOrderSingle message to submit new single-legged order
- OrderCancelRequest message to cancel submitted single or multi-leg order
- OrderCancel/ReplaceRequest message to replace single order

The FIX line sends Execution Report (MsgType = 8) messages to update the connected FIX client with order events:

- New order placed (ExecType = 0) – sent as a reply and referring to the NewOrderSingle messages to affirm the order is accepted in order book and working. The report is sent for every accepted order even for orders matched immediately.
- Order is cancelled (ExecType = 4) – sent as a reply to OrderCancelRequest message to acknowledge that the cancel is accepted OR an unsolicited cancel from the exchange indicating the original order is no longer working.
- Order is replaced (ExecType = 5) – sent as a reply and referring to OrderCancelReplaceRequest messages to affirm the replace is accepted. After receiving the message the original order is assumed to be replaced
- Order is rejected (ExecType = 8) – sent in case a new or cancel/replace order message is rejected
- Trade (ExecType = F) – sent whenever an order is filled completely or partially
- Done for Day (ExecType = 3) – sent for all currently open orders after the trading day closes
- Expired (ExecType = C) – sent for the expired Day orders

For rejected cancel requests the FairX API Order Cancel Reject (MsgType = 9)

For application-level rejection of any client request which cannot be reported with a well-formed 'rejected' Execution Report a Business Reject (MsgType = j) is sent



6.2. New Order Single (MsgType = D)

Used to send new single-legged orders for execution.

Tag	Name	FIX Type	Req	Description
<Header>			Y	MsgType="D"
1	Account	String(12)	Y	Unique account identifier.
11	ClOrdID	String(20)	Y	Unique order identifier assigned by client system. Client system must maintain uniqueness of this value for the life of the order.
38	OrderQty	Int(9)	Y	Order quantity. Must be a positive integer.
40	OrdType	Char(1)	Y	Order types. See order types (Tag40) Market Order types are not accepted during opening auction
44	Price	Price(20)	C	Required for limit or stop-limit orders. Designates the price per single contract unit.
54	Side	Char(1)	Y	Side of order. See Order Sides (Tag54)
55	Symbol	String(24)	Y	This tag contains the details of the instrument. Future Example: EUM20
167	SecurityType	String(6)	Y	Indicates security type. See Security Type (Tag167)
59	TimeInForce	Char(1)	N	Specifies how long the order remains in effect. If not present, DAY order is the default. See Order TimeInForce (Tag59) For FAK, MinQty can also be specified.
60	TransactTime	UTCTimest amp(21)	Y	Timestamp of the business event of submitting the order (e.g., click the submission button). UTC format YYYYMMDD-HH:MM:SS.sss
77	PositionEffect	Char(1)	N	Order position effect. See PositionEffect (Tag77) If not specified treated as "D"
99	StopPx	Price(20)	C	Stop price of the order. Required for stop and stop-limit orders.
110	MinQty	Int(9)	N	Minimum quantity of an order to be executed. This tag is used only when tag 59-TimeInForce=3 (Fill and Kill).
528	OrderCapacity	Int(1)	Y	The capacity of the firm placing the order (Agency/Principal) See OrderCapacity(Tag 528).
210	MaxShow	Qty(9)	N	Display quantity of an order to be shown in the order book at any given time.
432	ExpireDate	LocalMktD ate(8)	C	Required only if tag 59-TimeInForce=Good Till Date (GTD).
1028	ManualOrderIndi cator	Boolean(1)	Y	N' indicates the message was generated by automated trading logic. See ManualOrderIndicator (Tag1028)
1031	CustOrderHandli ngInst	String(1)	Y	Defines source of original order. See CustOrderHandlingInst (tag1031)

FairX

7928	SelfMatchPreventionID	Int(8)	C	This tag is required when market participants elect to use the optional Self Match Prevention functionality.
8000	SelfMatchPreventionStrategy	Char(1)	N	Indicates a cancel instruction when Self Match Prevention is triggered. To use tag 8000-SelfMatchPreventionStrategy, client systems must include tag 7928-SelfMatchPreventionID on the originating message See SelfMatchPreventionStrategy (tag 8000)
582	CustOrderCapacity	Char(1)	Y	Capacity of the customer. Used to indicate the Customer Type Indicator (CTI) Code. See CustOrderCapacity (tag582)
<Trailer>				End of message.

LIMIT order example:

BeginString = FIX.4.4
 BodyLength = ...
 MsgType = D
 MsgSeqNum = 3
 PossDupFlag =
 SenderCompID = EBR123
 SenderSubID = smithj
 SendingTime = 20191202-10:15:41.383
 TargetCompID = FAIRX
 TargetSubID = TEST
 Account = C123
 ClOrdID = 314bb362:109f840f9c0
 Symbol=EUM20
 SecurityType=FUT
 OrderQty = 100
 OrdType = 2
 Price = 1.10317
 Side = 1
 TimeInForce = 0
 TransactTime = 20171102-10:15:40.383
 PositionEffect = O
 CustomerOrFirm = 0
 OrderCapacity = A
 ManualOrderIndicator = Y
 CustOrderHandlingInst = Y
 SelfMatchPreventionID = 12347565
 SelfMatchPreventionStrategy = N
 CTICode = 4
 CheckSum = ...



6.3. Order Cancel Request (MsgType = F)

A firm may use this message to cancel the remaining quantity of any working single or multi-leg order.

Tag	Name	FIX Type	Req	Description
<Header>				MsgType="F"
1	Account	String(12)	Y	Unique account identifier.
11	ClOrdID	String(20)	Y	Unique order identifier assigned by client system. Client system must maintain uniqueness of this value for the life of the order.
37	OrderID	String(17)	Y	Unique ID assigned by FairX to identify orders.
41	OrigClOrdID	String(20)	N	Last accepted ClOrdID in the order chain.
54	Side	Char(1)	Y	Side of order. See Order Sides (Tag54)
55	Symbol	String(24)	Y	This tag contains the details of the instrument. Future Example: EUM20
167	SecurityType	String(6)	Y	Indicates security type. See Security Type (Tag167)
60	TransactTime	UTCTimestamp(21)	Y	Timestamp of the business event of submitting the order (e.g., click the submission button). UTC format YYYYMMDD-HH:MM:SS.sss
1028	ManualOrderIndicator	Boolean(1)	Y	N' indicates the message was generated by automated trading logic. See ManualOrderIndicator (Tag1028)
<Trailer>				End of message.

Cancel order request example:

BeginString = FIX.4.4
BodyLength = ...
MsgType = F
MsgSeqNum = 6
PossDupFlag =
SenderCompID = EBR123
SenderSubID = smithj
SendingTime = 20191202-10:15:41.383
TargetCompID = FAIRX
TargetSubID = TEST
Account = C123
ClOrdID = 65780bfc:901n328943d
OrderID = 5038
OrigClOrdID = 71fa0cda:109f847292d Side = 1
Symbol = EUM20
SecurityType = FUT
TransactTime = 20171102-10:22:24.725
ManualOrderIndicator = Y
Checksum = ...



6.4. Order Cancel/Replace Request (MsgType = G)

A firm may use this message to replace an order. The following restrictions exist:

- Triggered Stop and Stop-Limit orders cannot be replaced, only canceled
- Time in Force cannot be changed
- If a field cannot be changed, the replacing order gets rejected.

Tag	Name	FIX Type	Req	Description
<Header>				MsgType="G"
1	Account	String(12)	Y	Unique account identifier. Note: If this value is changed from the original value submitted in the New Order message, the order will lose priority in the order book.
11	ClOrdID	String(20)	Y	Unique order identifier assigned by client system. Client system must maintain uniqueness of this value for the life of the order.
37	OrderID	String(17)	Y	Unique ID assigned by FairX to identify orders.
38	OrderQty	Int(9)	Y	Order quantity. Must be a positive integer. Note: If this value is reduced from the original value submitted in the New Order message, the order will maintain priority in the order book. If this value is increased, the order will lose priority on the order book.
40	OrdType	Char(1)	Y	Order types. See order types (Tag40) Market Order types are not accepted during opening auction
41	OrigClOrdID	String(20)	N	Last accepted ClOrdID in the order chain.
44	Price	Price(20)	C	Required for limit or stop-limit orders. Designates the price per single contract unit.
54	Side	Char(1)	Y	Side of order. See Order Sides (Tag54)
55	Symbol	String(24)	Y	This tag contains the details of the instrument. Future Example: EUM20
167	SecurityType	String(6)	N	Indicates security type. See Security Type (Tag167)
59	TimeInForce	Char(1)	N	Specifies how long the order remains in effect. See Order TimeInForce (Tag59)
60	TransactTime	UTCTimest amp(21)	Y	Timestamp of the business event of submitting the order (e.g., click the submission button). UTC format YYYYMMDD-HH:MM:SS.sss
77	PositionEffect	Char(1)	N	Order position effect. See PositionEffect (Tag77) If not specified treated as "D"
99	StopPx	Price(20)	C	Stop price of the order. Required for stop and stop-limit orders. Note: If this value is changed from the original value submitted in the New Order message, the order will maintain priority in the order book.

528	OrderCapacity	Int(1)	Y	The capacity of the firm placing the order (Agency/Principal) See OrderCapacity(Tag 528).
210	MaxShow	Qty(9)	N	Display quantity of an order to be shown in the order book at any given time. Note: If this value is changed from the original value submitted in the New Order message, the order will maintain priority in the order book.
432	ExpireDate	LocalMktDate(8)	C	Required only if tag 59-TimeInForce=Good Till Date (GTD).
1028	ManualOrderIndicator	Boolean(1)	Y	N' indicates the message was generated by automated trading logic. See ManualOrderIndicator (Tag1028)
1031	CustOrderHandlingInst	String(1)	Y	Defines source of original order. See CustOrderHandlingInst (tag1031)
7928	SelfMatchPreventionID	Int(8)	C	This tag is required when market participants elect to use the optional Self Match Prevention functionality.
8000	SelfMatchPreventionStrategy	Char(1)	N	Indicates a cancel instruction when Self Match Prevention is triggered. To use tag 8000-SelfMatchPreventionStrategy, client systems must include tag 7928-SelfMatchPreventionID on the originating message See SelfMatchPreventionStrategy (tag 8000)
582	CustOrderCapacity	Char(1)	Y	Capacity of the customer. Used to indicate the Customer Type Indicator (CTI) Code. See CustOrderCapacity (tag582)
<Trailer>				End of message.

Cancel/Replace request of a LIMIT Order example:

BeginString = FIX.4.4
 BodyLength = ...
 MsgType = G
 MsgSeqNum = 15
 PossDupFlag =
 SenderCompID = EBR123
 SenderSubID = smithj
 SendingTime = 20191202-10:15:41.383
 TargetCompID = FAIRX
 TargetSubID = TEST
 Account = C123
 ClOrdID = 71fa0cda:109f847292d:-7ffe
 OrderID = 5095
 OrderQty = 150
 OrdType = 2
 OrigClOrdID = 71fa0cda:109f847292d:-7ffd
 Price = 1.10317
 Side = 1



Symbol=EUM20
SecurityType=FUT
TimeInForce = 2
TransactTime = 20191102-10:22:21.725
PositionEffect = O
OrderCapacity = A
ManualOrderIndicator = Y
CustOrderHandlingInst = Y
SelfMatchPreventionID = 12345678
SelfMatchPreventionStrategy = N
CTICode = 4
Checksum = ...



6.5. Execution Report: Accepted Order (MsgType = 8, ExecType = 0 or 5)

Reports with execution type New (0), Replace (5) represent an acknowledgement of successful acceptance of a New, Cancel/Replace order.

Tag	Name	FIX Type	Req	Description
<Header>				MsgType="8"
1	Account	String(12)	Y	Unique account identifier.
11	ClOrdID	String(20)	Y	Unique order identifier assigned by client system. Client system must maintain uniqueness of this value for the life of the order.
14	CumQty	Int(9)	Y	Contains cumulated traded quantity throughout lifespan of an order.
17	ExecID	String(40)	Y	Unique Exchange assigned execution report message identifier
37	OrderID	String(17)	Y	Unique exchange assigned order identifier
38	OrderQty	Int(9)	Y	Order quantity. Must be a positive integer.
39	OrdStatus	Char(1)	Y	Identifies order status as accepted, cancelled or replaced. See OrdStatus (tag 39)
40	OrdType	Char(1)	Y	Order types. See order types (Tag40) Market Order types are not accepted during opening auction
41	OrigClOrdID	String(20)	N	Last accepted ClOrdID in the order chain.
44	Price	Price(20)	C	Required for limit or stop-limit orders. Designates the price per single contract unit.
54	Side	Char(1)	Y	Side of order. See Order Sides (Tag54)
55	Symbol	String(24)	Y	This tag contains the details of the instrument. Future Example: EUM20
167	SecurityType	String(6)	N	Indicates security type. See Security Type (Tag167)
59	TimeInForce	Char(1)	N	Specifies how long the order remains in effect. If not present, DAY order is the default. See Order TimeInForce (Tag59) For FAK, MinQty can also be specified.
60	TransactTime	UTCTimest amp(21)	Y	Timestamp of the business event of submitting the order (e.g., click the submission button). UTC format YYYYMMDD-HH:MM:SS.sss
99	StopPx	Price(20)	C	Stop price of the order. Required for stop and stop-limit orders.
110	MinQty	Int(9)	N	Minimum quantity of an order to be executed. This tag is used only when tag 59-TimeInForce=3 (Fill and Kill).
150	ExecType	Char(1)	Y	Indicates type of execution report. See ExecType (tag 150)
151	LeavesQty	int(9)	Y	Amount of contract units open for further execution.
210	MaxShow	Qty(9)	N	Display quantity of an order to be shown in the order book at any given time.

FairX

378	ExecRestatementReason	Int(3)	N	Identifies origin of the order elimination. See ExecRestatementReason (tag 378)
432	ExpireDate	LocalMktDate(8)	C	Contains business date when order expires.
1028	ManualOrderIndicator	Boolean(1)	Y	N' indicates the message was generated by automated trading logic. See ManualOrderIndicator (Tag1028)
1031	CustOrderHandlingInst	String(1)	Y	Defines source of original order. See CustOrderHandlingInst (tag1031)
5979	RequestTime	Int(20)	N	Information carried on a response to convey the time (UTC) when the request was received by the Exchange
7928	SelfMatchPreventionID	Int(8)	C	This tag is required when market participants elect to use the optional Self Match Prevention functionality.
8000	SelfMatchPreventionStrategy	Char(1)	N	Indicates a cancel instruction when Self Match Prevention is triggered. To use tag 8000-SelfMatchPreventionStrategy, client systems must include tag 7928-SelfMatchPreventionID on the originating message See SelfMatchPreventionStrategy (tag 8000)
<Trailer>				End of message.



6.6. Execution Report: Order (MsgType = 8, ExecType = 4)

Reports with execution type Cancelled (4) represent an acknowledgement of successful Cancel, Unsolicited Cancel or Cancel/Replace order.

Tag	Name	FIX Type	Req	Description
<Header>				MsgType="8"
1	Account	String(12)	Y	Unique account identifier.
11	ClOrdID	String(20)	Y	Unique order identifier assigned by client system. Client system must maintain uniqueness of this value for the life of the order.
14	CumQty	Int(9)	Y	Contains cumulated traded quantity throughout lifespan of an order.
17	ExecID	String(40)	Y	Unique Exchange assigned execution report message identifier
37	OrderID	String(17)	Y	Unique exchange assigned order identifier
38	OrderQty	Int(9)	Y	Order quantity. Must be a positive integer.
39	OrdStatus	Char(1)	Y	Identifies order status as accepted, cancelled or replaced. See OrdStatus (tag 39)
40	OrdType	Char(1)	Y	Order types. See order types (Tag40) Market Order types are not accepted during opening auction
41	OrigClOrdID	String(20)	N	Last accepted ClOrdID in the order chain.
44	Price	Price(20)	C	Required for limit or stop-limit orders. Designates the price per single contract unit.
54	Side	Char(1)	Y	Side of order. See Order Sides (Tag54)
55	Symbol	String(24)	Y	This tag contains the details of the instrument. Future Example: EUM20
167	SecurityType	String(6)	N	Indicates security type. See Security Type (Tag167)
59	TimeInForce	Char(1)	N	Specifies how long the order remains in effect. If not present, DAY order is the default. See Order TimeInForce (Tag59) For FAK, MinQty can also be specified.
60	TransactTime	UTCTimest amp(21)	Y	Timestamp of the business event of submitting the order (e.g., click the submission button). UTC format YYYYMMDD-HH:MM:SS.sss
99	StopPx	Price(20)	C	Stop price of the order. Required for stop and stop-limit orders.
150	ExecType	Char(1)	Y	Indicates type of execution report. See ExecType (tag 150) 150=4 Cancelled
151	LeavesQty	int(9)	Y	Amount of contract units open for further execution.
210	MaxShow	Qty(9)	N	Display quantity of an order to be shown in the order book at any given time.
378	ExecRestatement Reason	Int(3)	N	Identifies origin of the order elimination. See ExecRestatementReason (tag 378)

FairX

432	ExpireDate	LocalMktDate(8)	C	Contains business date when order expires.
1028	ManualOrderIndicator	Boolean(1)	Y	N' indicates the message was generated by automated trading logic. See ManualOrderIndicator (Tag1028)
1031	CustOrderHandlingInst	String(1)	Y	Defines source of original order. See CustOrderHandlingInst (tag1031)
5979	RequestTime	Int(20)	N	Information carried on a response to convey the time (UTC) when the request was received by the Exchange
7928	SelfMatchPreventionID	Int(8)	C	This tag is required when market participants elect to use the optional Self Match Prevention functionality.
8000	SelfMatchPreventionStrategy	Char(1)	N	Indicates a cancel instruction when Self Match Prevention is triggered. To use tag 8000-SelfMatchPreventionStrategy, client systems must include tag 7928-SelfMatchPreventionID on the originating message See SelfMatchPreventionStrategy (tag 8000)
<Trailer>				End of message.



6.7. Execution Report: Stop Triggered (MsgType = 8, ExecType = L)

Sent when a stop order is triggered.

Tag	Name	FIX Type	Req	Description
<Header>				MsgType="8"
1	Account	String(12)	Y	Unique account identifier.
11	ClOrdID	String(20)	Y	Unique order identifier assigned by client system. Client system must maintain uniqueness of this value for the life of the order.
14	CumQty	Int(9)	Y	Contains cumulated traded quantity throughout lifespan of an order.
17	ExecID	String(40)	Y	Unique Exchange assigned execution report message identifier
37	OrderID	String(17)	Y	Unique exchange assigned order identifier
38	OrderQty	Int(9)	Y	Order quantity. Must be a positive integer.
39	OrdStatus	Char(1)	Y	Identifies order status as accepted, cancelled or replaced. See OrdStatus (tag 39)
40	OrdType	Char(1)	Y	Order types. See order types (Tag40) Market Order types are not accepted during opening auction
41	OrigClOrdID	String(20)	N	Last accepted ClOrdID in the order chain.
44	Price	Price(20)	C	Required for limit or stop-limit orders. Designates the price per single contract unit.
54	Side	Char(1)	Y	Side of order. See Order Sides (Tag54)
55	Symbol	String(20)	Y	This tag contains the details of the instrument. Future Example: EUM20
167	SecurityType	String(6)	N	Indicates security type. See Security Type (Tag167)
59	TimeInForce	Char(1)	N	Specifies how long the order remains in effect. If not present, DAY order is the default. See Order TimeInForce (Tag59) For FAK, MinQty can also be specified.
60	TransactTime	UTCTimestamp(21)	Y	Timestamp of the business event of submitting the order (e.g., click the submission button). UTC format YYYYMMDD-HH:MM:SS.sss
99	StopPx	Price(20)	C	Stop price of the order. Required for stop and stop-limit orders.
150	ExecType	Char(1)	Y	Indicates type of execution report. See ExecType (tag 150)
151	LeavesQty	int(9)	Y	Amount of contract units open for further execution.
210	MaxShow	Qty(9)	N	Display quantity of an order to be shown in the order book at any given time.
378	ExecRestatement Reason	Int(3)	N	Identifies origin of the order elimination. See ExecRestatementReason (tag 378)
432	ExpireDate	LocalMktDate(8)	C	Contains business date when order expires.

FairX

1028	ManualOrderIndicator	Boolean(1)	Y	N' indicates the message was generated by automated trading logic. See ManualOrderIndicator (Tag1028)
1031	CustOrderHandlingInst	String(1)	Y	Defines source of original order. See CustOrderHandlingInst (tag1031)
5979	RequestTime	Int(20)	N	Information carried on a response to convey the time (UTC) when the request was received by the Exchange
7928	SelfMatchPreventionID	Int(8)	C	This tag is required when market participants elect to use the optional Self Match Prevention functionality.
8000	SelfMatchPreventionStrategy	Char(1)	N	Indicates a cancel instruction when Self Match Prevention is triggered. To use tag 8000-SelfMatchPreventionStrategy, client systems must include tag 7928-SelfMatchPreventionID on the originating message See SelfMatchPreventionStrategy (tag 8000)
<Trailer>				End of message.



6.8. Execution Report: Trade (MsgType = 8, ExecType = F)

Sent by the FairX system to report order fill

Tag	Name	FIX Type	Req	Description
<Header>				MsgType=8
1	Account	String(12)	Y	Unique account identifier.
11	ClOrdID	String(20)	Y	Unique order identifier assigned by client system. Client system must maintain uniqueness of this value for the life of the order.
14	CumQty	Int(9)	Y	Contains cumulated traded quantity throughout lifespan of an order.
17	ExecID	String(40)	Y	Unique exchange assigned identifier of the execution
880	TrdMatchID	String (20)	Y	Unique exchange identifier to identify both sides of a match
31	LastPx	Price(20)	Y	Price at which order was filled.
32	LastQty	Int(9)	Y	Quantity filled.
37	OrderID	String(17)	Y	Unique exchange assigned order identifier
38	OrderQty	Int(9)	C	Quantity of order.
39	OrdStatus	Char(1)	Y	Indicates order status. See OrderStatus (tag 39) "1" - Partial Fill "2" - Complete Fill
40	OrdType	Char(1)	Y	Order types. See order types (Tag40)
41	OrigClOrdID	String(20)	N	Last accepted ClOrdID in the order chain.
44	Price	Price(20)	C	Price per single contract unit. All Execution Report messages sent in response to Market orders (with protection) and Stop orders (with protection) will include tag 44-Price populated with the Protection Price Limit (best available price +/- the protection points). If the order is not completely filled, the remaining Open Quantity will rest on the order book at the Protection Price Limit. Note: For spread trade Execution Reports, this tag is sent in the Execution Report – Fill Notice (35=8, 39=1 or 2) for the spread only and not the legs of the spread.
6	AvgPx	Price(20)	C	Calculated average price of all fills on this order
54	Side	Char(1)	Y	Side of order. See Order Sides (Tag54)
55	Symbol	String(24)	Y	This tag contains the details of the instrument. Future Example: EUM20
167	SecurityType	String(6)	N	Indicates security type. See Security Type (Tag167)
59	TimeInForce	Char(1)	N	Specifies how long the order remains in effect. If not present, DAY order is the default. See Order TimeInForce (Tag59)

60	TransactTime	UTCTimeStamp(21)	Y	UTC format YYYYMMDD-HH:MM:SS.sss
75	TradeDate	LocalMktDate(8)	Y	Indicates date of trade reference in this message in YYYYMMDD format.
150	ExecType	Char(1)	Y	Indicates type of execution report. See ExecType (tag 150). Always "F" for Trade
151	LeavesQty	Int(9)	C	Amount of contracts remaining for execution after this fill.
393	TotalNumSecurities	Int(3)	N	Contains the number of Leg Fill Acknowledgment messages sent with the spread summary. Sent for spread fill message only.
442	MultiLegReportingType	Int(1)	N	Indicates if acknowledgment message is sent for an outright, leg of spread or spread. See MultiLegReporting (tag 442)
527	SecondaryExecID	String(40)	C	Exchange assigned unique identifier; can be used to link spread summary fill notice with leg fill notice and trade cancel messages.
1028	ManualOrderIndicator	Boolean(1)	Y	Value sent on inbound message from client system indicating the order as sent manually or generated by automated trading logic.
1031	CustOrderHandlingInst	String(1)	Y	Defines source of original order. See CustOrderHandlingInst (tag1031)
1057	AggressorIndicator	Char(1)	C	Indicates if order was incoming or resting for the match event. See AggressorIndicator (tag 1057)
5979	RequestTime	Int(20)	N	Information carried on a response to convey the time (UTC) when the request was received by Exchange UTC format YYYYMMDD-HH:MM:SS.sss
<Trailer>				End of message.



6.9. Execution Report: Order is Rejected (MsgType=8, ExecType = 8)

Message notifies client system of a rejected order.

Tag	Name	FIX Type	Req	Description
<Header>				MsgType=8
1	Account	String(12)	Y	Unique account identifier.
11	ClOrdID	String(20)	Y	Unique order identifier assigned by client system. Client system must maintain uniqueness of this value for the life of the order.
14	CumQty	Int(9)	Y	Contains cumulated traded quantity throughout lifespan of an order.
17	ExecID	String(40)	Y	Unique exchange assigned identifier of the execution
37	OrderID	String(17)	Y	Unique exchange assigned order identifier
38	OrderQty	Int(9)	C	Quantity of order.
39	OrdStatus	Char(1)	Y	Indicates order status. See OrderStatus (Tag39) "8"=Rejected
40	OrdType	Char(1)	Y	Order types. See order types (Tag40)
41	OrigClOrdID	String(20)	N	Last accepted ClOrdID in the order chain.
44	Price	Price(20)	C	Price per single contract unit.
54	Side	Char(1)	Y	Side of order. See Order Sides (Tag54)
55	Symbol	String(20)	Y	This tag contains the details of the instrument. Future Example: EUM20
167	SecurityType	String(6)	N	Indicates security type. See Security Type (Tag167)
58	Text	String(200)	N	Text explanation of error code. See OrdRejReason (tag 103)
59	TimeInForce	Char(1)	N	Specifies how long the order remains in effect. If not present, DAY order is the default. See Order TimeInForce (Tag59)
60	TransactTime	UTCTimeStamp(21)	Y	UTC format YYYYMMDD-HH:MM:SS.sss
75	TradeDate	LocalMktDate(8)	Y	Indicates date of trade reference in this message in YYYYMMDD format.
103	OrdRejReason	Int(6)	N	Error code. See OrdRejReason (tag 103)
150	ExecType	Char(1)	Y	Type of execution report. See ExecType(tag 150) 8=Reject Ack
151	LeavesQty	Int(9)	C	Amount of contracts remaining for execution. Always 0 for reject
1028	ManualOrderIndicator	Boolean(1)	Y	Value sent on inbound message from client system indicating the order as sent manually or generated by automated trading logic.
1031	CustOrderHandlingInst	String(1)	Y	Defines source of original order. See CustOrderHandlingInst (tag1031)
<Trailer>				End of message.



6.10. Execution Report: Done for Day (MsgType = 8, ExecType = 3)

Reports with execution type 'Done for Day' (3) are sent by the exchange to the firms that need the status of their open orders after the trading session is closed.

Tag	Name	FIX Type	Req	Description
<Header>				MsgType=8
1	Account	String(12)	Y	Unique account identifier.
11	ClOrdID	String(20)	Y	Unique order identifier assigned by client system. Client system must maintain uniqueness of this value for the life of the order.
14	CumQty	Int(9)	Y	Contains cumulated traded quantity throughout lifespan of an order.
17	ExecID	String(40)	Y	Unique exchange assigned identifier of the execution
37	OrderID	String(17)	Y	Unique exchange assigned order identifier
38	OrderQty	Int(9)	C	Quantity of order.
39	OrdStatus	Char(1)	Y	Indicates order status. See OrderStatus (Tag39) "3"=Done for Day
40	OrdType	Char(1)	Y	Order types. See order types (Tag40)
41	OrigClOrdID	String(20)	N	Last accepted ClOrdID in the order chain.
44	Price	Price(20)	C	Price per single contract unit.
54	Side	Char(1)	Y	Side of order. See Order Sides (Tag54)
55	Symbol	String(24)	Y	This tag contains the details of the instrument. Future Example: EUM20
167	SecurityType	String(6)	N	Indicates security type. See Security Type (Tag167)
59	TimeInForce	Char(1)	N	Specifies how long the order remains in effect. If not present, DAY order is the default. See Order TimeInForce (Tag59)
60	TransactTime	UTCTimeStamp(21)	Y	UTC format YYYYMMDD-HH:MM:SS.sss
75	TradeDate	LocalMktDate(8)	Y	Indicates date of trade reference in this message in YYYYMMDD format.
150	ExecType	Char(1)	Y	Indicates type of execution report. See ExecType (tag 150). 3="Done for Day"
151	LeavesQty	Int(9)	C	Amount of contracts remaining for execution
1028	ManualOrderIndicator	Boolean(1)	Y	Value sent on inbound message from client system indicating the order as sent manually or generated by automated trading logic.
1031	CustOrderHandlingInst	String(1)	Y	Defines source of original order. See CustOrderHandlingInst (tag1031)
5979	RequestTime	Int(20)	N	Information carried on a response to convey the time (UTC) when the request was received by Exchange UTC format YYYYMMDD-HH:MM:SS.sss
<Trailer>				End of message.



6.11. Execution Report: Expired (MsgType = 8, ExecType = C)

Reports with execution type Expired (C) are sent by the exchange to the firms that need to track the expiration of their time-limited orders (such as DAY).

Tag	Name	FIX Type	Req	Description
<Header>			Y	MsgType="8"
1	Account	String(12)	Y	Unique account identifier.
11	ClOrdID	String(20)	Y	Unique order identifier assigned by client system. Client system must maintain uniqueness of this value for the life of the order.
14	CumQty	Int(9)	Y	Contains cumulated traded quantity throughout lifespan of an order.
17	ExecID	String(40)	Y	Unique exchange assigned identifier of the execution
37	OrderID	String(17)	Y	Unique exchange assigned order identifier
38	OrderQty	Int(9)	Y	Order quantity submitted by client.
39	OrdStatus	Char(1)	Y	Indicates order status. See OrdStatus (tag 39). "C" = Expired
40	OrdType	Char(1)	Y	Order types. See order types (Tag40)
41	OrigClOrdID	String(20)	N	Last accepted ClOrdID in the order chain.
44	Price	Price(20)	N	Price per unit of quantity.
54	Side	Char(1)	Y	Side of order. See Order Sides (Tag54)
55	Symbol	String(20)	Y	This tag contains the details of the instrument. Future Example: EUM20
167	SecurityType	String(6)	N	Indicates security type. See Security Type (Tag167)
59	TimeInForce	Char(1)	N	Specifies how long the order remains in effect. If not present, DAY order is the default. See Order TimeInForce (Tag59)
60	TransactTime	UTCTimest amp(21)	N	UTC format YYYYMMDD-HH:MM:SS.sss e.g. 20091216-19:21:41.109
110	MinQty	Int(9)	C	Sent in the Order Elimination message if the originating order was an FAK/FOK/MinQty that contained tag 110-MinQty.
150	ExecType	Char(1)	Y	Indicates type of execution report. See ExecType (tag 150). Always "C" = Elimination Ack
151	LeavesQty	Int(9)	Y	Quantity available for further execution. Always 0
1028	ManualOrderIndicator	Boolean(1)	Y	Value sent on inbound message from client system indicating the order as sent manually or generated by automated trading logic.
1031	CustOrderHandlingInst	String(1)	Y	Defines source of original order. See CustOrderHandlingInst (tag1031)
7928	SelfMatchPreventionID	Int(8)	N	This tag is required when market participants elect to use the optional Self Match Prevention functionality.
8000	SelfMatchPreventionStrategy	Char(1)	N	This tag is required when market participants elect to use the optional Self Match Prevention functionality.

FairX

5979	RequestTime	Int(20)	N	Information carried on a response to convey the time (UTC) when the request was received by Exchange UTC format YYYYMMDD-HH:MM:SS.sss
<Trailer>				End of message.



6.12. Order Cancel Reject (MsgType = 9)

Sent by the FairX system to notify the client on rejection of an Order Cancel Request OR Order Cancel/Replace.

Tag	Name	FIX Type	Req	Description
<Header>				MsgType="9"
1	Account	String(12)	Y	Unique account identifier
11	ClOrdID	String(20)	Y	Unique order identifier assigned by client system. Client system must maintain uniqueness of this value for the life of the order.
37	OrderID	String(17)	Y	Unique exchange assigned order identifier
39	OrdStatus	Char(1)	Y	Indicates order status. See OrdStatus (tag 39). "U" = Undefined
41	OrigClOrdID	String(20)	N	Last accepted ClOrdID in the order chain. If a value is included in tag 41 on order entry, the same value is returned. If not, the tag will contain '0'.
55	Symbol	String(24)	Y	This tag contains the details of the instrument. Future Example: EUM20
58	Text	String(200)	N	Text reason order cancel was rejected.
60	TransactTime	UTCTimest amp(21)	Y	UTC format YYYYMMDD-HH:MM:SS.sss e.g. 20091216-19:21:41.109
102	CXlRejReason	Int(6)	N	Cancel reject reason. See CXRejReason (tag 102)
434	CancelRejRespon seTo	Char(1)	Y	Identifies the type of request reject message is in response to. 1=Order Cancel Request 2=Order Cancel/Replace Request
5979	RequestTime	Int(20)	N	Information carried on a response to convey the time (UTC) when the request was received by the MSGW application. UTC Timestamps are sent in number of nanoseconds since Unix epoch synced to a master clock to microsecond accuracy.
<Trailer>				End of message.

6.13. Last ExecId Request (MsgType = F1)

Send this message to request the execId of the last (most recent) event sent by the trading system to this user/session. Can be used to determine if the client missed any events while disconnected. Also serves as a means of validating that the trading system is available and accepting requests.

Tag	Name	FIX Type	Req	Description
<Header>				MsgType="F1"
35	MsgType	String(2)	Y	Header tag identifying message type.
<Trailer>				End of message.

6.14. Last ExecId (MsgType = F2)

Sent in response to LastExecIdRequest.

Tag	Name	FIX Type	Req	Description
<Header>				MsgType="F2"
35	MsgType	String(2)	Y	Header tag identifying message type.
45	RefSeqNum	SeqNum(10)	Y	MsgSeqNum <34> of request message
17	ExecID	String(40)	Y	ExecId of last event sent to this user.
<Trailer>				End of message.

6.15. Event Resend Request (MsgType = F3)

Send this message to request order events in the specified range be resent. Since this is an application-level request, resent messages will have new sequence numbers and PossDupFlag (tag 43) will not be set; rather, PossResend (tag 97) will be set. Rejects (and any other message that does not contain an execId) will not be resent.

Tag	Name	FIX Type	Req	Description
<Header>				MsgType="F3"
35	MsgType	String(2)	Y	Header tag identifying message type.
22003	BeginExecId	String(40)	Y	Lower bound (inclusive) of execIds.
22004	EndExecId	String(40)	N	Upper bound (inclusive) of execIds. Resend all events up the last known event if not set.
<Trailer>				End of message.

6.16. Event Resend Complete (MsgType = F4)

Sent in response to a successful Event Resend Request following all resent events.

Tag	Name	FIX Type	Req	Description
<Header>				MsgType="F4"
35	MsgType	String(2)	Y	Header tag identifying message type.
45	RefSeqNum	SeqNum(10)	Y	MsgSeqNum <34> of request message
22005	ResentEventCount	Int	Y	Total number of events resent.
<Trailer>				End of message.



6.17. Event Resend Reject (MsgType = F5)

Sent in response to an Event Resend Request if the request cannot be fulfilled.

Tag	Name	FIX Type	Req	Description
<Header>				MsgType="F5"
35	MsgType	String(2)	Y	Header tag identifying message type.
45	RefSeqNum	SeqNum(10)	Y	MsgSeqNum <34> of request message
22006	EventResendReject Reason	Int	Y	Code identifying reject reason: 1 = BEGIN_EXEC_ID_TOO_SMALL 2 = END_EXEC_ID_TOO_LARGE 3 = RESEND_ALREADY_IN_PROGRESS 4 = TOO_MANY_RESEND_REQUESTS 5 = SERVER_ERROR
58	Text	String(200)	N	Free format text string.
<Trailer>				End of message.

6.18. Business Reject (MsgType = j)

Sent by the FairX system to notify the client of application-level rejection of client request which passed session-level validation but has business validation issues and cannot be reported with a well-formed Execution Report with Rejected status.

Tag	Name	FIX Type	Req	Description
<Header>				MsgType="8"
35	MsgType	String(2)	Y	Header tag identifying message type.
45	RefSeqNum	SeqNum(10)	Y	MsgSeqNum <34> of rejected message
58	Text	String(200)	N	Free format text string.
379	BusinessRejectRefID	String(32)	N	Value of business-level "ID" of the message being rejected.
380	BusinessRejectReason	Int(2)	Y	Code identifying reject reason 0=Other 1=Unknown ID 2=Unknown security 3=Unsupported message type 4=Application not available 5=Conditionally required field missing 6=Not authorized 7=Delivery to firm not available at this time
<Trailer>				End of message.

Market Data

7. Security Definition Messages

7.1. Security Definition (MsgType = d)

Tag	Name	FIX Type	Req	Description
35	MsgType	String	Y	35=d: Sent when contract states are updated
980	SecurityUpdate Action	Char	Y	Included in the message on the Incremental feed when a mid-week deletion or modification (i.e. extension) occurs. 980=A: Add represents Security Definition messages that are: Newly added during the current week Disseminated during the Sunday Startup period Resent by the system during the week 980=M: Modify represents modifications to a Security Definition 980=D: Delete represents deletions of a Security Definition
779	LastUpdateTime	UTCTimestamp	Y	Timestamp of when the instrument was last added, modified or deleted. UTC Timestamps are sent in number of nanoseconds since Unix epoch synced to a master clock to microsecond accuracy.
55	Symbol	String	Y	Instrument Name or Symbol.
167	SecurityType	String	C	See Security Type (Tag167)
762	SecuritySubType	String	C	See SecuritySubType (Tag762) Represents spread strategy if applicable
107	SecurityDesc	String	Y	Instrument Name
200	MaturityMonthYear	Month-year	C	This field provides the calendar month reflected in the instrument symbol (tag 55-Symbol in MDP 3.0; tag 107-SecurityDesc). Format YYYYMM (e.g., 201912) For futures spreads and options spreads, this field contains the first leg's calendar month reflected in the instrument symbol.
6937	Asset	String	Y	String field that indicates the underlying asset code (Product Code). Example: EU (Euro/USD FX Rate). Product Code was previously communicated in tag 1151-SecurityGroup.
461	CFIcode	String	Y	ISO standard instrument categorization code.
462	UnderlyingProduct	Int	Y	2=Commodity/Agriculture 4=Currency 5=Equity 12=Other 14=Interest Rate 15=FX Cash 16=Energy 17=Metals

1682	MDSecurityTradingStatus	Int	N	21 = Pre Open 27 = Pre Open (No Cancel) 17 = Ready to Trade 2 = Trading Halt 4 = Close 99 = Pause 18 = Expired 98 = Forbidden
15	Currency	Currency	Y	Identifies currency used for price.
120	SettlCurrency	Currency	C	Identifies currency used for settlement price, if different from trade price currency.
562	MinTradeVol	Qty	Y	The minimum trading quantity for a security.
1140	MaxTradeVol	Qty	Y	The maximum trading quantity for a security.
969	MinPriceIncrement	Float	Y	Minimum constant tick for the instrument.
864	NoEvents	NuminGroup	Y	Number of repeating EventType entries. Indicates number of repeating groups and length of each repeating group in the message.
→865	EventType	Int	Y	Code to represent the type of event. 5=Activation 7=Last eligible trade date
→1145	EventTime	String	Y	Trading Session Date of event. Format yyyyMMdd
996	UnitOfMeasure	String	C	Unit of measure for the product's original contract size. This will be populated for all products listed on FairX. USD unit of measure can be in U.S. dollars or cents.
1147	UnitOfMeasureQty	Float	C	This field contains the contract size for each instrument. Use in combination with tag 996-UnitofMeasure. For example: Euro futures -Tag 1147=10000 -Tag 996=Euro For variable-quantity products, the contract size reflects the original contract size, before the application of the multiplier
1149	HighLimitPrice	Price	N	Allowable high limit price for the trading day. A key parameter in validating order price. Used as the upper band for validating order prices. Orders submitted with prices above the upper limit will be rejected. This price protects off prices for quoting. Note: This value is indicative only and may not reflect the actual real-time high limit price.
1148	LowLimitPrice	Price	N	Allowable low limit price for the trading day. A key parameter in validating order price.

				Used as the lower band for validating order prices. Orders submitted with prices below the lower limit will be rejected. This price protects off prices for quoting. Note: This value is indicative only and may not reflect the actual real-time low limit price.
1143	MaxPriceVariation	Price	C	Differential value for price banding.
5796	TradingReferenceDate	LocalMktDate	C	Date of current trading session
734	PriorSettlPrice	Price	C	Prior Settlement Price (Settlement price of the prior trading session day)
730	SettlPrice	Price	C	Settlement Price
731	SettlPriceType	Int	C	Type of settlement price. The value of 2 (Theoretical) will be sent during the initial seeding of a new contract. This tag is sent in conjunction with Tag 734 (PriorSettlPrice)
555	NoLegs	NuminGroup	N	Number of legs that make up security (for spreads)
→ 600	LegSymbol	String	N	Symbol of underlying leg
→ 620	LegSecurityDesc	String	N	Name of underlying leg
→ 609	LegSecurityType	String	N	Security type of underlying leg
→ 624	LegSide	Char(1)	N	Side of underlying leg
→ 608	LegCFIcode	String	N	ISO standard instrument categorization code of underlying leg
→ 556	LegCurrency	Currency	N	Currency of underlying leg

7.2. SecurityListRequest

Tag	Name	FIX Type	Req	Description
35	MsgType	String	Y	35=x Full instrument contact state snapshot request
320	SecurityReqID	String	Y	Unique security request ID
559	SecurityListRequestType	String	Y	0 = Symbol
55	Symbol	String	Y	55=NA get all securities
167	SecurityType	String	C	See Security Type (Tag167) Not required if 55=NA

7.3. SecurityList

Tag	Name	FIX Type	Req	Description
35	MsgType	String	Y	35=y
320	SecurityReqID	String	Y	Unique security request ID
322	SecurityResponseID	String	Y	Unique response ID
560	SecurityRequestResult	String	Y	0 = ValidReq 1 = InvalidReq
393	TotNoRelatedSym	int	N	Total number of securities to be returned for this request
893	LastFragment	String	N	N = Not Last message Y = Last Message for SecurityListRequest
146	NoRelatedSym	int	Y	number of securities in the list
55	Symbol	String	Y	instrument code
167	SecurityType	String	Y	See Security Type (Tag167)
762	SecuritySubType	String	C	See SecuritySubType (Tag762) Represents spread strategy if applicable
107	SecurityDesc	String	Y	Instrument Name
200	MaturityMonthYear	Month-year	C	This field provides the calendar month reflected in the instrument symbol (tag 55-Symbol in MDP 3.0; tag 107-SecurityDesc). Format YYYYMM (e.g., 201912) For futures spreads and options spreads, this field contains the first leg's calendar month reflected in the instrument symbol.
6937	Asset	String	Y	String field that indicates the underlying asset code (Product Code). Example: EU (Euro/USD FX Rate) Product Code was previously communicated in tag 1151-SecurityGroup.
461	CFICode	String	Y	ISO standard instrument categorization code.
462	UnderlyingProduct	Int	Y	2=Commodity/Agriculture 4=Currency 5=Equity 12=Other 14=Interest Rate 15=FX Cash 16=Energy 17=Metals

1682	MDSecurityTradingStatus	Int	N	21 = Pre Open 27 = Pre Open (No Cancel) 17 = Ready to Trade 2 = Trading Halt 4 = Close 99 = Pause 18 = Expired 98 = Forbidden
15	Currency	Currency	Y	Identifies currency used for price.
120	SettlCurrency	Currency	C	Identifies currency used for settlement price, if different from trade price currency.
562	MinTradeVol	Qty	Y	The minimum trading quantity for a security.
1140	MaxTradeVol	Qty	Y	The maximum trading quantity for a security.
969	MinPriceIncrement	Float	Y	Minimum constant tick for the instrument.
864	NoEvents	NuminGroup	Y	Number of repeating EventType entries. Indicates number of repeating groups and length of each repeating group in the message.
→865	EventType	Int	Y	Code to represent the type of event. 5=Activation 7=Last eligible trade date
→1145	EventTime	String	Y	Trading Session Date of event. Format yyyyMMdd
231	ContractMultiplier	Int	C	Number of deliverable units per instrument, e.g., peak days in expiration month or number of calendar days in expiration month. The market data Security Definition (tag 35-MessageType=d) message for the variable quantity spread will be populated with the value '0' for tag 231-ContractMultiplier. The market data Security Definition (tag 35-MessageType=d) message is populated with values for the outright legs for tag 231-ContractMultiplier and customers must extract this value.
996	UnitOfMeasure	String	C	Unit of measure for the product's original contract size. This will be populated for all products listed on FairX. USD unit of measure can be in U.S. dollars or cents.
1147	UnitOfMeasureQty	Float	C	This field contains the contract size for each instrument. Use in combination with tag 996-UnitofMeasure. For example: For example: Euro futures -Tag 1147=10000 -Tag 996=Euro For variable-quantity products, the contract size reflects the original contract size, before the application of the multiplier

1149	HighLimitPrice	Price	N	Allowable high limit price for the trading day. A key parameter in validating order price. Used as the upper band for validating order prices. Orders submitted with prices above the upper limit will be rejected. This price protects off prices for quoting. Note: This value is indicative only and may not reflect the actual real-time high limit price.
1148	LowLimitPrice	Price	N	Allowable low limit price for the trading day. A key parameter in validating order price. Used as the lower band for validating order prices. Orders submitted with prices below the lower limit will be rejected. This price protects off prices for quoting. Note: This value is indicative only and may not reflect the actual real-time low limit price.
1143	MaxPriceVariation	Price	C	Differential value for price banding.
5796	TradingReferenceDate	LocalMktDate	C	Date of current trading session
734	PriorSettlPrice	Price	C	Prior Settlement Price (Settlement price of the prior trading session day)
730	SettlPrice	Price	C	Settlement Price
731	SettlPriceType	Int	C	Type of settlement price. The value of 2 (Theoretical) will be sent during the initial seeding of a new contract. This tag is sent in conjunction with Tag 734 (PriorSettlPrice)
555	NoLegs	NuminGroup	N	Number of legs that make up security (for spreads)
→600	LegSymbol	String	N	Symbol of underlying leg
→620	LegSecurityDesc	String	N	Name of underlying leg
→609	LegSecurityType	String	N	Security type of underlying leg
→624	LegSide	Char(1)	N	Side of underlying leg
→608	LegCFIcode	String	N	ISO standard instrument categorization code of underlying leg
→556	LegCurrency	Currency	N	Currency of underlying leg

8. Market Data Messages

8.1. MarketDataRequest

Tag	Name	FIX Type	Req	Description
35	MsgType	String	Y	35=V
262	MDReqID	String	Y	Unique ID assigned by client for Market Data Request To unsubscribe, the same ID must be sent with 263 = 2
263	SubscriptionRequest Type	int	Y	1 = Subscribe for snapshots and updates 2 = Unsubscribe
146	NoRelatedSym	NumInGroup	N	Number of securities to subscribe for If not specified, subscribes to all securities
→55	Symbol	String	N	instrument code
→167	SecurityType	String	N	See Security Type (Tag167)

8.2. MarketDataRequestReject

Tag	Name	FIX Type	Req	Description
35	MsgType	String	Y	35=Y
262	MDReqID	String	Y	Unique ID assigned by client for Market Data Request
281	MDReqRejReason	String	N	Numerical reason for rejection of Market Data Request
58	Text	String	N	Free format text string describing reason for rejection

8.3. MarketDataSnapshotFullRefresh

Tag	Name	FIX Type	Req	Description
35	MsgType	String	Y	35=W
262	MDReqID	String	Y	Unique ID assigned by client for Market Data Request
55	Symbol	String	Y	instrument symbol
167	SecurityType	String	Y	See Security Type (Tag167)
762	SecuritySubType	String	C	See SecuritySubType (Tag762) Represents spread strategy if applicable
268	NoMDEntries	int	Y	Number of entries in group
→ 269	MDEntryType	int	Y	See MD Entry Types (tag 269)
→ 278	MDEntryID	String	N	Market data identifier MDEntryID is unique on a per instrument, session, and entry type basis An active entry with the same MDEntryID (278) should be replaced
→ 270	MDEntryPx	Price	N	The all-in rate
→ 271	MDEntrySize	Qty	N	Entry size/quantity

→ 110	MinQty	Qty	N	Minimum fill size associated with the amount
→ 60	TransactTime	UTCTimest amp(24)	Y	GMT time of event, in microseconds
→ 1023	MDPriceLevel	int	N	1 or 2. Price level in book (implies only)
→ 286	OpenCloseSettl Flag	Int	C	Flag describing Open Price entry 0=Daily Open Price 5=Indicative Opening Price (IOP)
→ 5796	TradingReferen ceDate	LocalMktD ate	C	Trading session date corresponding to the MDEntryType. This value will have the current trading date except for open interest and closing price (where it will usually be the previous trading date)

8.4. MarketDataIncrementalRefresh

Tag	Name	FIX Type	Req	Description
35	MsgType	String	Y	35=X
262	MDReqID	String	Y	Unique ID assigned by client for Market Data Request To unsubscribe, the same ID must be sent with 263 = 2
266	AggregatedBook	char	Y	N = Non-aggregated market data
268	NoMDEntries	int	Y	Number of entries in group
→ 279	MDUpdateActio n	int	Y	0 = New 1 = Change 2 = Delete
→ 269	MDEntryType	int	Y	See MD Entry Types (tag 269)
→ 278	MDEntryID	String	Y	Market data identifier MDEntryID is unique on a per instrument, session, and entry type basis MDUpdateAction = 0/1 (New/Change) should replace any active entry with the same MDEntryID (278) MDUpdateAction = 2 (Delete) indicates any active entry with same MDEntryID should be deleted
→ 55	Symbol	String	Y	instrument symbol
→ 167	SecurityType	String	Y	See Security Type (Tag167)
→ 762	SecuritySubTyp e	String	C	See SecuritySubType (Tag762) Represents spread strategy if applicable
→ 270	MDEntryPx	Price	N	The all-in rate Required when MDUpdateAction = New (0) or Change (1)
→ 271	MDEntrySize	Qty	N	Required when MDUpdateAction = New (0) or Change (1)

FairX

→ 110	MinQty	Qty	N	Minimum fill size associated with the amount
→ 60	TransactTime	UTCTimest amp(24)	Y	GMT time of event, in microseconds
→ 1023	MDPriceLevel	int	N	1 or 2. Price level in book (implies only)
→ 286	OpenCloseSettl Flag	Int	C	Flag describing Open Price entry 0=Daily Open Price 5=Indicative Opening Price (IOP)
→ 5796	TradingReferenc eDate	LocalMktD ate	C	Trading session date corresponding to the MDEntryType. This value will have the current trading date
→ 5797	AggressorSide	Int	C	Indicates the aggressor side. During certain periods (Pre-open, Halt, etc.) trades can occur without aggressors 0=No Aggressor 1=Buy 2=Sell This value is sent for 269=2 (trades)



9. Tag Values

9.1. Order Status (tag 39)

Value	Description
0	New
1	Partially filled
2	Filled
3	Done for day
4	Cancelled
5	Modify
8	Rejected
U	Undefined
C	Expired

9.2. Order Types (tag 40)

Value	Description
1	Market
2	Limit
3	Stop
4	Stop Limit

9.3. Order Sides (tag 54 and tag 624)

Value	Description
1	Buy Side
2	Sell Side

9.4. Order Time-in-force (tag 59)

Value	Description
0	Day
1	Good Till Cancel (GTC)
3	Immediate Or Cancel (IOC)
4	Fill or Kill (FOK)

9.5. Position Effect (tag 77)

Value	Description
O	Open
C	Close
D	Default

9.6. Cancel Reject Reason (tag 102)

Value	Description
0	Too late to cancel
1	Unknown order

9.7. Order Reject Reason (tag 103)

Value	Description
1	Unknown symbol
2	Exchange closed
6	Duplicate order
18	Invalid price increment (submitted price precision exceeds the one supported for the instrument)
99	Other

9.8. Execution Types (tag 150)

Value	Description
0	New order Ack
3	Done for Day
4	Cancel Ack
5	Modify Ack
8	Reject Ack
F	Trade
L	Stop Trigger
H	Trade Cancel Ack
C	Elimination Ack



9.9. Security Types (tag 167 and 609)

Value	Description
FUT	Future
OPT	Option

9.10. MD Entry Types (tag 269)

Value	Description
0	Bid
1	Offer
2	Trade
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	Trading Session Traded Volume
C	Open Interest
E	Implied Bid
F	Implied Offer
J	Empty Book

9.11. Session Reject Reason Codes (tag 373)

Value	Description
0	Invalid tag number
1	Required tag missing
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
9	CompID problem
99	Other



9.12. ExecRestatementReason (tag 378)

Value	Description
8	Exchange
100	Cancel on Disconnect
103	Cancel Oldest (resting) due to Self Match Prevention
104	Cancel from Exchange Credit Controls violation
105	Cancel from Exchange Website
106	Cancel from Risk Management API
107	Cancel Newest (Aggressing) due to Self Match Prevention
108	Cancel due to resting order quantity less than min lot size
109	Cancel both due to Self Match Prevention

9.13. MultiLegReporting (tag 442)

Value	Description
1	Outright
2	Leg of Spread
3	Spread

9.14. OrderCapacity (tag 528)

Identifies the origin of the order (the capacity of the firm placing the order).

Value	Description
A	Agency. Order placed by customer
P	Principal. Order placed by the firm

9.15. CustOrderCapacity (tag 582)

Customer Type Indicator as defined by the NFA

Value	Description
1	CTI 1: Transactions initiated and executed by an individual member for his/her own account, for an account he/she controls, or for an account in which he/she has ownership or financial interest.
2	CTI 2: Transactions executed for the proprietary account of a clearing member or non-clearing member firm.
3	CTI 3: Transactions where an individual member or authorized trader executes for the personal account of another individual member, for an account the other individual member controls or for an account in which the other individual member has ownership or financial interest.
4	CTI 4: Any transaction not meeting the definition of CTI 1, 2 or 3. (These should be non-member customer transactions).

9.16. SecuritySubType (tag 762)

Value	Description
SP	Standard Calendar Spreads

9.17. CustomerOrderHandlingInst (tag 1031)

Value	Description
W	Desk
Y	Electronic (Default)
C	Vendor provided platform Billed by Executing Broker
G	Sponsored Access via exchange API
H	Premium Algorithmic trading provided by executing broker
D	Other

9.18. AggressorIndicator (tag 1057)

Value	Description
Y	Match aggressor
N	Resting at match

9.19. Manual or Automated flow (tag 1028)

Value	Description
Y	Manual
N	Automated

9.20. SelfMatchPreventionStrategy (tag 8000)

Value	Description
N	Cancel aggressing order
O	Cancel resting order
Q	Cancel both aggressing and resting orders