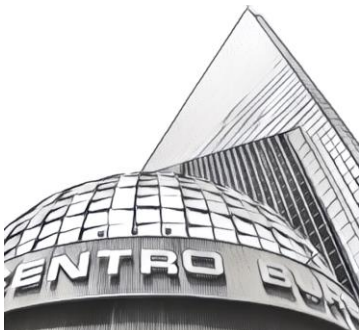


# MexDer

## New MexFix Trading Specifications



Version: 1.2

Last update: 2018-04-24

## Change Log

Version	Author	Change	Date
1.0	Francisco Javier Ibarra C DGA Tecnología Grupo BMV	First official version	March 2017
1.1	Jose A Gonzalez G DGA Tecnología Grupo BMV	Tag Text (58) was limited up 5 characters when is used to indicate "Reference"  Reject codes added to Tag BusinessRejectReason (380) in BusinessMessageReject (j) message.	June 2017
1.2	Jose A Gonzalez G Francisco Javier Ibarra C DGA Tecnología Grupo BMV	Tag TradeIDRed(20127) was added to Execution Report message (8) to indicate the trade registration number of the corresponding strategy trade. Applies only to the execution report for leg trades.	April 2018

## COMMENTS AND SUGGESTIONS

*Please forward any comment or suggestion to [ftsc@grupobmv.com.mx](mailto:ftsc@grupobmv.com.mx)*

Information contained in this document is subject to change without prior notice. The companies, names and data used in the examples are not real, unless otherwise stated. You may not reproduce or transmit any part of this document in any way, nor by any means, either electronic or mechanic, for any purpose, without MexDer's express written consent.

## Content

<b>Change Log</b>	<b>2</b>
<b>Content</b>	<b>3</b>
1. Introduction	7
1.1. Purpose	7
1.2. Target	7
1.3. Conventions	7
1.4. Related Documents	8
2. Implementation	9
2.1. Description	9
2.2. Ignored tags	9
2.3. Unsupported tags	9
2.4. String Type Length	9
2.5. Message Maximum Length	9
2.6. Encryption	9
2.7. FIX Protocol	9
3. FIX Session	10
3.1. Application Level Synchronization	10
3.2. List of Messages	11
3.2.1. Standard Header	11
3.2.2. Standard Message Trailer	12
3.2.3. Logon (MsgType = A)	12
3.2.4. Logout (MsgType = 5)	12
3.2.5. HeartBeat (MsgType = 0)	13
3.2.6. TestRequest (MsgType = 1)	13
3.2.7. ResendRequest (MsgType = 2)	13
3.2.8. SequenceReset (MsgType = 4)	14

3.2.9. Reject (MsgType = 3)	14
3.3. FIX 4.4 Delimitations and Adaptations	16
3.4. FIX Session Identification	17
4. General Conventions for Application Messages	18
4.1. Order Identification	18
4.1.1. ClOrdID in Orders Generated by the Trader	18
4.1.1. ClOrdID in Orders Generated by the Exchange (Unsolicited Orders)	18
4.1.1. OrderID	19
4.2. Trade Identification	19
4.2.1. ExecID	19
4.2.2. SecondaryExecID	19
4.3. Party Block	19
5. Generic Messages at Application Level	20
5.1. Introduction	20
5.2. Business Message Reject (MessageType = j)	20
6. Trading Application Messages	21
6.1. Order Entry (New Orders)	21
6.1.1. Description	21
6.1.2. Supported Order Types	22
6.1.3. New Order Message Flow	22
6.1.3.1. New Order Accepted by MexFix and the Trading Engine	22
6.1.3.2. New Order Rejected	23
6.1.3.3. New Order Accepted and Canceled by the Engine	23
6.2. Order Modification	24
6.2.1. Description	24
6.2.2. Order Modification Message Flow	25
6.2.2.1. Order Modification Accepted	25
6.2.2.1. Order Modification Rejected	26
6.2.2.2. Order Canceled by an order modification	26
6.3. Order Cancelation	28
6.3.1. Description	28

6.3.2.	Order Cancellation Message Flow	28
6.3.2.1.	Order Cancellation Accepted	28
6.3.2.2.	Order Cancellation Rejected	28
6.4.	Mass Order Cancellation	29
6.4.1.	Description	29
6.4.2.	Status of massive cancellation request	29
6.4.3.	ClOrdID Tag	29
6.4.4.	Mass Cancel Order Flow	30
6.4.4.1.	Mass Cancel Request Accepted	30
6.4.4.2.	Mass Cancel Request Rejected	30
6.5.	Order Status	31
6.5.1.	Description	31
6.5.1.1.	Query Status about a specific order	31
6.5.1.2.	Query Status about a set of orders	32
6.5.2.	Order Status Message Flow	32
6.5.2.1.	Order Status	32
6.5.2.2.	Order Mass Status	33
6.6.	Order Acknowledges	33
6.7.	Trade Notification	33
6.8.	Message Definitions	34
6.8.1.	NewOrderSgls (MsgType = D)	34
6.8.2.	OrderCancelReplaceRequest (MsgType = G)	36
6.8.3.	OrderCancelRequest (MsgType = F)	38
6.8.4.	ExecutionReport (MsgType = 8)	39
6.8.5.	OrderCancelReject (MsgType = 9)	44
6.8.6.	OrderStatusRequest (MsgType = H)	45
6.8.7.	OrderMassStatusRequest (MsgType = AF)	46
6.8.8.	OrderMassCancelRequest (MsgType = q)	47
6.8.9.	OrderMassCancelReport (MsgType = r)	48
7.	Cross Trades	49
7.1.	Introduction	49

7.2.	Cross Orders _____	49
7.3.	Cross Executions _____	49
7.4.	CrossID _____	49
7.5.	Cross Order Message Flow _____	50
7.5.1.	CrossOrder Accepted _____	50
7.5.2.	CrossOrder Rejected _____	51
7.6.	Message Definition _____	52
7.6.1.	NewCrossOrder (MsgType = s) _____	52
8.	Event Notification _____	54
9.	BlockTrades _____	54
9.1.	Introduction _____	54
9.2.	Block Trade Acceptance _____	54
9.3.	BlockTrade Message Flow _____	55
9.3.1.	BlockTrade Accepted by Buyer and the Exchange _____	55
9.3.2.	BlockTrade Declined (by the buyer) _____	56
9.3.3.	BlockTrade Rejected by the Exchange _____	57
9.4.	Message Definition _____	58
9.4.1.	Trade Capture Report (MsgType = AE) Sent by Members _____	58
9.4.2.	Trade Capture Report Ack (MsgType = AR) _____	60
9.4.3.	Trade Capture Report (MsgType = AE) Sent by the Exchange _____	61
10.	Appendix _____	63
10.1.	Appendix A _____	63

## 1. Introduction

### 1.1. Purpose

The present document quotes the specification and flow of messages of the interface that is relevant to the functionality of MexFix. This solution is based on the FIX (Financial Information eXchange) protocol standard, version 4.4.

### 1.2. Target

This document is addressed to those members willing to connect through FIX to MexDer.

### 1.3. Conventions

Any message related to the solution is duly specified in its entirety. Certain MexFix aspects differ from the FIX standard; therefore, tables specifying the tags contained in each message include information like the following:

Column	Meaning
Tag	Field Number. Fields added to the message by MexFix show an asterisk ("*") following this number.
Name	Field name according to FIX standard.
Req	"Y" means the field is required; "N" means the field is optional. "Y*" means the field is required in MexFix implementation, but optional in FIX standard 4.4.
Valid values	Field valid values in the context of the message. It may be a list of values or a range of numeric values, for instance ">=3, <= 10". This column also contains the field default value, for the optional fields that may require it.  To avoid confusions with the terms, in the values related to codes the FIX original value has been kept and therefore it has not been translated.
Format	Type of field data. This is one of the types defined by FIX, or one of those types with some additional restriction. String(n) is a String type with a maximum of n characters, or in some cases with exactly n characters. If you need more information on the String type please refer to point 2.4.
Description	Field description in the context of the message.
Timestamp data types	For all timestamp data types, the valid format will be either: <ul style="list-style-type: none"> <li>- YYYYMMDD-HH:MM:SS (whole seconds) or</li> <li>- YYYYMMDD-HH:MM:SS.sss (milliseconds)</li> </ul>

## 1.4. Related Documents

#	Title	Author	Version
1	Financial Information Exchange Protocol (FIX) 4.4 with errata 20030618	FIX Committee	June 18, 2003
2	Layout MexFix EIS English version 2 3.docx (For MarketData Specs)	MexDer	October, 2012
3	Coding Tables	MexDer	



## **2. Implementation**

### **2.1. Description**

Given the fact that some modifications were made to FIX standard, the limitations implemented are detailed below:

### **2.2. Ignored tags**

In some cases, MexFix may ignore the content of some tags of the input messages. If this is the case, it is clearly explained in the tags description in each message specification.

### **2.3. Unsupported tags**

Unsupported tags in a message have not been included in its description.

Messages sent to MexFix should not contain unsupported tags. Messages sent by MexFix never contain unsupported fields.

Unsupported tags will be ignored and MexDer will not return them back in the acknowledgement.

### **2.4. String Type Length**

FIX standard imposes no maximum length restriction on the String type. In the implementation carried out by MexFix, the maximum length for this type has been set to 255 characters.

In some fields, a lower maximum length of this value has been set. In these cases, the type is presented as String(n), where “n” is the maximum number of field characters. Sometimes “n” is the exact length of the field, in such case, it shall be explicitly mentioned in the “Valid Values” column.

### **2.5. Message Maximum Length**

The maximum length of messages sent or received by MexFix is up to 4096 bytes.

### **2.6. Encryption**

MexFix does not use the encryption defined by FIX standard (by means of SecureData and SecureDataLen fields in the heading of the message).

### **2.7. FIX Protocol**

FIX protocol's version 4.4 is the only one that may be used with the implementation of MexFix.

### **3. FIX Session**

MexFix complies with the specifications in FIX standard 4.4 as far as Session level is concerned. In this way, this paragraph will show only the new messages layout due to certain MexFix delimitations. Some other differences, arising out of the protocol adaptation made by MexFix, will be shown afterwards. To find more details as to the way in which sessions are established and aspects related to FIX session level, it is necessary to refer to the relevant FIX documentation which is not rewritten in order to avoid any doubt about it.

#### **3.1. Application Level Synchronization**

Session synchronization will be achieved according to the FIX Protocol specification.

Messages coming from an explicit request of repetition (requested with a Resend Request message, as shown in session 3.2.7) will contain the “Y” value in the field PossDupFlag stating such situation.

Missing or pending messages, if any, received by the engine from the trading engine (while the session was disconnected), will be delivered immediately after the logon and synchronization.

The group of messages mentioned in this section corresponds with the following type of messages:

- Execution Report with ExecType with New (“0”), Replace (“5”), Cancelled (“4”) and Trade (“F”, “G” y “H”) values.
- Execution Report with ExecType with Rejected (“8”) value that informs about the rejection of an application on behalf of the MexDer’s trading engine.

## 3.2. List of Messages

### 3.2.1. Standard Header

This header should be included in all incoming or outgoing messages.

Tag	Name	Req	Valid Values	Format	Description
8	BeginString	Y	FIX 4.4	String	It indicates the beginning of a new message. It contains the FIX protocol version. It is always the first field of the message.
9	BodyLength	Y		int	Message length in bytes, from the end of this field until, and including, the previous limit character of the CheckSum field. It is always the second field of the message.
35	MsgType	Y	All message types supported by FIX	String	It identifies the type of message. It is always the third field of the message.
49	SenderCompID	Y		String	Identifier of the entity that sends the message. It has to contain the client code in the messages sent by the client's application.  Messages sent by the exchange to the client will contain: MEXFIX.
56	TargetCompID	Y		String	Identifier of the entity to whom the message is targeted. It contains the client code in the messages sent by MexFix.  Messages sent by the client to the exchange will contain: MEXFIX.
34	MsgSeq Num	Y		int	Sequence number of the message within the current FIX session.
50	SenderSubID	Y*	For further detail regarding market codes, see Table 17 in the "Coding Tables" document.	String	The messages sent by MexFix contain the code assigned to the Market to which the connection was established. In messages sent to MexFix it must contain the trader code with which the FIX session was started.
57	TargetSubID	Y*	For further detail regarding market codes, see Table 17 in the "Coding Tables" document.	String	The messages sent by MexFix contain the trader code to whom it is addressed. In messages sent to MexFix it must contain the Market code with which the connection was established.
43	PossDupFlag	N	N = Sending of the original message (default value) Y = Possibly duplicated	Boolean	It indicates if it is the first time, within the FIX session, that a message is sent ( "N") or if it is sending again the same message ("Y"), because of an explicit request on the other behalf or because there is a doubt about the

					reception of the original message.
52	SendingTime	Y		UTC Timestamp	Sending time of the message.
122	OrigSendingTime	N		UTC Timestamp	Sending time of the original message. Required in a resending. A message is considered a resending if the field PossDupFlag = "Y" and if the MsgType field is not a "4" (SequenceReset).

### 3.2.2. Standard Message Trailer

This trailer should be included in all incoming or outgoing messages and calculated according de FIX Specification.

Tag	Name	Req	Valid Values	Format	Description
10	Checksum	Y		String(3)	Message Checksum, computed as it is described in the standards. It is always the last field of the message and its length is 3 bytes accurately.

### 3.2.3. Logon (MsgType = A)

The Logon message is used by the client to start a session and to be accepted by the gateway.

Tag	Name	Req	Valid Values	Format	Description
	Standard Header	Y	MsgType = A		
98	EncryptMethod	Y	0 = None	int	Ignored by MexFix
108	HeartBtInt	Y	60	int	Sending interval of the connection verification message (Heartbeat message) expressed in seconds.
553	Username	Y*		String	User identifier assigned by MexDer.
554	Password	Y*		String	Password assigned by MexDer.
	Standard Trailer	Y			

### 3.2.4. Logout (MsgType = 5)

The Logout message is used by both parties whether to request or notify the end of the communication session or to accept such request.

Tag	Name	Req	Valid Values	Format	Description
	Standard Header	Y	MsgType = 5		
58	Text	N		String	Explanatory text
	Standard Trailer	Y			

### 3.2.5. HeartBeat (MsgType = 0)

The Heartbeat message is used by both parties to indicate that the connection remains active.

Tag	Name	Req	Valid Values	Format	Description
	Standard Header	Y	MsgType = 0		
112	TestReqID	N		String	If the message is the response to a Test Request message, it may contain the same value than the original TestReqID field.
	Standard Trailer	Y			

### 3.2.6. TestRequest (MsgType = 1)

The Test Request message is used by both parties to request the sending of the Heartbeat message.

Tag	Name	Req	Valid Values	Format	Description
	Standard Header	Y	MsgType = 1		
112	TestReqID	Y		String	Petition identifier. It has to be included in the Heartbeat message answer.
	Standard Trailer	Y			

### 3.2.7. ResendRequest (MsgType = 2)

The Resend Request message can be used by both parties to request the resent of messages that have not been received.

Tag	Name	Req	Valid Values	Format	Description
	Standard Header	Y	MsgType = 2		
7	BeginSeqNo	Y	Valid sequence number	int	Sequence number of the first message of the rank of messages of which the resent is requested. It must contain a lower value than the last sequence number received.
16	EndSeqNo	Y	0 = Infinite number of valid sequence	int	Sequence number of the last message of the rank of messages of which the resent is requested. It must contain a lower value than the last number of the received sequence. If the request is only of one message EndSeqNo = BeginSeqNo. If the request is of all messages from a given one EndSeqNo = 0
	Standard Trailer	Y			

### 3.2.8. SequenceReset (MsgType = 4)

The Sequence reset message is used by both parties to fill in the blank spaces in the messages that are being sent, through the re-assignment of the sequence number.

Tag	Name	Req	Valid Values	Format	Description
	Standard Header	Y	MsgType = 4		Have in mind that PossDupFlag must contain the "Y" value
123	GapFillFlag	Y*	Y = It indicates that the message is to fill in a blank space	Boolean	For further information consult the document with the FIX 4.4 specifications
36	NewSeqNo	Y		int	Sequence number of the message which will be sent
	Standard Trailer	Y			

### 3.2.9. Reject (MsgType = 3)

The Reject message is used by MexFix for rejecting a message that does not comply with the FIX protocol specified in this document.

Tag	Name	Req	Valid Values	Format	Description
	Standard Header	Y	MsgType = 3		
45	RefSeqNum	Y		int	Sequence number of the rejected message
371	RefTagID	N		int	The tag number of the FIX field being referenced.
372	RefMsgType	N		int	The MsgType (35) of the FIX message being referenced.
373	SessionRejectReason	N	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	int	It is the code that indicates the reason of rejection

			7= Decryption problem  8 = Signature problem  9 = CompID problem  10 = SendingTime accuracy problem  11= Invalid MsgType  13 = Tag appears more than once  14 = Tag specified out of required order  15 = Repeating group fields out of order  16 = Incorrect NumInGroup count for repeating group  17 = Non "data" value includes field delimiter (SOH character)  99 = Other		
58	Text	N		String	It contains a more specific description of the reason of rejection
	Standard Trailer	Y			

### **3.3. FIX 4.4 Delimitations and Adaptations**

- When a request to start a session (Logon message) is rejected, MexFix will always send a Logout message as an answer.
- Fields SenderSubID and TargetSubID in the heading of messages (Standard Message Header) are required.
- PossDupFlag tag in incoming messages is not supported. All messages labeled with PossDupFlag tag will be dropped.
- FIX encryption method is not supported.
- ResetSeqNumFlag tag in Logon messages is not supported.
- Username and Password tags are required in Logon message.
- The field NextExpectedMsgSeqNum (789) of the Logon message (A) is not supported. The sequence synchronization mechanism must be executed through the Resend Request message ("2") as was always made by the FIX protocol.
- MexDer uses a heartbeat interval of 60 seconds.
- Valid formats for Timestamp data types are:
  - YYYYMMDD-HH:MM:SS (whole seconds)
  - YYYYMMDD-HH:MM:SS.sss (milliseconds)



### **3.4. FIX Session Identification**

Once a communication session has been established, MEXFIX identifies the associated FIX session using four tags in the Logon message sent by the initiator:

- SenderCompID
- SenderSubID
- TargetCompID
- TargetSubID

SenderCompID identifies the member and SenderSubID identifies the trader. TargetCompID together with TargetSubID identify the market.

No more than one FIX session can exist at a time with the same values.

The SenderCompID, SenderSubID, TargetCompID and TargetSubID tags are present in all the FIX messages. All the messages belonging to the same FIX session must have the same values in these tags (according to the FIX Specification for the inbound and outbound messages).

## 4. General Conventions for Application Messages

### 4.1. Order Identification

#### 4.1.1. ClOrdID in Orders Generated by the Trader

Any message related to an order (new, cancellation, modification) sent by the client should contain a unique identifier in the tag ClOrdID. As stated in the standard, the uniqueness of these identifiers should be maintained during the trading session.

When MexFix accepts a message for the first time, the client receives an acknowledging message with the same ClOrdID.

ClOrdID field assigned by the client must be up to 10-character length. Leading and trailing spaces will be removed if any.

#### 4.1.1. ClOrdID in Orders Generated by the Exchange (Unsolicited Orders)

When an order is created, modified or cancelled by the Exchange, and assigned to a specific member, a special ClOrdID will be generated.

This ID will follow the below specification:

**ECO|BROnnnnnn**

Where:

- ECO|BRO – Internal identifier. Could be either ECO or BRO.
- Nnnnnn – Numeric value.

*This kind of order is identified by specifying the tag SolicitedFlag tag with value "N".*

In this case the client is responsible for handling properly this unsolicited order and to create the proper data structures to support order created by the exchange.

### 4.1.1. OrderID

The OrderID field is an order identifier, unique per trading day, assigned by the MexDer 's trading engine.

## 4.2. Trade Identification

### 4.2.1. ExecID

The ExecID field is not a trade identifier. It is an identifier assigned to each Execution Report message, without duplicates during the whole FIX session.

### 4.2.2. SecondaryExecID

The SecondaryExecID tag contains the trade registration number. This is the code assigned by the trading engine to the trade. This value is unique per trading day.

## 4.3. Party Block

Party Block is used in several application messages to identify the parties involved in the trading operation.

The detailed definition of the messages containing this block incorporates the block as shown below: The list of possible values is restricted according to the message particular characteristics.

Tag	Name	Req	Valid Values	Format	Description
	Start <Parties>				
453	NoPartyIDs	N	1, 2	NumInGroup	
448	PartyID	Y		String(40)	Member code assigned by MexDer or Operator name.
447	PartyIDSource	Y	D = Proprietary/ Custom code	char	Indicates the coding followed in the PartyID field
452	PartyRole	Y	13 = Order Origination Firm ( <i>Member code</i> )  118 = Operator ( <i>The person who has the capabilities and authorization to take certain actions</i> )	int	Indicates the role taken by the referenced party in the PartyID field.  If NoPartyIDs = 1, PartyRole should be 13 and PartyID should contain the member code.  If NoPartyIDs = 2, are expected two repeating groups. One with PartyRole = 13 and PartyID with the member code; and one with PartyRole = 118 and the PartyID with an additional identifier related to responsible person (operator).

## 5. Generic Messages at Application Level

### 5.1. Introduction

This chapter deals with messages at the application level covering: message rejection by MexFix.

### 5.2. Business Message Reject (MessageType = j)

The BusinessMessageReject message will be sent when a non-defined message is received.

BusinessReject messages sent to MEXFIX will be ignored.

Tag	Name	Req	Valid Values	Format	Description
	Standard Header	Y	MsgType = j		
45	RefSeqNum	Y		int	Sequence number of the rejected message
372	RefMsgType	Y		String	MsgType of rejected message.
379	BusinessRejectRefID	N		String	Optional identifier from rejected message.
380	BusinessRejectReason	Y	0 = Other  2 = Unknown Security  3 = Unsupported Message Type  5 = Conditionally required field missing  6 = Not Authorized	int	Code of rejection.
58	Text	N		String	It contains a more specific description of the reason of rejection
	Standard Trailer	Y			

## 6. Trading Application Messages

Order management functionality groups several functions. From the point of view of a FIX client, the following are the functions involved:

- Order entry
- Order modification
- Order cancellation
- Massive order cancellation
- Request and reception of order status reports
- Cross Orders
- Block Trades
- Unsolicited orders and trades

Each one of these functions is dealt in a separate paragraph of this chapter. Method of use, list of related messages, message flow and additions or delimitations applied to this implementation are described for every function involved.

Everything described in this chapter is valid for both, simple or strategy contracts.

### 6.1. Order Entry (New Orders)

#### 6.1.1. Description

The FIX client uses this functionality to submit orders to the trading engine.

Once an order has been accepted, it may be modified, cancelled or executed. These subjects are dealt with more detail in other paragraphs of this chapter.

In addition to the habitual tags of an order like price or quantity, the client may include a free text (up to 5 chars) in the Text field. This text remains related to the order and is included in the Execution Report messages related to such order. **The information received in this tag is sent to the back office.**

Within MexDer trading engine, every order is related to an account (assigned by MexDer). FIX clients must include this information in the Account tag. If, when entering a new order, the account is not specified, the order is rejected. An account may be modified during its life-cycle in the way described in section [Order Modification](#).

There are several tags related to order identification. If you need more information refer to paragraph [Order Identification](#).

### 6.1.2. Supported Order Types

When sending an order, the order type is specified by the combination of the OrdType and TimeInForce tags. The below table, show the allowed combinations.

Order Type	OrdType	TimeInForce
Limit	2 (Limit)	0 (DAY)
Limit (Immediate or Cancel)	2 (Limit)	3 (IOC)
At Settlement Price	1 (Market)	7 (At the Close)

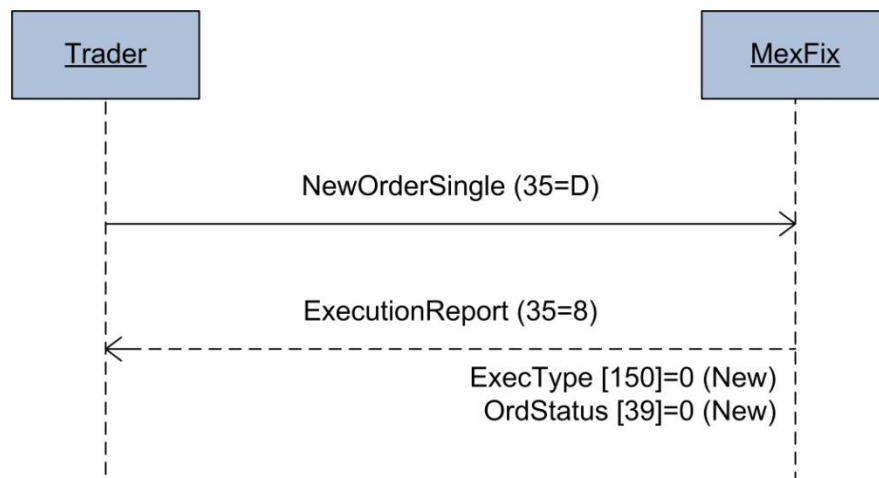
Any other combination of those tags will be rejected.

### 6.1.3. New Order Message Flow

In the diagrams below, values appearing after “Execution Report” correspond to the ExecType and OrdStatus tags, respectively.

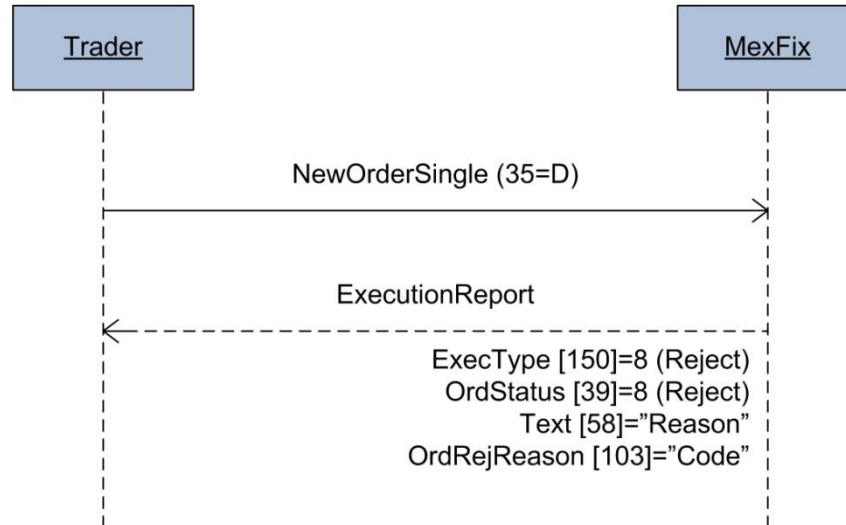
#### 6.1.3.1. New Order Accepted by MexFix and the Trading Engine

This picture belongs to the typical Order Entry case.



### 6.1.3.2. New Order Rejected

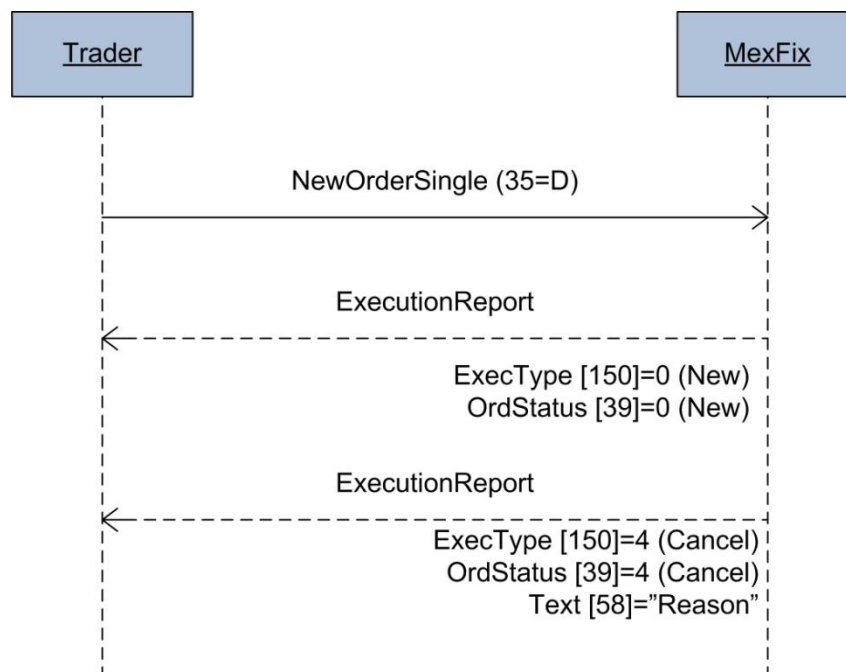
This case belongs to the typical rejection of a NewOrderSingle message.



### 6.1.3.3. New Order Accepted and Canceled by the Engine

Under some circumstances the order is fully accepted and acknowledged but immediately rejected by the engine. Possible reasons are:

- Attempting to trade with itself, without “auto-trade” permission.
- New orders with type IOC (Immediate or Cancel) without any passive order to trade with.



## 6.2. Order Modification

### 6.2.1. Description

Once an order has been sent to the market as a New Order and before its being fully filled or canceled, some of its attributes may be modified.

The following are the modifiable attributes of an order:

- Account
- Quantity
- Price
- Text (reference to the order assigned by client)

The modification request is made by means of the OrderCancelReplaceRequest message also called Order Modification Request message.

In each modification message a unique ClOrdID tag should be specified, like in a new order entry message. The order to be modified is identified by the OrigClOrdID tag.

When a modification request is accepted, the ClOrdID provided replaces the one in the OrigClOrdID tag as the active order.

The order identifier, OrderID, is not affected and maintains its value after the modification **unless a change in position occurs**.

As a rule, the tags specified in the modification request message, replaces the tags of the order being modified. Tags not specified are not modified.

In addition to the ClOrdID tag and the tags with values to be modified, the FIX standard requires the presence of some redundant tags: Symbol, Side, Order Type and HandlInst. These tags should be present in the modification request message with the same values as the order to be modified. If any of these values differ, the request is rejected with an OrderCancelReject.



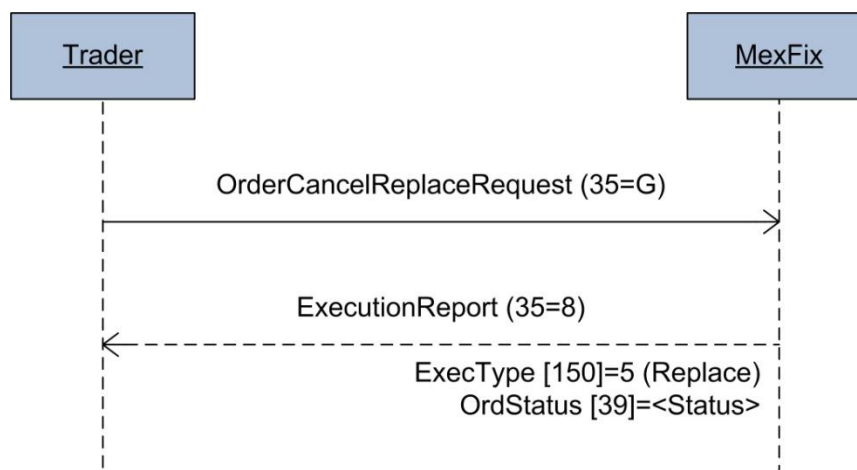
The next table shows the allowed operations for Order Modifications:

			Allowed	Position Changed
Change	Price	Worst	No	-
		Improved	Yes	Yes
	Quantity	Decrease	Yes	No
		Increase	No	-
	Reference	Change	Yes	No
	Account	Change	Yes	No

## 6.2.2. Order Modification Message Flow

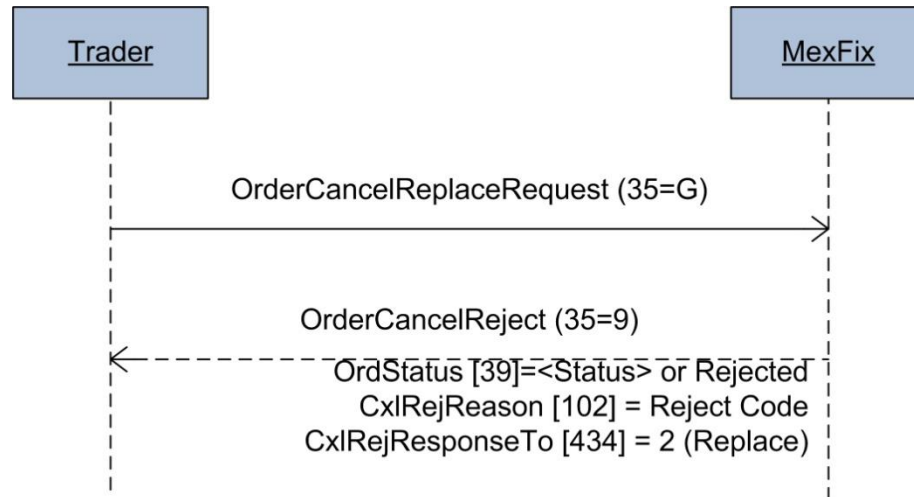
In the diagrams below, values appearing after “Execution Report” correspond to the ExecType and OrdStatus tag, respectively. When OrdStatus is shown as “<status>” it refers to the current status of the order.

### 6.2.2.1. Order Modification Accepted



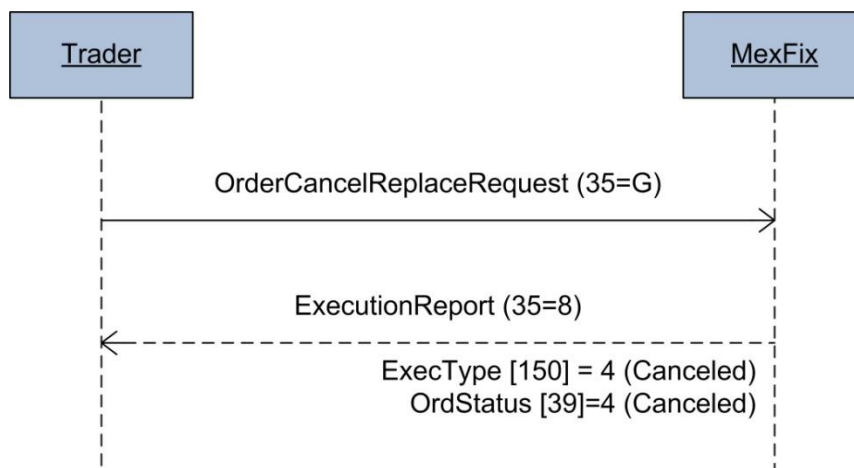
### 6.2.2.1. Order Modification Rejected

The status “Rejected” in the OrdStatus tag could be sent in case that an Order Modification is attempted on a non-existent order or when the order modification is not allowed due business rules



### 6.2.2.2. Order Canceled by an order modification

Some times during an order modification, the operation generates an order cancelation. For example when a modification is sent and, right before the modification gets in the engine, the order is partially executed, so at the time when the modification is processed by the engine if the LeavesQty becomes less or equal to zero then the order is canceled immediately by the engine.



Example:

	Buyer	Bid (Open Qty)	Ask (Open Qty)	Seller
1	New Buy IPC DC15, 100@44000	100 @ 44000		
2			70@44000	New Sell IPC DC15, 70 @ 44000
3	Trade (fill) 70 @ 44000	30 @ 44000		Trade (fill) 70 @ 44000
4	Modify Buy IPC DC15, 20@44000			

The last modification will result in an order cancelation in the buy side due the order of 100lots were partially executed with 70 lots (30 lots left) and a modification is attempted to decrease from 100 to 20 lots The engine executes the following operation in an order modification:  $\text{OrderModificationQty} = (\text{FilledQty} + \text{OpenQty})$ ; the quantity must be greater than zero If the client wants to maintain the order active in the market..

In this example the result of this operation in the buy side is:  $\text{OpenQty} = \text{OrderModificationQty} - \text{FilledQty}$ ;  $\text{OpenQty} = 20 - 70 = -50$ , which does not comply with the premise of must be greater than zero.

## 6.3. Order Cancellation

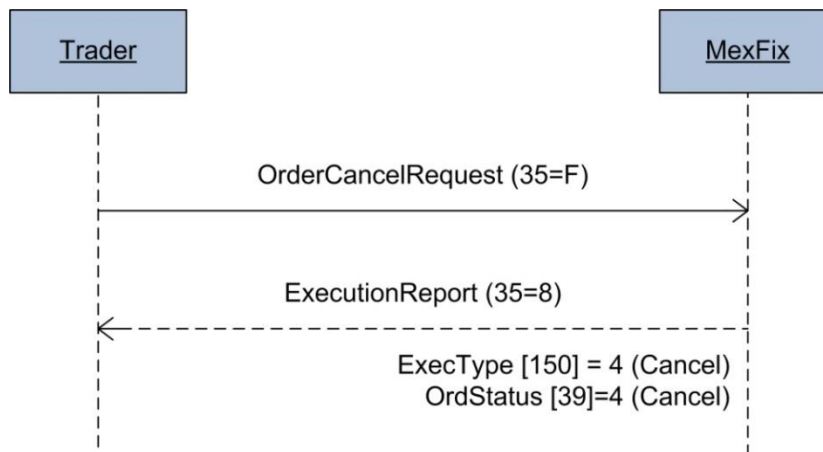
### 6.3.1. Description

Once an order has been sent to the market and is not fully filled, the order may be canceled.

### 6.3.2. Order Cancellation Message Flow

In the diagrams below, values appearing after “Execution Report” correspond to the ExecType and OrdStatus fields, respectively. When OrdStatus is shown as “<status>” it refers to the current status of the order.

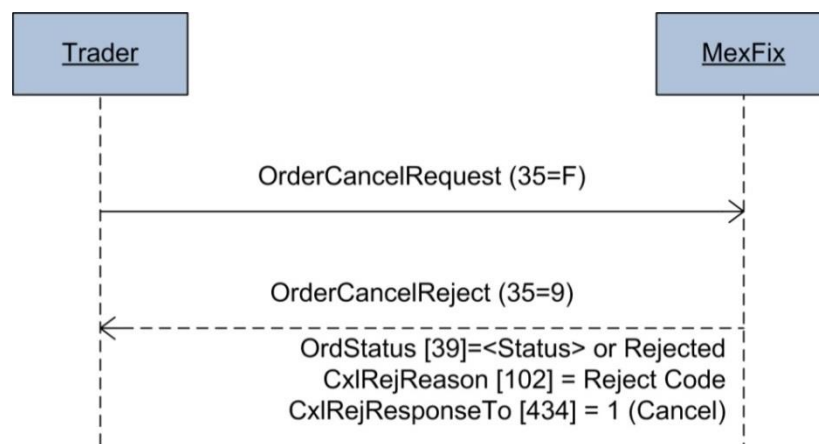
#### 6.3.2.1. Order Cancellation Accepted



#### 6.3.2.2. Order Cancellation Rejected

When an OrderCancel message is sent to the market, and it cannot be processed by the engine, a rejection will be sent.

The status “Rejected” in the OrdStatus tag could be sent in case that an Order Cancellation is attempted on a non-existent order.



## 6.4. Mass Order Cancellation

### 6.4.1. Description

This function allows the simultaneous cancellation of all orders for a fix connection.

The selection criteria are based in the tag MassCancelRequestType according to this:

Tag Value	Description	Explanation	Required Tags
1	Cancel orders for a security	Cancel orders that match the security identification block, all fields required to uniquely qualify the security should be specified.	<b>Required:</b>  SecurityID
7	Cancel all orders	Cancel all orders for the firm identified using this FIX connection	No other tag should be specified.

### 6.4.2. Status of massive cancelation request

Whether the massive cancellation request is accepted or rejected, the trading engine sends an Order Mass Cancel Report message. When the request is rejected, the MassCancelResponse tag contains the value "0". If accepted, it contains the value "7", even when there are no orders matching the selected criteria.

The accepting message should not be deemed a cancellation confirmation. For this purpose, the trading engine sends an Execution Report message for all the orders which were canceled.

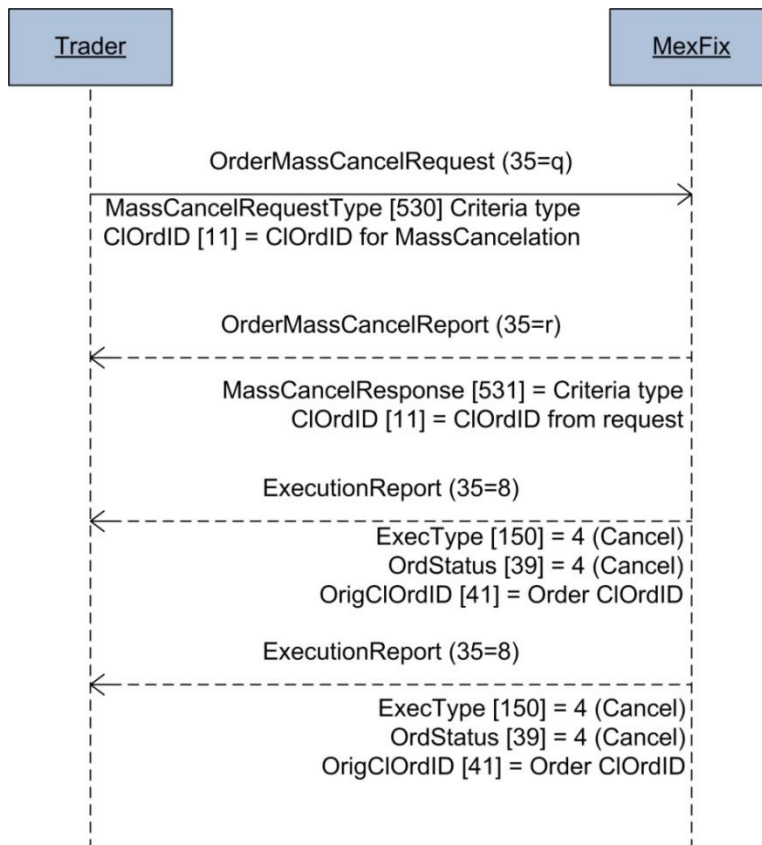
### 6.4.3. ClOrdID Tag

In the relevant Execution Reports messages in which cancellations are notified, there is an OrigClOrdID field identifying unambiguously each one of the orders cancelled.

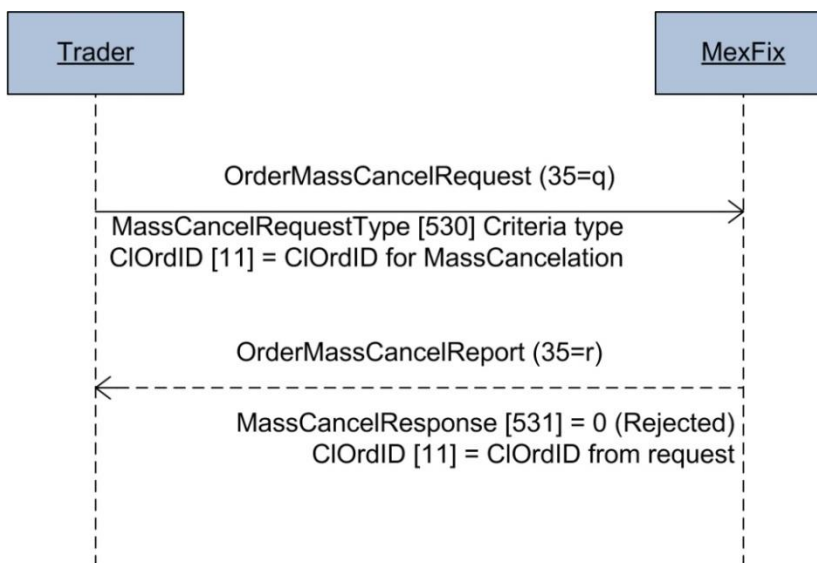
Please take into account that, following the Standard, the ClOrdID field of these messages will contain the same value in all of them, which corresponds to the ClOrdID assigned to the request in the Order Mass Cancel Request message. Therefore, it should be noted that from that moment the cancelled orders will have the same ClOrdID.

## 6.4.4. Mass Cancel Order Flow

### 6.4.4.1. Mass Cancel Request Accepted



### 6.4.4.2. Mass Cancel Request Rejected



## 6.5. Order Status

### 6.5.1. Description

In this section is depicted the order query functionality. Covered by two functions:

- **Query Status about a specific order.** This allows querying about a specific order using its ClOrdID.
- **Query Status about a set of orders.** This allows querying about a set of orders based on certain selection criteria.

Both functionalities are limited to the orders sent to the market during the current trading session.

In both cases, the answer is formed by a unique Execution Report message for each one of the orders, reflecting the last order status. If there is an error in the query, it is rejected by means of a Business Reject message.

#### 6.5.1.1. Query Status about a specific order

When querying about a specific order, the ClOrdID used should coincide with the last one relevant to the order. The query of a ClOrdID that has been replaced, by means of an order cancellation or modification, will be rejected through a Business Reject message.

Unlike most of the messages related to order management, in the Order Status Request message, the ClOrdID field should contain the value of the order being queried.

Please take into account that in an order status request, FIX standard requires the presence of two redundant fields: Symbol and Side. These fields' values should coincide with those of the original order.

The query status about a specified order may be used by the intermediary of a cross trade to query the status of such application. In this case, the query should be made using the ClOrdID of any of the legs of the cross trade.

An order status query by means of an Order Status Request message may return more than one Execution Report message, when the ClOrdID present in the query coincides with the one used in a massive cancellation, since in this case all orders cancelled by the same request share the ClOrdID identifier. There is more information regarding [massive cancellation](#).

### 6.5.1.2. Query Status about a set of orders

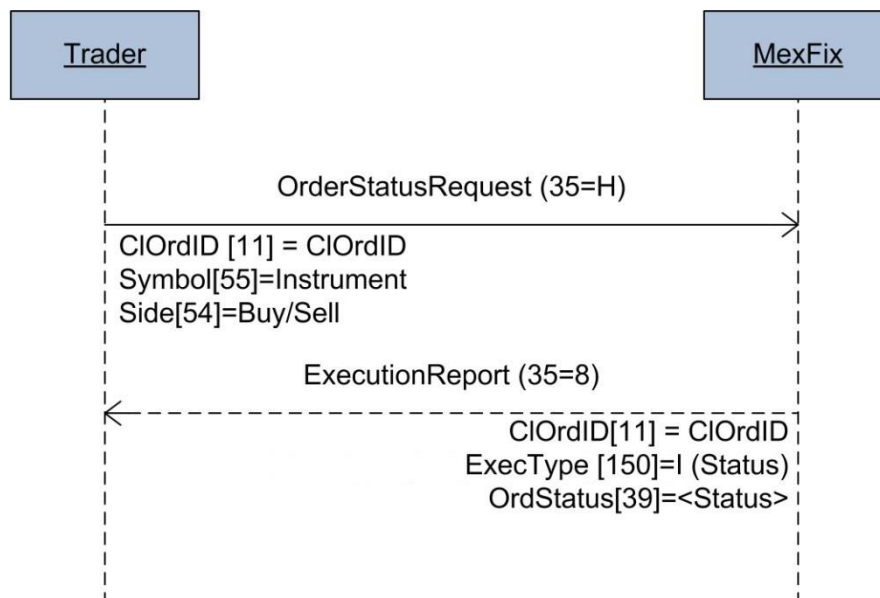
The query status about a set of orders always refers to orders of the same trader and therefore it does not include orders in which such trader acts as an intermediary.

The selection criteria are based in the tag MassStatusRequestType according to this table:

Tag Value	Description	Explanation	Required Tags
1	Status for all orders for an specific security.	Return status of orders that match the security identification block, all fields required to uniquely qualify the security should be specified.	<b>Required:</b> SecurityID
7	Status for all orders.	Return status of all orders for the firm identified using this FIX connection	No tag required.

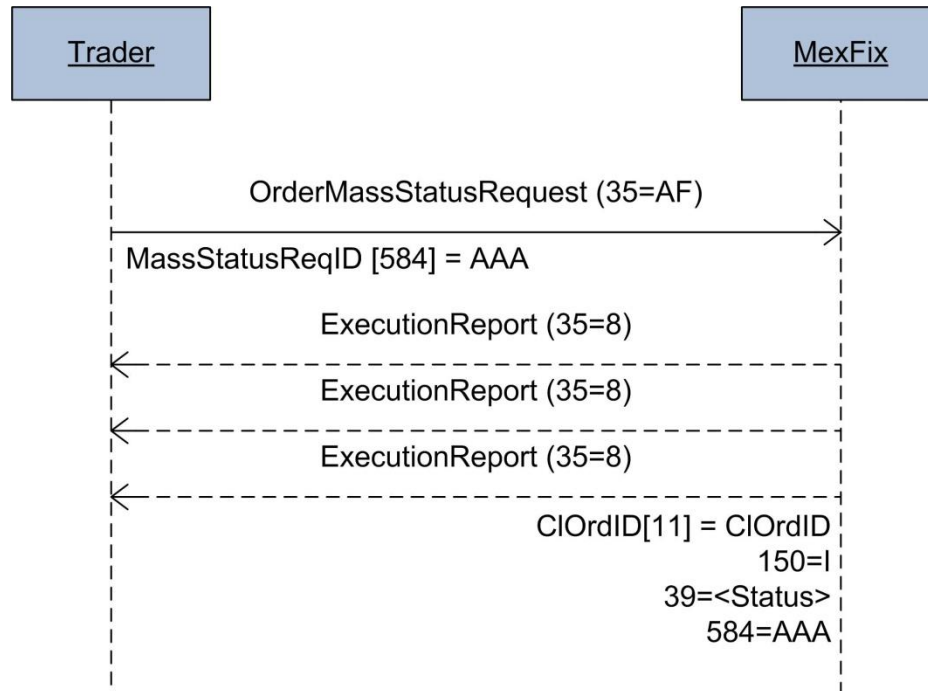
## 6.5.2. Order Status Message Flow

### 6.5.2.1. Order Status





### 6.5.2.2. Order Mass Status



## 6.6. Order Acknowledges

Once an order request is submitted through MexFix (NewOrderSingle, OrderCancelReplaceRequest, OrderCancelRequest or NewCrossOrder messages), the engine will accept or reject the order by sending ExecutionReport or OrderCancelReplaceReject messages.

## 6.7. Trade Notification

When an order is fully or partially filled, MexFix notifies this operation through an ExecutionReport message with the ExecType field = "F" (Trade).

Also, the trade can be corrected ExecType="G" or Canceled ExecType = "H" by the exchange.

When the Execution Report message is used to notify the execution of an order, it contains the customized ExchangeTradeType (5681) tag specifying the tradetype. Refer to Table 19 in the "Coding Tables" document where a list of possible values of this field and its meaning is included.

## 6.8. Message Definitions

### 6.8.1. NewOrderSigle (MsgType = D)

Tag	Name	Req	Valid Values	Format	Description
	Standard Header	Y	MsgType = D		
11	ClOrdID	Y		String(10)	Order unique identifier.
	Start <Parties>				
453	NoPartyIDs	Y	2	NumInGroup	
-> 448	PartyID	Y		String(40)	Member code assigned by MexDer or Operator name.
-> 447	PartyIDSource	Y	D = Proprietary/ Custom code	char	Indicates the coding followed in the PartyID field
-> 452	PartyRole	Y	13 = Order Origination Firm  118 = Operator	int	Indicates the role taken by the referenced party in the PartyID field.  It should be expected two repeating groups. One with PartyRole = 13 and PartyID with the member code; and one with PartyRole = 118 and the PartyID with an additional identifier related to the responsible person (operator).
	End <Parties>				
1	Account	Y		String(5)	Account assigned by MexDer.
21	HandlInst	Y	1 = Automated execution order, private.	char	
18	ExecInst	N	H = Reinstate on System Failure (default)  Q = Cancel on System Failure	MultipleValueString	Order persistence.  Indicates the action taken by the trading engine in adisconnection event.  Value "Q" means to cancel the pending volume.  Not informing this tag or value "H" means the order will remain in the order book.
	Start <Instrument>				
55	Symbol	Y		String (16)	Contract code
	End <Instrument>				
54	Side	Y	1 = Buy  2 = Sell	char	
60	TransactTime	Y		UTCTimestamp	Time at which the request is made. Milliseconds precision.

## MexFix 3.0 Trading Specification

	Start <OrderQtyData>				
38	OrderQty	Y		Qty	Order quantity (volume).
	End <OrderQtyData>				
40	OrdType	Y	1 = Market 2 = Limit	char	Order type.
44	Price	N		Price	Order price.  Required if OrdType = 2.
59	TimeInForce	N	0 = Day (default) 3 = Immediate or Cancel (IoC) 7 = At the close	char	Specifies how long the order is active in the market.
58	Text	N		String(5)	Reference to the order assigned by the client.
	Standard Trailer	Y			

## 6.8.2. OrderCancelReplaceRequest (MsgType = G)

Tag	Name	Req	Valid Values	Format	Description
	Standard Header	Y	MsgType = G		
11	CIOrdID	Y		String(10)	Modification identifier. It becomes the order identifier when the modification is processed.
41	OrigCIOrdID	Y		String(10)	CIOrdID of the order to be Replaced.
	Start <Parties>				
453	NoPartyIDs	Y	2	NumInGroup	
-> 448	PartyID	Y		String(40)	Member code assigned by MexDer or Operator name
-> 447	PartyIDSource	Y	D = Proprietary/ Custom code	char	Indicates the coding followed in the PartyID field
-> 452	PartyRole	Y	13 = Order Origination Firm  118 = Operator	int	Indicates the role taken by the referenced party in the PartyID field.  It should be expected two repeating groups. One with PartyRole = 13 and PartyID with the member code; and one with PartyRole = 118 and the PartyID with an additional identifier related to the responsible person (operator).
	End <Parties>				
1	Account	Y		String(5)	Account assigned by MexDer.
21	HandlInst	Y	1 = Automated execution order, private	char	It should contain the same value as the one specified in the original order.
	Start <Instrument>				
55	Symbol	Y		String (16)	Contract code. It should contain the same value as the one specified in the original order.
	End <Instrument>				
54	Side	Y	1 = Buy  2 = Sell	char	It should contain the same value as the one specified in the original order.
60	TransactTime	Y		UTCTimestamp	Time at which the request is made.
	Start <OrderQtyData>				
38	OrderQty	N		Qty	Order quantity (volume).  New total volume (including the volume already executed).

## MexFix 3.0 Trading Specification

					<p>For instance, if an original order covered 20 contracts, five of which have been partially executed, and the original order is to be reduced to 14 contracts, this field should be filled with the value 19.</p> <p>If this value is not specified, the volume remains unchanged.</p>
	End <OrderQtyData>				
40	OrdType	Y	1 = Market 2 = Limit	char	<p>Order type.</p> <p>It should contain the same value as the one specified in the original order.</p>
44	Price	N		Price	Order price.
58	Text	N		String(5)	Reference to the order assigned by client.
	Standard Trailer	Y			

### 6.8.3. OrderCancelRequest (MsgType = F)

Tag	Name	Req	Valid Values	Format	Description
	Standard Header	Y	MsgType = F		
11	CIOrdID	Y		String(10)	Cancelation identifier.  It becomes the order identifier when the cancellation is processed.
41	OrigCIOrdID	Y		String(10)	CIOrdID of the order to be canceled.
	Start <Parties>				
453	NoPartyIDs	Y	2	NumInGroup	
-> 448	PartyID	Y		String(40)	Member code assigned by MexDer or Operator name.
-> 447	PartyIDSource	Y	D = Proprietary/ Custom code	char	Indicates the coding followed in the PartyID field
-> 452	PartyRole	Y	13 = Order Origination Firm  118 = Operator	int	Indicates the role taken by the referenced party in the PartyID field.  It should be expected two repeating groups. One with PartyRole = 13 and PartyID with the member code; and one with PartyRole = 118 and the PartyID with an additional identifier related to the responsible person (operator).
	End <Parties>				
	Start <Instrument>				
55	Symbol	Y		String (16)	Contract code. It should contain the same value as the one specified in the original order.
	End <Instrument>				
54	Side	Y	1 = Buy  2 = Sell	char	It should contain the same value as the one specified in the original order.
60	TransactTime	Y		UTCTimesta mp	Time at which the request is made.
	Standard Trailer	Y			

### 6.8.4. ExecutionReport (MsgType = 8)

Tag	Name	Req	Valid Values	Format	Description
	Standard Header	Y	MsgType = 8		
37	OrderID	Y		String	Unique order identifier.  It contains "NONE" if ExecType field = 8 (Rejected)
17	ExecID	Y		String	ExecutionReport unique identifier assigned by MexFix.  It contains 0 if the ExecType field = "I" (Order Status)
527	SecondaryExecID	N		String(13)	Trade registration number Unique identifier of the order partial or full execution.
571	TradeReportID*	N		String(10)	Contains the BlockTrade identifier.  May be present when ExecType = F and the trade belongs to a block trade.
11	CIOrdID	N		String(10)	CIOrdID sent by the client.
41	OrigCIOrdID	N		String(10)	The OrigCIOrdID that was sent by the client.  It is present only when the related message is a cancellation or modification request.
5681	ExchangeTradeType	N	Refer to Table 19 in the "Coding Tables" document.	String(2)	It is present only when the ExecType field = "F" (Trade)
790	OrdStatusReqID	N		String	It contains the same value as the one specified in the Order Status Request message related.  It is only present if the related message is of this type and contained this field.
584	MassStatusReqID	N		String	Related message identifier.  It is present only when the related message is an Order Mass Status Request.
911	TotNumReports	N		Int	It indicates the total number of messages comprehended by the answer to an Order Mass Status Request.

## MexFix 3.0 Trading Specification

					It is present only when an answer to this type of request is involved.
912	LastRptRequested	N		Boolean	It is used to indicate that this is the last message sent as an answer to an Order Mass Status Request.
	Start <Parties>				
453	NoPartyIDs	N	1	NumInGroup	
-> 448	PartyID	Y		String(40)	Member code assigned by MexDer.
-> 447	PartyIDSource	Y	D = Proprietary/ Custom code	char	Indicates the coding followed in the PartyID field
-> 452	PartyRole	Y	13 = Order Origination Firm	int	Indicates the role taken by the referenced party in the PartyID field.
	End <Parties>				
548	CrossID	N		String(10)	Identifier of the New Order Cross Message.  It is present only if the related message is of this type.
549	CrossType	N		Int	It is the same value specified in the New Order Cross message.  It is present only if the related message is of this type.
150	ExecType	Y	0 = New  4 = Canceled  5 = Replace  8 = Rejected  C = Expired  F = Trade  I = Order Status  G = TradeCorrect  H = TradeCancel	char	It indicates the related message status
39	OrdStatus	Y	0 = New  1 = Partially Filled  2 = Filled  4 = Canceled  8 = Rejected  C = Expired	char	It reports the current order status.
103	OrdRejReason	N	0 =	Int	Rejection reason.



			Broker/Exchange option  1 = Unknown Symbol  2 = Exchange closed  3 = Order exceeds limit  4 = Too late to enter  5 = Unknown order  6 = Duplicate Order  11 = Unsupported order characteristic  13 = Incorrect quantity  15 = Unknown account(s)  99 = Other		It is present if the ExecType field = 8.
1	Account	Y		String(5)	Account related to the order.  Assigned by MexDer.
	Start <Instrument> Symbol	Y		String (16)	Contract code related to the order.
	End <Instrument> Side	Y	1 = Buy  2 = Sell	char	It indicates if the order is to buy or sell.
	Start <OrderQtyData> OrderQty	Y		Qty	Order quantity (volume) as indicated in the NewOrder message or in the modification message.
	End <OrderQtyData> OrdType	N	1 = Market  2 = Limit	char	Order type.
44	Price	N		Price	Order price.
377	SolicitedFlag	N	Y = Was solicited (default) N = Was not solicited	Boolean	Whenever the order is created by MexDer's trading desk, this tag will contain 'N'.
59	TimelnForce	N	0 = Day (default)	char	Specifies how long the order remains in the market.

## MexFix 3.0 Trading Specification

			3 = Immediate or Cancel (IoC)  7 = At the close		
18	ExecInst	N	H = Reinstate on System Failure (default)  Q = Cancel on System Failure	MultipleValueString	Order persistence.  Indicates the action taken by the trading engine in the event of a disconnection.  Value "Q" means to cancel the pending volume.  Not informing this tag or value "H" means the order will remain in the order book.
32	LastQty	N		Qty	Trade quantity (volume).  It is present if ExecType = F.
31	LastPx	N		Price	Trade price.  It is present if ExecType = F.
151	LeavesQty	Y		Qty	Pending order quantity (volume).  It contains 0 when the ExecType field = 4 (Canceled).
14	CumQty	Y		Qty	Total quantity filled.
6	AvgPx	Y		Price	Average price of all the executions of this order.  This field should not be considered when the CumQty field = 0.
60	TransactTime	Y		UTCTimestamp	Time at which the request is made.
58	Text	N		String	Contains the same value (if any) as the one sent in the order messages (D, G, s, AE).  In case of unsolicited executions, the exchange can specify some value in this tag.
442	MultiLegReportingType	N	1 = Single Security char (default value)  2 = Individual leg of a multi-leg security  3 = Multi-leg security	char	It indicates if the trade being informed deals with a single contract, a time-spread contract or a time-spread leg.
*20127	TradeIDRef	N		String	It is present only if tag 442 = 2.  Contains the trade registration number (SecondaryExecID) of

					the corresponding strategy trade.
	Standard Trailer	Y			

### 6.8.5. OrderCancelReject (MsgType = 9)

Tag	Name	Req	Valid Values	Format	Description
	Standard Header	Y	MsgType = 9		
37	OrderID	Y		String	Unique order identifier.  It contains "NONE" if CxlRejReason (102) = "Unknown order".
11	ClOrdID	Y		String(10)	ClOrdID sent by the client.
41	OrigClOrdID	Y		String(10)	The OrigClOrdID that was sent by the client.
39	OrdStatus	Y	0 = New 1 = Partially Filled 2 = Filled 4 = Canceled 8 = Rejected	char	Order Status.  It contains 8 (Rejected) if CxlRejReason field = 1(Unknown order).
40	OrdType	N	1 = Market 2 = Limit	char	Order type.
60	TransactTime	N		UTCTimestamp	Time in which this rejection message was generated.
434	CxlRejResponseTo	Y	1 = Order Cancel Request 2 = Order Cancel / Replace Request	char	Type of message being answered.
102	CxlRejReason	N	0 = Too late to cancel 1 = Unknown order 2 = Exchange Option 6 = Duplicate ClOrdID (11) received 99 = Other	int	Reason for rejection.  When the value is 99, the Text field contains an explanatory text.
58	Text	N		String	Explanatory text of rejection reason.
	Standard Trailer	Y			

### 6.8.6. OrderStatusRequest (MsgType = H)

Tag	Name	Req	Valid Values	Format	Description
	Standard Header	Y	MsgType = H		
11	ClOrdID	Y		String(10)	ClOrdID sent by the client.
	Start <Parties>				
453	NoPartyIDs	N	1, 2	NumInGroup	
-> 448	PartyID	Y		String(40)	Member code assigned by MexDer or Operator name.
-> 447	PartyIDSource	Y	D = Proprietary/ Custom code	char	Indicates the coding followed in the PartyID field
-> 452	PartyRole	Y	13 = Order Origination Firm  118 = Operator	int	Indicates the role taken by the referenced party in the PartyID field.  If NoPartyIDs = 1, PartyRole should be 13 and PartyID should contain the member code.  If NoPartyIDs = 2, are expected two repeating groups. One with PartyRole = 13 and PartyID with the member code; and one with PartyRole = 118 and the PartyID with an additional identifier related to the responsible person (operator).
	End <Parties>				
790	OrdStatusReqID	N		String(10)	Request identification optional code. If present, it will be returned in the relevant Execution Report answer message.
	Start <Instrument>				
55	Symbol	Y		String (16)	It should contain the same value as the order queried.
	End <Instrument>				
54	Side	Y	1 = Buy  2 = Sell	char	It should contain the same value as the order queried.
	Standard Trailer	Y			

### 6.8.7. OrderMassStatusRequest (MsgType = AF)

Tag	Name	Req	Valid Values	Format	Description
	Standard Header	Y	MsgType = AF		
584	MassStatusReqID	Y		String(10)	Unique ID of OrderMassStatusRequest assigned by the client.
585	MassStatusRequestType	Y	1 = Status for all orders for a specific security.  7 = Status for all orders.	char	Selection Type.
	Start <Parties>				
453	NoPartyIDs	N	1, 2	NumInGroup	
-> 448	PartyID	Y		String(40)	Member code assigned by MexDer or Operator name.
-> 447	PartyIDSource	Y	D = Proprietary/ Custom code	char	Indicates the coding followed in the PartyID field
-> 452	PartyRole	Y	13 = Order Origination Firm  118 = Operator	int	Indicates the role taken by the referenced party in the PartyID field.  If NoPartyIDs = 1, PartyRole should be 13 and PartyID should contain the member code.  If NoPartyIDs = 2, are expected two repeating groups. One with PartyRole = 13 and PartyID with the member code; and one with PartyRole = 118 and the PartyID with an additional identifier related to the responsible person (operator).
	End <Parties>				
	Start <Instrument>				
48	SecurityID	N	If you need more detail on contract groups, refer to Table 20 in the "Coding Tables" document.	String	Contract group.  Required if MassStatusRequestType = 1.
22	SecurityIDSource	N	8 = Exchange Symbol	String	It is required if the SecurityID has been specified.
	End <Instrument>				
60	TransactTime	Y		UTCTimestamp	
	Standard Trailer	Y			

### 6.8.8. OrderMassCancelRequest (MsgType = q)

Tag	Name	Req	Valid Values	Format	Description
	Standard Header	Y	MsgType = q		
11	ClOrdID	Y		String(10)	Unique identifier of this OrderMassCancelRequest message.
530	MassCancelRequestType	Y	1 = Cancel orders for a specific security.  7 = Cancel all orders.	char	Selection Type.
	Start <Parties>				
453	NoPartyIDs	Y	2	NumInGroup	
-> 448	PartyID	Y		String(40)	Member code assigned by MexDer or Operator name.
-> 447	PartyIDSource	Y	D = Proprietary/ Custom code	char	Indicates the coding followed in the PartyID field
-> 452	PartyRole	Y	13 = Order Origination Firm  118 = Operator	int	Indicates the role taken by the referenced party in the PartyID field.  It should be expected two repeating groups. One with PartyRole = 13 and PartyID with the member code; and one with PartyRole = 118 and the PartyID with an additional identifier related to the responsible person (operator).
	End <Parties>				
	Start <Instrument>				
48	SecurityID	N	If you need more detail on contract groups, refer to Table 20 in the "Coding Tables" document.	String	Contract group.  Required if MassCancelRequestType = 1.
22	SecurityIDSource	N	8 = Exchange Symbol	String	It is required if the SecurityID has been specified.
	End <Instrument>				
60	TransactTime	Y		UTCTimestamp	
	Standard Trailer	Y			

### 6.8.9. OrderMassCancelReport (MsgType = r)

Tag	Name	Req	Valid Values	Format	Description
	Standard Header	Y	MsgType = r		
11	ClOrdID	N		String(10)	ClOrdID specified in the OrderMassCancelRequest message.
37	OrderID	Y		String	Unique order identifier for the OrderMassCancelRequest message.
530	MassCancelRequestType	Y	1 = Cancel orders for a security.  7 = Cancel all orders.	char	It contains the same value as the one specified in the request.
531	MassCancelResponse	Y	0 = Cancel Request Rejected.  1 = Cancel orders for a security.  7 = Cancel all orders.	char	It contains the value 1 or 7 if the cancellation was accepted.  It contains 0 if the request was rejected.  If the value is 0, the MassCancelRejectReason field contains the reason for rejection.
532	MassCancelRejectReason	N	1 = Invalid or unknown Security.  99 = other	char	Reason for rejection. It is present if the MassCancelResponse field = 0.  If the value is 99, the Text field contains an explanatory text of the reason for rejection.
58	Text	N		String	Reason of rejection.
	Standard Trailer	Y			



## 7. Cross Trades

### 7.1. Introduction

Cross trades are trades made between clients outside the market order book. This functionality allows members to request the registration of these trades with MexFix.

### 7.2. Cross Orders

A cross order should be reported using the message type 35=s and including the information regarding the price, quantity as mandatory fields, a reference is optional in any side. The message must also contain the buyer and seller which will correspond to the member who is reporting the cross order in the market.

### 7.3. Cross Executions

The only cross trade supported by MexFix is "all or nothing" (CrossType = 1). A cross trade is fully executed or rejected, but it is never partially executed. Therefore, the quantity indicated for both parties should coincide; otherwise, the order is rejected.

When the order is processed by the trading engine, the cross trade is fully executed or rejected (due to market rules or invalid information). When it's executed, an ExecutionReport message is sent to each one of the cross participants (sell and buy side) indicating the trade execution or the rejection.

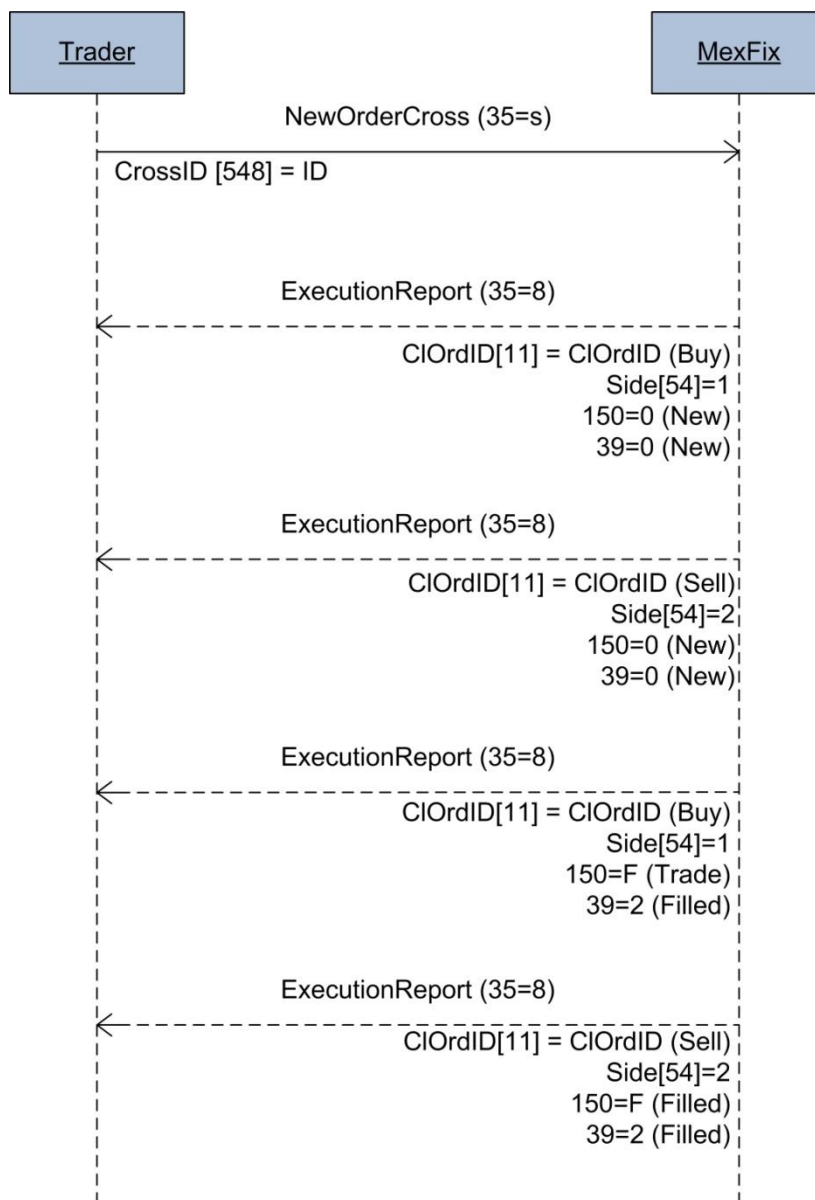
### 7.4. CrossID

The CrossID tag is a unique identifier for each cross trade. It should be assigned by the trader.

## 7.5. Cross Order Message Flow

In this section, is depicted the valid flow for CrossTrades.

### 7.5.1. CrossOrder Accepted



## 7.5.2. CrossOrder Rejected



## 7.6. Message Definition

### 7.6.1. NewCrossOrder (MsgType = s)

Tag	Name	Req	Valid Values	Format	Description
	Standard Header	Y	MsgType = s		
548	CrossID	Y		String(10)	Unique identifier of the New Order Cross message.
549	CrossType	Y	1 = All or None	int	Type of cross trade.
550	CrossPrioritization	Y	0 = None	int	No party is prioritized.
552	NoSides	Y	2	NumInGroup	Both parties (buyer and seller) are specified.
-> 54	Side	Y	1 = Buy 2 = Sell	char	
-> 11	ClOrdID	Y		String(10)	Unique order identifier for the member entering the cross trade.  It should be different for the buying and selling party.
->	Start <Parties>				
-> 453	NoPartyIDs	Y	2	NumInGroup	
->-> 448	PartyID	Y		String(40)	Member code assigned by MexDer or Operator name.
->-> 447	PartyIDSource	Y	D = Proprietary/ Custom code	char	Indicates the coding followed in the PartyID field
->-> 452	PartyRole	Y	13 = Order Origination Firm  118 = Operator	int	Indicates the role taken by the referenced party in the PartyID field.  It should be expected two repeating groups. One with PartyRole = 13 and PartyID with the member code; and one with PartyRole = 118 and the PartyID with an additional identifier related to the responsible person (operator).
->	End <Parties>				
-> 1	Account	Y		String(5)	Account assigned by MexDer.
->	Start <OrderQtyData>				
-> 38	OrderQty	Y*		Qty	Order quantity (volume) (it

## MexFix 3.0 Trading Specification

					should be the same for both parties).
->	End <OrderQtyData>				
-> 58	Text	N	AV = Sell self-entry  AC = Buy self-entry  Any other value will be considered as a cross trade.	String(7)	When it is required to indicate a "Self-entry", the first 2 characters will be used with "AV or AC" leaving only available characters from 3 to 7 for the "reference".  When a "Self-entry" is not indicated, characters from 1 to 5 will be used for the "reference".
	Start <Instrument>				
55	Symbol	Y		String(16)	Contract code
	End <Instrument>				
21	HandlInst	Y	1	char	
60	TransactTime	Y		UTCTimeStamp	Date and time, in UTC format, in which the transaction is carried out.
40	OrdType	Y	2 = Limit	char	Only "Limit" orders are allowed for cross trades.
44	Price	Y		Price	
	Standard Trailer	Y			

## 8. Event Notification

All event notification will be sent throughout the market data sessions.

## 9. BlockTrades

### 9.1. Introduction

A block trade takes place when two Members negotiate a trade out of the electronic market and want to register that operation in the Exchange.

As an operation rule, Block Trades should always be initiated or registered by the seller, to later be accepted or rejected by the buyer.

Block Trade operation can only be initiated by the seller using message “Trade Capture Report”, the Exchange will answer with message “Trade Capture Report Ack” to indicate Block trade was received. The Exchange will execute the required validations sending necessary error or success messages.

If a block trade is accepted by the Exchange, it will send a “Trade Capture Report” message to the buyer with the information of the operation that the seller is requesting, and then the buyer should accept or reject the operation. In case of rejection, the buyer should notify this action with the message “Trade capture Report”; the trading engine will generate a message to the seller indicating that the block trade operation was not accepted by the buyer.

In case of the operation is accepted, the buyer will specify the account and a reference (optional) in the “Trade capture report” message to notify this event.

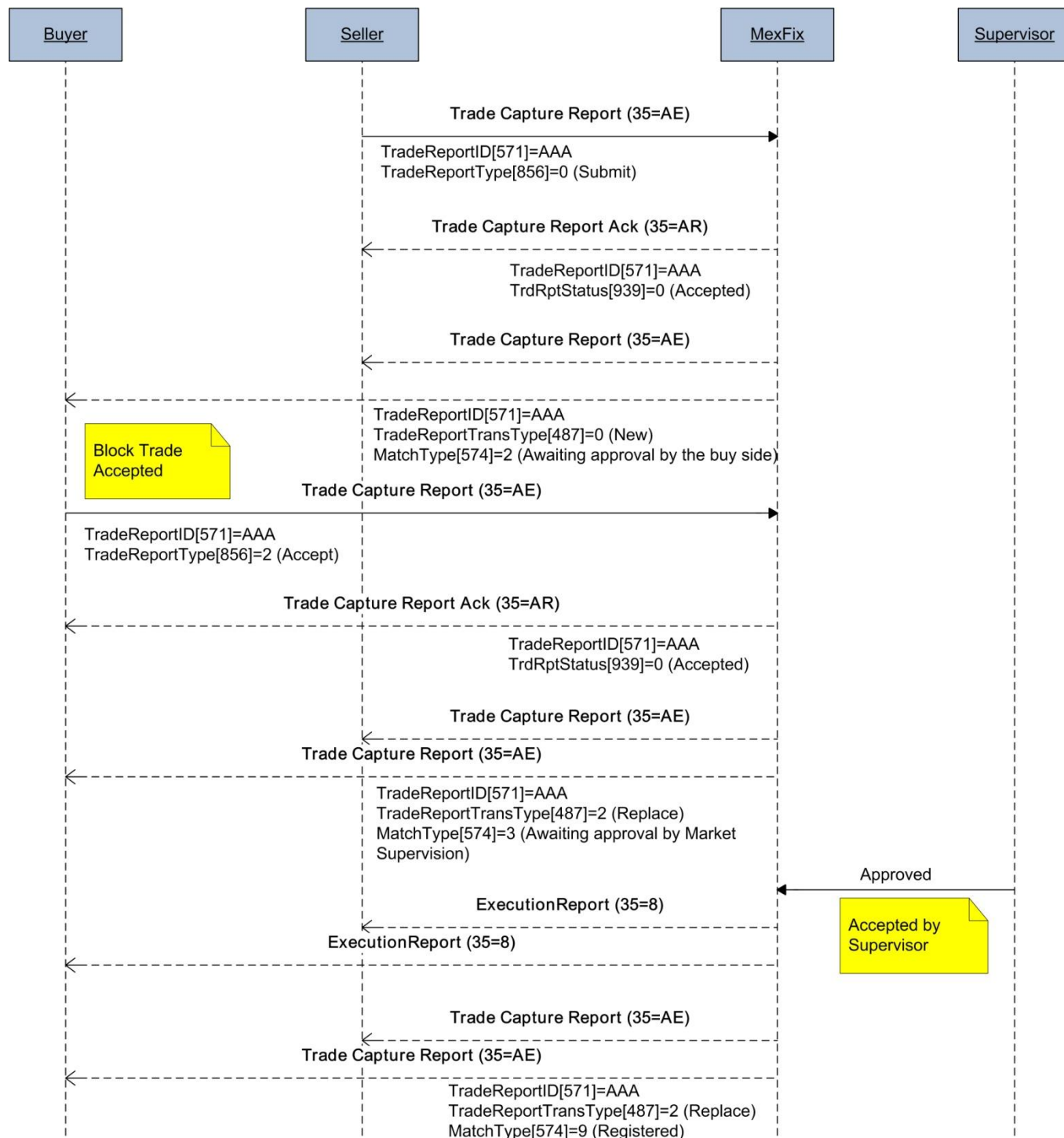
The buyer will have a period of time, (defined by MexDer), to confirm the Block Trade operation once it is submitted by the seller. If the period of time is over and no confirmation has been received from the buyer, either to accept or reject the block trade, the Exchange will cancel it automatically, sending a message to the seller and buyer indicating the Block Trade has been canceled.

### 9.2. Block Trade Acceptance

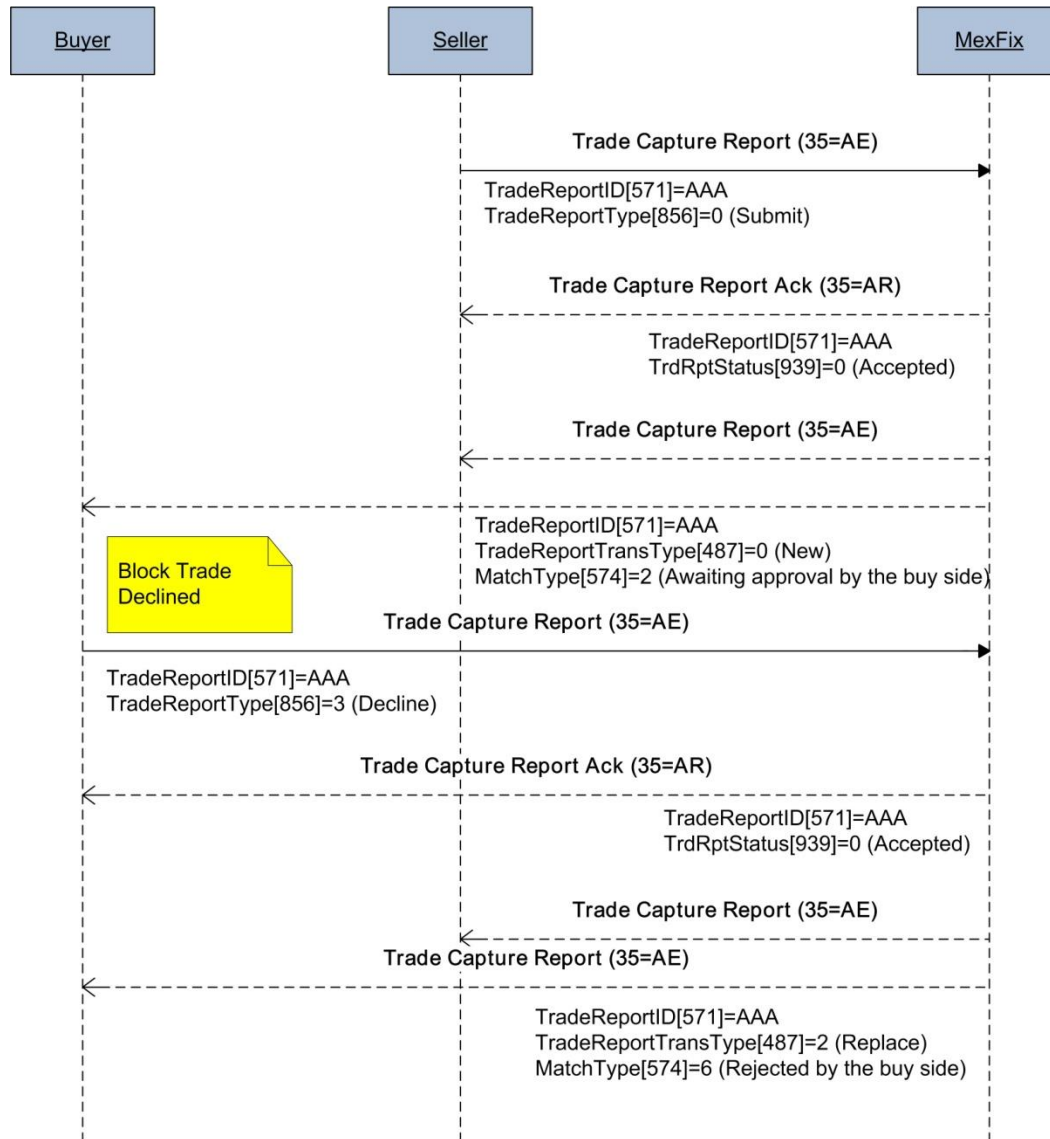
Once the Block Trade negotiation has been completed (both sides) and it is approved by the Exchange, the buyer and seller will receive an ExecutionReport message (ExecType = “F”, Trade) in order to notify the execution. The SecondaryTradeReportID assigned by Exchange is indicated in the CrossID field of Execution Report message.

## 9.3. BlockTrade Message Flow

### 9.3.1. BlockTrade Accepted by Buyer and the Exchange

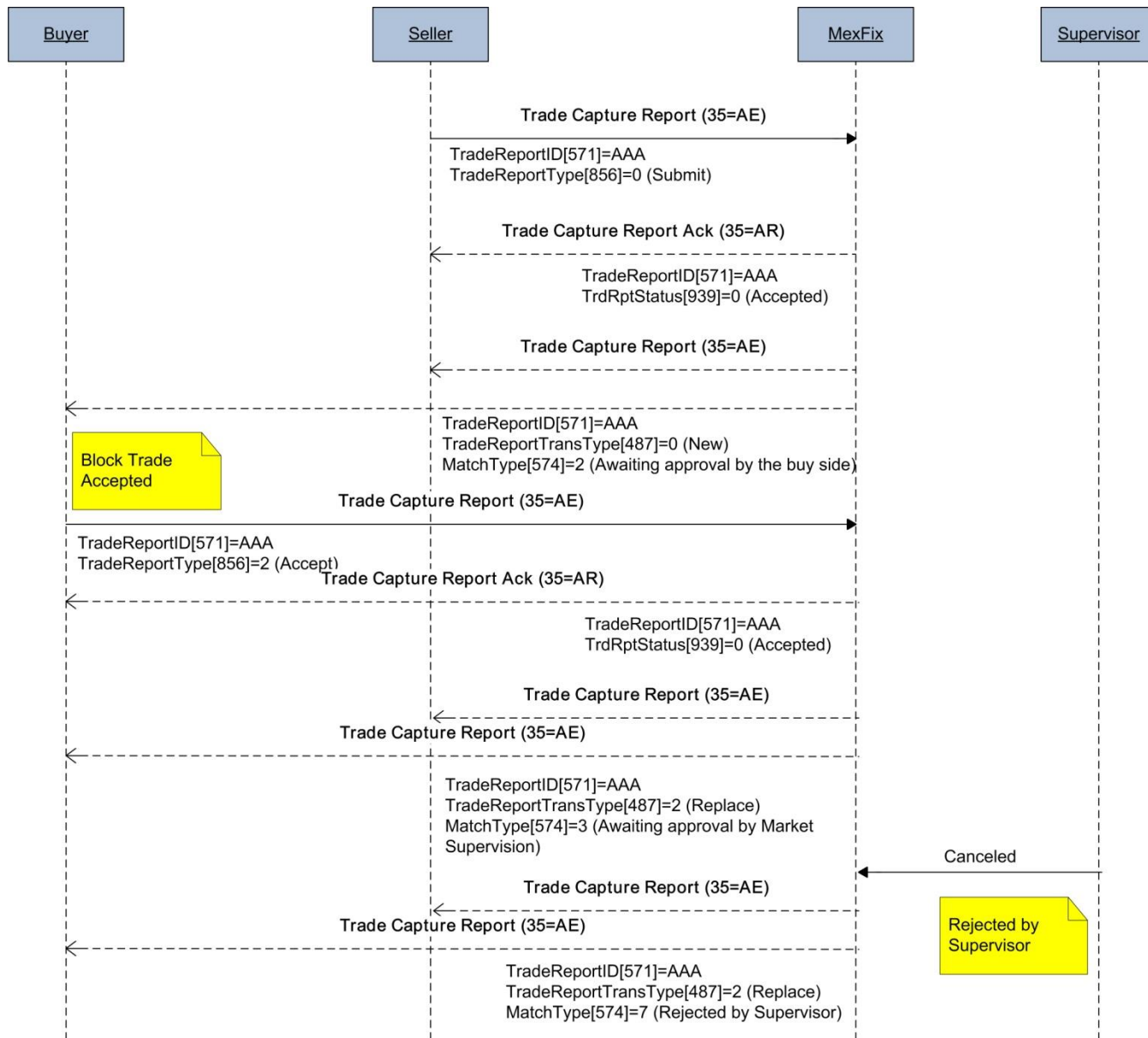


### 9.3.2. BlockTrade Declined (by the buyer)





### 9.3.3. BlockTrade Rejected by the Exchange



## 9.4. Message Definition

### 9.4.1. Trade Capture Report (MsgType = AE) Sent by Members

Message used (by seller and buyer) to negotiate a Block Trade in the Exchange.

Tag	Name	Req	Valid Values	Format	Description
	Standard Header	Y	MsgType = AE		
571	TradeReportID	Y		String(10)	Unique identifier for each Trade Capture Report message, created by the seller.  Unique per FIX session.
856	TradeReportType	Y	0 = Submit  2 = Accept  3 = Decline  6 = Trade Report Cancel	int	Type of Trade Report:  0 (Submit): This is the value indicated by the seller when it sends the initial block trade request.  2 (Accept): Used by the buyer to accept a block trade.  3 (Decline): Used by the buyer to reject a block trade.  6 (Cancel): Used by the seller to cancel the block trade request before the buyer accept it.
570	PreviouslyReported	Y	Y,N	Boolean	Indicates if the block trade was notified to the counterparty.
	Start <Instrument>				
55	Symbol	Y		String(18)	
	End <Instrument>				
32	LastQty	Y	> 0	Qty	Block trade quantity.
31	LastPx	Y		Price	Block trade Price.
552	NoSides	Y	2	NumInGroup	
-> 54	Side	Y	1 = Buy  2 = Sell	char	Position that the party takes in the cross trade.
-> 37	OrderID	Y	NONE	String	Always 'NONE'.
	Start <Parties>				
-> 453	NoPartyIDs	Y	2	NumInGroup	
->-> 448	PartyID	Y		String(40)	Member code assigned by MexDer or Operator name.
->-> 447	PartyIDSource	Y	D = Proprietary/ Custom code	char	Indicates the coding followed in the PartyID

## MexFix 3.0 Trading Specification

					field
->-> 452	PartyRole	Y	13 = Order Origination Firm  118 = Operator	int	Indicates the role taken by the referenced party in the PartyID field.  It should be expected two repeating groups. One with PartyRole = 13 and PartyID with the member code; and one with PartyRole = 118 and the PartyID with an additional identifier related to the responsible person (operator).
	End <Parties>				
-> 1	Account	N		String	Account assigned by MexDer.
-> 58	Text	N		String(5)	Reference.
	Standard Trailer	Y			

### 9.4.2. Trade Capture Report Ack (MsgType = AR)

Message used by MexDer 's Fix Engine as a reply to a Trade Capture Report message. It contains the same information as the Trade Captured Report received, including the trade registration number assigned by the exchange, the indication if the block trade was accepted or rejected and the rejection reason.

Tag	Name	Req	Valid Values	Format	Description
	Standard Header	Y	MsgType = AR		
571	TradeReportID	Y		String(10)	Contains the same information as the corresponding Trade Capture Report.
487	TradeReportTransType	N	0 = New 1 = Cancel 2 = Replace	int	Contains the same information as the corresponding Trade Capture Report.
856	TradeReportType	N	0 = Submit 2 = Accept 3 = Decline 6 = Trade Report Cancel	int	Contains the same information as the corresponding Trade Capture Report.
818	SecondaryTradeReportID	N		String	Trade number assigned by the exchange.
939	TrdRptStatus	N	0 = Accepted 1 = Rejected	int	Trade Report Status.  Indicates if the Trade Capture Report was accepted or rejected by the Exchange.
751	TradeReportRejectReason	N	Refer to appendix A for a complete list of error codes.		Identifies the rejection reason.  Only present when TrdRptStatus [939] = 1.
	Start <Instrument>				
55	Symbol	Y		String	Contains the same information as the corresponding Trade Capture Report.
	End <Instrument>				
32	LastQty	Y		Qty	Contains the same information as the corresponding Trade Capture Report.
31	LastPx	Y		Price	Contains the same information as the corresponding Trade

					Capture Report.
552	NoSides	Y	2	NumInGroup	
-> 54	Side	Y	1 = Buy 2 = Sell	char	
-> 37	OrderID	Y	NONE	String	
	Start <Parties>				
-> 453	NoPartyIDs	Y	1	NumInGroup	
->-> 448	PartyID	Y		String(40)	Member code assigned by MexDer.
->-> 447	PartyIDSource	Y	D = Proprietary/ Custom code	char	Indicates the coding followed in the PartyID field
->-> 452	PartyRole	Y	13 = Order Origination Firm	int	Indicates the role taken by the referenced party in the PartyID field.
	End <Parties>				
-> 1	Account	N		String	Contains the same information as the corresponding Trade Capture Report.
-> 58	Text	N		String	Contains the same information as the corresponding Trade Capture Report.
	Standard Trailer	Y			

### 9.4.3. Trade Capture Report (MsgType = AE) Sent by the Exchange

Message used by the exchange to report the status of a block trade negotiated by buyer and seller parties.

Tag	Name	Req	Valid Values	Format	Description
	Standard Header	Y	MsgType = AE		
571	TradeReportID	Y		String(10)	Unique identifier for each Trade Capture Report message, created by the seller.
487	TradeReportTransType	N	0 = New 1 = Cancel 2 = Replace	int	0 (New): Indicates an initial block trade request  1 (Cancel): Indicates the block trade request has been cancelled.  2 (Replace): Indicates the block trade request has been modified (i.e. because has been accepted by the counterparty member).
570	PreviouslyReported	Y	Y,N	Boolean	Indicates if the block trade was notified to the

**MexFix 3.0**  
**Trading Specification**

					counterparty.
818	SecondaryTradeReportID	N		String	Trade number assigned by the exchange
	Start <Instrument>				
55	Symbol	Y		String	Contract code.
	End <Instrument>				
32	LastQty	Y	> 0	Qty	Blocktrade quantity.
31	LastPx	Y		Price	Blocktrade price.
574	MatchType	N	2 = Awaiting approval by the buy side.  3 = Awaiting approval by Market Supervision.  4 = Cancelled. 6 = Rejected by the buy side.  7 = Rejected by Market Supervision.  9 = Registered  A = Cancelled by the system.  B = Rejected by the system.	String	Describes the block trade status.
552	NoSides	Y	2	NumInGroup	
-> 54	Side	Y	1 = Buy  2 = Sell	char	
-> 37	OrderID	Y	NONE	String	
	Start <Parties>				
-> 453	NoPartyIDs	Y	1	NumInGroup	
->-> 448	PartyID	Y		String(40)	Member code assigned by MexDer.
->-> 447	PartyIDSource	Y	D = Proprietary/ Custom code	char	Indicates the coding followed in the PartyID field
->-> 452	PartyRole	Y	13 = Order Origination Firm	int	Indicates the role taken by the referenced party in the PartyID field.
	End <Parties>				
-> 1	Account	N		String	Contains the same information as the corresponding Trade Capture Report.
-> 58	Text	N		String	Contains the same information as the corresponding Trade Capture Report.
	Standard Trailer	Y			

## 10. Appendix

### 10.1. Appendix A

Block trade error codes

Error Code	Description
1	Invalid Party Information
2	Unknown instrument
3	Suspended instrument
99	Others
4000	Invalid seller reference
4001	Invalid buyer reference
4002	Invalid minimum quantity
4003	Price exceeds market filter
4004	Seller not granted to trade with the instrument
4005	Buyer does not exist
4006	Buyer not granted to trade with the instrument
4007	Member is suspended
4008	Message out of sequence
4009	Invalid Market Configuration
4010	Market Closed
4012	Cannot accept or decline with market closed
4013	Cannot modify seller blocktrade data
4014	Invalid sequences for message
4015	Invalid Seller Account
4016	Invalid Buyer Account
4017	Invalid minimum quantity for stapled
4018	Missing TradeReportID
4019	Missing TradeReportType
4021	Missing Symbol
4022	Missing Seller Code
4023	Missing Buyer Code
4024	Missing seller account
4025	Missing BlockTrade Parameters
4026	Invalid quantity
4027	Invalid price
4028	Missing buyer account
5001	Existing TradeReportID

End of Document