Common Customer Gateway Solutions

FIX 5.0 (SP1) Protocol

Specification

Version 2.1

28 January 2015

Document History

VERSION	DATE	SUMMARY OF CHANGES
1.0	2014-06-30	Release version
1.1	2014-08-07	Update parties In trade ER, added contraFirm and contraGiveupFirm. In Outbound AE, added contraGiveupFirm Add CrossID and Account fields to ER msg Remove OrderRestrictions and ExecRestatementReason fields Remove MatchType field from ER msg Update TrdType field values
2.0	2014-10-07	Added Buy Back in Tag 54 Side Field Enumeration
2.0.1	2014-10-14	Updated Appendix C - to add 'Z' as possible value for tag 54-Side Updated 5.3.1 to add side is amendable to and from BuyIn and ShortSell
2.0.2	2015-01-22	Updated section 4.1.1 *Logon (A)* for out of range of 108-HeartBtInt. Updated Table 7 - *Logon*. Tag 1137-DefaultAppVerID will not be validated. Updated Table 18 - *Order Cancel/Replace Request.* 54-Side will not be validated. Added section 6 - *Reference Data* for news. Added news related tags in Table 32 – *Field Enumerations Sorted By Tag Name*. Updated text description for tag 572-TradeReportRefID and added tag 572 in Table 25 - *Trade Capture Report to confirm / decline*. Appendix C - 'Z' value for tag 54-Side updated.
2.1	2015-01-28	Updated Table 25. TradeReportRefID tag is mandatory. Updated Table 24. TradeReportRefID tag is only for initiator. Updated One-party Block Sale workflow. Added Cross Block Sale workflow in section 7.2.3 *Workflow for Cross Report.* Figure 7 - *One-Party cancel Report (or time out) before counterparty confirms* updated. Updated Table 32 for tag 1473 value. Market Place News should be XCHG.

Contents

Tables

1 Introduction to X-stream INET FIX

This document provides the Financial Information Exchange (FIX) specification for the X-stream INET trading platform provided by OMX Technology AB, subsidiary of NASDAQ OMX Group, Inc.

X-stream FIX supports the 5.0 (SP1) protocol. The X-stream FIX server will also support FIX 4.4 clients. However, deprecated FIX 4.4 fields will be replaced with the equivalent FIX 5.0 fields.

It is assumed that the user of this manual is familiar with the FIX protocol standard (which can be found at www.fixprotocol.org).

2 Session Information

X-stream FIX expects the client application to fully comply with the FIX 5.0 specification.

The first message should be a logon message. No additional message should be transmitted until validation of the logon message and SenderCompID (49) is complete.

X-stream FIX does not support encryption or compression.

2.1 FIX Admin and Infrastructure Messages Supported

The standard FIX administrative messages are supported by the X-stream FIX server.

Table 1 – FIX Admin Messages Supported

MESSAGE NAME	MSGTYPE
Heartbeat	0
Logon	A
Test Request	1
Resend Request	2
Reject	3
Sequence Reset	4
Logout	5

Additionally the Business Reject Message is supported to indicate an application message that cannot be processed by the X-stream FIX server that cannot be rejected by another more suitable message.

Table 2 – FIX Infrastructure Messages Supported

HEADING	MSGTYPE
Business Message Reject	j

2.2 SenderCompID and TargetCompID

FIX clients should send these tags in the message header.

Table 3 – FIX Client to X-stream FIX Server

TAG	NAME	REQUIRED	FORMAT	COMMENTS
49	SenderCompID	Y	String	The ID of the FIX client agreed with the Exchange.
56	TargetCompID	Y	String	The ID of the Exchange.

A FIX client should expect to receive these tags in the message header from the X-stream FIX server at the Exchange.

Table 4 – X-stream FIX server to FIX Client

TAG	NAME	REQUIRED	FORMAT	COMMENTS
49	SenderCompID	Y	String	The ID of the Exchange.
56	TargetCompID	Y	String	The ID of the FIX client agreed with the Exchange.

3 FIX Application Messages

X-stream FIX supports the following FIX protocol application messages.

Table 5 - FIX Inbound Application Messages (to the Exchange)

MESSAGE NAME	MSGTYPE	COMMENTS
New Order Single	D	Used by participants to submit orders for execution.
New Order Cross	s	Used by participants to submit crossing orders for execution.
Order Cancel Request	F	Request to cancel a live order.
Order Cancel / Replace Request	G	Request to amend or cancel a live order.
Trade Capture Report	AE	Report, accept or decline an off market trade.

Table 6 – FIX Outbound Application Messages (from the Exchange)

MESSAGE NAME	MSGTYPE	COMMENTS
Execution Report	8	Accept or reject for message D, F or G, order expiry, trade or restatement of overnight orders - if GTD or GTC orders supported.
Order Cancel Reject	9	Failure of message F or G.
Trade Capture Report Ack	AR	Initial response validating or invalidating a submitted Trade Capture Report.
Trade Capture Report	AE	Publish a pending, declined or accepted off market trade.

4 FIX Message Definitions

4.1 Session

4.1.1 Logon (A)

The logon message authenticates a user establishing a connection to a remote system. The logon message must be the first message sent by the application requesting to initiate a FIX session.

Table 7 – Logon

TAG	FIELD NAME	REQ'D	COMMENTS	FORMAT
Standard Header		Y	MsgType = A	
98	EncryptMethod	Y	(Always unencrypted)	Int
108	HeartBtInt	Y	Note same value used by both sides	Int
141	ResetSeqNumFlag	N	Indicates both sides of a FIX session should reset sequence numbers	Boolean
553	Username	N	The FIX connector username, as agreed with the exchange	String (30)
554	Password	N	The FIX connector password. No security exists without transport level encryption.	String (10)
1137	DefaultApplVerID	Y	Specifies the service pack release being applied by default to the message at the session level. The only valid value is '8' = FIX50SP1. This value will not be validated.	String
58	Text	N	Free format text string	String (30)
Standard Trailer		Y		

The FIX gateway accepts HeartBtInt (108) range which is configured by the exchange. If the client logon with HeartBtInt less than the minimum range value then the FIX gateway will set HeartBtInt to a minimum. Similarly, if the client logon with HeartBtInt greater than the maximum range value then the FIX gateway will set HeartBtInt to a maximum.

Please contact the exchange for the valid range.

4.1.2 Logout (5)

The logout message initiates or confirms the termination of a FIX session. Disconnection without the exchange of logout messages should be interpreted as an abnormal condition.

The logout format is as follows.

Table 8 – Logout

TAG	FIELD NAME	REQ'D	COMMENTS	FORMAT
Standard Header		Y	MsgType = 5	

TAG	FIELD NAME	REQ'D	COMMENTS	FORMAT
58	Text	N	Free format text string	String (30)
Standard Trailer		Y		

4.1.3 Reject (3)

The reject message should be issued when a message is received but cannot be properly processed due to a session-level rule violation. An example of when a reject may be appropriate would be the receipt of a message with invalid basic data (e.g. MsgType=&) which successfully passes de-encryption, CheckSum and BodyLength checks. As a rule, messages should be forwarded to the trading application for business level rejections whenever possible.

Rejected messages should be logged and the incoming sequence number incremented.

The reject format is as follows.

Table 9 – Reject

TAG	FIELD NAME	REQ'D	COMMENTS	FORMAT
Standard Header		Y	MsgType = 3	
45	RefSeqNum	Y	MsgSeqNum of rejected message	SeqNum
371	RefTagID	N	The tag number of the FIX field being referenced.	Int
372	RefMsgType	N	The MsgType of the FIX message being referenced.	String (2)
373	SessionRejectReason	N	Code to identify reason for a session-level Reject message.	Int
58	Text	N	Free format text string	String (200)
Standard Trailer		Y		

4.1.4 Resend Request (2)

The resend request is sent by the receiving application to initiate the retransmission of messages. This function is utilized if a sequence number gap is detected, if the receiving application lost a message, or as a function of the initialization process.

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

The resend request format is as follows.

Table 10 – Resend Request

TAG	FIELD NAME	REQ'D	COMMENTS	FORMAT
Standard Header		Y	MsgType = 2	
7	BeginSeqNo	Y		SeqNum
16	EndSeqNo	Y		SeqNum
Standard Trailer		Y		

4.1.5 Sequence Reset (Gap Fill) (4)

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

Gap Fill mode

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

During normal resend processing, the sending application may choose not to send a message (e.g. an aged order). During normal resend processing, a number of administrative messages are skipped and not resent (such as Heart Beats, Test Requests). Gap Fill mode is indicated by GapFillFlag (tag 123) field = "Y". If the GapFillFlag field is present (and equal to "Y"), the MsgSeqNum should conform to standard message sequencing rules (i.e. the MsgSeqNum of the Sequence Reset GapFill mode message should represent the beginning MsgSeqNum in the GapFill range because the remote side is expecting that next message sequence number).

Reset mode

Reset mode involves specifying an arbitrarily higher new sequence number to be expected by the receiver of the Sequence Reset-Reset message, and is used to establish a FIX session after an unrecoverable application failure.

Reset mode is indicated by the GapFillFlag (tag 123) field = "N" or if the field is omitted. The Sequence Reset format is as follows.

Table 11 – Sequence Reset

TAG	FIELD NAME	REQ'D	COMMENTS	FORMAT
Standard Header		Y	MsgType = 4	
123	GapFillFlag	N		Boolean
36	NewSeqNo	Y		SeqNum
Standard Trailer		Y		

4.1.6 Test Request (1)

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

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

Table 12 – Test Request

TAG	FIELD NAME	REQ'D	COMMENTS	FORMAT
Standard Header		Y	MsgType = 1	
112	TestReqID	Y		String (30)
Standard Trailer		Y		

4.1.7 Heartbeat (0)

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

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

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

The heartbeat format is as follows.

Table 13 – Heartbeat

TAG	FIELD NAME	REQ'D	COMMENTS	FORMAT
Standard Header		Y	MsgType = 0	
112	TestReqID	N	Required when the heartbeat is the result of a Test Request message.	String (30)
Standard Trailer		Y		

4.2 Infrastructure

4.2.1 Business Message Reject (j)

The Business Message Reject message can reject an application-level message which fulfils session-level rules and cannot be rejected via any other means. Note if the message fails a session-level rule (e.g. body length is incorrect), a session-level Reject message should be issued.

Table 14 – Business Message Reject

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
StandardHeader		Y	MsgType = j (lowercase)	
45	RefSeqNum	N	MsgSeqNum of rejected message	SeqNum
372	RefMsgType	Y	The MsgType of the FIX message being referenced.	String (2)
379	BusinessRejectRefID	N	The value of the business-level "ID" field on the message being referenced. Required unless the corresponding ID field (see list above) was not specified.	String (20)
380	BusinessRejectReason	Y	Code to identify reason for a Business Message Reject message. Code to identify reason for a Business Message Reject message.	Int
58	Text	N	Free format text string	String (200)

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
Standard Trailer		Y		

4.3 Application

4.3.1 New Order Single (D)

The new order message type is used by institutions wishing to electronically submit securities orders for execution.

Table 15 – New Order Single

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
StandardHeader		Y	MsgType = D	
11	ClOrdID	Y	Unique identifier for Order as assigned by the buy-side (institution, broker, intermediary etc.) (identified by SenderCompID (49) or OnBehalfOfCompID (5) as appropriate). Uniqueness must be guaranteed within a single trading day. Firms, particularly those which electronically submit multi-day orders, trade globally or throughout market close periods, should ensure uniqueness across days, for example by embedding a date within the ClOrdID field.	String (20)
Component block <Parties>		Y	Insert here the set of "Parties" (firm identification) fields. See *Table 30 – Parties Component* Block. 12 – Executing Trader (required) 14 – Giveup Clearing Firm (optional, `give up')	
Component block <Instrument>		Y	Insert here the set of "Instrument" (symbology) fields. See *Table 29 – Instrument Component Block.* Must include Symbol (55) and SecuritySubType (762).	
1	Account	Y	Trade Account.	String (14)
528	OrderCapacity	N	The capacity of the firm placing the valid order. This field is ignored.	Char
18	ExecInst	N	Instructions for order handling. Used to move an order to the private order book or into the market.	MultipleChar Value (1)
38	OrderQty	Y	Quantity ordered. This value represents the number of shares for equities.	Qty
40	OrdType	Y	Indicates the type of order.	Char
44	Price	Y/N	Required for all limit order types – not required for Market orders.	Price
54	Side	Y	Side of the market.	Char

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
60	TransactTime	Y	Time of order creation by Trader. This field is not processed by the Exchange nor is it used as a mechanism to place an order at a future time.	UTCTimeStamp
110	MinQty	Y/N	Specifies the minimum fill quantity.	Qty
59	TimeInForce	N	Specifies how long the order remains in effect. Absence of this field indicates a 'day' order.	Char
432	ExpireDate	Y/N	Conditionally required if TimeInForce = GTD	LocalMktDate
1138	DisplayQty	N	Replaces 'MaxFloor' and specifies the disclosed volume on hidden/iceberg orders. This is a V5.0 tag value.	Qty
58	Text	N	Free Text.	String (30)
Component block <TriggeringInstruction>		N	Insert here the set of "TriggeringInstruction" (symbology) fields. See *Table 31 – TriggeringInstruction Component Block*.	
Standard Trailer		Y		

4.3.2 New Order Cross (s)

The New Order Cross type is used to submit a cross order into a market. The cross order contains two order sides (a buy and a sell). The cross order is identified by its CrossID.

Table 16 – New Order Cross

TAG	FIELDNAME		REQ'D	COMMENTS	FORMAT
StandardHeader			Y	MsgType = s	
548	CrossID		Y	Identifier for a cross order. Must be unique during a given trading day.	String (20)
549	CrossType		Y	Type of cross being submitted to a market. Must be 1 (Cross AON).	Int
550	CrossPrioritization		Y	Indicates if one side or the other of a cross order should be prioritized. Must be 0 (None).	Int
Start of Component block, expanded in line < SideCrossOrdModGrp >					
552	NoSides		Y	Must be 2	NumInGrp
→	54	Side	Y	Side of order	Char
→	11	ClOrdID	Y	Unique identifier of the order as assigned by institution or by the intermediary with closest association with the investor.	String (20)
→	Component block <Parties>		Y	Insert here the set of "Parties" (firm identification) fields. See *Table 30 – Parties Component* Block.	

TAG	FIELDNAME		REQ'D	COMMENTS	FORMAT
				12 – Executing Trader (required) 14 – Giveup Clearing Firm (optional, `give up')	
→	1	Account	Y	Trade Account.	String (14)
→	38	OrderQty	Y	Quantity ordered. This value represents the number of shares for equities. Must be the same for both sides.	Qty
→	528	OrderCapacity	N	Designates the capacity of the firm placing the order. This field is ignored.	Char
→	58	Text	N	Free Text.	String (30)
End of Component block, expanded in line < SideCrossOrdModGrp >					
Component block <Instrument>			Y	Insert here the set of "Instrument" (symbology) fields. See *Table 29 – Instrument Component Block*. Must include Symbol (55) and SecuritySubType (762).	
40	OrdType		Y	Indicates the type of order. Must be 2 (Limit).	Char
44	Price		Y	Price of the cross. Required.	Price
60	TransactTime		Y	Time of order creation by Trader. This field is not processed by the Exchange nor is it used as a mechanism to place an order at a future time.	UTCTimeStamp
59	TimeInForce		Y	Specifies how long the order remains in effect. Must be 3 (IOC).	Char
Standard Trailer			Y		

4.3.3 Order Cancel Request (F)

The order cancel request message requests the cancellation of **all** of the remaining quantity of an existing order. Note that the Order Cancel/Replace Request should be used to partially cancel (reduce) an order. The request will only be accepted if the order can successfully be withdrawn from the Exchange without executing.

A cancel request is assigned a ClOrdID and is treated as a separate entity. If rejected, the ClOrdID of the cancel request will be sent in the Cancel Reject message, as well as the ClOrdID of the actual order in the OrigClOrdID field. The ClOrdID assigned to the cancel request must be unique amongst the ClOrdID assigned to regular orders and replacement orders.

The format of the cancel request message is:

Table 17 – Order Cancel Request

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
StandardHeader		Y	MsgType = F	
11	ClOrdID	Y	Unique identifier for Order as assigned by the buy-side (institution, broker, intermediary etc.) (identified by SenderCompID (49) or OnBehalfOfCompID (5) as appropriate). This identifier represents the unique identifier for the Order Cancel Request. Uniqueness must be guaranteed within a single trading day. Firms, particularly those which electronically submit multi-day orders, trade globally or throughout market close periods, should ensure uniqueness across days, for example by embedding a date within the ClOrdID field.	String (20)
37	OrderID	N	Unique order identifier as assigned by X-stream that identifies the Order to be changed.	String (18)
41	OrigClOrdID	Y/N	ClOrdID (11) of the previous non-rejected order (NOT the initial order of the day) when cancelling or replacing an order. Required when referring to orders that where electronically submitted over FIX or otherwise assigned a ClOrdID. Mandatory if OrderID (37) is not set.	String (20)
Component block <Parties>		N	Insert here the set of "Parties" (firm identification) fields. See *Table 30 – Parties Component* Block. 12 – Executing Trader (required)	
Component block <Instrument>		Y	Insert here the set of "Instrument" (symbology) fields. See *Table 29 – Instrument Component Block*. Must include Symbol (55) and SecuritySubType (762). These are ignored by the exchange.	
38	OrderQty	Y	Order quantity. This is ignored.	Qty
54	Side	Y	Side of the market. This is ignored.	Char
60	TransactTime	Y	Time this order request was initiated. This field is not processed by the Exchange nor is it used as a mechanism to cancel an order at a future time.	UTCTimeStamp
58	Text	N	Free Text.	String (30)
Standard Trailer		Y		

4.3.4 Order Cancel/Replace Request (G)

The order cancel/replace request is used to change the parameters of an existing order. All of the application-level fields in the original order should be retransmitted with the original values in the Order Cancel/Replace Request, except the fields that are being changed.

Do not use this message to cancel the remaining quantity of an outstanding order, use the Order Cancel Request message for this purpose.

Cancel/Replace will be used to change any valid attribute of an open order (i.e. reduce/increase quantity, change limit price, change instructions, etc.).

Table 18 – Order Cancel/Replace Request

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
StandardHeader		Y	MsgType = G	
11	ClOrdID	Y	Unique identifier for Order as assigned by the buy-side (institution, broker, intermediary etc.) (identified by SenderCompID (49) or OnBehalfOfCompID (5) as appropriate). Uniqueness must be guaranteed within a single trading day. Note that this identifier will be used in ClOrdID field of the Cancel Reject message if the replacement request is rejected.	String (20)
37	OrderID	N	Unique identifier of most recent order as assigned by the Exchange.	String (18)
41	OrigClOrdID	Y/N	ClOrdID(11) of the previous non-rejected order (NOT the initial order of the day) when cancelling or replacing an order. Required when referring to orders that where electronically submitted over FIX or otherwise assigned a ClOrdID. Mandatory if OrderID (37) is not set.	String (20)
Component block <Parties>		N	Insert here the set of "Parties" (firm identification) fields. See *Table 30 – Parties Component* Block. 12 – Executing Trader (required) 14 – Giveup Clearing Firm (optional, `give up')	
Component block <Instrument>		Y	Insert here the set of "Instrument" (symbology) fields. See *Table 29 – Instrument Component Block*. Must match original order.	
1	Account	N	Trade Account.	String (14)
528	OrderCapacity	N	The capacity of the firm placing the valid order. This field is ignored.	Char
18	ExecInst	N	Instructions for order handling. Used to move an order to the private order book or	MultipleChar Value (1)

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
			into the market.	
38	OrderQty	Y	Quantity ordered. This value represents the number of shares for equities.	Qty
40	OrdType	Y	Indicates the type of order to change to (must follow rules of the Exchange).	Char
44	Price	Y/N	Required for all limit order types – not required for Market orders.	Price
54	Side	Y	Side of the market. The value will not be validated.	Char
60	TransactTime	Y	Time of execution/order creation. This field is not processed by the Exchange nor is it used as a mechanism to amend an order at a future time.	UTCTimeStamp
110	MinQty	N	Specifies the minimum fill quantity.	Qty
59	TimeInForce	N	Specifies how long the order remains in effect.	Char
432	ExpireDate	Y/N	Conditionally required if TimeInForce = GTD	LocalMktDate
1138	DisplayQty	N	Replaces 'MaxFloor' and specifies the disclosed volume on hidden/iceberg orders. This is a V5.0 tag value.	Qty
58	Text	N	Free Text.	String (30)
Component block <TriggeringInstruction>		N	Insert here the set of "TriggeringInstruction" (symbology) fields. See *Table 31 – TriggeringInstruction Component Block*.	
Standard Trailer		Y		

4.3.5 Execution Report (8)

The execution report message is used to:

1. Confirm the receipt of an order
2. Confirm changes to an existing order (i.e. accept cancel and replace requests)
3. Report order status information (including GTC & GTD order restatement at market start)
4. Report fill information on working orders
5. Report fill information on tradeable or restricted tradeable quotes
6. Report on rejected order
7. Report on orders activated/deactivated by Market Control

Table 19, entitled 'Execution Report Returned Tags Based On Scenario' follows the Execution Report message description and provides information on which tags are returned in an Execution Report message based on various order management scenarios.

Table 19 – Execution Report

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
StandardHeader		Y	MsgType = 8	
11	ClOrdID	Y/N	Unique identifier for Order as assigned by the buy-side (institution, broker, intermediary etc.) (identified by SenderCompID (49) or OnBehalfOfCompID (5) as appropriate). Uniqueness must be guaranteed within a single trading day. Firms, particularly those which electronically submit multi-day orders, trade globally or throughout market close periods, should ensure uniqueness across days, for example by embedding a date within the ClOrdID field. Required when referring to orders that where electronically submitted over FIX or otherwise assigned a ClOrdID(11).	String (20)
17	ExecID	Y	Unique identifier of execution message as assigned by the Exchange.	String (18)
37	OrderID	Y	OrderID is required to be unique for each chain of orders.	String (18)
41	OrigClOrdID	Y/N	Conditionally required for response to a Cancel or Cancel/Replace request	String (20)
548	CrossID	N	Identifier for a cross order. Must be unique during a given trading day.	String (20)
150	ExecType	Y	Type of Execution being reported. Describes the specific ExecutionRpt (i.e. Pending Cancel) while OrdStatus (39) will always identify the current order status (i.e. Partially Filled).	Char
Component block <Parties>		N	Insert here the set of "Parties" (firm identification) fields. See *Table 30 – Parties Component* Block. 11 = Order Origination Trader 36 = Entering trader 12 = Executing Trader 1 = Executing firm 14 - Giveup Clearing Firm For trade ExecutionReport: 17 = ContraFirm 20 = Contra Giveup Clearing Firm	
Component block <Instrument>		Y	Insert here the set of "Instrument" (symbology) fields. See *Table 29 – Instrument Component Block*. Must include Symbol (55) and	

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
			SecuritySubType (762).	
1	Account	Y	Trade Account.	String (14)
528	OrderCapacity	N	The capacity of the firm placing the valid order.	Char
6	AvgPx	N	Calculated average price for all fills on this order during the day. If not available then the value reflects the trade price for this fill.	Price
14	CumQty	Y	Total matched quantity.	Qty
31	LastPx	N	Price of this fill.	Price
32	LastQty	N	Quantity (e.g. shares) bought/sold on this fill.	Qty
38	OrderQty	N	Quantity ordered.	Qty
110	MinQty	N	Minimum fill quantity.	Qty
39	OrdStatus	Y	Describes the current state of an order.	Char
40	OrdType	N	OrderType	Char
44	Price	N	Price on order.	Price
54	Side	Y	Side of order.	Char
59	TimeInForce	N	Indicates time in force techniques that are valid for the specified market segment. Absence of this field indicates a 'day' order.	Char
60	TransactTime	Y	Time of execution/order creation (expressed in Universal Time Coordinated (UTC), also known as GMT.	UTCTimeStamp
75	TradeDate	N	Indicates date of trade referenced in this message in YYYYMMDD format.	LocalMktDate
432	ExpireDate	Y/N	Conditionally required if TimeInForce = GTD.	LocalMktDate
64	SettlDate	N	Specific date of trade settlement Settlement Date is in YYYYMMDD format.	LocalMktDate
103	OrdRejReason	N	For optional use with ExecType = 8 (Rejected). Code to identify reason for order rejection.	Int
151	LeavesQty	Y	Quantities open for further execution. If the OrdStatus is Cancelled, DoneForTheDay, Expired or Rejected (in which case the order is no longer active) then LeavesQty could be 0, otherwise LeavesQty = OrderQty – CumQty.	Qty

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
381	GrossTradeAmt	N	Total amount traded expressed in units of currency. Calculated on Price*LastQty	Amt
880	TrdMatchID	N	Identifier assigned by the trading system for a trade. This is the X-stream trade id.	String (21)
1057	AggressorIndicator	N	Used to identify whether the order initiator is an aggressor or not in the trade. Valid during continuous trading only. This is a V5.0 tag value.	Boolean
1138	DisplayQty	N	Replaces 'MaxFloor' and specifies the disclosed volume on hidden/iceberg orders. This is a V5.0 tag value. This field is always returned as part of a fill or partial fill for all order types. For non-hidden/iceberg orders this field will contain the same value as LeavesQty (151).	Qty
58	Text	N	Free Text. On an error condition, this will specify X-stream generated error message.	String (200)
Component block <TriggeringInstruction>		N	Insert here the set of "TriggeringInstruction" (symbology) fields. See *Table 31 – TriggeringInstruction Component Block*.	
797	CopyMsgIndicator	N	Indicates Drop Copy	Boolean
Standard Trailer		Y		

4.3.6 Order Cancel Reject (9)

The order cancel reject message is issued by the Exchange upon receipt of a cancel request or cancel/replace request message which cannot be honoured. Filled orders cannot be changed.

When rejecting a Cancel/Replace Request (or Cancel Request), the Cancel Reject message should provide the ClOrdID which was specified on the Cancel/Replace Request (or Cancel Request) message for identification, and the OrigClOrdId should be that of the last accepted order except in the case of CxlRejReason = "Other".

Refer to the Text (58) field for specific information on the reason for the rejection.

The order cancel reject message format is as follows.

Table 20 – Order Cancel Reject

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
StandardHeader		Y	MsgType = 9	
11	ClOrdID	Y	Unique identifier for Order as assigned by sell-side (e.g. exchange, ECN). If CxlRejReason="Unknown order" specify "NONE".	String (20)
37	OrderID	Y	Unique identifier of most recent order as assigned	String (18)

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
			by the Exchange. If CxlRejReason="Unknown order", specify "NONE".	
39	OrdStatus	Y	Describes the current status of the order	Char
41	OrigClOrdID	Y/N	ClOrdID (11) of the previous non-rejected order (NOT the initial order of the day) when cancelling or replacing an order. Required when referring to orders that where electronically submitted over FIX or otherwise assigned a ClOrdID.	String (20)
60	TransactTime	N	Time of order cancellation request rejection by the Exchange.	UTCTimeStamp
102	CxlRejReason	Y	Code to identify reason for cancel rejection. Only '99' (Other) will be returned. Refer to 'text' (58) for exact reason for rejection.	Int
434	CxlRejResponseTo	Y	Identifies the type of request that a Cancel Reject is in response to.	Char
58	Text	N	Specify X-stream generated error message.	String (200)
Standard Trailer		Y		

5 Order Management

5.1 Unique ClOrdID (11)

X-stream may not check for uniqueness of ClOrdID (11) on New Order Single, Order Cancel/Replace Request and Order Cancel Request messages. Firms submitting order transactions via FIX interface must ensure a unique ClOrdID is entered on these transactions. If a firm has multiple FIX connections, then ClOrdID should be unique across all FIX connections for that firm.

5.2 Order Identification

A FIX order is identified either by its current client assigned ClOrdID (11), or by the X-stream OrderID (37).

OrderID should be used to identify an order across FIX connections, even if they belong to the same firm. The X-stream OrderID is guaranteed to be unique for all order durations including over-night orders.

If OrderID is used on an OrderCancelRequest or OrderCancel/ReplaceRequest message, OrigClOrdID (41) should be set to "NONE". OrderID is unique for every order.

Note: OrderID(37) may be changed by the exchange after order amendment.

5.3 Order Modification via Order Cancel/Replace Request

Order modification is accomplished through the use of the Order Cancel/Replace Request message. An order modification is not a delta change to order instructions. The values set in the Cancel Replace represent the requested new order state. An Execution Report will relay the new state of the order.

A new ClOrdID (11) must be provided in the Order Cancel/Replace Request message.

5.3.1 Order Attributes allowed to change

Although the FIX protocol allows for virtually all of the Order attributes to be changed, there are a defined number of amendable fields that X-stream allows.

The following attributes are allowed to change:

- ClOrdID (11)
- OrderQty (38)
- DisplayQty (111)
- Price (44)
- OrdType (40)
- TimeInForce (59)
- ExpireDate (432)
- PartyID (448) where PartyRole (452) = 14 (Giveup Clearing Firm)
- Account (1)
- MinQty (110)
- Text (58)

- TriggerPrice (1102)
- ExecInst (18)
- Side (54) – from buy to BuyIn and vice versa and from sell to short sell and vice versa

Note: Any change to the price or trigger price of an order, or increasing quantities will result in the order losing its priority in the market.

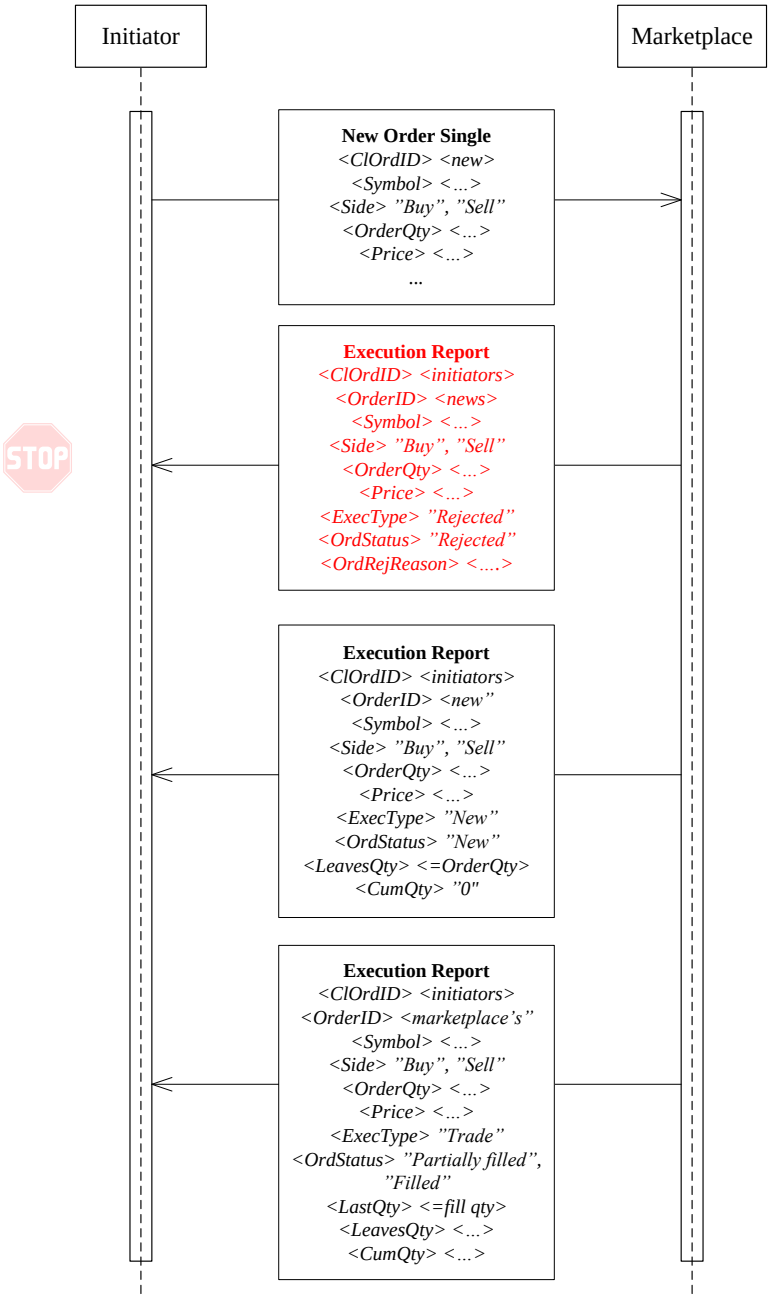
5.4 Order Cancellation

- If the user wishes to cancel a single previously sent order, the Order Cancel Request message is used.
- Execution Reports are issued relaying the status of every canceled order.
- In some cases orders may be cancelled in the system without prior request by the user. These will be sent as unsolicited Execution Reports to the client.
- The system will generate cancel messages (Execution Report – IOC Order Cancel) for every IOC order.

5.5 Workflows

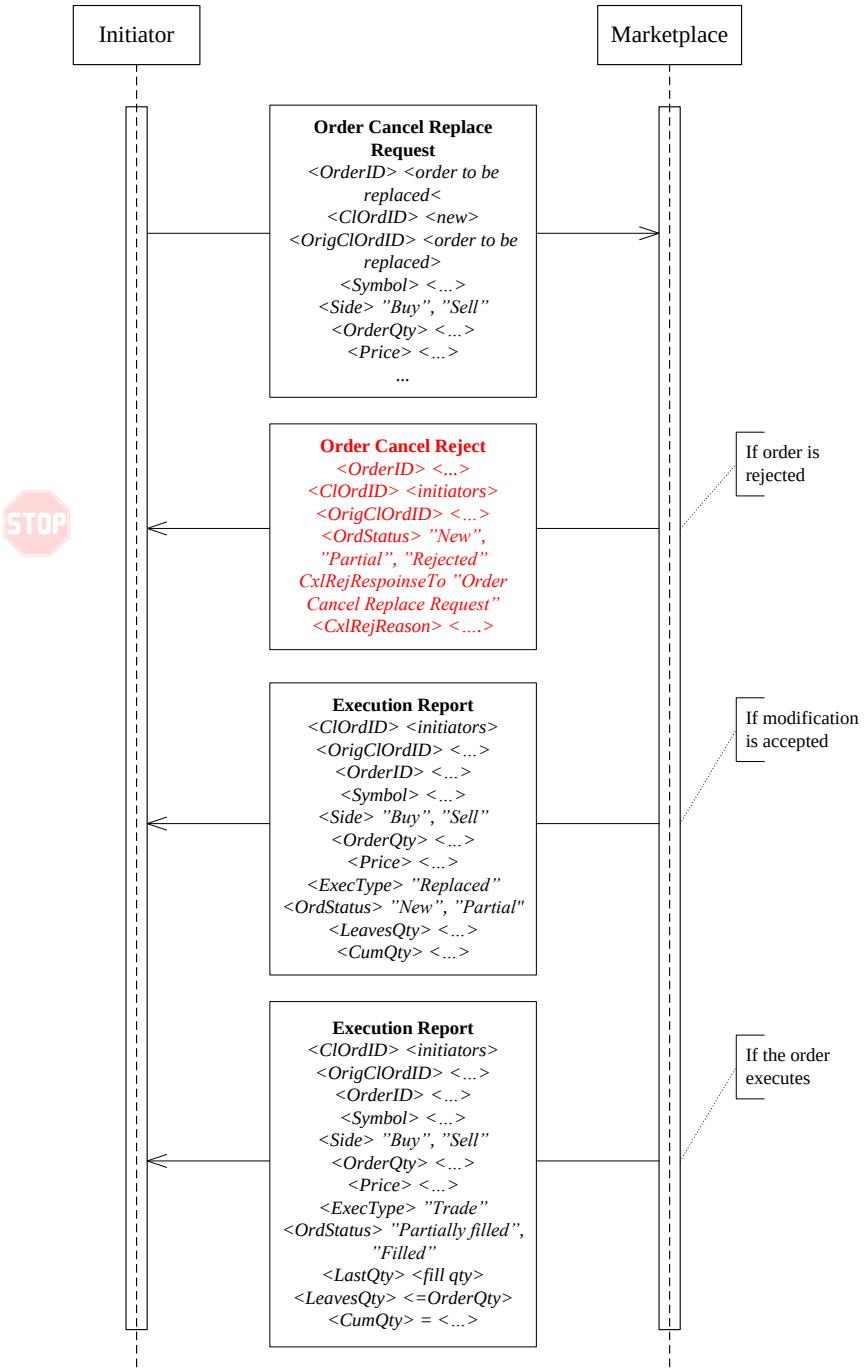
5.5.1 Entering of an New Order

Figure 1 – New Order Entry Workflow



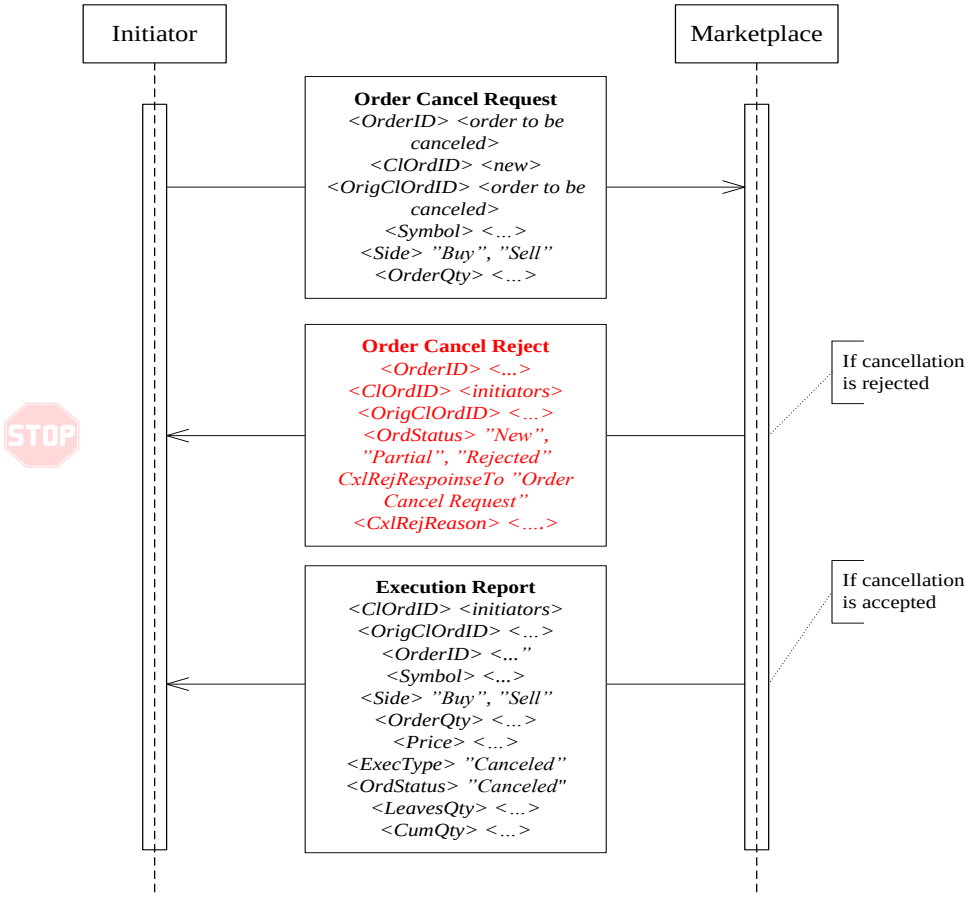
5.5.2 Modification of an Order

Figure 2 – Order Modification Workflow



5.5.3 Order Cancellation

Figure 3 – Order Cancellation Workflow



5.5.4 Order Status

Order state changes are divulged in Execution Report messages. Every state change is communicated in a separate Execution Report. The OrdStatus (39) field specifies the state.

Figure 4 – Order Status States

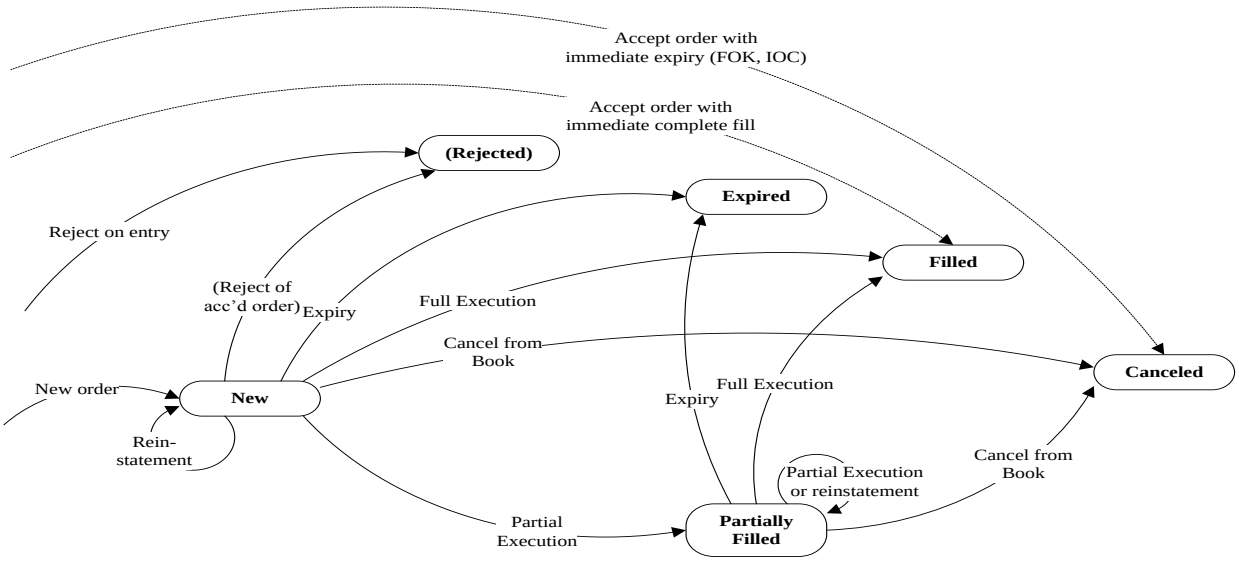


Table 21 – Execution Report Returned Tags Based On Scenario

	ClOrdID (11)	*ExecID (17)*	*OrderID (37)*	*OrigClOrdID (41)*	*ExecType (150)*	*OrderCapacity (528)*	*Parties*	*Instrument*	*Triggering Instruction*	*Account (1)*	*AvgPX (6)*	*CumQTY (14)*	*LastPX (31)*	*LastQTY (32)*	*MinQty (110)*	*OrderQty (38)*	*OrdStatus (39)*	*OrdType (40)*	*Price (44)*	*Side (54)*	*TimeInForce (59)*	*TransactTime (60)*	*TradeDate (75)*	*DisplayQty (1138)*	*ExpireDate (432)*	*SettleDate (64)*	*CxlRejReason (102)*	*CxlRejReaResponsetTo (434)*	*OrdRejReason (103)*	*LeavesQTY (151)*	*GrossTradeAmt (381)*	*TrdMatchID (880)*	*AggressorIndicator (1057)*	*Text (58)*
New Order Single	R	R	R		R	C	R	R	C	C		R	C	C	C	R	R	R	R	R	C	R		C	C					R				C
Order Cancel Pending	C	R	R	R	R			R				R				R	R			R	C			C						R				C
Order Cancel / Replace	C	R	R	R	R	C	R	R	C	C		R	C	C	C		R		R	R	C	R		C	C					R				C
Order Cancel / Replace Reject	C		R	R				R									R			R	C	R		C	C		R	R	R						C
Order Cancelled	C	R	R	R	R	C		R				R	C	C		R	R	C	R	R	C	R		C						R				C
Order Filled	R	R	R		R		R	R	C	C	R	R	R	R	C	R	R	C	R	R	C	R		C		R				R	R	R	R	C
Order Partially Filled	R	R	R		R		R	R	C	C	R	R	R	R	C	R	R	C	R	R	C	R		C	C	R				R	R	R	R	C
Order Rejected	C	R	R		R			R				R				C	R			R	C	R		C	C				R				C

C = Conditional - Based on input transaction/query (or error condition)

R = Returned as part of Execution Report message

6 Reference Data

The reference data category consists of the following message:

- News

6.1 News (B)

The news message is a general free format message between the participant and the exchange. The message contains flags to identify the news item's urgency and to allow sorting by Subject Company (symbol).

Table 22 – News

TAG	FIELDNAME		REQ'D	COMMENTS	FORMAT
StandardHeader			Y	MsgType = B	
42	OrigTime		N	Time if message origination. Always expressed in UTC time.	UTCTimeStamp
61	Urgency		N	Urgency Flag.	Char
148	Headline		Y	Specifies the headline text	String
1473	NewsCategory		N	Category of news message.	int
358	EncodedHeadlineLen		N	Byte length of encoded (non-ASCII characters) EncodedHeadline (359) field.	Length
359	EncodedHeadline		N	Encoded (non-ASCII characters) representation of the Headline (148) field in the encoded format specified via the MessageEncoding (347) field. If used, the ASCII (English) representation will also be specified in the Headline field.	Data
Start of Component block, expanded in line < InstrmtGrp >					
146	NoRelatedSym		N	Specifies the number of repeating symbols (instruments) specified	NumInGroup
→	55	Symbol	N	Unique marketplace assigned identifier number for an order book.	String
End of Component block, expanded in line < InstrmtGrp >					
Start of Component block, expanded in line < LinesOfTextGroup >					
33	NoLinesOfText		Y	Specifies the number of repeating lines of text.	NumInGrp
→	58	Text	Y	Free format text string	String
→	354	Encoded TextLen	N	Must be set if EncodedText field is specified and must immediately precede it. Byte Length of encoded (non-ASCII) characters.	Length
→	355	Encoded Text	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding.	Data

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
End of Component block, expanded in line < LinesOfTextGroup >				
149	URLLink	N	A URL (Uniform Resource Locator) link to additional information (i.e. http://www.XYZ.com/research.html)	String
StandardTrailer		Y		

7 Trade Capture Reporting

Trade Capture reports are used for a variety of purposes and include:

- Relaying trades to counterparties of the trade. Those messages are outbound from the exchange.
- Reporting of privately negotiated trades (Block Trades). Those messages may be inbound or outbound.

7.1 Trade Capture Messages

The Trade Capture category of messages consists of the following:

- Trade Capture Report
- Trade Capture Report Ack

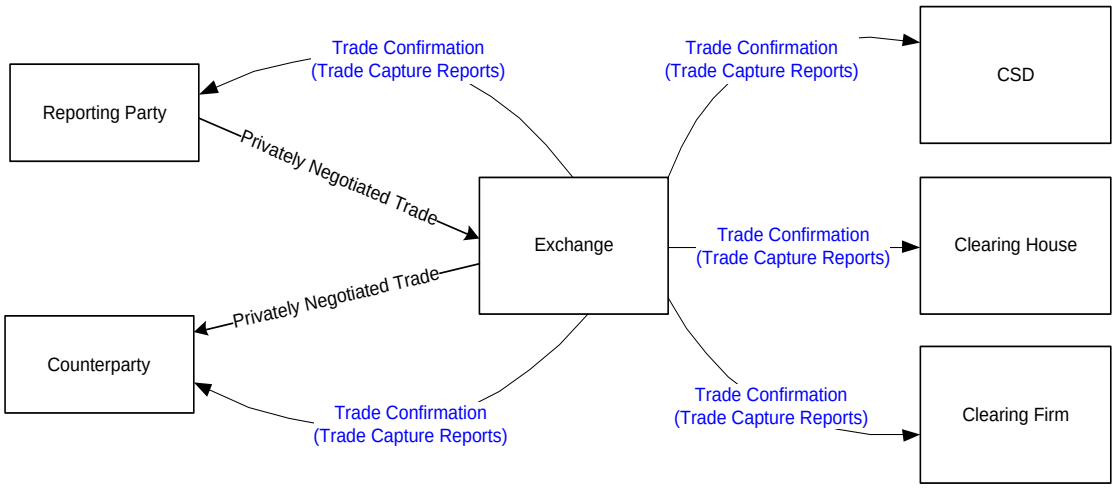
7.2 Workflows

7.2.1 Trade Capture Diagram for Privately Negotiated Trade, One-Party Report for Pass-through to Counterparty

The deal is struck between two parties, one of whom has an obligation to report the trade. The counterparty does not have agreement with the reporting party, so he must acknowledge the trade. The reporting party sends the trade report to the market. The market informs the counterparty of the report and the counterparty then accepts the trade. The exchange confirms the Confirmed Trade to all involved parties. The FIX Trade Capture Report is used for all involved messages.

When submitting a one sided (crossing) Trade Capture Report, the submitter must fill in details of both sides in two TrdCapRptSideGrps. The Exchange will either confirm the TradeCaptureReport, or reject with TradeCaptureReportAck.

Figure 5 – Diagram for Privately Negotiated Trade



7.2.2 Workflow for One-Party Report for Pass-through to Counterparty

TradeReportID (571) is used to identify a trade capture report. The initiator, the exchange and the counterparty maintain their own set of TradeReportID (571) values. The TradeReportID used by the exchange will not change during the whole TradeCaptureReport life cycle. TradeReportRefID (572) is used to identify the TradeReportID (571) from the previously received TradeCaptureReport.

1. The initiator (seller) sends a TradeCaptureReport (AE) to the exchange with a unique TradeReportID (571). The uniqueness of TradeReportID will not be checked by the exchange.
2. The exchange will send to the initiator a TradeCaptureReportAck (AR) with the same TradeReportID (571) as received from the initiator, either rejecting or accepting their TradeCaptureReport.
3. If accepted, the exchange will also send to the initiator a TradeCaptureReport (AE) with a new, exchange assigned TradeReportID (571), a new ExecID (17) and a TradeReportRefID (572) set to the initiator's TradeReportID (571). The exchange will also send a TradeCaptureReport (AE) to the counterparty, with a new TradeReportID (571) and a new ExecID (17). The TradeReportID (571) will be different for buyer and seller, however the ExecID (17) will be the same.
4. Both initiator and counterparty can withdraw the TradeCaptureReport with a new TradeReportID (571), and a TradeReportRefID (572) set to the same TradeReportID (571) as received in step 3. In response, the exchange will send a TradeCaptureReport (AE) with the same TradeReportID (571) as in step 3.
5. The counterparty can confirm the TradeCaptureReport with a new TradeReportID (571), and a TradeReportRefID (572) set to the same TradeReportID (571) as received in step 3. In response, the exchange will send a TradeCaptureReport to both parties, with the same TradeReportID (571) as in step 3. The confirmed TradeCaptureReport will have a new TrdMatchID (880) – this is the unique exchange trade identifier.
6. If both initiator and counterparty do not respond within a timeout period then the TradeCaptureReport (AE) will be cancelled. The exchange will send a TradeCaptureReport (AE) with the same TradeReportID (571) as in step 3, and with TradeReportType (856) set to "Defaulted".

Figure 6 - One-Party Report for Pass-through to Counterparty

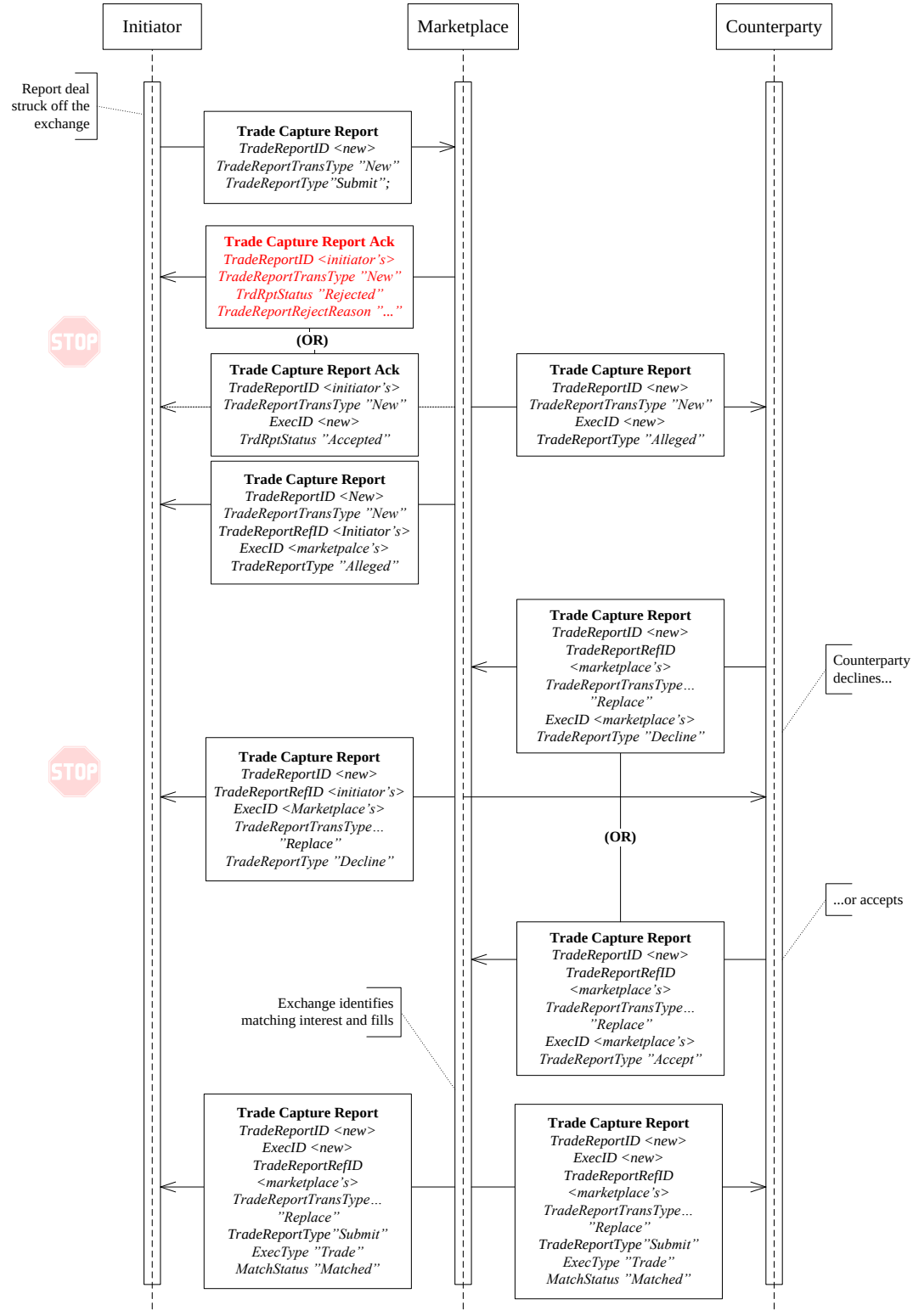
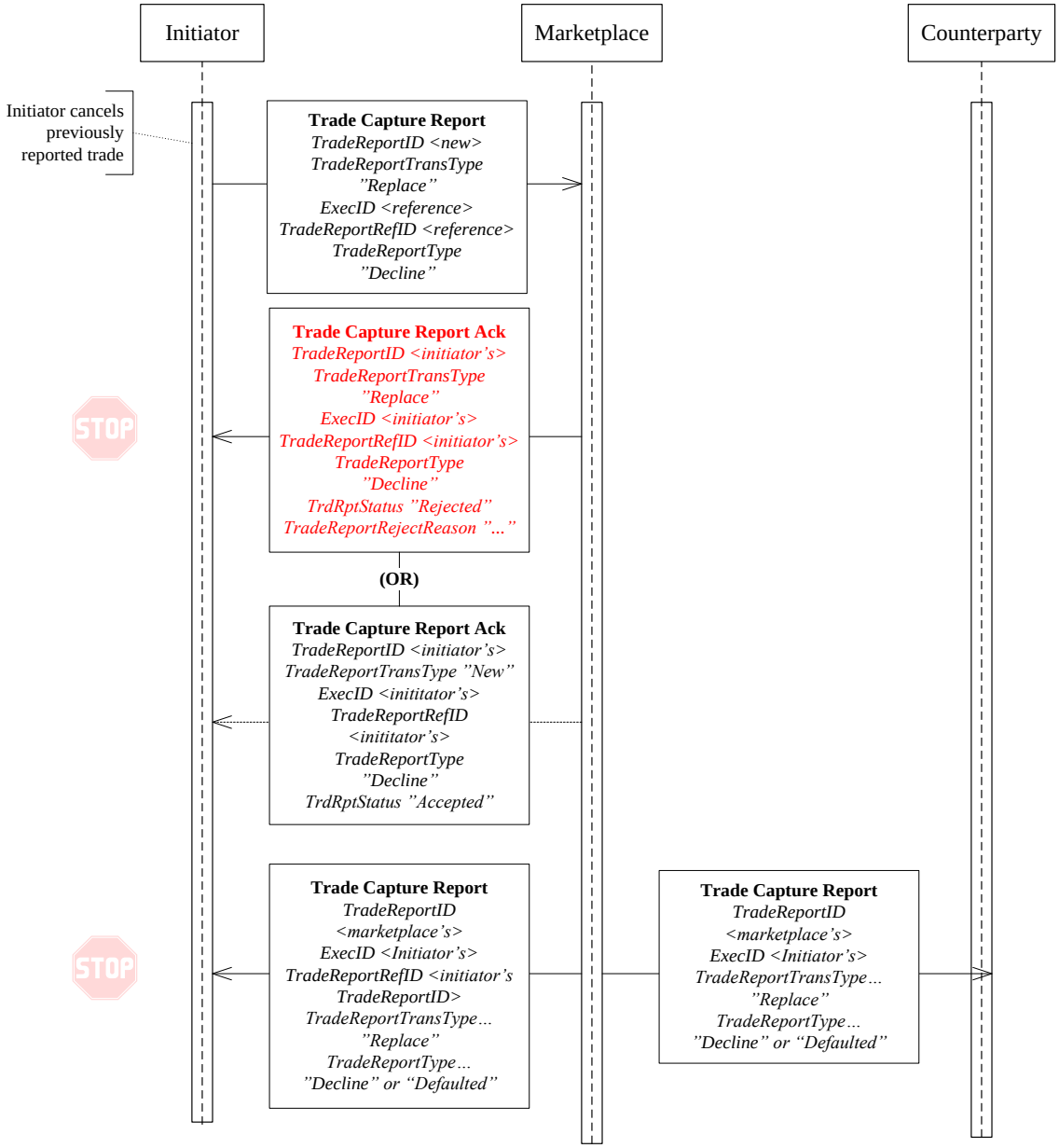


Figure 7 - One-Party cancel Report (or time out) before counterparty confirms



7.2.3 Workflow for Cross Report

TradeReportID (571) is used to identify a trade capture report. The initiator and the exchange maintain their own set of TradeReportID (571) values. The TradeReportID used by the exchange will not change during the whole TradeCaptureReport life cycle. TradeReportRefID (572) is used to identify the TradeReportID (571) from the previously received TradeCaptureReport.

1. The initiator sends a TradeCaptureReport (AE) to the exchange with a unique TradeReportID (571). The uniqueness of TradeReportID will not be checked by the exchange.
2. The exchange will send to the initiator a TradeCaptureReportAck (AR) with the same TradeReportID (571) as received from the initiator, either rejecting or accepting their TradeCaptureReport.
3. If accepted (match confirmed), the exchange will also send to the initiator a TradeCaptureReport (AE) with a new, exchange assigned TradeReportID (571), a new ExecID (17) and a TradeReportRefID (572) set to the initiator's TradeReportID (571). The confirmed TradeCaptureReport will have a new TrdMatchID (880) – this is the unique exchange trade identifier.

7.3 Trade Capture Report (AE)

The Trade Capture Report message can be:

- Sent by an initiator to the exchange to report a privately negotiated trade (Block Trade)
- Sent by the exchange to parties to inform them that the privately negotiated trade has been initiated
- Sent by a counterparty to the exchange to confirm or decline the privately negotiated trade
- Sent by the exchange to parties to inform them of the confirmed or declined privately negotiated trade

7.3.1 Submitting a Trade Capture Report to the Exchange

The Intiator should send a Trade Capture Report (AE) to the exchange to report a privately negotiated trade (Block Trade).

Table 23 – Trade Capture Report

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
StandardHeader		Y	MsgType = AE	
571	TradeReportID	Y	Unique identifier for the Trade Capture Report.	String (20)
487	TradeReportTransType	N	Identifies Trade Report message transaction type 0 = New	Int
856	TradeReportType	N	Type of Trade Report 0 = Submit	Int
Component block <Instrument>		Y	Insert here the set of "Instrument" (symbology) fields. See *Table 29 – Instrument Component Block*. Must include Symbol (55) and	

TAG	FIELDNAME		REQ'D	COMMENTS	FORMAT
				SecuritySubType (762).	
31	LastPx		Y	Trade Price.	Price
32	LastQty		Y	Trade Quantity	Qty
60	TransactTime		N	Time the transaction represented by this Trade Capture Report occurred	UTCTime Stamp
75	TradeDate		N	Date of the trade being reported. This is ignored by the exchange.	LocalMkt Date
64	SettlDate		N	Specific date of trade settlement (SettlementDate) in YYYYMMDD format.	LocalMkt Date
Start of Component block, expanded in line < TrdCapRptSideGrp >					
552	NoSides		Y	Number of sides 1 – For two party trade capture report. 2 – For crossing trade capture report.	Int
→	54	Side	Y	Side of order	Char
→	Component block <Parties>		Y	Insert here the set of "Parties" (firm identification) fields. See *Table 30 – Parties Component* Block. 12 – Executing Trader (required, buy side value used only for crossing) 14 – Giveup Clearing Firm (optional, `give up') 37 = Contra trader (optional; used for two party trade capture report to specify the counterparty) 20 = Contra Giveup Clearing Firm executing firm (optional; 'give-up', used for two party trade capture report)	
→	1	Account	Y	Trade Account.	String (14)
→	528	OrderCapacity	N	Designates the capacity of the firm placing the order. This field is ignored.	Char
→	58	Text	N	Free format text.	String (30)
End of Component block, expanded in line < TrdCapRptSideGrp >					
StandardTrailer			Y		

7.3.2 Exchange reporting a Trade Capture Report to Parties

The exchange sends a Trade Capture Report (AE) to all involved parties to inform them that a privately negotiated trade (Block Trade) has either been initiated, confirmed or declined.

Table 24 – Trade Capture Report

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
StandardHeader		Y	MsgType = AE	
571	TradeReportID	Y	Unique identifier for the Trade Capture Report.	String (20)
17	ExecID	N	Exchanged assigned Execution ID	String (18)
487	TradeReportTransType	N	Identifies Trade Report message transaction type 0 = New 1 = Cancel 2 = Replace	Int
856	TradeReportType	N	Type of Trade Report 0 = Submit 1 = Alleged 3 = Decline 6 = Trade Report Cancel	Int
828	TrdType	N	Type of Trade. 0 = Regular Block Sale 1 = Special Block Sale 52 = Trade with Impact (Market Control) 46 = Trade without Impact (Market Control)	Int
150	ExecType	N	Type of Execution being reported. F – Trade H – Trade Cancel	Char
572	TradeReportRefID	N	The TradeReportID that is being referenced for some action, such as confirmation or cancellation. Refer to *Figure 6* and workflow section above for detail scenario when this tag is used.	String (20)
573	MatchStatus	N	The status of this trade with respect to matching or comparison	Char
574	MatchType	N	The point in the matching process at which this trade was matched.	String (1)
Component block <Instrument>		Y	Insert here the set of "Instrument" (symbology) fields. See *Table 29 – Instrument Component Block*. Must include Symbol (55) and SecuritySubType (762).	
31	LastPx	Y	Trade Price.	Price
32	LastQty	Y	Trade Quantity	Qty

TAG	FIELDNAME		REQ'D	COMMENTS	FORMAT
60	TransactTime		N	Time the transaction represented by this Trade Capture Report occurred	UTCTime Stamp
75	TradeDate		N	Date of this trade.	LocalMkt Date
64	SettlDate		N	Specific date of trade settlement (SettlementDate) in YYYYMMDD format.	LocalMkt Date
381	GrossTradeAmt		N	Total amount traded expressed in units of currency. Includes accrued Interest for convertible bonds and fixed income.	Amt
880	TrdMatchID		N	Identifier assigned by the trading system for a trade.	String (21)
Start of Component block, expanded in line < TrdCapRptSideGrp >					
552	NoSides		Y	Number of sides. Should be 2.	Int
→	54	Side	Y	Side of order	Char
→	Component block <Parties>		Y	Insert here the set of "Parties" (firm identification) fields. See *Table 30 – Parties Component* Block. 36 = Entering trader 12 – Executing Trader (required) 14 – Giveup Clearing Firm (optional, `give up') 7 – Entering Firm 37 = Contra trader 20 = Contra Giveup Clearing Firm executing firm ('give-up') 17 = Contra firm	
→	1	Account	N	Trade Account.	String (14)
→	528	OrderCapacity	N	Designates the capacity of the firm placing the order	Char
→	58	Text		Free format text.	String (30)
End of Component block, expanded in line < TrdCapRptSideGrp >					
797	CopyMsgIndicator		N	Indicates Drop Copy	Boolean
StandardTrailer			Y		

7.3.3 Confirm or withdraw/decline a Trade Capture Report to the Exchange

A counterparty should send a Trade Capture Report (AE) to the exchange to either confirm or decline a pending privately negotiated trade (Block Trade).

Table 25 – Trade Capture Report to confirm / decline

TAG	FIELDNAME		REQ'D	COMMENTS	FORMAT
StandardHeader			Y	MsgType = AE	
571	TradeReportID		Y	Unique identifier for the Trade Capture Report.	String (20)
487	TradeReportTransType		N	Identifies Trade Report message transaction type 2 = Replace	Int
572	TradeReportRefID		Y	The TradeReportID that is being referenced for some action, such as confirmation or cancellation.	String (20)
856	TradeReportType		N	Type of Trade Report 2 = Accept 3 = Decline	Int
Component block <Instrument>			Y	Insert here the set of "Instrument" (symbology) fields. See *Table 29 – Instrument Component Block*. Not verified by the exchange.	
31	LastPx		Y	Trade Price. Not verified by the exchange.	Price
32	LastQty		Y	Trade Quantity. Not verified by the exchange.	Qty
60	TransactTime		N	Time the transaction represented by this Trade Capture Report occurred	UTCTime Stamp
75	TradeDate		N	Date of the trade being reported. This is ignored by the exchange.	LocalMkt Date
64	SettlDate		N	Specific date of trade settlement (SettlementDate) in YYYYMMDD format.	LocalMkt Date
Start of Component block, expanded in line < TrdCapRptSideGrp >					
552	NoSides		Y	Number of sides. Should be 1.	Int
→	54	Side	Y	Side of order. Not verified by the exchange.	Char
→	Component block <Parties>		N	Insert here the set of "Parties" (firm identification) fields for confirming a Trade Capture Report. See *Table 30 – Parties Component* Block. 12 – Executing Trader (required) 14 – Giveup Clearing Firm (optional)	
→	1	Account	N	Trade Account. Required for confirming a Trade Capture Report.	String (14)
→	528	OrderCapacity	N	Designates the capacity of the firm placing the order. This field is ignored.	Char
→	58	Text	N	Free format text.	String (30)

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
End of Component block, expanded in line < TrdCapRptSideGrp >				
StandardTrailer		Y		

7.4 Trade Capture Report Ack (AR)

The Trade Capture Report Ack message can be:

- Sent by the exchange to acknowledge trade capture reports received from a counterparty
- Sent by the exchange to reject a trade capture report received from a counterparty

Table 26 – Trade Capture Report Ack

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
StandardHeader		Y	MsgType = AR	
571	TradeReportID	Y	Unique identifier for the Trade Capture Report	String (20)
487	TradeReportTransType	N	Identifies Trade Report message transaction type 0 = New 1 = Cancel 2 = Replace	Int
856	TradeReportType	N	0 = Submit 1 = Alleged 2 = Accept 3 = Decline	
939	TrdRptStatus	N	0 = Accepted 1 = Rejected	Int
17	ExecID	N	Exchanged assigned Execution ID (Trade Identifier)	String (18)
60	TransactTime	N	Time the transaction represented by this Trade Capture Report Ack occurred	UTCTime Stamp
751	TradeReportRejectReason	N	Reason for Rejection of Trade Report	int
572	TradeReportRefID	N	The TradeReportID that is being referenced for some action, such as confirmation or cancellation	String (20)
58	Text	N	If TradeReportRejectReason is set, text of reason	String (200)
StandardTrailer		Y		

Appendix A - Standard Header and Trailer

A.1 Standard Header

The standard message header format is as follows.

Table 27 – Standard Message Header

TAG	FIELD NAME	REQ'D	COMMENTS	FORMAT
8	BeginString	Y	FIXT.1.1 (always unencrypted, must be first field in message)	String
9	BodyLength	Y	(Always unencrypted, must be second field in message)	Length
35	MsgType	Y	(Always unencrypted, must be third field in message)	String
1128	ApplVerID	N	Specifies the service pack release being applied at the message level. The only valid value is '8' = FIX50SP1	String
49	SenderCompID	Y	(Always unencrypted). Identifies the firm sending the message.	String
56	TargetCompID	Y	(Always unencrypted). Identifies the firm receiving the message.	String
115	OnBehalfOfCompID	N	Trading partner company ID used when sending messages via a third party (Can be embedded within encrypted data section). Not supported.	String
116	OnBehalfOfSubID	N	Trading partner SubID used when delivering messages via a third party (Can be embedded within encrypted data section). Not supported.	String
144	OnBehalfOfLocationID	N	Trading partner LocationID (i.e. geographic location and/or desk) used when delivering messages via a third party. (Can be embedded within encrypted data section). Not supported.	String
128	DeliverToCompID	N	Trading partner company ID used when sending messages via a third party (Can be embedded within encrypted data section). Not supported.	String
34	MsgSeqNum	Y	(Can be embedded within encrypted data section.)	SeqNum
50	SenderSubID	N	Assigned value used to identify specific message originator (e.g. desk, trader, etc.)	String

TAG	FIELD NAME	REQ'D	COMMENTS	FORMAT
142	SenderLocationID	N	Sender's LocationID (i.e. geographic location and/or desk) (Can be embedded within encrypted data section.)	String
57	TargetSubID	N	"ADMIN" reserved for administrative messages not intended for a specific user. Assigned value used to identify specific individual or unit intended to receive the message.	String
143	TargetLocationID	N	Trading partner LocationID (i.e. geographic location and/or desk) (Can be embedded within encrypted data section.)	String
129	DeliverToSubID	N	Trading partner SubID used when delivering messages via a third party. (Can be embedded within encrypted data section). Not supported.	String
145	DeliverToLocationID	N	Trading partner LocationID (i.e. geographic location and/or desk) used when delivering messages via a third party. (Can be embedded within encrypted data section). Not supported.	String
43	PossDupFlag	N	Always required for retransmitted messages, whether prompted by the sending system or as the result of a resend request. (Can be embedded within encrypted data section.)	Boolean
97	PossResend	N	Required when message may be duplicate of another message sent under a different sequence number. (Can be embedded within encrypted data section.)	Boolean
52	SendingTime	Y	Can be embedded within encrypted data section.	UTCTimeStamp
122	OrigSendingTime	N	Required for message resent as a result of a ResendRequest. If data is not available set to same value as SendingTime (can be embedded within encrypted data section.)	UTCTimeStamp
347	MessageEncoding	N	Type of message encoding (non-ASCII (non-English) characters) used in a message's "Encoded" fields.	String
369	LastMsgSeqNumProcessed	N	Not supported	SeqNum

A.2 Standard Trailer

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

The standard message trailer format is as follows.

Table 28 – Standard Message Trailer

TAG	FIELD NAME	REQ'D	COMMENTS	FORMAT
10	CheckSum	Y	(Always unencrypted, always last field in message)	String

Appendix B - Component Blocks

B.1 Instrument (symbology) Component Block

The Instrument component block contains all the fields commonly used to describe a security or instrument. Typically the elements in this component block are considered static data of a security which may be commonly found in a security master database (reference database). The Instrument component block can be used to describe any asset type supported by FIX.

The Instrument component, when part of a transaction that is inbound to the Exchange may only contain the following fields:

- Symbol (55) e.g. 'MEG'
- SecuritySubType (762) e.g. 'N'

Table 29 – Instrument Component Block

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
55	Symbol	Y	Unique exchange assigned identifier number for an order book. Required for inbound transactions to the Exchange except for OrderCancelReplaceRequest(G), OrderCancelRequest(F) and Order Status Request(H). In X-stream this is the security code.	String (20)
762	SecuritySubType	Y	In X-stream, this field is used to specify board on which Symbol is listed. Valid values: 'N' = Normal board 'O' = Oddlot board 'I' = Index board	String (1)

B.2 Parties (firm identification) Component Block

The Parties component is used to provide identifiers for parties involved in the transaction (e.g. firm, trader, Exchange, etc.).

The Parties component block is used to identify and convey information on the entities both central and peripheral to the financial transaction represented by the FIX message containing the Parties Block. The Parties block allows many different types of entities to be expressed through use of the PartyRole field and identifies the source of the PartyID through the PartyIDSource. Entities can encompass the following PartyRole (452) values:

- EnteringTrader (36) – The user that last actioned the order. Corresponding PartyID (448) tag has a maximum length of 30 characters
- EnteringFirm (7) – The firm corresponding to the EnteringTrader. Corresponding PartyID (448) tag has a maximum length of 30 characters
- ExecutingFirm (1) – The firm corresponding to the ExecutingTrader. Corresponding PartyID (448) tag has a maximum length of 30 characters
- ContraTrader (37) – The counterparty trader. Corresponding PartyID (448) tag has a maximum length of 30 characters
- ContraFirm (17) – The firm corresponding to the counterparty trader. Corresponding PartyID (448) tag has a maximum length of 30 characters

- OrderOriginationTrader (11) The user that originally entered the order. Corresponding PartyID (448) tag has a maximum length of 30 characters
- ExecutingTrader (12) – The trader that currently owns the order. Corresponding PartyID (448) tag has a maximum length of 30 characters
- GiveupClearingFirm (14) – The give-up firm for the order. Corresponding PartyID (448) tag has a maximum length of 30 characters
- ContraGiveupClearingFirm (20) – The contra give-up firm for the order. Corresponding PartyID (448) tag has a maximum length of 30 characters

Table 30 – Parties Component Block

TAG	FIELDNAME		REQ'D	COMMENTS	FORMAT
453	NoPartyIDs		N	Repeating group below should contain unique combinations of PartyID, PartyIDSource, and PartyRole	NumInGrp
→	448	PartyID	N	Used to identify source of PartyID. Required if PartyIDSource is specified. Required if NoPartyIDs > 0.	String (30)
→	447	PartyIDSource	N	Used to identify class source of PartyID value. Required if PartyID is specified. Required if NoPartyIDs > 0.	Char
→	452	PartyRole	N	Identifies the type of PartyID (e.g. Executing Broker). Required if NoPartyIDs > 0.	Int

B.2.1 Examples

Firm and individual User for whom the transaction applies:

- Give-up Firm (for outbound messages)

  PartyID (448) = "…" – the identifier of the give-up firm

  PartyIDSource (447) = "…" – the type of identifier used

  PartyRole (452) = "14" – Giveup Clearing Firm

- Executing Trader (for outbound messages)

  PartyID (448) = "…" – the identifier of the user

  PartyIDSource (447) = "…" – the type of identifier used

  PartyRole (452) = "12" – Executing Trader

B.3 TriggeringInstruction Component Block

The TriggeringInstruction component block specifies the conditions under which an order will be triggered by related market events as well as the behavior of the order in the market once it is triggered.

Table 31 – TriggeringInstruction Component Block

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
1100	TriggerType	Y	Defines when the trigger will hit, i.e. the action specified by the trigger instructions will come into effect. Valid values:	char

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
			1 = Partial Execution	
1107	TriggerPriceType	Y	The type of price that the trigger is compared to. Valid values: 2 = Last Trade	char
1102	TriggerPrice	Y	The price at which the trigger should hit.	Price

Appendix C - Field Enumerations Sorted By Tag Name

Table 32 – Field Enumerations Sorted By Tag Name

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
1057	AggressorIndicator	N	Used to identify whether the order initiator is an aggressor or not in the trade. Valid values: Y – Order initiator is aggressor N – Order initiator is passive	Boolean
380	BusinessRejectReason	Y	Valid values: 0 – Other 1 – Unknown ID 2 – Unknown Security 3 – Unknown Message Type 4 – Application not available 5 – Conditionally required field missing 6 – Not Authorized	Int
102	CxlRejReason	N	Identifies the reason for the cancel rejection. Valid values: 1 – Unknown order 6 – Duplicate order (e.g. duplicate CLOrdID) 99 – Other. Refer to returned Text (58) field for exact reason for rejection.	Int
434	CxlRejResponseTo	Y	Identifies the type of request that a Cancel Reject is in response to. Valid values are: 1 – Order Cancel Request 2 – Order Cancel/Replace Request	Char
549	CrossType	Y	Type of cross being submitted to a market. Valid values: 1 – Cross AON	int
550	CrossPrioritization	Y	Indicates if one side or the other of a cross order should be prioritized. Valid values: 0 – None	int
18	ExecInst	N	Instructions for order handling. Valid values: S – Suspend (move to private book) q – Unsuspend (enter into the market) Note: 'S' and 'q' are mutually exclusive	Multiple Char Value
150	ExecType	Y	Type of Execution being reported. Describes the specific ExecutionRpt	Char

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
			(i.e. Pending Cancel) while OrdStatus (39) will always identify the current order status (i.e. Partially Filled) Valid values: 0 – New 4 – Cancelled 5 – Replaced 7 – Stopped 8 – Rejected C – Expired F – Trade (partial fill or fill) H – Trade Cancel	
573	MatchStatus	N	The status of this trade with respect to matching or comparison 0 = compared, matched or affirmed 1 = uncompared, unmatched, or unaffirmed	Char
574	MatchType	N	1 – One-Party Trade Report 2 – Two-Party Trade Report	String
1473	NewsCategory	N	Category of news message (newsType) Valid values are: 1 – Market Place News (XCHG) 3 – Technical News (TECH) 99 – Other News (OTHR) *** NASDAQ Specific *** 94 – Disclosure (DISC) 95 – BlockSale (BLCK) 96 – NonSensitive News (NSEN) 97 – Sensitive News (SENS) 98 – System News (SYST) NOTE: If NewsType is unknown then this tag will not present in the news message.	int
528	OrderCapacity	N	Specifies the capacity of the firm placing the order. *** NASDAQ Defined *** A – Agent (client orders) P – Principal (house orders) S – Institutional G – Group O – Other M – Market Maker	Char

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
			L – Related Party E – Error T – Tax Exempt D – Special Account Retail F – Special Account Institutional	
103	OrdRejReason	N	For optional use with ExecType = 8 (Rejected). Code to identify reason for order rejection. Valid values are: 5 – Unknown order 6 – Duplicate order (e.g. duplicate CLOrdID) 99 – Other. Refer to returned Text (58) field for exact reason for rejection.	Int
39	OrdStatus	Y	Describes the current state of an order. Valid values are: 0 – New 1 – Partially filled 2 – Filled 4 – Cancelled 5 – Replaced 8 – Rejected C – Expired *** NASDAQ Defined *** U – Order is Unplaced X – Order with trigger in the book but not active (e.g. Order has not been triggered). Z – Private Order (suspended)	Char
40	OrdType	Y	Indicates the type of order. Valid values are: 1 – Market – The Price (44) field is not used, the order executes against the best prices order on the opposite side. 2 – Limit – The Price (44) field is specified and the order will execute at this price or better. 3 – Stop – The Price (44) field is not used. The TriggeringInstruction block is required. 4 – Stop Limit – The Price (44) field is specified. The TriggeringInstruction block is required.	Char
447	PartyIDSource	N	Used to identify class source of PartyID value. Required if PartyID is specified. Required if NoPartyIDs > 0. Valid values are:	Char

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
			C – Participant identifier	
452	PartyRole	N	Identifies the type of PartyID (e.g. Executing Broker). Required if NoPartyIDs > 0. Valid values are: 1 – Executing Firm 7 – Entering Firm 11 - Order Origination Trader 12 - Executing Trader 14 - Giveup Clearing Firm ('Give-up') 17 – Contra Firm 20 – Contra Giveup Clearing Firm ('Give-up') 36 – Entering trader 37 – Contra trader	Int
373	SessionRejectReason	N	Code to identify reason for a session-level Reject message. Valid values: 0 – Invalid Tag Number 1 – Required Tag Missing 2 – Tag not defined for this msg type 3 – Undefined tag 4 – Tag specified without a value 5 – Value incorrect for this tag 6 – Incorrect data format for value 9 – CompID 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 (character)	Int
54	Side	Y	Optional qualifier used to indicate the side of the market. Valid values are: 1 – Buy 2 – Sell 5 – Short Sell *** NASDAQ Defined *** Z – Buy Back	Char
59	TimeInForce	N	Indicates time in force techniques that are valid for the specified market segment. Valid values are: 0 – Day 1 – Good till cancelled (GTC) 3 – Immediate or Cancel (IOC/FaK) 6 – Good till date (GTD) 8 – Session	Char

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
487	TradeReportTransType	N	Identifies Trade Report message transaction type 0 – New 1 – Cancel 2 – Replace	Int
856	TradeReportType	N	Type of Trade Report 0 – Submit 1 – Alleged 2 – Accept 3 – Decline 6 – Trade Report Cancel 8 - Defaulted	Int
828	TrdType	N	Type of Trade. 0 – Regular Trade 1 – Block Trade	Int

Appendix D - FIX Data Types

Data types (with the exception of those of type "data") are mapped to ASCII strings as follows.

int	Sequence of digits without commas or decimals and optional sign character (ASCII characters "-" and "0" - "9"). The sign character utilizes one byte (i.e. positive int is "99999" while negative int is "-99999"). Note that int values may contain leading zeros (e.g. "00023" = "23"). Examples: 723 in field 21 would be mapped int as |21=723|. -723 in field 12 would be mapped int as |12=-723| The following data types are based on int.	
	Length	int field representing the length in bytes. Value must be positive.
	TagNum	int field representing a field's tag number when using FIX "Tag=Value" syntax. Value must be positive and may not contain leading zeros.
	SeqNum	int field representing a message sequence number. Value must be positive.
	NumInGroup	int field representing the number of entries in a repeating group. Value must be positive.
	DayOfMonth	int field representing a day during a particular month (values 1 to 31).
float	Sequence of digits with optional decimal point and sign character (ASCII characters "-", "0" - "9" and "."); the absence of the decimal point within the string will be interpreted as the float representation of an integer value. All float fields must accommodate up to fifteen significant digits. The number of decimal places used should be a factor of business/market needs and mutual agreement between counterparties. Note that float values may contain leading zeros (e.g. "00023.23" = "23.23") and may contain or omit trailing zeros after the decimal point (e.g. "23.0" = "23.0000" = "23" = "23."). Note that fields which are derived from float may contain negative values unless explicitly specified otherwise. The following data types are based on float.	
	Qty	float field capable of storing either a whole number (no decimal places) of "shares" (securities denominated in whole units) or a decimal value containing decimal places for non-share quantity asset classes (securities denominated in fractional units).
	Price	float field representing a price. Note the number of decimal places may vary. For certain asset classes, prices may be negative values. For example, prices for options strategies can be negative under certain market conditions (see FIX Specifications Volume 7: FIX Usage by Product for asset classes that support negative price values).
	PriceOffset	float field representing a price offset, which can be mathematically added to a "Price". Note the number of decimal places may vary and some fields such as LastForwardPoints may be negative.
	Amt	float field typically representing a Price times a Qty
	Percentage	float field representing a percentage (e.g. 0.05 represents 5% and 0.9525 represents 95.25%). Note the number of decimal places may vary.
char	Single character value, can include any alphanumeric character or punctuation except the delimiter.	

	All char fields are case sensitive (i.e. m != M). The following fields are based on char.	
	Boolean	char field containing one of two values: 'Y' = True/Yes 'N' = False/No
String	Alpha-numeric free format strings, can include any character or punctuation except the delimiter. All String fields are case sensitive (i.e. morstatt != Morstatt).	
	MultipleCharValue	string field containing one or more space delimited single character values (e.g. |18=2 A F|).
	MultipleStringValue	string field containing one or more space delimited multiple character values (e.g. |277=AV AN A|).
	Country	string field representing a country using ISO 3166 Country code (2 character) values (see FIX Specifications Volume 6 - Appendix 6-B).
	Currency	string field representing a currency type using ISO 4217 Currency code (3 character) values (see FIX Specifications Volume 6 - Appendix 6-A).
	Exchange	string field representing a market or exchange using ISO 10383 Market Identifier Code (MIC) values (see FIX Specifications Volume 6 - Appendix 6-C).
	MonthYear	string field representing month of a year. An optional day of the month can be appended or an optional week code. Valid formats: YYYYMM YYYYMMDD YYYYMMWW Valid values: YYYY = 0000-9999; MM = 01-12; DD = 01-31; WW = w1, w2, w3, w4, w5.
	UTCTimestamp	string field representing Time/date combination represented in UTC (Universal Time Coordinated, also known as "GMT") in either YYYYMMDD-HH:MM:SS (whole seconds) or YYYYMMDD-HH:MM:SS.sss (milliseconds) format, colons, dash, and period required. Valid values: * YYYY = 0000-9999, MM = 01-12, DD = 01-31, HH = 00-23, MM = 00-59, SS = 00-60 (60 only if UTC leap second) (without milliseconds). * YYYY = 0000-9999, MM = 01-12, DD = 01-31, HH = 00-23, MM = 00-59, SS = 00-60 (60 only if UTC leap second), sss=000-999 (indicating milliseconds). Leap Seconds: Note that UTC includes corrections for leap seconds, which are inserted to account for slowing of the rotation of the earth. Leap second insertion is declared by the International Earth Rotation Service (IERS) and has, since 1972, only occurred on the night of Dec. 31 or Jun 30. The IERS considers March 31 and September 30 as secondary dates for leap second insertion, but has never utilized these dates. During a leap second insertion, a UTCTimestamp field may read "19981231-23:59:59", "19981231-23:59:60", "19990101-00:00:00". (see

	http://tycho.usno.navy.mil/leapsec.html)
UTCTimeOnly	string field representing Time-only represented in UTC (Universal Time Coordinated, also known as "GMT") in either HH:MM:SS (whole seconds) or HH:MM:SS.sss (milliseconds) format, colons, and period required. This special-purpose field is paired with UTCDateOnly to form a proper UTCTimestamp for bandwidth-sensitive messages. Valid values: HH = 00-23, MM = 00-60 (60 only if UTC leap second), SS = 00-59. (without milliseconds) HH = 00-23, MM = 00-59, SS = 00-60 (60 only if UTC leap second), sss=000-999 (indicating milliseconds).
UTCDateOnly	string field representing Date represented in UTC (Universal Time Coordinated, also known as "GMT") in YYYYMMDD format. This special-purpose field is paired with UTCTimeOnly to form a proper UTCTimestamp for bandwidth-sensitive messages. Valid values: YYYY = 0000-9999, MM = 01-12, DD = 01-31.
LocalMktDate	string field representing a Date of Local Market (as opposed to UTC) in YYYYMMDD format. This is the "normal" date field used by the FIX Protocol. Valid values: YYYY = 0000-9999, MM = 01-12, DD = 01-31.
Data	string field containing raw data with no format or content restrictions. Data fields are always immediately preceded by a length field. The length field should specify the number of bytes of the value of the data field (up to but not including the terminating SOH). **Caution**: The value of one of these fields may contain the delimiter (SOH) character. Note that the value specified for this field should be followed by the delimiter (SOH) character as all fields are terminated with an "SOH".

Appendix E - PSE FIX 4.4 Differences

The following is a summary of the notable differences between the PSE FIX 4.4 and X-stream FIX 5.0 order management messages. It is not an exhaustive list.

- PSE FIX 4.4 implements a custom message flow for trade reporting (Trade Capture Report and Trade Capture Report Ack messages), while X-stream FIX 5.0 follows the FIX 5.0 standard
- PSE FIX 4.4 implements the OrderCancel/ReplaceRequest in a custom manner such that certain fields are not required to be resubmitted when modifying an order (e.g OrderQty, Price), while X-stream FIX 5.0 follows the FIX 5.0 standard where all fields must be resubmitted
- X-stream FIX 5.0 uses that same messages as PSE FIX 4.4, with some updates to individual fields as shown below

Table 33 – PSE FIX 4.4 vs X-stream FIX 5.0 Field Differences

PSE FIX 4.4	X-stream FIX 5.0
N/A	DefaultApplVerID (1137) for FIX 5.0 logon validation
Symbol (55)	Split out to Symbol (55) plus SecuritySubType (762)
PartyRole=24 (customer account)	Account (1), limited to 14 characters ('TC' prefix removed)
MaxFloor (111)	DisplayQty (1138)
StopPx (99)	TriggeringInstruction block
N/A	AggressorIndicator (1057) to provide passive/active trade information
N/A	ExecInst (18) to allow the use of the private order book
N/A	CopyMsgIndicator (797) for drop copy sessions
PartyRole (452)	Values differ, for more information refer to section *B.2 Parties (firm identification) Component Block*