

FIX SPECIFICATION

FIX SPECIFICATION FOR X-STREAM FOR DSE



Version 1.4
Date 06 July 2022
File DSE X-stream FIX Specification v1-4.docm



Copyright © 2022 Nasdaq Technology AB

All Rights Reserved.

No part of this document may be photocopied, reproduced, stored in a retrieval system, or transmitted, in any form or by any means whether, electronic, mechanical, or otherwise without the prior written permission of Nasdaq Technology AB.

No warranty of accuracy is given concerning the contents of the information contained in this publication.

To the extent permitted by law no liability (including liability to any person by reason of negligence) will be accepted by Nasdaq, Inc, its subsidiaries or employees for any direct or indirect loss or damage caused by omissions from or inaccuracies in this document.

Nasdaq Technology AB reserves the right to change details in this publication without notice.

Windows is a trademark and Microsoft, MS-DOS, and Windows NT are registered trademarks of Microsoft Corporation.

Other product and company names herein may be the trademarks of their respective owners.

Document History

VERSION	DATE	SUMMARY OF CHANGES
0.1	2013-09-30	First revision for Order Entry and Executions.
0.2	2013-10-18	Added Trade Capture/Negotiated Deal functionality
0.3	2014-04-01	Added order management Account Type and Deposit Account (Party Block), add WON specifics, added Howla indicator to MatchType, added Order Status Request.
1.0	2014-04-17	Released to client.
1.1	2014-06-11	Added sections on Private Orders and 'on behalf of' order management. Changes Account to be max 16 length. Removed LinesOfText component block.
1.2	2014-07-16	Update to Trade Capture messages and inclusion of Position Maintenance section. ClOrdId is Optional in OrderStatus(H).
1.3	2014-10-09	Removed tag SecondaryClOrdID (526) from section 4.3.4 <i>ExecutionReport (8)</i> . Section 5.6 <i>Order Suspend and Release (Private order)</i> updated.
1.4	2022-07-06	Reviewed the FIX specification. No additional changes are required for CR15. NASDAQ OMX copyrights and logos updated to Nasdaq throughout document.

Contents

1	Introduction to X-stream INET FIX	6
2	Session Information	7
2.1	FIX Admin and Infrastructure Messages Supported	7
2.2	SenderCompID and TargetCompID	7
3	FIX Application Messages	8
4	FIX Message Definitions	9
4.1	Session	9
4.1.1	Logon (A)	9
4.1.2	Logout (5)	9
4.1.3	Reject (3)	9
4.1.4	Resend Request (2)	10
4.1.5	Sequence Reset (Gap Fill) (4)	10
4.1.6	Test Request (1)	11
4.1.7	Heartbeat (0)	11
4.2	Infrastructure	12
4.2.1	Business Message Reject (j)	12
4.3	Application	12
4.3.1	New Order Single (D)	12
4.3.2	Order Cancel Request (F)	14
4.3.3	Order Cancel/Replace Request (G)	15
4.3.4	Execution Report (8)	16
4.3.5	Order Cancel Reject (9)	19
4.3.6	Order Status Request (H)	20
5	Order Management	22
5.1	Unique ClOrderID (11)	22
5.2	Order Identification	22
5.3	Order Modification via Order Cancel/Replace Request	22
5.3.1	Order Attributes allowed to change	22
5.4	Order Cancellation	23
5.5	Order Suspend and Release (Private order)	23
5.6	On-Behalf Order Management	23
5.7	Workflows	24
5.7.1	Entering of an New Order	24
5.7.2	Modification of an Order	25
5.7.3	Order Cancellation	26

5.7.4	Order Status States.....	26
6	Trade Capture Reporting.....	29
6.1	Trade Capture Messages.....	29
6.2	Workflows.....	29
6.2.1	Trade Capture Diagram for Privately Negotiated Trade, One-Party Report for Pass-through to Counterparty	29
6.2.2	Workflow for One-Party Report for Pass-through to Counterparty	30
6.3	Trade Capture Report (AE).....	33
6.3.1	Submit a Trade Capture Report to marketplace.....	33
6.3.2	Marketplace report a Trade Capture Report to parties	34
6.3.3	Confirm or withdraw/decline a Trade Capture Report	35
6.4	Trade Capture Report Ack (AR)	36
7	Position Maintenance.....	38
7.1	PositionReport (AP)	38
	Appendix A - Standard Header and Trailer	39
A.1	Standard Header	39
A.2	Standard Trailer	41
	Appendix B - Component Blocks	42
B.1	Instrument (symbology) Component Block	42
B.2	Parties Component Block	42
B.2.1	Examples	43
B.3	YieldData Component Block	44
	Appendix C - Field Enumerations Sorted By Tag Name.....	45
	Appendix D - FIX Data Types	49

Figures

Figure 1 – New Order Entry Workflow	24
Figure 2 – Order Modification Workflow	25
Figure 3 – Order Cancellation Workflow.....	26
Figure 4 – Order Status States.....	27
Figure 5 – Diagram for Negotiated Trade.....	29
Figure 6 - One-Party Report for Pass-through to Counterparty	31
Figure 7 - One-Party cancel (or time out) before counterparty confirms	32

Tables

Table 1 – FIX Admin Messages Supported	7
Table 2 – FIX Infrastructure Messages Supported.....	7

Table 3 – FIX Client to X-stream FIX Server.....	7
Table 4 – X-stream FIX server to FIX Client.....	7
Table 5 - FIX Inbound Application Messages	8
Table 6 – FIX Outbound Application Messages.....	8
Table 7 – Logon	9
Table 8 – Logout	9
Table 9 – Reject.....	10
Table 10 – Resend Request.....	10
Table 11 – Sequence Reset.....	11
Table 12 – Test Request.....	11
Table 13 – Heartbeat	12
Table 14 – Business Message Reject	12
Table 15 – New Order Single	12
Table 16 – Order Cancel Request.....	14
Table 17 – Order Cancel/Replace Request	15
Table 18 – Execution Report	17
Table 19 – Order Cancel Reject	20
Table 20 – Order Status Request.....	20
Table 21 – Execution Report Returned Tags Based On Scenario.....	28
Table 22 – Trade Capture Report to marketplace.....	33
Table 23 – Trade Capture Report to parties	34
Table 24 – Trade Capture Report to confirm/decline	35
Table 25 – Trade Capture Report Ack.....	36
Table 26 – PositionReport.....	38
Table 27 – Standard Message Header	39
Table 28 – Standard Message Trailer	41
Table 29 – Instrument Component Block	42
Table 30 – Parties Component Block	43
Table 31 – YieldData Component Block	44
Table 32 – Field Enumerations Sorted By Tag Name	45

1 Introduction to X-stream INET FIX

This document provides the Financial Information Exchange (FIX) specification for the X-stream INET trading platform.

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

MESSAGE NAME	MSGTYPE	COMMENTS
New Order Single	D	Used by participants to submit 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.
Order Status Request	H	Request for querying the details of an order.
Trade Capture Report	AE	Report off market trade.

Table 6 – FIX Outbound Application Messages

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	Response to a Trade Capture Report.
Trade Capture Report	AE	Trade Report.

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	String
554	Password	N	The FIX connector password. No security exists without transport level encryption.	String
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.	String
58	Text	N	Free format text string	String
Standard Trailer		Y		

The FIX gateway accepts HeartBtInt(108) range from 10 to 60. If client HeartBtInt is out of this range, the server will reply with the last valid value, or the default value (60) if it is the first logon of the day.

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	
58	Text	N	Free format text string	String
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
373	SessionRejectReason	N	Code to identify reason for a session-level Reject message.	Int
58	Text	N	Free format text string	String
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
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
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
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
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
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	

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
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. Maximum length 20 characters.	String
Component block <Parties>		Y	Insert here the set of "Parties" fields. Must include PartyRole = 5 for Investor (Deposit) Account.	
Component block <Instrument>		Y	Insert here the set of "Instrument" (symbology) fields.	
1	Account	Y	Trade Account. Maximum length 16 characters.	String
529	OrderRestrictions	Y	Account Type.	Char
18	ExecInst	N	Instructions for order handling. Note, whenever using this value to specify a Whole or None order the Minimum Quantity field must not be set..	Char
38	OrderQty	Y	Quantity ordered. This value represents the number of shares for equities or par, face or nominal value for Fixed Income instruments.	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, or yield based products. For yield based products, this is the dirty price and includes accrued interest. Price (44) and Yield (236) are mutually exclusive.	Price
54	Side	Y	Side of the market	Char
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. Note, for Whole or None orders this field must not be set.	Qty
236	Yield	Y/N	Yield percentage – This is the yield equivalent of the price. Price (44) and Yield (236) are mutually exclusive.	Percentage
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
432	ExpireDate	Y/N	Conditionally required if TimeInForce = GTD and ExpireTime is not specified.	LocalMktDate

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
126	ExpireTime	Y/N	Conditionally required if TimeInForce = 'Good till Date/Time'.	UTCTimeStamp
1138	DisplayQty	N	Specifies the disclosed volume on hidden/iceberg orders.	Qty
58	Text	N	Free Text. Maximum length 30 characters.	String
Standard Trailer		Y		

4.3.2 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 16 – 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. Maximum length 20 characters.	String
37	OrderID	N	Unique order identifier as assigned by X-stream that identifies the Order to be changed. Maximum length 18 characters.	String
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 were electronically submitted over FIX or otherwise assigned a ClOrdID. Maximum length 20 characters. Mandatory if OrderID (37) is not set.	String

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
Component block <Parties>		N	Insert here the set of "Parties" (firm identification) fields.	
Component block <Instrument>		Y	Insert here the set of "Instrument" (symbology) fields.	
38	OrderQty	Y	Order quantity	Qty
54	Side	Y	Side of the market.	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
Standard Trailer		Y		

4.3.3 Order Cancel/Replace Request (G)

The order cancel/replace request is used to change the parameters of an existing order.

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 17 – 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. Maximum length 20 characters.	String
37	OrderID	N	Unique identifier of most recent order as assigned by the Exchange. Maximum length 18 characters.	String
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. Maximum length 20 characters. Mandatory if OrderID (37) is not set.	String
Component block <Parties>		N	Insert here the set of "Parties" fields.	

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
Component block <Instrument>		Y	Insert here the set of "Instrument" (symbology) fields. Must match original order	
1	Account	N	Trade Account.	String
529	OrderRestrictions	N	Account type.	Char
18	ExecInst	N	Instructions for order handling. Note, whenever using this value to specify a Whole or None order the Minimum Quantity field must not be set.	Char
38	OrderQty	Y	Quantity ordered. This value represents the number of shares for equities or par, face or nominal value for Fixed Income instruments.	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, but not for yield based products. For yield based products, this is the dirty price and includes accrued interest. Price (44) and Yield (236) are mutually exclusive.	Price
54	Side	Y	Side of the market.	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. Note, for Whole or None orders this field must not be set.	Qty
59	TimeInForce	N	Indicates time in force techniques that are valid for the specified market segment.	Char
432	ExpireDate	Y/N	Conditionally required if TimeInForce = GTD and ExpireTime is not specified.	LocalMktDate
126	ExpireTime	Y/N	Conditionally required if TimeInForce = 'Good till Date/Time'.	UTCTimeStamp
236	Yield	N	Yield percentage.	Percentage
1138	DisplayQty	N	Specifies the disclosed volume on hidden/iceberg orders.	Qty
58	Text	N	Free Text. Maximum length 30 characters.	String
Standard Trailer		Y		

4.3.4 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
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.

If an Order Status Request is issued for an order with an OrdStatus(39) of either Cancelled, Expired or Filled, only mandatory fields will be provided in resulting Execution Reports. Non-mandatory fields will not be provided.

Table 18 – 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 were electronically submitted over FIX or otherwise assigned a ClOrdID(11). In the case of quotes can be mapped to the QuoteID (117) tag of a Mass Quote.	String
17	ExecID	Y	Unique identifier of execution message as assigned by the Exchange (will be 0 (zero) for ExecType=I (Order Status)).	String
18	ExecInst	N	Instructions for order handling.	Char
37	OrderID	Y	OrderID is required to be unique for each chain of orders.	String
41	OrigClOrdID	Y/N	Conditionally required for response to a Cancel or Cancel/Replace request	String
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

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
790	OrdStatusReqID	N	Required if responding to and if provided on the Order Status Request message. Echo back the value provided by the requester.	String
Component block <Parties>		N	Insert here the set of "Parties" (firm identification) fields.	
Component block <Instrument>		Y	Insert here the set of "Instrument" (symbology) fields.	
1	Account	N	Trade Account.	String
529	OrderRestrictions	N	Account type.	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 and ExpireTime is not specified.	LocalMktDate
126	ExpireTime	Y/N	Conditionally required if TimeInForce = GTD and ExpireDate is not specified.	UTCTimestamp
64	SettlDate	N	Specific date of trade settlement Settlement Date is in YYYYMMDD format.	LocalMktDate

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
574	MatchType	N	The point in the matching process at which this trade was matched.	String
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
236	Yield	N	Yield percentage (Fixed Income only)	Percentage
381	GrossTradeAmt	N	Total amount traded expressed in units of currency. Calculated on Price*LastQty	Amt
880	TradeMatchID	N	Identifier assigned by the trading system for a trade. This is the X-stream trade id.	String
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
797	CopyMsgIndicator	N	Drop Copy	Boolean
Standard Trailer		Y		

4.3.5 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 19 – 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
37	OrderID	Y	Unique identifier of most recent order as assigned by the Exchange. If CxlRejReason="Unknown order", specify "NONE".	String
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 were electronically submitted over FIX or otherwise assigned a ClOrdID.	String
60	TransactTime	Y	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
Standard Trailer		Y		

4.3.6 Order Status Request (H)

The order status request message is used by the broker/participant to generate an order status message back from the Exchange.

If an Order Status Request is issued for an order that is either cancelled, expired or fully filled, only mandatory fields will be provided in resulting Execution Reports. Non-mandatory fields will not be provided.

Table 20 – Order Status Request

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
StandardHeader		Y	MsgType = H	
11	ClOrdID	N	Corresponds to the ClOrdID (11) of the order whose status is being requested (if it exists). Conditionally required if the OrderID(37) is not provided. Either OrderID or ClOrdID must be provided. Maximum length 20 characters.	String
37	OrderID	N	Conditionally required if ClOrdID(11) is not provided. Either OrderID (37) or ClOrdID (11) must be provided. Maximum length 18 characters.	String

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
790	OrdStatusReqID	N	Optional, can be used to uniquely identify a specific Order Status Request message. Echoed back on Execution Report if provided.	String
55	Symbol	Y	Unique marketplace assigned identifier number for an order book. This field is not processed by the Exchange.	String
54	Side	Y	Side of order. This field is not processed by the Exchange. If order status request is failed, tag Side (54) will be set to 1 – Buy in the rejecting execution report.	Char
StandardTrailer		Y		

5 Order Management

5.1 Unique ClOrderId (11)

X-stream will 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 unique ClOrdId(11) is entered on these transactions. If a firm has multiple FIX connections, then ClOrdId(11) should be unique across all FIX connections for that firm.

When an action (order modification or order cancellation) is requested on a ClOrderId that happens to be duplicated, only the last order identified by ClOrderId is affected.

5.2 Order Identification

A FIX order is identified by either by its current ClOrderId using OrigClOrdID (41) for each FIX connection, or by X-stream OrderID (37) for the whole system.

OrderID(37) should be used to identify an order between 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 X-stream OrderID (37) is used, OrigClOrdID(41) should be set to "NONE". OrderID (37) is unique for every order.

Note: OrderID (37) maybe changed 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 ClOrderId 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:

- OrderQty (38)
- DisplayQty (111)
- Price (44)
- OrdType (40)
- TimeInForce (59)
- Yield (236)
- ExpireDate (432)
- ExpireTime (126)
- ExecInst (18).

Note: Any change to the 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/Fok Order Cancel) for every IOC and FoK order.

5.5 Order Suspend and Release (Private order)

Order can be suspended by specifying 'S' in ExecInst (18) in NewOrderSingle or Order Cancel Replace Request. Once suspended, the order will not match with other orders. OrdStatus (39) for such orders will be set to 'Z' – Private Order.

Order can be released by specifying 'q' in ExecInst (18) in Order Cancel Replace Request. Once released from suspension, the order will be able to match other orders.

5.6 On-Behalf Order Management

The FIX session may be used for Order Management in two ways:

- The FIX userId is both operator and user for the transaction.
- The FIX userId is the operator Id operates 'on-behalf of' the user given in SenderSubID (50).

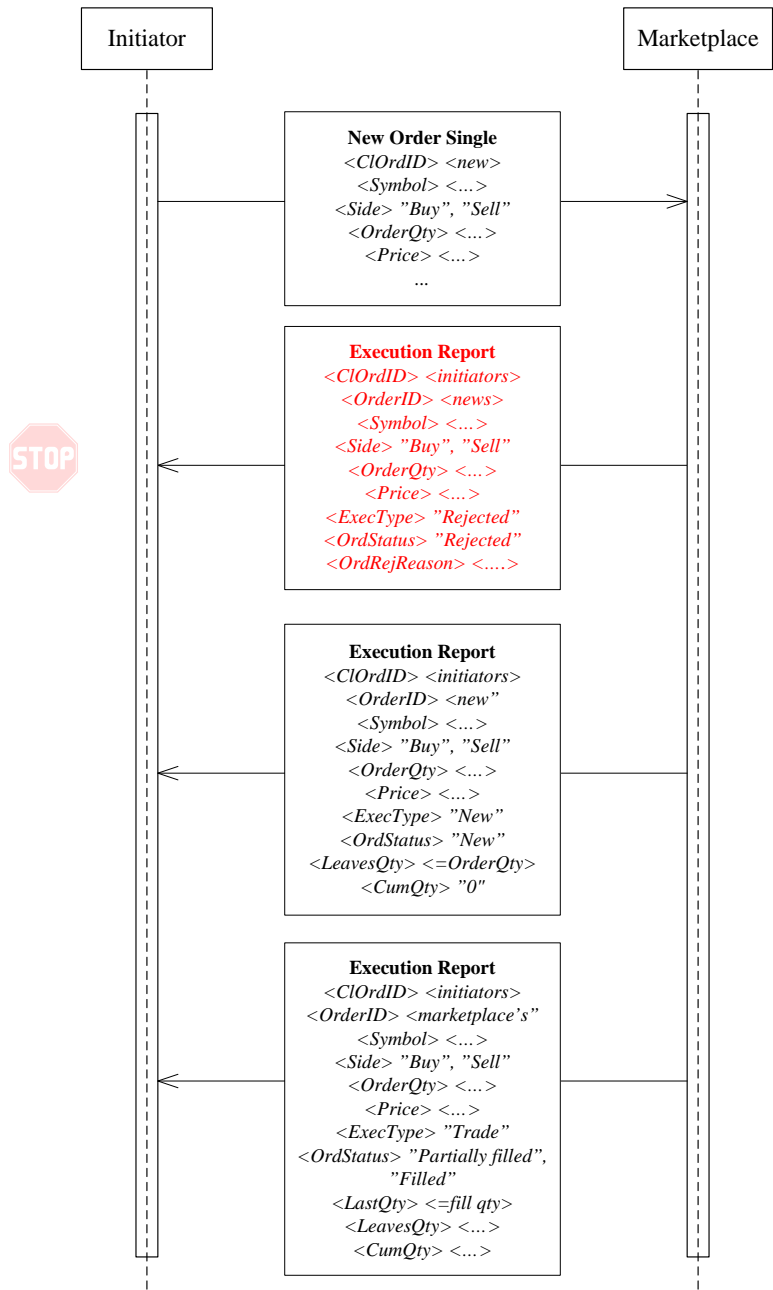
DSE FIX connections will operate in 'on-behalf of' mode.

A FIX order message with SenderSubID (50) will send two usernames to the X-stream backend – OperatorId and UserId. X-stream first checks that the OperatorId, the owner of the FIX session, has permission to enter messages 'on-behalf' of the UserId from the SenderSubID (50). The transaction is then processed with the permissions of UserId.

5.7 Workflows

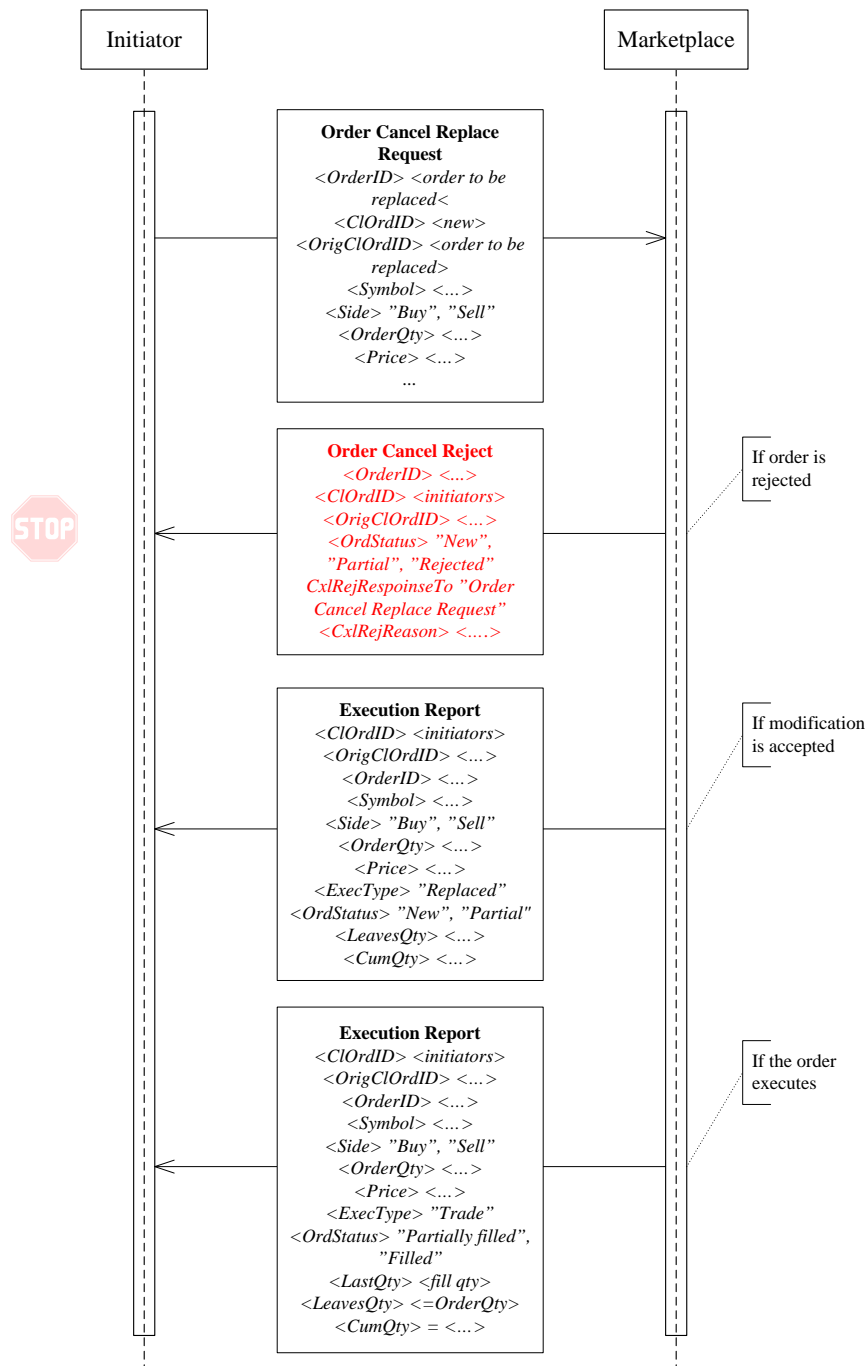
5.7.1 Entering of an New Order

Figure 1 – New Order Entry Workflow



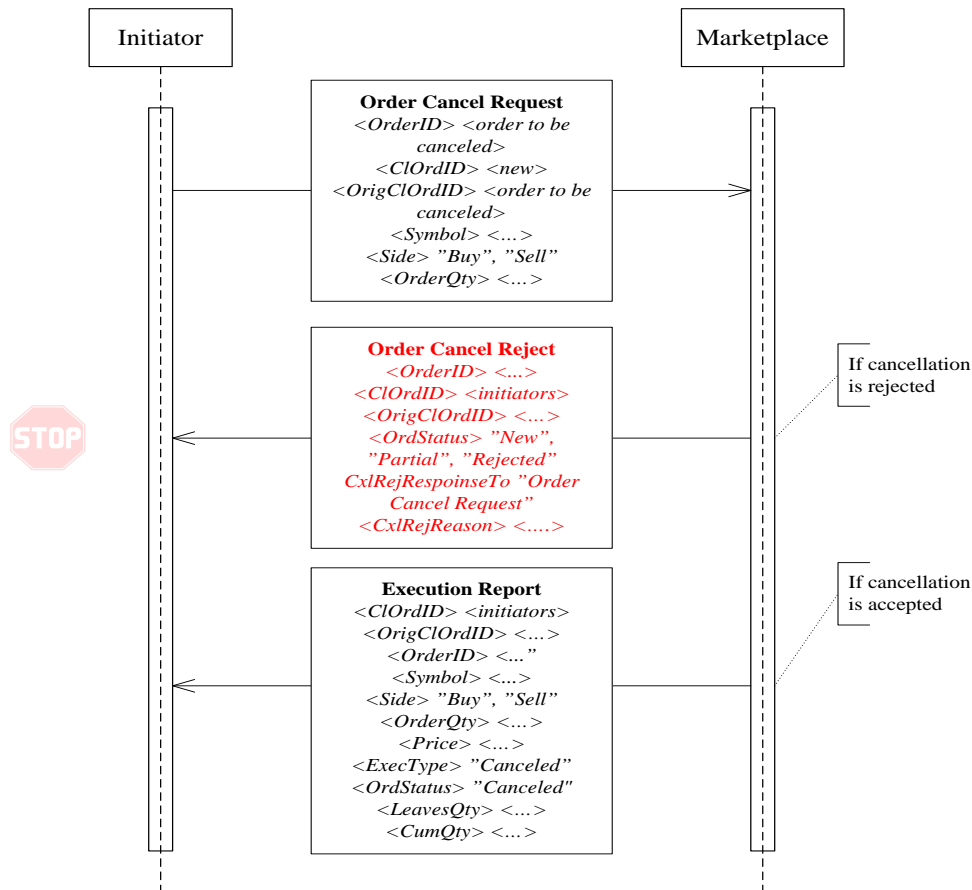
5.7.2 Modification of an Order

Figure 2 – Order Modification Workflow



5.7.3 Order Cancellation

Figure 3 – Order Cancellation Workflow



5.7.4 Order Status States

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

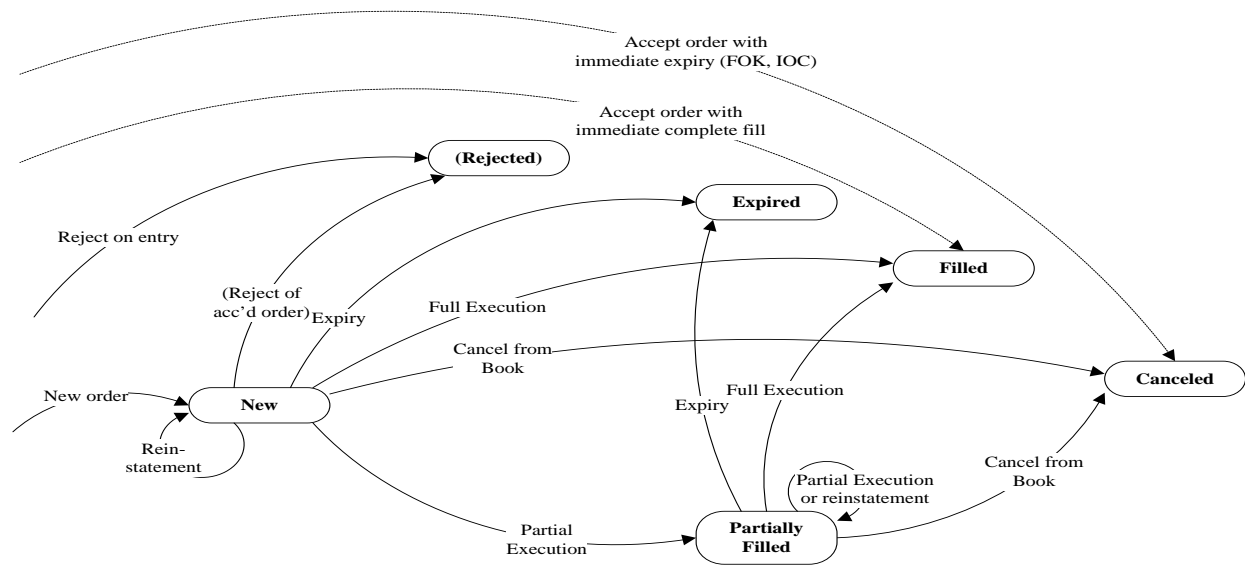


Table 21 – Execution Report Returned Tags Based On Scenario

	ClOrdID (11)	ExecID (17)	ExecInst (18)	OrderID (37)	OrigClOrdID (41)	ExecType (150)	SecondaryClOrdID (526)	MassStatusReqID (584)	TotNumReports (911)	OrdStatusReqID (790)	Parties	Instrument	Trigger 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)	ExpireTime (126)	SettleDate (64)	SecuritySubType (762)	CxlRejReason (102)	CxlRejReaResponseTo (434)	OrdRejReason (103)	LeavesQTY (151)	Yield (236)	GrossTradeAmt (381)	TradeMatchID (880)	AggressorIndicator (1057)	Text (58)
New Order Single	R	R	C	R		R	C				R	R	C	C		R	C	C	C	R	R	R	R	R	C	R		C	C	C		C				R	C				C
Order Cancel Pending	C	R		R	R	R						R				R				R	R			R	C			C							R						C
Order Cancel / Replace	C	R	C	R	R	R		C	C		R	R	C	C		R	C	C	C		R			R	R	C	R		C	C	C		C			R	C				C
Order Cancel / Replace Reject	C			R	R							R									R			R	C	R		C	C	C			R	R	R						C
Order Cancelled	C	R		R	R	R		C	C			R				R	C	C		R	R	C	R	R	C	R		C				C				R					C
Order Filled	R	R	C	R		R					R	R	C	C	R	R	R	R	C	R	R	C	R	R	C	R		C				R	R			R	C	R	R	R	C
Order Partially Filled	R	R	C	R		R					R	R	C	C	R	R	R	R	C	R	R	C	R	R	C	R		C	C	C		R	R			R	C	R	R	R	C
Order Rejected	C	R		R		R						R				R				C	R			R	C	R		C	C	C					R						C

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

R = Returned as part of Execution Report message

6 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 marketplace.
- Reporting of privately negotiated trades. Those messages are inbound to the marketplace but may also be used as outbound.

6.1 Trade Capture Messages

The Trade Capture category of messages consists of the following:

- Trade Capture Report
- Trade Capture Report Ack

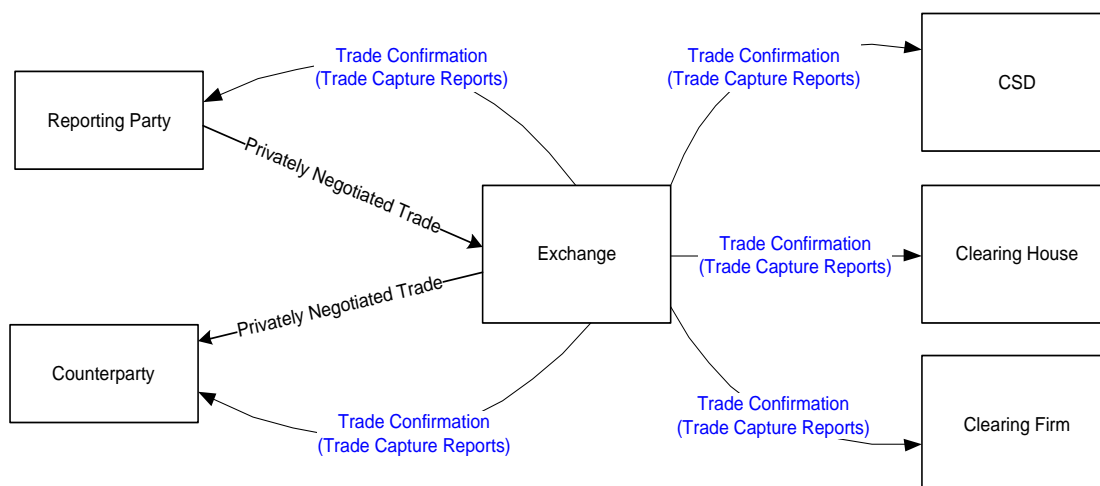
6.2 Workflows

6.2.1 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 marketplace 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 Negotiated Trade



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

TradeReportID (571) is used to identify a trade capture report. The initiator, the marketplace and the counterparty maintains their own set of TradeReportID (571). The TradeReportID used by the marketplace will not change during the whole TradeCaptureReport life cycle.

TradeReportRefID (572) is used to identify the TradeReportID (571) in the received TradeCaptureReport.

1. The initiator (usually the seller) sends TradeCaptureReport (AE) to the marketplace with a unique TradeReportID (571). The uniqueness of TradeReportID will not be checked by the exchange
2. If rejected, the marketplace will send to the initiator a TradeCaptureReport Ack(AR) with the same TradeReportID (571) as received from the initiator.
3. If accepted, the marketplace will 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 marketplace 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. The marketplace 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. The marketplace will send a TradeCaptureReport to both parties, with the same TradeReportID (571) in step 3. The confirmed TradeCaptureReport will have a new TrdMatchID (880) – the trade identifier.

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

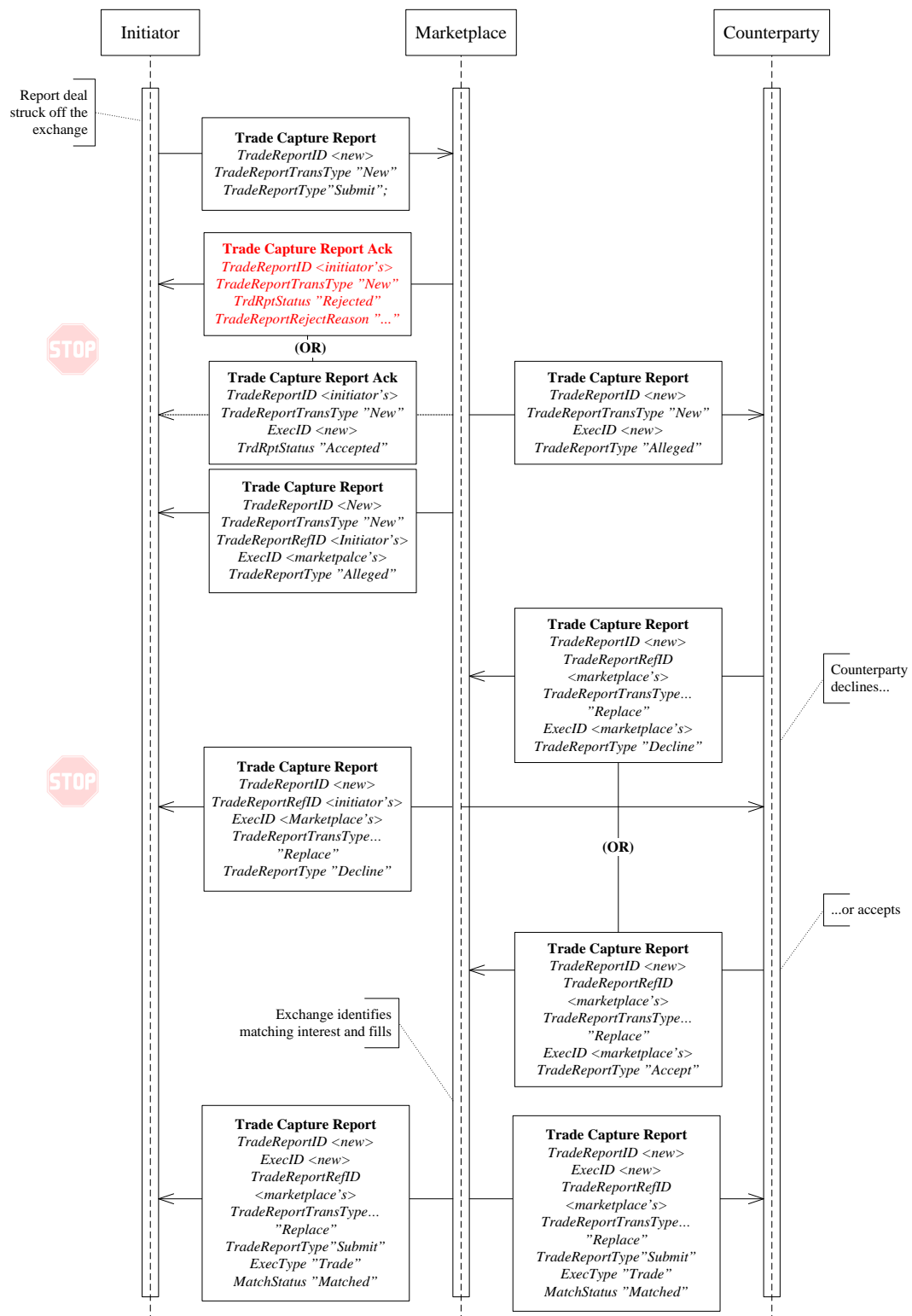
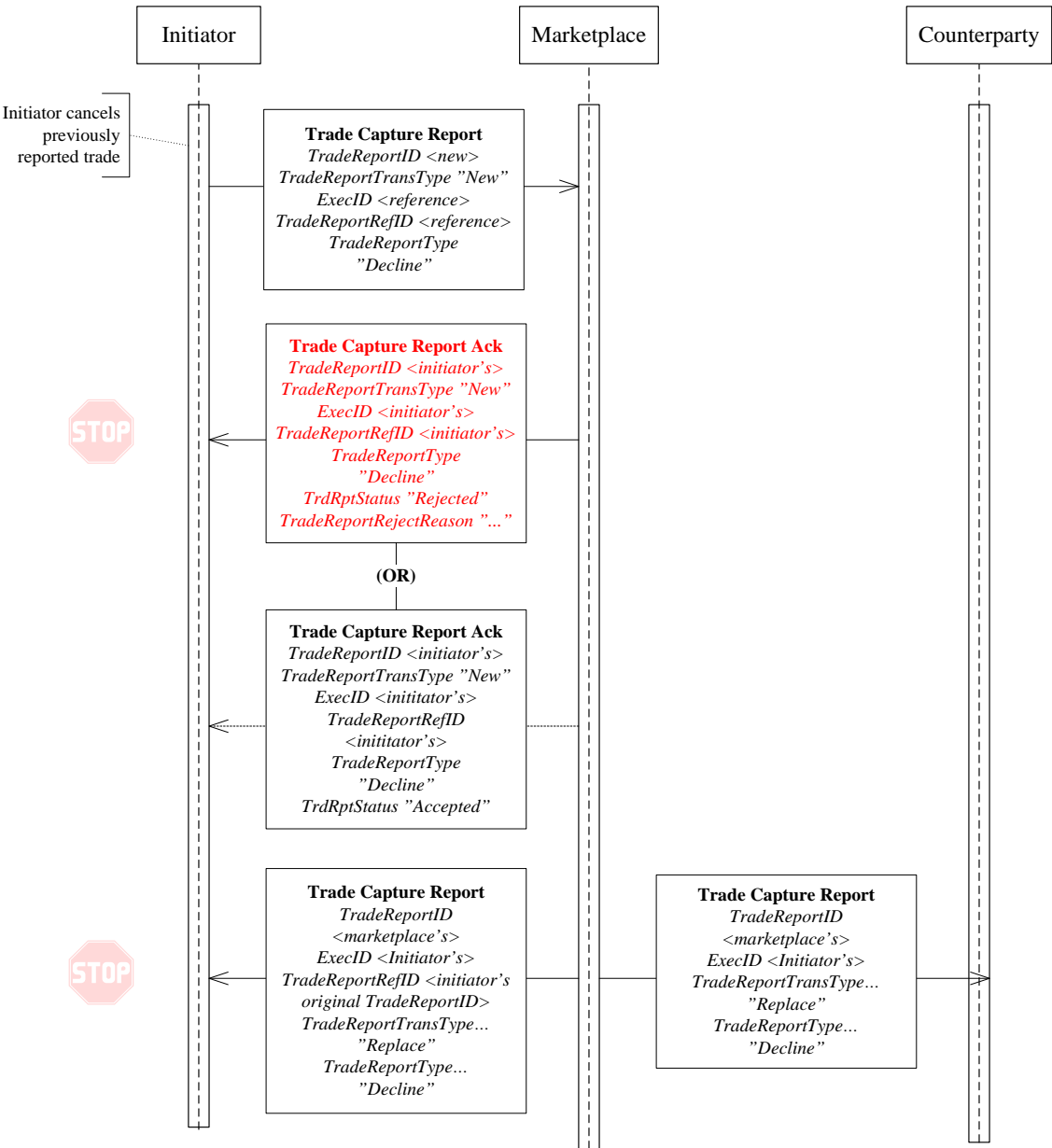


Figure 7 - One-Party cancel (or time out) before counterparty confirms



6.3 Trade Capture Report (AE)

6.3.1 Submit a Trade Capture Report to marketplace

The Initiator should send a Trade Capture Report (AE) to the marketplace to report an off market trade.

Table 22 – Trade Capture Report to marketplace

TAG	FIELDNAME		REQ'D	COMMENTS	FORMAT
StandardHeader			Y	MsgType = AE	
571	TradeReportID		Y	Unique identifier for the Trade Capture Report. Maximum length 20 characters.	String
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	
31	LastPx		Y	Trade Price. Dirty price for yield traded bonds.	Price
32	LastQty		Y	Trade Quantity	Qty
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>		N	Insert here the set of "Parties" (firm identification) fields 5 = Investor ID 37 = Contra trader (used for two party trade capture report to specify the counterparty)	
→	1	Account	Y	Trade Account. Maximum length 12 characters.	String
→	529	OrderRestrictions	N	Used to specify Account Type	Multiple CharValue
End of Component block, expanded in line < TrdCapRptSideGrp >					
797	CopyMsgIndicator		N	Indicates Drop Copy	Boolean
StandardTrailer			Y		

6.3.2 Marketplace report a Trade Capture Report to parties

Table 23 – Trade Capture Report to parties

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
StandardHeader		Y	MsgType = AE	
571	TradeReportID	Y	Unique identifier for the Trade Capture Report. Maximum length 20 characters.	String
17	ExecID	N	Exchanged assigned Execution ID	String
487	TradeReportTransType	N	Identifies Trade Report message transaction type 0 = New 2 = Replace	Int
856	TradeReportType	N	Type of Trade Report 0 = Submit (confirmed/matched trade) 1 = Alleged (unconfirmed trade) 3 = Decline (unconfirmed trade is declined or withdrawn) 6 = Trade Report Cancel (matched/confirmed trade is cancelled by the marketplace)	Int
828	TrdType	N	Type of Trade. 0 = Regular Trade 22 = Privately Negotiated Trades	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 correction or cancellation. Maximum length 20 characters.	String
573	MatchStatus	N	The status of this trade with respect to matching or comparison 0 = compared, matched or affirmed 1 = uncomparing, unmatched, or unaffirmed	Char
574	MatchType	N	The point in the matching process at which this trade was matched.	String
Component block <Instrument>		Y	Insert here the set of "Instrument" (symbology) fields	
31	LastPx	Y	Trade Price. Dirty price for yield traded bonds.	Price
32	LastQty	Y	Trade Quantity	Qty
60	TransactTime	N	Time the transaction represented by this Trade Capture Report occurred	UTCTime Stamp

TAG	FIELDNAME		REQ'D	COMMENTS	FORMAT
64	SettlDate		N	Specific date of trade settlement (Settlement Date) in YYYYMMDD format	LocalMkt Date
75	TradeDate		N	Used when reporting other than current day trades	LocalMkt Date
381	GrossTradeAmt		N	Total amount traded expressed in units of currency. Includes accrued Interest for convertible bonds and fixed income.	Amt
880	TradeMatchID		N	Identifier assigned by the trading system for a trade. Maximum length 21 characters.	String
Start of Component block, expanded in line < TrdCapRptSideGrp >					
552	NoSides		Y	Number of sides	Int
→	54	Side	Y	Side of order	Char
→	Component block <Parties>		N	Insert here the set of “Parties” (firm identification) fields	
→	1	Account	N	Trade Account. Maximum length 12 characters.	String
→	529	OrderRestrictions	N	Used to specify Account Type	Multiple CharVal ue
End of Component block, expanded in line < TrdCapRptSideGrp >					
797	CopyMsgIndicator		N	Indicates Drop Copy	Boolean
StandardTrailer			Y		

6.3.3 Confirm or withdraw/decline a Trade Capture Report

Table 24 – 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. Maximum length 20 characters.	String
487	TradeReportTransType	N	Identifies Trade Report message transaction type 2 = Replace	Int
856	TradeReportType	N	Type of Trade Report 2 = Accept 3 = Decline	Int
Component block <Instrument>		Y	Insert here the set of "Instrument" (symbology) fields. Not verified by the marketplace.	

TAG	FIELDNAME		REQ'D	COMMENTS	FORMAT
31	LastPx		Y	Trade Price. Not verified by the marketplace.	Price
32	LastQty		Y	Trade Quantity. Not verified by the marketplace.	Qty
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 marketplace.	Char
→	Component block <Parties>		N	Insert here the set of "Parties" (firm identification) fields for confirming a Trade Capture Report. 5 = Investor ID	
→	1	Account	N	Trade Account. Maximum length 12 characters. Required for confirming a Trade Capture Report.	String
→	529	OrderRestrictions	N	Used to specify Account Type for confirming a Trade Capture Report.	Multiple CharValue
End of Component block, expanded in line < TrdCapRptSideGrp >					
StandardTrailer			Y		

6.4 Trade Capture Report Ack (AR)

The Trade Capture Report Ack message is used to acknowledge a trade capture report from the reporter of a privately negotiated trade.

Table 25 – 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
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

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
17	ExecID	N	Exchanged assigned Execution ID	String
751	TradeReportRejectReason	N	Reason for Rejection of Trade Report	int
572	TradeReportRefID	N	The TradeReportID that is being referenced for some action, such as correction or cancellation	String
58	Text	N	If TradeReportRejectReason is set, text of reason	String
StandardTrailer		Y		

7 Postion Maintenance

7.1 PositionReport (AP)

A PositionReport(AP) reports the current trading limit value for a firm. An initial report or reports may be available at the start of the day when a connector first logs on.

When a Connector logs on the Logon reply will have a message sequence number that reflects any stored PositionReport(s) that have already been generated. The Connector can request these with a ResendRequest.

A PositionReport may be received intra-day if a limit is changed. The MessageEventSource(1011) is used to identify the intra-day report.

Table 26 – PositionReport

TAG	FIELDNAME		REQ'D	COMMENTS	FORMAT
StandardHeader			Y	MsgType = AP	
721	PosMaintRptID		Y	Unique ID for Report	String
715	ClearingBusinessDate		Y	Today's date	UTCDate
1011	MessageEventSource		N	'New' for initial report, 'Update' for an intra-day update to the limit	String
Component block <Parties>			Y	EnteringFirm	Parties
Start of Component block < PositionAmountData >					
753	NoPosAmt		Y/N	Specifies the number of repeating Position Amounts. Set to 1.	NumInGrp
→	707	PosAmtType	N	LMT = Firm Trading Limit	String
→	708	PosAmt	N	Amount value	Int
End of Component block, expanded in line < LinesOfTextGroup >					
Standard Trailer			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 data elements in this component block are considered the 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)
- SecuritySubType (762)

X-stream may support a non-ASCII representation of a security. In such a case the tags EncodedSecurityDescLen (350) and EncodedSecurityDesc (351) are used with MessageEncoding (347) set in the FIX Standard Header.

Table 29 – Instrument Component Block

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
55	Symbol	Y	Unique marketplace 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).	String
762	SecuritySubType	Y/N	In X-stream, this field is used to specify board on which Symbol is listed. This field is equivalent to MarketSegmentID (1300) in Security List (y) and Security Definition (d)	String
350	EncodedSecurityDescLen	N	Must be set if EncodedSecurityDesc field is specified and must immediately precede it.	Length
351	EncodedSecurityDesc	N	Encoded (non-ASCII characters) representation of the SecurityDesc field in the encoded format specified via the MessageEncoding field.	Data

B.2 Parties 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:

- EnteringTrader (36) - Corresponding PartyID (448) tag has a maximum length of 30 characters
- EnteringFirm (7) - Corresponding PartyID (448) tag has a maximum length of 30 characters

- ContraTrader (37) - Corresponding PartyID (448) tag has a maximum length of 30 characters
- ContraFirm (17) - Corresponding PartyID (448) tag has a maximum length of 30 characters
- ExecutingTrader (12) - Corresponding PartyID (448) tag has a maximum length of 30 characters
- ExecutingFirm (1) - Corresponding PartyID (448) tag has a maximum length of 30 characters
- OrderOriginationTrader (11)- Corresponding PartyID (448) tag has a maximum length of 30 characters
- ClearingFirm (4) - Corresponding PartyID (448) tag has a maximum length of 30 characters
- Investor ID (5) - Corresponding PartyID (448) tag has a maximum length of 32 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
→	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:

- Broker Firm (for outbound messages)
 - PartyID = "..." – the identifier of the firm
 - PartyIDSource = "..." – the type of identifier used
 - PartyRole = "1" – Executing Firm
- User (for outbound messages)
 - PartyID = "..." – the identifier of the user
 - PartyIDSource = "..." – the type of identifier used
 - PartyRole = "12" – Executing Trader

In cases the transaction is entered on behalf of the real owner and the marketplace validates authorization in those cases:

- Broker Firm (For outbound messages)
 - PartyID = "..." – the identifier of the firm on behalf of the real owner
 - PartyIDSource = "..." – the type of identifier used
 - PartyRole = "7" – Entering Firm

- User (for outbound messages)
 - PartyID = "..." – the identifier of the user on behalf of the real owner
 - PartyIDSource = "..." – the type of identifier used
 - PartyRole = "36" – Entering Trader

B.3 YieldData Component Block

The YieldData component block conveys yield information for a given Fixed Income security.

Table 31 – YieldData Component Block

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
235	YieldType	N	Type of yield.	String
236	Yield	N	Yield percentage	Percentage
696	YieldRedemptionDate	N	Date to which the yield has been calculated (i.e. maturity, par call or current call, pre-refunded date).	LocalMktDate
697	YieldRedemptionPrice	N	Price to which the yield has been calculated.	Price

Appendix C - Field Enumerations Sorted By Tag Name

Table 32 – Field Enumerations Sorted By Tag Name

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
1	Account	Y	Trade Account. Maximum length 16 characters.	String
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
18	ExecInst	N	Instructions for order handling. Valid values: 'S' – Suspend order 'q' – Release from suspension (mutually exclusive with S) 'w' – Whole or None	Char
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) Valid values: 0 – New 3 – Done for day 4 – Cancelled 5 – Replaced	Char

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
			6 – Pending Cancel (e.g. result of Order Cancel Request) 7 – Stopped 8 – Rejected C – Expired F – Trade (partial fill or fill) H – Trade Cancel I – Order Status	
574	MatchType	N	1 = One-Party Trade Report 2 = Two-Party Trade Report 4 = Auto-match *** Nasdaq Defined *** F1 = One-Party DVP Trade Report F2 = Two-Party DVP Trade Report F4 = DVP Auto-match	
529	OrderRestrictions	N	*** Nasdaq Defined *** Specifies the Account Type. N – Investor F – Foreign Investor I – Institution O – Foreign Institution R – NRB (Non Resident Bangladeshi)	
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 *** X – Order with trigger in the book but not active (e.g. Order has not been triggered). Z – Private Order	Char

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
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. *** Exchange Defined *** Z – Market at best	Char
447	PartyIDSource	N	Used to identify class source of PartyID value. Required if PartyID is specified. Required if NoPartyIDs > 0. Valid values are: C – Participant identifier	Char
452	PartyRole	N	Identifies the type of PartyID (e.g. Executing Broker). Required if NoPartyIDs > 0. Valid values are: 1 – Executing Firm 3 – Client ID 4 – Clearing Firm 5 – Investor ID (Deposit Account) 7 – Entering Firm 11 – Order origination trader 12 – Executing Trader 13 – Order origination firm 17 – Contra Firm 36 – Entering trader 37 – Contra trader 44 – Order Entry Operator ID	Int
54	Side	Y	Optional qualifier used to indicate the side of the market. Valid values are: 1 – Buy 2 – Sell 5 – Short Sell	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 3 – Immediate or Cancel (IOC) 4 – Fill or Kill (FoK) 6 – Good till date (GTD) *** Nasdaq Defined *** S - Session	Char
336	TradingSessionID	Y	Identifier for Trading Session. A trading session spans an extended	String

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
			period of time that can also be expressed informally in terms of the trading day. Usage is determined by market or counterparties. For trading session related messages TradingSessionID will map to the current trading session state (e.g. pre-open, open, closed, etc.)	

Appendix D - FIX Data Types

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

int	<p>Sequence of digits without commas or decimals and optional sign character (ASCII characters "-", "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").</p> <p>Examples:</p> <p>723 in field 21 would be mapped int as 21=723 .</p> <p>-723 in field 12 would be mapped int as 12=-723 </p> <p>The following data types are based on int.</p>	
	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	<p>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.").</p> <p>Note that fields which are derived from float may contain negative values unless explicitly specified otherwise. The following data types are based on float.</p>	
	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	<p>Single character value, can include any alphanumeric character or punctuation except the delimiter. All char fields are case sensitive (i.e. m != M).</p>	

	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	<p>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 UTCTimeOnly to form a proper UTCTimestamp for bandwidth-sensitive messages.</p> <p>Valid values:</p> <p>HH = 00-23, MM = 00-60 (60 only if UTC leap second), SS = 00-59. (without milliseconds)</p> <p>HH = 00-23, MM = 00-59, SS = 00-60 (60 only if UTC leap second), sss=000-999 (indicating milliseconds).</p>
UTCTimeOnly	<p>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.</p> <p>Valid values:</p> <p>YYYY = 0000-9999, MM = 01-12, DD = 01-31.</p>
LocalMktDate	<p>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.</p> <p>Valid values:</p> <p>YYYY = 0000-9999, MM = 01-12, DD = 01-31.</p>
Data	<p>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).</p> <p>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".</p>