

FIX SPECIFICATION

X-STREAM FIX SPECIFICATION FOR BVC



Version	4.4
Date	29 May 2019
File	BVC X-stream FIX Specification v4-4.docx



Copyright © 2017 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
3.5	2016-10-28	Updated for X-stream INET. Release version.
3.6	2017-01-27	Added BidYield and OfferYield tags to Quote and QuoteStatusReport messages. Changed FIX5 tag DisplayQty to MaxFloor in table 16. Added AccruedInterestAmt tag to ExecutionReport and TradeCaptureReport messages. Added AccruedInterestRate tag to SecurityList and SecurityListUpdateReport messages. Added FillYield tag to ExecutionReport message. Clarified Price field description. Clarified Side description for RFQs/Quotes. Added Appendix E.
3.7	2017-02-17	Added DisplayQty tag back to ExecutionReport message to provide visible balance. Added Cancel On Connection Loss value 'o' to ExecInst tag. Corrected ExecInst tag type from char to MultipleCharValue. Added Order Mass Action Request (CA) and Order Mass Action Report (BZ) messages.
3.8	2017-03-23	Added MassActionScope value 99 for Order Mass Action Request message, clarified message description. Added ReturnValue tag to Execution Report and Trade Capture Report messages. Added DirtyPrice tag to Execution Report, Trade Capture Report, Market Data Snapshot and Market Data Incremental Refresh messages. Added TotalNetValue tag to Trade Capture Report message. Clarified Yield tag description for Execution Report message. Clarified LastPx tag description for Trade Capture Report message. Clarified Market Data Information Request and Responses tables. Clarified MDEntryPx, LastPx and FirstPrice tag usage for Market Data messages. Removed AccruedInterestRate tag. Clarified AgreementID tag in FinancingDetails Component Block. Clarified UnderlyingSymbol tag in UnderlyingInstrument Component Block.

VERSION	DATE	SUMMARY OF CHANGES
		<p>Modified UnderlyingSecurityType values in UnderlyingInstrument Component Block.</p> <p>Added new FIX Message Examples to Appendix E.</p> <p>Updated Quote Response message description to match workflow.</p> <p>Clarified Repo tag usage for New Order Single, Order Cancel/Replace Request and Trade Capture Report messages.</p>
3.9	2017-04-10	<p>Added StartCash (to replace TotalNetValue) and EndCash tags (to replace ReturnValue)</p> <p>Added 'Pass' value to QuoteRespType for declining quotes</p>
3.10	2017-04-21	Updated examples
3.11	2017-07-27	<p>Clarified use of Stipulations block, OrderQty and StartCash tags in New Order Single, Order Cancel/Replace and Trade Capture Report messages</p> <p>Corrected Quote Status Report MsgType value</p> <p>Clarified Side tag usage for crossing in Trade Capture Report message</p> <p>Added ClientID and SponsoringFirm as valid PartyRole values for Trade Capture Report message</p> <p>Added ValidUntilTime tag to Trade Capture Report message</p> <p>Clarified that fields may be ignored in Trade Capture Report for confirm/decline</p> <p>Added MarketPriceType tag to Security List and Security List Update messages</p> <p>Added ContractMultiplier tag to Instrument block</p>
3.12	2017-08-04	<p>Clarified the usage of the TradeReportRefID tag.</p> <p>Renamed FirstPrice tag to FirstPx.</p>
3.13	2017-09-14	<p>Clarified that a crossing TradeCaptureReport doesn't need confirmation in section 6.3.1</p> <p>Updated ClientID usage in Parties block for TradeCaptureReport to be used as a mandatory OTC agreement time</p> <p>Removed TrdRegTimestamps block from TradeCaptureReport</p> <p>Added SponsoringFirm to Parties block in TradeCaptureReport confirmation</p> <p>Clarified fields used in Market Data tables in section 7.2</p> <p>Clarified SecurityListUpdateReport usage</p> <p>Clarified TriggerPriceType valid values in TriggeringInstruction block</p>
3.14	2018-07-28	<p>Specified that SenderSubID (50) must be set by clients for all order operations</p> <p>Added Trade Cancel Request workflow and messages to section 6</p>
3.15	2018-10-26	Included trade status information via the Text (58) tag on Market Data messages
4.1	2018-01-24	<p>Improved description of Stipulations block in ExecutionReport message</p> <p>Improved description of FirstPx(1025) tag usage for Market Data</p> <p>Removed obsolete table 'Execution Report Returned Tags Based On Scenario'</p> <p>Added OrderQty(38) to QuoteRequestReject message</p> <p>Added LTQ, Off-market LTP, Off-market shares traded, Off-market value traded, % change from prev close, Prev Settle Price to Market Data messages</p> <p>Added PriceDelta(811) tag to MarketDataIncrementalRefresh message</p> <p>Added StrategyType(2141) and LiquidityValue(404) tags to SecurityList messages and Factor(228), SecurityGroup(1151), AssetType(1940), ShortSaleRestriction(1687), and ISIN to Instrument component block for SecurityList use</p>

VERSION	DATE	SUMMARY OF CHANGES
		<p>Added 3 = ClientID as a valid PartyRole(452) value for QuoteRequest, Quote and QuoteStatusReport messages</p> <p>Added TrdSubType(829) to ExecutionReport and TradeCaptureReport messages</p>
4.2	2019-01-29	<p>Changed OrderQty, StartCash and Stipulations block usage to be the same for both buy and sell Fixed Income orders, on the NewOrderSingle and OrderCancel/ReplaceRequest messages</p> <p>Added TrdType(828) tag to ExecutionReport and Market Data [W & X] messages</p> <p>Added AggressorSide(2446) tag to Market Data [W & X] messages</p> <p>Added SettlDate(64), TrdType(828) and AggressorSide(2446) tags to trade Market Data usage in table 33</p>
4.3	2019-05-16	<p>LastMsgSeqNumProcessed (369) can be sent on a Logoff message</p> <p>Added Order Transfer sections 3.6 and 3.8.4</p> <p>Clarified behaviour of SenderSubID (50) and TargetSubID (57)</p> <p>Added PartyRole (452) values '991' (Transfer to trader), '992' (Transfer operator) and '999' (Pre-arranged Firm)</p> <p>Added OrderOwnershipIndicator (2679) tag to Order Cancel/Replace Request and Execution Report</p> <p>Added MatchType (574) tag to Execution Report and Market Data messages</p> <p>Added MatchType (574) value '99' (Pre-arranged)</p> <p>Added ExecType (150) values 'o' (Transfer offered), 'r' (Transfer rejected), 't' (Transferred) and 'w' (Transfer withdrawn)</p>
4.4	2019-05-29	Clarified section 3.6 OrderTransfer

Contents

1	X-stream FIX Specification for X-stream.....	12
1.1	Document Structure.....	12
1.2	FIX Messages Supported	12
2	Session and Infrastructure Messages	16
2.1	FIX Session Establishment	16
2.1.1	Logon and Authentication.....	16
2.1.2	Logon Failures and Account Locking	16
2.1.3	FIX Session SenderCompID, Username and Passwords	16
2.1.4	Changing FIX Session Passwords	16
2.1.5	Encryption.....	17
2.1.6	FIX Session Logon Confirmation and Logout	17
2.1.6.1	Logon Confirmation (Session Authenticated)	17
2.1.6.2	Logout (Authentication Failure).....	17
2.2	Logon (A)	17
2.3	Logout (5)	18
2.4	Reject (3)	18
2.5	Resend Request (2)	19
2.6	Sequence Reset (Gap Fill) (4)	19
2.7	Test Request (1).....	20
2.8	Heartbeat (0)	20
2.9	Business Message Reject (j).....	21
3	Order Management	22
3.1	Unique ClOrdId (11).....	22
3.2	Order Identification.....	22
3.3	Order Modification via Order Cancel/Replace Request	22
3.3.1	Order Attributes allowed to change.....	22
3.4	Order Cancellation	23
3.5	Order Suspend and Release (Private order).....	23
3.6	Order Transfer	23
3.7	Users in FIX Order Management	24
3.8	Workflows.....	25
3.8.1	Entering of a New Order.....	25
3.8.2	Modification of an Order.....	26
3.8.3	Order Cancellation.....	27
3.8.4	Order Transfer	28

3.8.5	Order Status.....	29
3.9	New Order Single (D).....	29
3.10	Order Cancel Request (F)	31
3.11	Order Cancel/Replace Request (G).....	32
3.12	Order Cancel Reject (9).....	36
3.13	Order Status Request (H)	36
3.14	Order Mass Action Request (CA)	37
3.15	Order Mass Action Report (BZ)	38
3.16	Execution Report (8).....	39
4	Mass Quote Management.....	45
4.1	Unique QuoteID (117).....	45
4.2	Workflows.....	45
4.2.1	Entering of a mass quote	45
4.2.2	Quote Entry Cancel	46
4.3	Mass Quote (i)	46
4.4	Mass Quote Acknowledgement (b)	47
5	Request for Quote (RFQ)	50
5.1	Quote Request (R)	50
5.2	Quote Request Reject (AG)	51
5.3	Quote (S)	52
5.4	Quote Cancel (Z)	53
5.5	Quote Status Report (AI).....	54
5.6	Quote Reponse (AJ)	56
5.7	Workflows.....	57
5.7.1	Request for Quote	57
6	Trade Capture Reporting.....	58
6.1	Trade Capture Messages.....	58
6.2	Workflows.....	58
6.2.1	Trade Capture Workflow for Multiple Counterparties	58
6.2.2	Trade Capture Diagram for Privately Negotiated Trade, One-Party Report for Pass-through to Counterparty	58
6.2.3	Workflow for One-Party Report for Pass-through to Counterparty	59
6.2.4	Trade Cancellation Request Workflow	61
6.3	Trade Capture Report (AE).....	64
6.3.1	Submitting a Trade Capture Report to the Exchange to initiate a trade	64
6.3.2	Exchange reporting a Trade Capture Report to Parties	66

6.3.3	Confirm or withdraw/decline a Trade Capture Report to the Exchange for a trade	69
6.3.4	Submitting a Trade Capture Report to the Exchange to request a trade cancellation.....	71
6.3.5	Confirm or withdraw/decline a Trade Capture Report to the Exchange for a trade cancellation.....	72
6.4	Trade Capture Report Ack (AR)	73
7	Market Data	74
7.1	Workflows.....	75
7.1.1	Subscribing to and Receiving Market Data	75
7.1.2	Subscribing to and Receiving Trading Session Status	76
7.1.3	Subscribing to and Receiving Security Status.....	77
7.1.4	Receiving News.....	78
7.2	General Information for Market Data Requests and Responses	78
7.2.1	Market Data Requests Based on Category	78
7.2.2	Market Data Requests.....	78
7.2.3	Wildcard Security Specification	78
7.2.4	Response Enumerations Extensions	79
7.2.5	Order Information Request and Responses	79
7.2.6	Trade Information Request and Responses	80
7.2.7	Index Information Request and Responses	82
7.2.8	Security Statistics Requests and Responses.....	82
7.2.9	The order of Market Data Responses	84
7.3	Market Data Request (V)	84
7.4	Market Data Request Reject (Y)	85
7.5	Market Data Snapshot/Full Refresh (W)	85
7.6	Market Data Incremental Refresh (X)	88
7.6.1	Maintaining the Order Book	88
7.7	Trading Session Status Request (g)	91
7.8	Trading Session Status (h).....	92
7.9	Security List Request (x)	93
7.10	Security List (y).....	93
7.11	Security List Update Report (BK)	96
7.12	Security Status Request (e)	99
7.13	Security Status (f)	100
7.14	News (B)	100
	Appendix A - Standard Header and Trailer	102
A.1	Standard Header	102
A.2	Standard Trailer	104

Appendix B - Component Blocks	105
B.1 Instrument (symbology) Component Block	105
B.2 Parties Component Block	107
B.2.1 Examples	108
B.3 YieldData Component Block	109
B.4 TriggeringInstruction Component Block	109
B.5 LinesOfTextGrp Component Block	110
B.6 FinancingDetails Component Block	111
B.7 Underlying Instrument Component Block	111
B.8 Stipulations Component Block	112
Appendix C - Field Enumerations	113
C.1 Field Enumerations Sorted by Tag Value	113
Appendix D - FIX Data Types	124
Appendix E - FIX Message Examples	127
E.1 Logon	127
E.2 Order Entry	127
E.3 Order Modification	127
E.4 Order Cancellation	127
E.5 Order Execution (Fixed Income)	128
E.6 Negotiated Deal Execution (Fixed Income)	128
E.7 RFQ Execution (Fixed Income)	129
E.8 Repo Order Execution (Fixed Income)	131
E.9 Repo Negotiated Deal Execution (Fixed Income)	133

Figures

Figure 1 – New Order Entry Workflow	25
Figure 2 – Order Modification Workflow	26
Figure 3 – Order Cancellation Workflow	27
Figure 4 – Order Transfer Workflow	28
Figure 5 – Order Status States	29
Figure 6 – Mass Quote workflow	45
Figure 7 – Request for Quote Workflow	57
Figure 8 – Trade Capture High Level Workflow	58
Figure 9 – Diagram for Privately Negotiated Trade	59
Figure 10 – One-Party Trade Report for Pass-through to Counterparty	60
Figure 11 – One-Party Trade Report decline (or time out) before counterparty confirms	61
Figure 12 – Trade Cancel Request workflow	63
Figure 13 – Subscribing and Receiving Market Data Workflow	75
Figure 14 – Subscribing and Receiving Trading Session Status Workflow	76
Figure 15 – Subscribing to and Receiving Security Status Workflow	77
Figure 16 – Receiving News Workflow	78

Tables

Table 1 – FIX Messages Supported	12
Table 2 – Logon	17
Table 3 – Logout	18
Table 4 – Reject	18
Table 5 – Resend Request	19
Table 6 – Sequence Reset	20
Table 7 – Test Request	20
Table 8 – Heartbeat	21
Table 9 – Business Message Reject	21
Table 10 – New Order Single	29
Table 11 – Order Cancel Request	32
Table 12 – Order Cancel/Replace Request	33
Table 13 – Order Cancel Reject	36
Table 14 – Order Status Request	37
Table 15 – Order Mass Action Request	37

Table 16 - Order Mass Action Report.....	38
Table 17 - Execution Report	39
Table 18 - Mass Quote.....	46
Table 19 - Mass Quote Acknowledgement	47
Table 20 - Quote Request	50
Table 21 - Quote Request Reject.....	52
Table 22 - Quote.....	52
Table 23 - Quote Cancel	54
Table 24 - Quote Status Report	54
Table 25 - Quote Reponse.....	56
Table 26 - Trade Capture Report (inbound to the Exchange) to initiate a trade	64
Table 27 - Trade Capture Report (outbound from the Exchange).....	66
Table 28 - Trade Capture Report (inbound to the Exchange) to confirm/decline a trade	69
Table 29 - Trade Capture Report (inbound to the Exchange) to request a trade cancellation	71
Table 30 - Trade Capture Report (inbound to the Exchange) to confirm/decline a trade cancellation.....	72
Table 31 - Trade Capture Report Ack.....	73
Table 32 - Order Information Response Tags.....	79
Table 33 - Trade Information Returned Tags	80
Table 34 - Index Information Returned Tags	82
Table 35 - Security Statistics Returned Tags.....	82
Table 36 - Market Data Request.....	84
Table 37 - Market Data Request Reject	85
Table 38 - Market Data Snapshot/Full Refresh	86
Table 39 - Market Data Incremental Refresh	89
Table 40 - Trading Session Status Request.....	92
Table 41 - Trading Session Status	92
Table 42 - Security List Request	93
Table 43 - Security List.....	93
Table 44 - Security List Update Report.....	96
Table 45 - Security Status Request	99
Table 46 - Security Status.....	100
Table 47 - News	100

Table 48 – Standard Message Header	102
Table 49 – Standard Message Trailer	104
Table 50 – Instrument Component Block	105
Table 51 – Investment Firm Parties Component Block	108
Table 52 – YieldData Component Block	109
Table 53 – TriggeringInstruction Component Block	109
Table 54 – LinesOfTextGrp Component Block	110
Table 55 – FinancingDetails Component Block	111
Table 56 – UnderlyingInstrument Component Block	111
Table 57 – Stipulations Component Block	112
Table 58 – Field Enumerations Sorted by Tag Value	113

1 X-stream FIX Specification for X-stream

This document provides the X-stream FIX message specification supporting version 4.4 of the FIX protocol specification. Some messages and tags post FIX 4.4 are coloured in blue.

The messages specified in this document should support existing X-stream clients that are using FIX V4.4 engine implementation.

1.1 Document Structure

This document is divided into the following chapters:

- Chapter 2 – Session and Infrastructure messages.
- Chapter 3 – Order Management
- Chapter 4 – Mass Quote Management
- Chapter 5 – Request for Quote
- Chapter 6 – Trade Capture Reporting
- Chapter 7 – Market Data
- Appendix A describes the FIX Standard Header and Standard Trailer.
- Appendix B describes the various component blocks used throughout this document.
- Appendix C describes the enumerations for fields with multiple valid values.
- Appendix D describes the FIX data types referred to in the 'format' column of each message description.

1.2 FIX Messages Supported

The following table lists the FIX messages that are supported by the X-stream FIX Gateway. X-stream FIX Gateway can be configured to support order management (OM) message only, or support market data (MD) messages only.

Table 1 – FIX Messages Supported

MESSAGE NAME	MESSAGE TYPE	GATEWAY		MESSAGE DIRECTION	MESSAGE FUNCTION
		OM	MD		
Administrative messages					
Logon	A	Y	Y	Inbound Outbound	Identifies and authenticates a user/member establishing a connection to the gateway.
Logout	5	Y	Y	Inbound Outbound	Used to terminate a FIX session.
Reject	3	Y	Y	Inbound Outbound	Response message providing notification regarding messages that cannot be processed by the gateway or FIX Client.
Resend Request	2	Y	Y	Inbound Outbound	Initiates a re-transmission of messages from the gateway.

MESSAGE NAME	MESSAGE TYPE	GATEWAY		MESSAGE DIRECTION	MESSAGE FUNCTION
		OM	MD		
Sequence Reset (Gap Fill)	4	Y	Y	Inbound Outbound	Message has two modes: Sequence Reset - Gap Fill and Sequence Reset - Reset.
Test Request	1	Y	Y	Inbound Outbound	Verifies sequence numbers or communications line status.
Heartbeat	0	Y	Y	Inbound Outbound	Monitors gateway status during periods of inactivity.
Business Message Reject	j	Y	Y	Outbound	Rejects any application message that cannot be processed by the Gateway and cannot be rejected via another message.
Order Management messages					
New Order Single	D	Y	N	Inbound	Used by institutions wishing to electronically submit securities orders for execution.
Order Cancel Request	F	Y	N	Inbound	Request to cancel all of the remaining quantity of an existing order.
Order Cancel / Replace Request	G	Y	N	Inbound	Request message to change the details of an existing order.
Order Cancel Reject	9	Y	N	Outbound	Reject message for an Order Cancel / Replace Request or Order Cancel Request that cannot be honoured.
Order Status Request	H	Y	N	Inbound	Request for querying the details of an order.
Order Mass Action Request	CA	Y	N	Inbound	Request for cancelling multiple orders.
Order Mass Action Report	BZ	Y	N	Outbound	Responds with the action that has resulted from an Order Mass Action Request.
Execution Report	8	Y	N	Outbound	Responds with the action X-stream has taken in response to a new or existing order including acknowledges Order Cancel and Cancel / Replace Requests, order history requests and report fills to orders. Fills against orders are reported via the Execution Report message, as are trade cancels.

MESSAGE NAME	MESSAGE TYPE	GATEWAY		MESSAGE DIRECTION	MESSAGE FUNCTION
		OM	MD		
RFQ Quote Management messages					
Quote Request	R	Y	N	Inbound/Outbound	Used to initiate an RFQ, and distribute it to recipients
Quote Request Reject	AG	Y	N	Outbound	Reject message for an invalid Quote Request
Quote	S	Y	N	Inbound/Outbound	Used by an RFQ recipient to respond with a Quote, and to pass the Quote to the initiator
Quote Cancel	Z	Y	N	Inbound	Used by an RFQ initiator to cancel an active RFQ
Quote Status Report	AI	Y	N	Outbound	Sent by the exchange to initiators and recipients to inform them of the status of an RFQ or Quote
Quote Response	AJ	Y	N	Inbound	Used by an RFQ initiator to accept a Quote
Mass Quote messages					
Mass Quote	i	Y	N	Inbound	Used by market makers to submit market maker order/quotes for multiple securities.
Mass Quote Acknowledgment	b	Y	N	Outbound	Mass Quote Acknowledgement is used as the application level response to a Mass Quote message.
Trade Capture Reporting messages					
Trade Capture Report	AE	Y	N	Inbound/Outbound	Used to report matched trades and negotiated trades, as well as facilitate trade cancel requests.
Trade Capture Report Ack	AP	Y	N	Outbound	Initial response validating or invalidating a submitted Trade Capture Report.
Market Data messages					
Market Data Request	V	N	Y	Inbound	Requests current best market information in a market for a security.
Market Data Request Reject	Y	N	Y	Outbound	Rejects market data request messages that cannot be honoured due to business or technical reasons.

MESSAGE NAME	MESSAGE TYPE	GATEWAY		MESSAGE DIRECTION	MESSAGE FUNCTION
		OM	MD		
Market Data Snapshot / Full Refresh	W	N	Y	Outbound	Responds to the Market Data Request message with the current best market information for a security.
Market Data Incremental Refresh	X	N	Y	Outbound	Used for Market Data incremental updates.
Trading Session Status Request	g	Y	Y	Inbound	Request information on the status of a market.
Trading Session Status	h	Y	Y	Outbound	Responds with the current status of a market.
Security Status Request	e	Y	Y	Inbound	Requests the status of a security. One or more Security Status message are returned as a result of a Security Status Request.
Security Status	f	Y	Y	Outbound	Responds with the current state of a security that is currently listed. May be filtered by board.
News	B	Y	Y	Outbound	Contains bulletin messages initiated by the Exchange.
Reference Data messages					
Security List Request	x	Y	Y	Inbound	Requests a list of securities from the Exchange that match criteria provided in the request.
Security List	y	Y	Y	Outbound	Responds with a list of securities that match the criteria specified in a Security List Request.
Security List Update	BK	Y	Y	Outbound	Responds with updates to the Security List.

2 Session and Infrastructure Messages

This section defines the FIX Session and Infrastructure messages. This section also describes the FIX Session establishment actions.

The FIX Session Level messages are:

- Logon
- Logout
- Reject
- Resend Request
- Sequence Reset (Gap Fill)
- Test Request
- Heartbeat

The FIX Infrastructure messages are:

- Business Message Reject

2.1 FIX Session Establishment

2.1.1 Logon and Authentication

A FIX session must be established with X-stream before the exchange of business messages is allowed. The session is established using the Logon message and part of session establishment processing includes the authentication of the initiator. This requires that a valid SenderCompID (49) which defines the party initiating the session, and an optional password, is provided in the Logon message which can be used for security authentication purposes. A FIX session will not be established if authentication processing fails.

2.1.2 Logon Failures and Account Locking

All logon failures return a Logout message with an appropriate reason code and may include additional text which provides additional information regarding the failure. If the session initiator fails to authenticate with the X-stream system within a defined number of attempts, the account will be locked and all subsequent logon attempts will be rejected. If logons are disabled by the marketplace, a failure to logon will not cause the account to be locked but only rejected. Any other causes for authentication failure will cause the account to be locked after a defined number of failed attempts. To unlock the account requires marketplace operations to reset the account and assign a new password.

2.1.3 FIX Session SenderCompID, Username and Passwords

The SenderCompID (49) and Username (553) are always required for authentication with X-stream and must be included in the Logon message. The SenderCompID, Username and Password (554) can have a maximum length of 32 characters.

The Password field may be omitted from the logon message only if the database password is not encrypted (also see the note about changing the password below).

2.1.4 Changing FIX Session Passwords

Passwords can be changed using the FIX session Login messages and the following will apply:

- Passwords can be changed programmatically using the Logon message only at session establishment and only while they are valid. If the password has expired or cannot be

changed programmatically due to limitations of the Participant's FIX implementation then they must be changed manually by Business or Technical Operations.

- Note that once a password is changed in this manner then it will no longer be optional in the Logon message since it will be encrypted in the database.
- To change the FIX session password at logon time both the current password and the new password must be included in the FIX Logon message. In addition, the SenderCompID (49), Username (553) and the current password must be valid otherwise authentication will fail.
- The current password is sent using the Password (554) field in the Logon message. It should be noted that the password will transit external and internal X-stream networks in plain text if encryption is not utilized (refer to Section 2.1.5 regarding encryption).
- Providing the SenderCompID, UserName (553) and current password are valid, the new password is checked against the password policy for compliance. If the new password complies, it is updated in the X-stream database and becomes the password to be used for the next session logon. If the new password does not comply with the password policy then an error status and message is returned in the Logon confirm message. However, regardless of whether the new password complies or not with the password policy the FIX session will be established if the SenderCompID, UserName and existing password are still valid.

2.1.5 Encryption

X-stream FIX for X-stream does not currently support either password or message encryption over FIX sessions. If encryption is required then hardware based encryption must be used.

2.1.6 FIX Session Logon Confirmation and Logout

Upon receipt of a Logon message and after successful authentication, a Logon message is returned as an acknowledgement indicating that a session has been established. If a session logon has failed for any reason a FIX Logout message is returned. Both the FIX Logon and Logout messages include fields which are used to return status and text information pertaining to either a successful or failed session logon.

2.1.6.1 Logon Confirmation (Session Authenticated)

Both the Logon and Logout message contain the Text(58) field which provide additional may be returned in the Logon notification message. For example, the text information returned may indicate the application version in use, why the new password did not comply with password policies, the number of days until the password expires or other information deemed relevant by the marketplace.

2.1.6.2 Logout (Authentication Failure)

Failure to establish a session with X-stream for any reason will return a Logout message. The Text(58) field in the Logout message may contain additional useful information regarding the reason for the Logout message being returned.

2.2 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 2 – Logon

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

TAG	FIELD NAME	REQ'D	COMMENTS	FORMAT
98	EncryptMethod	Y	0 (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	Y/N	Specifies a different username or userID to use for authentication (required inbound)	String
554	Password	N	Note: minimal security exists without transport-level encryption	String
925	NewPassword	N	Specifies a new password when required.	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 default value (60).

2.3 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 3 – Logout

TAG	FIELD NAME	REQ'D	COMMENTS	FORMAT
Standard Header		Y	MsgType = 5 Exchange may set LastMsgSeqNumProcessed (369)	
58	Text	N	Free format text string	String
Standard Trailer		Y		

2.4 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 4 – Reject

TAG	FIELD NAME	REQ'D	COMMENTS	FORMAT
Standard Header		Y	MsgType = 3	
45	RefSeqNum	Y	MsgSeqNum of rejected message	SeqNum

TAG	FIELD NAME	REQ'D	COMMENTS	FORMAT
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		

2.5 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 5 – Resend Request

TAG	FIELD NAME	REQ'D	COMMENTS	FORMAT
Standard Header		Y	MsgType = 2	
7	BeginSeqNo	Y	Message sequence number of first message in range to be resent	SeqNum
16	EndSeqNo	Y	Message sequence number of last message in range to be resent. If request is for a single message BeginSeqNo (7) = EndSeqNo. If request is for all messages subsequent to a particular message, EndSeqNo = "0" (representing infinity).	SeqNum
Standard Trailer		Y		

2.6 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 6 – Sequence Reset

TAG	FIELD NAME	REQ'D	COMMENTS	FORMAT
Standard Header		Y	MsgType = 4	
123	GapFillFlag	N	Indicates that the Sequence Reset message is replacing administrative or application messages which will not be resent. N = Sequence Reset, Ignore Msg Seq Num Y = Gap Fill Message, Msg Seq Num Field Valid	Boolean
36	NewSeqNo	Y	New sequence number	SeqNum
Standard Trailer		Y		

2.7 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 7 – Test Request

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

2.8 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 8 – 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		

2.9 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 9 – 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		

3 Order Management

The order management category consists of the following messages:

- New Order Single
- Order Cancel Request
- Order Cancel Replace Request
- Order Cancel Reject
- Order Status Request
- Order Mass Action Request
- Order Mass Action Report
- Execution Report

The figures below describe the workflow for new order entry, order cancellation, order modification and order status.

3.1 Unique ClOrdId (11)

Firms submitting order transactions via FIX interface must ensure unique ClOrdId(11) is entered on these transactions. A New Order Single, Cancel or Cancel/Replace message with a non-unique ClOrdID will be rejected.

Order Cancel or Cancel/Replace message using the last used ClOrdID will also be rejected.

3.2 Order Identification

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

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

Both OrderID and ClOrderId should be unique for the current trading day as well as for orders carrying over from previous trading days.

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

3.3 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 should be provided in the Order Cancel/Replace Request message.

3.3.1 Order Attributes allowed to change

Although the FIX protocol allows for virtually all of the Order attributes to be changed, there are limitations as to what the back-end X-stream system allows. The following attributes are allowed to change:

- OrderQty (38)
- MaxFloor (111)
- Price (44)

- OrdType (40)
- TimeInForce (59)
- Yield (236)
- ExpireDate (432)
- ExpireTime (126)
- Account (1)
- ExecInst (18)
- Side (54) – only allowed to change from Sell to Short Sell and vice versa
- TriggerPrice (1102)
- PartyID for PartyRole=3 (Client ID)
- PartyID for PartyRole=999 (Pre-arranged Firm)
- StartCash (921)
- StipulationValue (234)

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

3.4 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.

3.5 Order Suspend and Release (Private order)

Order can be created or public orders suspended by specifying 'z' in ExecInst (18) in NewOrderSingle or Order Cancel/Replace Request. Once suspended, the order is removed from the order book and will not match with other orders. Order can be released by specifying 'y' in ExecInst (18) in Order Cancel/Replace Request. Once released from suspension, the order will be placed back into the order book and is able to match other orders. If 'z' or 'y' is used in Order Cancel/Replace Request to suspend or release an order, all other order attributes should still be present in the messages.

3.6 Order Transfer

The owner of an order can initiate a transfer to another trader (the transferee) within their own firm, submitting an Order Cancel/Replace Request message with OrderOwnershipIndicator (2679) = '3' (Submit transfer) and the transferee specified in the Parties block with PartyRole (452) = '991' (Transfer to trader). Any other changes to the order's existing values will be ignored by the exchange.

Execution Report messages for the order will then be sent by the exchange to both the initiator and transferee, with ExecType = 'o' (Transfer offered). The pending transferee is indicated in the Parties block with PartyRole (452) = '991' (Transfer to trader), while the initiator is indicated with PartyRole (452) = '992' (Transfer operator).

The transferee can accept or decline the transfer by submitting an Order Cancel/Replace Request message with OrderOwnershipIndicator (2679) = '97' (Reject transfer), or '99' (Accept transfer).

Further Execution Report messages will then be sent by the exchange with ExecType = 'r' (Transfer rejected), or 't' (Transferred). If the order is transferred, the transferee will now be identified as the new owner of the order via the TargetSubID (57) field.

The initiator of the transfer can withdraw a pending transfer by submitting an Order Cancel/Replace Request message with OrderOwnershipIndicator (2679) = '98' (Withdraw transfer).

Execution Report messages would then be sent by the exchange with ExecType = 'w' (Transfer withdrawn).

3.7 Users in FIX Order Management

FIX connections can send an order on behalf of a valid trader. The trader ID should be supplied in SenderSubID (50). Orders will be rejected if the FIX session is not allowed to submit them for this trader. Execution Report (8) will contain TargetSubID (57) set to the inbound message's SenderSubID (50). Additionally, the Parties repeating block in Execution Report (8) will contain the trader ID in tag PartyID (448), with tag PartyRole (452) set to ExecutingTrader (12).

Every FIX connection is also associated with a connection user. The exchange will provide the user ID for each FIX connection, based on the port number that FIX connection connects to. This is used in the Logon (A) message's Username (553) field.

In response to a FIX order message, the Parties repeating block in Execution Report (8) will contain the FIX user ID in tag PartyID (448), with tag PartyRole (452) set to EnteringTrader (36).

If an order is amended, EnteringTrader (36) will then indicate the user that last modified the order. For example:

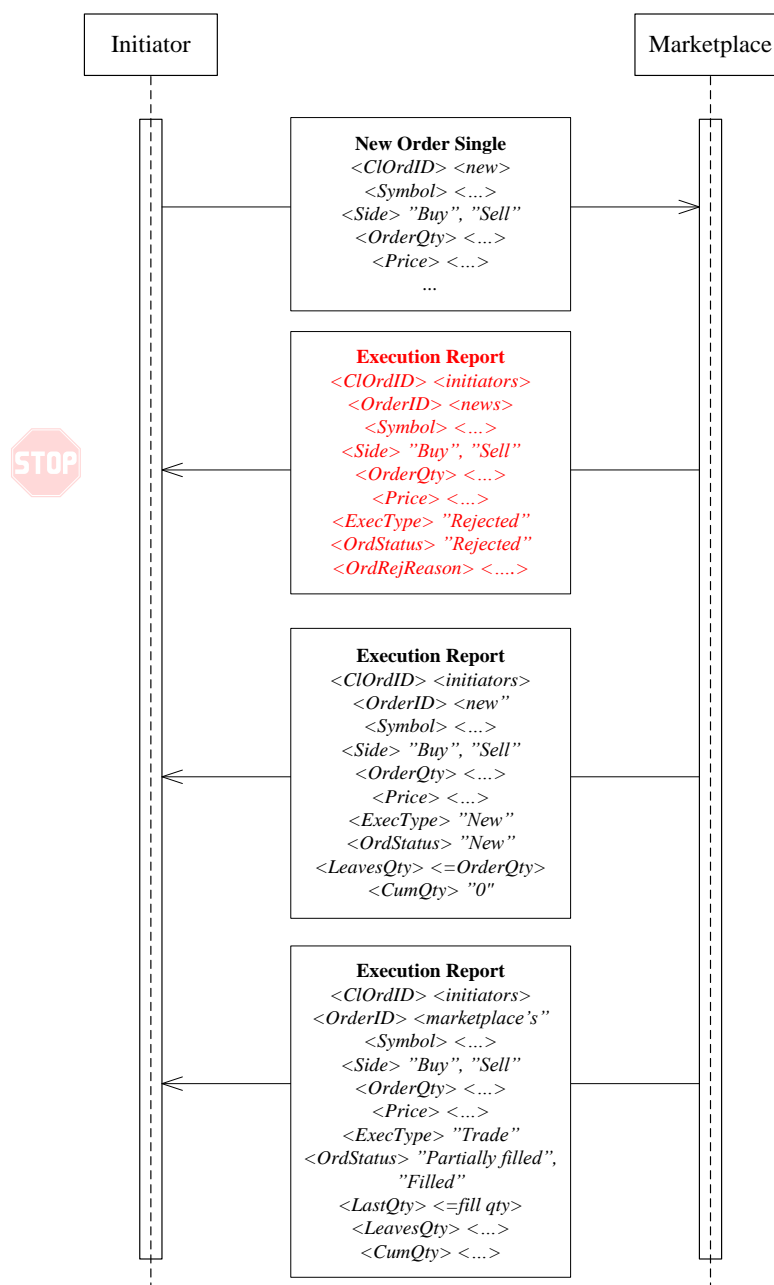
1. Market controller cancelled an order for a trader:
 - EnteringTrader = Market Controller ID
 - ExecutingTrader = Trader ID.
2. Trader modified an order from the workstation:
 - EnteringTrader = Trader ID
 - ExecutingTrader = Trader ID.
3. Trader entered an order via FIX:
 - EnteringTrader = FIX connection user ID
 - ExecutingTrader = FIX Trader ID.

If EnteringTrader (36) does not appear in an Execution Report, then it was the system that modified the order.

3.8 Workflows

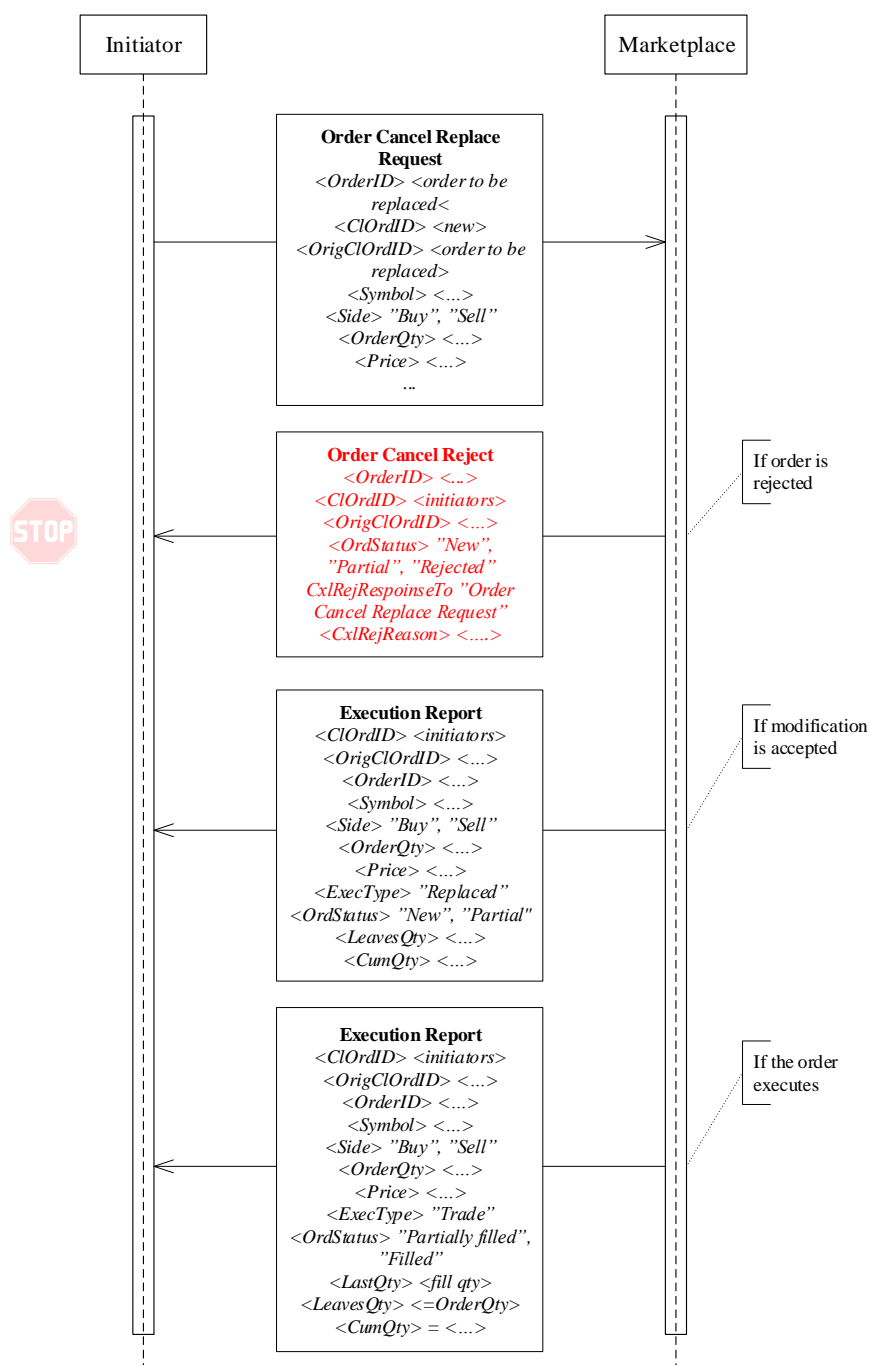
3.8.1 Entering of a New Order

Figure 1 – New Order Entry Workflow



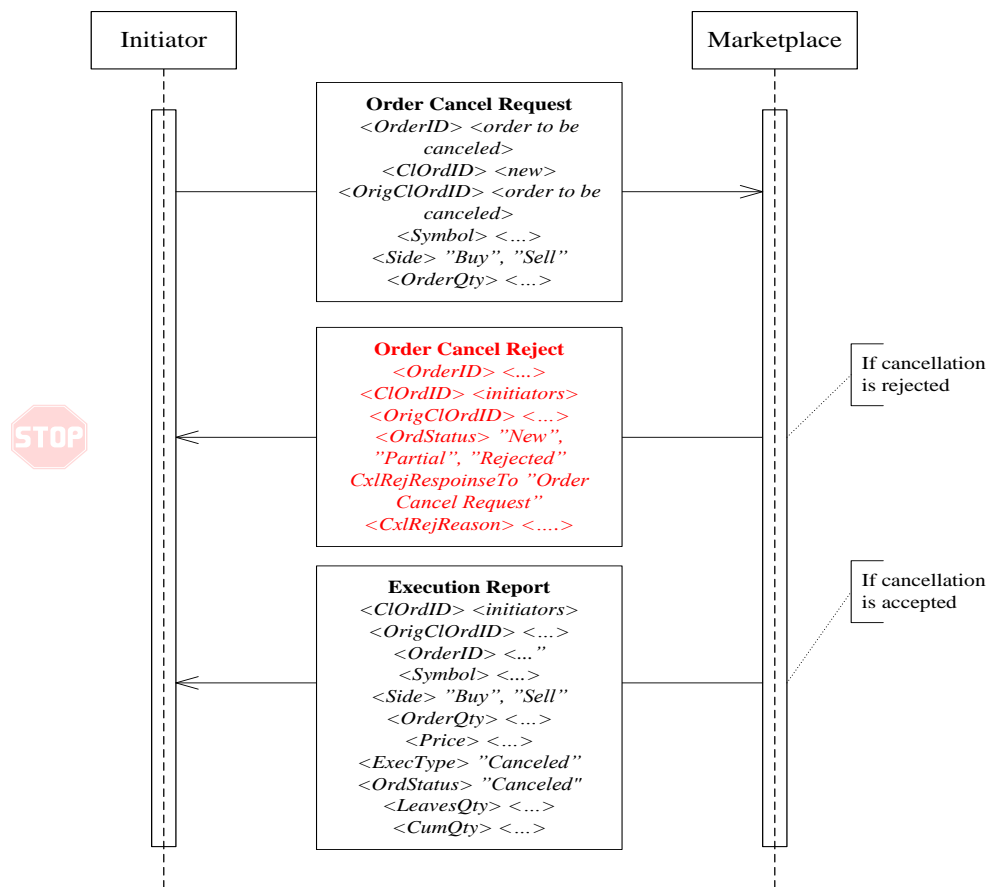
3.8.2 Modification of an Order

Figure 2 – Order Modification Workflow



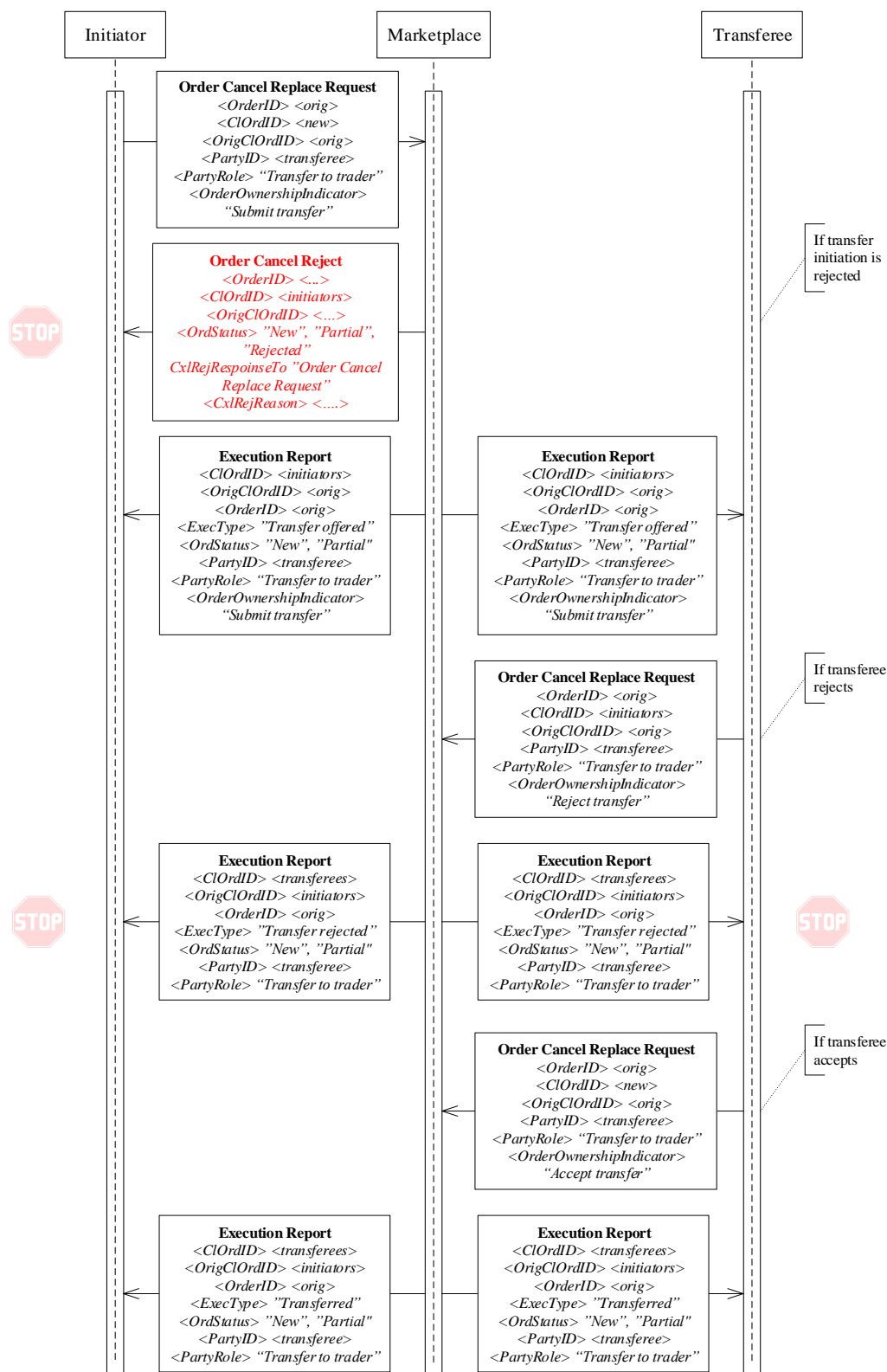
3.8.3 Order Cancellation

Figure 3 – Order Cancellation Workflow



3.8.4 Order Transfer

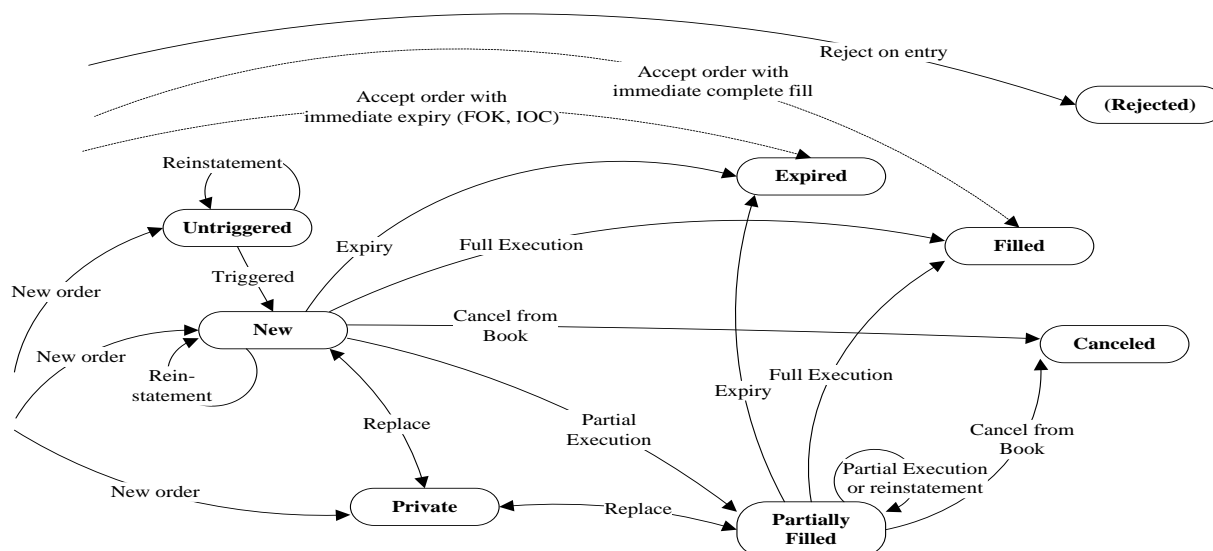
Figure 4 – Order Transfer Workflow



3.8.5 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 5 – Order Status States



3.9 New Order Single (D)

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

Table 10 – New Order Single

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
StandardHeader		Y	MsgType = D SenderSubID (50) is required (executing trader)	
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

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
	Component block <Parties>	N	Insert here the set of "Parties" (firm identification) fields. Valid values for PartyRole: 3 = ClientID 10 = Settlement Location (Depository, required for Fixed Income) 19 = Sponsoring Firm 999 = Pre-arranged Firm	
	Component block <Instrument>	Y	Insert here the set of "Instrument" (symbology) fields. For REPO/TTV/SBB this should be the underlying Equity or Fixed Income security.	
	Component block <FinancingDetails>	N	Insert here the set of "FinancingDetails" fields Required for REPO/TTV/SBB orders.	
	Component block <TriggeringInstruction>	N	Insert here the set of "TriggeringInstruction" fields.	
	Component block <Stipulations>	N	Insert here the set of "Stipulations" fields. Valid values for StipulationType: 'REPOPRICE' – used for the following repurchase orders: <ul style="list-style-type: none"> Equity REPO/TTV buy/sell (optional, if specified must be equal to the current Repo Price for the security) Fixed Income REPO buy/sell (required) 'CLEANPRICE' – used for the following repurchase orders: <ul style="list-style-type: none"> Fixed Income TTV/SBB buy/sell (required) 	
1	Account	N	Specifies Investor Account.	String
18	ExecInst	N	Instructions for order handling. Note for an AON order, the Minimum Quantity field must be equal to the total quantity.	MultipleCharValue
38	OrderQty	Y	Quantity ordered. This value represents the number of shares for equities or par, face or nominal value for Fixed Income instruments. Used for the following repurchase orders: <ul style="list-style-type: none"> Equity sell REPO Equity buy/sell TTV Fixed Income buy/sell REPO/TTV/SBB For all other repurchase orders this field will be ignored.	Qty
40	OrdType	Y	Indicates the type of order.	Char

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
921	StartCash	N	Used for the following repurchase orders, specifies the total value of the order: <ul style="list-style-type: none"> Equity buy REPO (required) Equity sell REPO (optional, can only be lowered from calculated total) Fixed Income buy/sell REPO (optional, can only be lowered from calculated total) 	Amt
44	Price	Y/N	Required for all limit order types – not required for Market orders, or yield based products. For all REPO/TTV/SBB orders, this is the repurchase rate. For yield based products that allow entry by price or yield, this is the clean price. For products traded by dirty price this is the dirty price. Price (44) and Yield (236) are mutually exclusive.	Price
54	Side	Y	Side of the market (buy or sell).	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. Required if an All or None quantity condition and must be equal to the total quantity.	Qty
236	Yield	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. Must be 'session' for a REPO/TTV/SBB order.	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
111	MaxFloor	N	Specifies the disclosed volume on hidden/iceberg orders.	Qty
Standard Trailer		Y		

3.10 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 11 – Order Cancel Request

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
StandardHeader		Y	MsgType = F SenderSubID (50) is required (cancelling trader)	
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
37	OrderID	N	Unique order identifier as assigned by X-stream that identifies the Order to be changed.	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.	String
54	Side	Y	Side of the market. This field is not processed by the Exchange.	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		

3.11 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 12 – Order Cancel/Replace Request

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
StandardHeader		Y	MsgType = G SenderSubID (50) is required (amending trader)	
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
37	OrderID	N	Unique identifier of most recent order as assigned by the Exchange.	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	String
Component block <Parties>		N	Insert here the set of "Parties" (firm identification) fields. Valid value for PartyRole: 3 = ClientID 991 = Transfer to trader 999 = Pre-arranged Firm	
Component block <Instrument>		Y	Insert here the set of "Instrument" (symbology) fields. This is ignored by the exchange.	
Component block <TriggeringInstruction>		N	Insert here the set of "TriggeringInstruction" fields.	

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
Component block <Stipulations>		N	<p>Insert here the set of "Stipulations" fields.</p> <p>Valid values for StipulationType:</p> <p>'REPOPRICE' – used for the following repurchase orders:</p> <ul style="list-style-type: none"> Equity REPO/TTV buy/sell (optional, if specified must be equal to the current Repo Price for the security) Fixed Income REPO buy/sell (required) <p>'CLEANPRICE' – used for the following repurchase orders:</p> <ul style="list-style-type: none"> Fixed Income TTV/SBB buy/sell (required) 	
1	Account	N	Specifies Investor Account.	String
18	ExecInst	N	Instructions for order handling. Note for an AON order, the Minimum Quantity field must be equal to the total quantity.	MultipleChar Value
38	OrderQty	Y	<p>Quantity ordered. This value represents the number of shares for equities or par, face or nominal value for Fixed Income instruments.</p> <p>Used for the following repurchase orders:</p> <ul style="list-style-type: none"> Equity sell REPO Equity buy/sell TTV Fixed Income buy/sell REPO/TTV/SBB <p>For all other repurchase orders this field will be ignored.</p>	Qty
40	OrdType	Y	Indicates the type of order to change to (must follow rules of the Exchange).	Char
921	StartCash	N	<p>Used for the following repurchase orders, specifies the total value of the order:</p> <ul style="list-style-type: none"> Equity buy REPO (required) Equity sell REPO (optional, can only be lowered from calculated total) Fixed Income buy/sell REPO (optional, can only be lowered from calculated total) 	Amt

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
44	Price	Y/N	Required for all limit order types, but not for yield based products. For REPO/TTV/SBB orders, this is the repurchase rate. For yield based products, this is the clean price. For products traded by dirty price this is the dirty price. 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.	UTCTime Stamp
110	MinQty	N	Specifies the minimum fill quantity	Qty
236	Yield	N	Yield percentage.	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. Must be 'session' for a REPO/TTV/SBB order.	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'.	UTCTime Stamp
111	MaxFloor	N	Specifies the disclosed volume on hidden/iceberg orders.	Qty
2679	OrderOwnershipIndicator	N	Change ownership of an order to a specific party. If specified, a Parties block entry with PartyRole = 991 (Transfer to trader) must also be specified. Valid values: 3 = Submit transfer (by initiator) 97 = Reject transfer (by recipient) 98 = Withdraw transfer (by initiator) 99 = Accept transfer (by recipient)	Int
Standard Trailer		Y		

3.12 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 13 – Order Cancel Reject

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
StandardHeader		Y	MsgType = 9 TargetSubID (57) identifies the rejected trader	
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	Free format text string	String
Standard Trailer		Y		

3.13 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.

An Order Status Request issued for an order that has been expired or cancelled without filling any part may be rejected by the exchange.

Table 14 – Order Status Request

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
StandardHeader		Y	MsgType = H SenderSubID (50) is required (requesting trader)	
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.	String
37	OrderID	N	Conditionally required if ClOrdID(11) is not provided. Either OrderID (37) or ClOrdID (11) must be provided.	String
790	OrdStatusReqID	N	Optional, can be used to uniquely identify a specific Order Status Request message. Echoed back on Execution Report if provided.	String
54	Side	Y	Side of the market. Not validated.	Char
StandardTrailer		Y		

3.14 Order Mass Action Request (CA)

The Order Mass Action Request message can be used to request the cancellation of a group of orders that match the criteria specified within the request. Mass order status request is not supported. This is a FIX V5.0 message.

Note: The SenderSubID (50) tag must be specified when using this message. This should contain the ID of the trader who is actioning the request.

Table 15 - Order Mass Action Request

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
StandardHeader		Y	MsgType = CA SenderSubID (50) is required (cancelling trader)	
11	ClOrdID	Y	Unique ID of Order Mass Action Request as assigned by the institution. 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
1373	MassActionType	Y	Specifies the type of action requested. This is a V5.0 tag value. Valid values: 3 = Cancel orders	Int

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
1374	MassActionScope	Y	Specifies the scope of the action. This is a V5.0 tag value. Valid values: 1 = All orders belonging to the specified trader for the specified security/board 7 = All orders belonging to the specified trader 99 = All orders that the specified trader has permission to cancel	Int
Component block <Instrument>		N	Insert here the set of "Instrument" (symbology) fields.	
60	TransactTime	Y	Time of mass order action request by Trader. This field is not processed by the Exchange nor is it used to schedule an action at a future time.	UTCTimeStamp
Standard Trailer		Y		

3.15 Order Mass Action Report (BZ)

The Order Mass Action Report is used to acknowledge an Order Mass Action Request. Note that each order that is affected by the Order Mass Action Request is acknowledged with a separate Execution Report. This is a FIX V5.0 message.

Table 16 - Order Mass Action Report

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
StandardHeader		Y	MsgType = BZ TargetSubID (57) identifies the cancelling trader	
11	ClOrdID	N	ClOrdID provided on the Order Mass Action Request.	String
1369	MassActionReportID	Y	Unique Identifier for the Order Mass Action Report. This is a V5.0 tag value.	String
1373	MassActionType	Y	Specifies the type of mass action requested. This is a V5.0 tag value.	Int
1374	MassActionScope	Y	Specifies scope of Order Mass Action Request. This is a V5.0 tag value.	Int
1375	MassActionResponse	Y	Indicates the action taken by the counterparty order handling system as a result of the Action Request. This is a V5.0 tag value. Valid values: 0 = Rejected 1 = Accepted	Int

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
1376	MassActionRejectReason	N	Indicates why Order Mass Action Request was rejected. Required if MassActionResponse = 0 Reason Order Mass Action Request was rejected. This is a V5.0 tag value. Valid values: 0 = Mass Action not supported 1 = Invalid or unknown Security (or board) 99 = Other	Int
533	TotalAffectedOrders	N	Optional field used to indicate the total number of orders affected by the Order Mass Action Request	Int
Component block <Instrument>		N	Insert here the set of "Instrument" (symbology) fields.	
60	TransactTime	N	Time of the action.	UTCTimeStamp
58	Text	N	Free format text string	String
Standard Trailer		Y		

3.16 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
8. Report on orders with triggers that have been activated. Refer to Appendix B.4 for additional details on Triggered Orders.

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 17 – Execution Report

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
StandardHeader		Y	MsgType = 8 TargetSubID (57) identifies the Executing Trader (owner) of the order	

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
11	ClOrdID	Y/N	<p>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).</p> <p>In the case of RFQ Quotes can be mapped to the the QuoteRespID(693) / QuoteID(117) tag of a Quote.</p>	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.	MultipleChar Value
37	OrderID	Y	OrderID is required to be unique for each chain of orders. Will be '0' if no OrderID available, or for an RFQ Quote that has traded.	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
526	SecondaryClOrdID	N	<p>In the case of mass quotes can be mapped to:</p> <p>- QuoteEntryID(299) of a Mass Quote.</p>	String
198	SecondaryOrderID	N	The previous exchange-generated unique OrderID (37) value for a regular order, provided when the OrderID (37) changes (due to order amendment or hidden order balance refresh etc.)	String
1080	RefOrderID	N	The ID reference to the order being hit or taken. Used to identify an RFQ Quote that has traded. These identifiers are distinct from regular orders.	String

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
1081	RefOrderIDSource	N	Used to specify the source of the RefOrderID for the RFQ Quote. Valid values: '5' = QuoteID (117) '6' = QuoteReqID (131)	Char
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. Valid values for PartyRole are listed in Appendix B.2 <i>Parties Component Block</i> .	
Component block <Instrument>		Y	Insert here the set of "Instrument" (symbology) fields.	
Component block < FinancingDetails >		N	Insert here the set of "FinancingDetails" fields for REPO/TTV/SBB orders.	
Component block <UnderlyingInstrument>		N	Insert here the set of "UnderlyingInstrument" (symbology) fields.	
Component block <TriggeringInstruction>		N	Insert here the set of "TriggeringInstruction" fields.	
Component block <Stipulations>		N	Insert here the set of "Stipulations" fields. Used to disseminate the Repo Price for a REPO or Equity TTV, or the Clean Price for a Fixed Income TTV/SBB order or fill.	
1	Account	N	Specifies Investor Account.	String
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. For REPO/TTV/SBB orders this specifies the repurchase rate.	Price
32	LastQty	N	Quantity (e.g. shares) bought/sold on this fill.	Qty
38	OrderQty	N	Quantity ordered.	Qty
921	StartCash	N	For REPO/TTV/SBB orders only, specifies the total value of the order.	Amt

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
922	EndCash	N	For REPO/TTV/SBB orders only, specifies the return value of the order.	Amt
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. For REPO/TTV/SBB orders this is the repurchase rate. For yield based products that allow entry by price or yield, this is the clean price. For products traded by dirty price this is the dirty price.	Price
20102	DirtyPrice	N	The dirty price for the order or fill (only for Fixed Income products traded by clean price). This is a custom tag.	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.	UTCTime Stamp
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.	UTCTime Stamp
63	SettlType	N	Indicates the order settlement period.	String
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
236	Yield	N	Yield percentage for the order (Fixed Income only). This is the yield equivalent of the Price (or RepoPrice/CleanPrice, if specified).	Percentage
1623	FillYield	N	Yield percentage for this fill (Fixed Income only)	Percentage
381	GrossTradeAmt	N	Total amount traded expressed in units of currency.	Amt
880	TrdMatchID	N	Identifier assigned by the trading system for a trade. This is the X-stream trade id.	String
828	TrdType	N	Indicates the type of trade. If not specified on a fill it is a regular trade. Valid values: 99 = RFQ Trade	Int
829	TrdSubType	N	If specified, this indicates whether the trader is on the buy or sell side of a fill.	Int
824	TradeLegRefID	N	Reference to the TrdMatchID of an associated REPO/TTV/SBB trade at start and return date.	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
574	MatchType	N	Indicates whether a fill was pre-arranged. Valid values: 99 = Pre-arranged	String
111	MaxFloor	N	Specifies the maximum disclosed volume for a hidden/iceberg order.	Qty
2679	OrderOwnershipIndicator	N	Indicates that the order has a transfer pending to the user specified in the Parties block entry with PartyRole = 991 (Transfer to trader). Valid values: 3 = Submit transfer (i.e. transfer pending)	Int
1138	DisplayQty	N	Indicates the current volume disclosed on a hidden/iceberg order. Not set if unavailable or inapplicable. This is a V5.0 tag value.	Qty
58	Text	N	On an error condition, this will specify X-stream generated error message.	String
797	CopyMsgIndicator	N	Drop Copy	Boolean

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
	Standard Trailer	Y		

4 Mass Quote Management

Market makers can submit mass quotes through X-stream FIX. The mass quote management category consists of the following messages:

- Mass Quote
- Mass Quote Acknowledgement

Note: Quote Management is not supported.

4.1 Unique QuoteID (117)

X-stream will not check for uniqueness of QuoteID (117) on a MassQuote message. Firms submitting order transactions via FIX interface must ensure a unique QuoteID (117) is entered on these transactions.

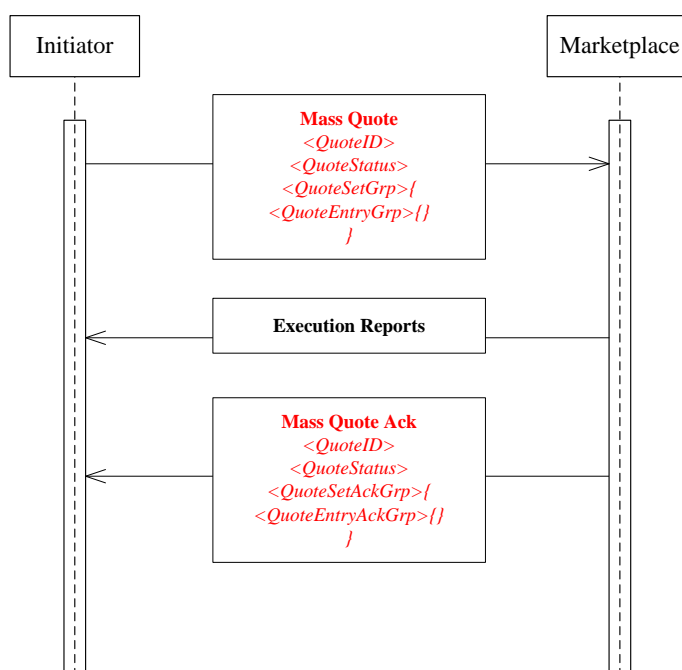
4.2 Workflows

4.2.1 Entering of a mass quote

Mass Quote contains a group of Quote Sets. A Quote Set contains a group of Quote Entries. Each quote entry can submit a pair of bid and offer quotes on a single instrument. Existing quotes in the instrument will be replaced.

Mass Quote Acknowledgement is used as the application level response to a Mass Quote message. It reports the status of a Mass Quote, as well as the result of every Quote Set and Quote Entry.

Figure 6 – Mass Quote workflow



Execution Reports will be generated for quotes individually. The ClOrdID(11) field will be set to QuoteID(117), the SecondaryClOrdID(526) will be set to QuoteEntryID(299). Every quote will

be assigned an unique OrderID(37), which can be used in Order Cancel Replace Request(G) to amend the quote. However OrigClOrdID(41) can not be used to amend or cancel a quote.

4.2.2 Quote Entry Cancel

All open quotes can be cancelled on the buy and/or sell side of a security by sending a quote entry with bid and/or offer prices and sizes all set to zero in a Mass Quote message. In this case the Mass Quote Acknowledgement will indicate the cancellation.

If a quote is cancelled by the system or expired, an ExecutionReport will be generated.

4.3 Mass Quote (i)

The Mass Quote message can contain quotes for multiple securities to support applications that allow for the mass quoting of an option series. Two levels of repeating groups have been provided to minimize the amount of data required to submit a set of quotes for a class of options (e.g. all option series for IBM).

X-stream FIX supports only one QuoteSetGrp per Mass Quote message. Fragmented Mass Quote messages will be rejected - NoQuoteEntries(295) should always be the same as TotNoQuoteEntries(304) in the same QuotSetGrp.

Table 18 – Mass Quote

TAG	FIELDNAME		REQ'D	COMMENTS	FORMAT
StandardHeader			Y	MsgType = i SenderSubID (50) is required (executing trader)	
117	QuoteID		Y	Unique identifier for quote	String
1	Account		N	Specifies Investor Account	String
Component block <Parties>			N	Insert here the set of "Parties" Valid value for PartyRole: 3 = ClientID	
Start of Component block, expanded in line < QuotSetGrp >					
296	NoQuoteSets		Y	The number of sets of quotes in the message. Must be 1.	NumInGrp
→	302	QuoteSetID	Y	Identifier for the Quote Set. Echoed back by the exchange in a Mass Quote Acknowledgement message. Must be the first field in the repeating group.	String
→	304	TotNoQuoteEntries	Y	Total number of quotes for the quote set across all messages. Should be the sum of all NoQuoteEntries in each message that has repeating quotes that are part of the same quote set.	Int
→	Start of Component block, expanded in line < QuotEntryGrp >				
→	295	NoQuoteEntries	Y	The number of quotes for this QuotSet that follow in this message. Maximum allowed is 5.	NumInGrp

TAG	FIELDNAME			REQ'D	COMMENTS	FORMAT
→	→	299	QuoteEntryID	Y	Unique identifier for the quote.	String
→	→	Component block <Instrument>		Y	Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages."	
→	→	132	BidPx	N	Bid price/rate.	Price
→	→	133	OfferPx	N	Offer price/rate.	Price
→	→	134	BidSize	N	Quantity of bid	Qty
→	→	135	OfferSize	N	Quantity of offer	Qty
→	→	62	ValidUntilTime	N	Indicates expiration time of quote (always expressed in UTC (Universal Time Coordinated, also known as "GMT"))	UTCTimest amp
→	End of Component block, expanded in line < QuotEntryGrp >					
End of Component block, expanded in line < QuotSetGrp >						
Standard Trailer				Y		

4.4 Mass Quote Acknowledgement (b)

Mass Quote Acknowledgement is used as the application level response to a Mass Quote message. The Mass Quote Acknowledgement contains a field for reporting the reason in the event that the entire quote is rejected (QuoteRejectReason[300]). The Mass Quote Acknowledgement also contains a field for each quote that is used in the event that the quote entry is rejected (QuoteEntryRejectReason[368]).

Table 19 – Mass Quote Acknowledgement

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
StandardHeader		Y	MsgType = b TargetSubID (57) identifies the executing trader	
117	QuoteID	Y	Unique identifier for quote	String
297	QuoteStatus	N	Status of the mass quote acknowledgement. Valid values: 0 = Accepted 9 = Rejected	Int
300	QuoteRejectReason	N	Reason Quote was rejected.	Int
1	Account	N	Specifies Investor Account.	String
58	Text	N	Free format text string	String
Start of Component block, expanded in line < QuotSetAckGrp >				

TAG	FIELDNAME			REQ'D	COMMENTS	FORMAT
296	NoQuoteSets			Y	The number of sets of quotes in the message	NumInGrp
→	302	QuoteSetID		Y	Identifier for the Quote Set. Must be the first field in the repeating group.	String
→	304	TotNoQuoteEntries		N	Total number of quotes for the quote set across all messages. Should be the sum of all NoQuoteEntries in each message that has repeating quotes that are part of the same quote set. Required if NoQuoteEntries > 0	Int
→	893	LastFragment		N	Indicates whether this is the last fragment in a sequence of message fragments. Only required where message has been fragmented.	Boolean
→	Start of Component block, expanded in line < QuotEntryAckGrp >					
→	295	NoQuoteEntries		Y	The number of quotes for this QuotSetAck that follow in this message.	NumInGrp
→	→	299	QuoteEntryID	Y	Unique identifier for the quote entry across the complete set of all quotes for a given quote provider.	String
→	→	Component block <Instrument>		Y	Instrument component received in QuotEntryGrp	
→	→	132	BidPx	N	Bid price/rate.	Price
→	→	133	OfferPx	N	Offer price/rate.	Price
→	→	134	BidSize	N	Quantity of bid	Qty
→	→	135	OfferSize	N	Quantity of offer	Qty
→	→	62	ValidUntilTime	N	Indicates expiration time of quote (always expressed in UTC (Universal Time Coordinated, also known as "GMT"))	UTCTimestamp
→	→	1167	QuoteEntryStatus	N	Status of quote entry action. Valid values: 0 = Accepted 5 = Rejected	Int
→	→	368	QuoteEntryRejectReason	N	Reason Quote Entry was rejected.	Int
→	End of Component block, expanded in line < QuotEntryAckGrp >					

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
End of Component block, expanded in line < QuotSetAckGrp >				
Standard Trailer		Y		

5 Request for Quote (RFQ)

A Request for Quote (RFQ) can be issued by a trader (the 'RFQ initiator') interested in ascertaining the market for an instrument. These requests are then distributed by the exchange to the listed participants (the 'RFQ quoters') as per the initiators message. The receivers of these requests can then submit quotes back to the exchange, and if approved by the initiator they will produce a trade.

Once initiated, an RFQ can only be accepted or cancelled. It cannot be amended.

All of QuoteReqID, QuoteID, QuoteRespID and ClOrdID must be unique.

5.1 Quote Request (R)

A Quote Request [R] message can be sent by an RFQ initiator to the exchange. The initiator will receive a Quote Request Reject [AG] message if the request failed, otherwise a Quote Status Report [AI] message will be received back by the initiator, confirming the successful request submission.

If the request is valid, the exchange will send Quote Request [R] messages to the participants listed in the initiator's Quote Request [R] message.

To cancel a currently active RFQ a Quote Request [R] message can be submitted by the RFQ initiator with OrderQty (38) set to 0.

Table 20 – Quote Request

TAG	FIELDNAME		REQ'D	COMMENTS	FORMAT
StandardHeader			Y	MsgType = R SenderSubID (50) is required (entering trader)	
131	QuoteReqID		Y	Unique identifier for quote request. It's an exchange generated Id when the quote message is sent to participants.	String
Start of Component block, expanded in line < QuoteReqGrp >					
146	NoRelatedSym		Y	The number of sets of related symbols (instruments) in Request. Only value 1 is supported	NumInGrp
→	Component block <Instrument>		Y	The instrument that the trader wishes to receive prices for.	
→	1	Account	Y/N	Specifies Investor Account. Not published to the receivers of the Quote Request.	String

TAG	FIELDNAME		REQ'D	COMMENTS	FORMAT
→	54	Side	N	Side for the RFQ initiator (always), indicating whether the initiator is buying or selling. Valid values: 1 = Buy 2 = Sell If not set, the quote request is for both sides.	Char
→	38	OrderQty	Y	The maximum amount that the initiator wishes to trade (this limit will be enforced). Set this to 0 to cancel an active Quote Request.	Qty
→	62	ValidUntilTime	N	The time that this RFQ will expire. If this time has passed when received by the system the RFQ will be rejected.	UTCTimestamp
→	Component block <Parties>		N	Insert here the set of "Parties" fields. Up to 5 'Contra' parties can be specified for published the Quote Request to. If not specified Quote Request is published to all FIX participants. Valid values for PartyRole: 3 = ClientID 10 = Settlement Location (Depository) 17 = Contra Firm 37 = Contra Trader Sent by the exchange only, to indicate the RFQ initiator: 36 = Entering Trader 7 = Entering Firm 12 = Executing Trader 1 = Executing Firm	
End of Component block, expanded in line < QuotReqGrp >					
Standard Trailer			Y		

5.2 Quote Request Reject (AG)

The Quote Request Reject [AG] message can be sent by the exchange to reject a Quote Request [R] message.

Table 21 – Quote Request Reject

TAG	FIELDNAME		REQ'D	COMMENTS	FORMAT
StandardHeader			Y	MsgType = AG TargetSubID (57) identifies the rejected trader	
131	QuoteReqID		Y	Unique identifier for quote request to be rejected.	String
658	QuoteRequestRejectReason		Y	Reason Quote was rejected. Valid values: 99 = Other	
Start of Component block, expanded in line < QuotReqRjctGrp >					
146	NoRelatedSym		Y	Should always be 1.	NumInGrp
→	Component block <Instrument>		Y		
→	38	OrderQty	Y	Echoed from a Quote Request message.	Qty
End of Component block, expanded in line < QuotReqRjctGrp >					
58	Text		N	Error message when exchange rejects a quote request	String
Standard Trailer			Y		

5.3 Quote (S)

The Quote [S] message can be sent to the exchange by a quoter in response to a Quote Request [R] message submitted by an RFQ initiator. The quoter will receive a Quote Status Report [AI] message indicating whether the quote submission succeeded or was rejected.

If the quote is valid, the exchange will forward the Quote [R] message to the initiator.

All quotes are tradeable.

Fixed Income securities may only be quoted by yield, while all other markets may only be quoted by price.

Table 22 – Quote

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
StandardHeader		Y	MsgType = S SenderSubID (50) is required when sent to the exchange (RFQ initiator) TargetSubID (57) identifies the RFQ initiator when sent by the exchange	
131	QuoteReqID	Y	The ID of the Quote Request [R] that this is in response to.	String
117	QuoteID	Y	A unique Quote ID assigned by the Dealer for the quote. It's an exchange generated Id when the quote message is sent to RFQ initiator.	String

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
Component block <Instrument>		Y	Insert here the set of "Instrument" (symbology) fields. This is ignored by the exchange.	
1	Account	Y/N	Specifies Investor Account. Not published to the receivers of the Quote.	
132	BidPx	N	The Bid Price being provided by the quoter (non-Fixed Income only, not valid for Fixed Income). Must be specified with BidSize tag 134.	Price
632	BidYield	N	The Bid Yield being provided by the quoter (Fixed Income only, not valid for non-Fixed Income). Must be specified with BidSize tag 134.	Percentage
133	OfferPx	N	The Offer Price being provided by the quoter (non-Fixed Income only, not valid for Fixed Income). Must be specified with OfferSize tag 135.	Price
634	OfferYield	N	The Offer Yield being provided by the quoter (Fixed Income only, not valid for non-Fixed Income). Must be specified with OfferSize tag 135.	Percentage
134	BidSize	N	The notional amount of the bid, conditionally required if BidPx tag 132 or BidYield tag 632 are specified. Must be the same value as OfferSize tag 135 if both a bid and offer are provided.	Qty
135	OfferSize	N	The notional amount of the offer, conditionally required if OfferPx tag 133 or OfferYield tag 634 are specified. Must be the same value as BidSize tag 134 if both a bid and offer are provided.	Qty
62	ValidUntilTime	N	The time when the quote will expire.	UTCTimestamp
Component block <Parties>		Y/N	Insert here the set of "Parties" fields. Valid values for PartyRole: 3 = ClientID 10 = Settlement Location (Depository) Sent by the exchange only, to indicate the quoter: 17 = Contra Firm 37 = Contra Trader	
Standard Trailer		Y		

5.4 Quote Cancel (Z)

The Quote Cancel [Z] message can be sent by an RFQ quoter to the exchange to cancel an active quote that was previously entered using the Quote [S] message.

If successful, the exchange will forward the QuoteCancel [Z] message to the RFQ initiator, indicating that the quote is no longer valid. The exchange will send a Quote Status Report [AI]

message back to the RFQ quoter in response to the Quote Cancel [Z] message, indicating whether the cancel succeeded or was rejected.

Table 23 – Quote Cancel

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
StandardHeader		Y	MsgType = Z SenderSubID (50) is required (cancelling trader)	
117	QuoteID	Y/N	Conditionally required when QuoteCancelType(298) = 5 (cancel quote specified in QuoteID).	String
298	QuoteCancelType	Y	Identifies the type of Quote Cancel request. The following values are supported: 5 = Cancel quote specified in QuoteID	Int
Standard Trailer		Y		

5.5 Quote Status Report (AI)

The quote status report message is sent by the exchange as a response to a Quote Request [R] message, Quote [S] message, Quote Cancel [Z] message, or Quote Response [AJ] message.

Table 24 – Quote Status Report

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
StandardHeader		Y	MsgType = AI TargetSubID (57) identifies the RFQ initiator or the quoter	
131	QuoteReqID	N	Unique identifier for quote request (RFQ).	String
117	QuoteID	N	A unique Quote ID assigned by the Dealer for the quote. It's an exchange generated Id when the quote message is sent to RFQ initiator.	String
298	QuoteCancelType	N	Identifies the type of Quote Cancel request.	Int
693	QuoteRespID	N	A unique identifier specified by the initiator.	String
Component block <Instrument>		Y	Insert here the set of "Instrument" (symbology) fields.	
54	Side	N	Side for the RFQ initiator (always), indicating whether the initiator is buying or selling. Valid values: 1 = Buy 2 = Sell If not set, the initiator has requested a quote for both sides.	Char
1	Account	N	Specifies Investor Account.	

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
38	OrderQty	N	The maximum amount that the initiator wishes to trade.	Qty
132	BidPx	N	The Bid Price being provided by the quoter (non-Fixed Income only). Must be specified with BidSize tag 134.	Price
632	BidYield	N	The Bid Yield being provided by the quoter (Fixed Income only). Must be specified with BidSize tag 134.	Percentage
133	OfferPx	N	The Offer Price being provided by the quoter (non-Fixed Income only). Must be specified with OfferSize tag 135.	Price
634	OfferYield	N	The Offer Yield being provided by the quoter (Fixed Income only). Must be specified with OfferSize tag 135.	Percentage
134	BidSize	N	The notional amount of the bid, conditionally required if BidPx tag 132 or BidYield tag 632 are specified. Must be the same value as OfferSize tag 135 if both a bid and offer are provided.	Qty
135	OfferSize	N	The notional amount of the offer, conditionally required if OfferPx tag 133 or OfferYield tag 634 are specified. Must be the same value as BidSize tag 134 if both a bid and offer are provided.	Qty
62	ValidUntilTime	N	The time when the quote will expire.	UTCTimestamp
297	QuoteStatus	N	Identifies the status of the RFQ or quote action. Valid values: 9 = Rejected (last action was rejected) 10 = Pending (RFQ entered) 17 = Canceled (RFQ/Quote cancelled) 16 = Active (Quote entered) 19 = Pending End Trade (a matched Quote awaiting Exchange approval, should it be required)	Int
300	QuoteRejectReason	N	Reason quote or RFQ action was rejected	Int
58	Text	N	Message regarding the status of the quote.	String

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
	Component block <Parties>	N	Insert here the set of "Parties" fields. Valid values for PartyRole: 3 = ClientID 10 = Settlement Location (Depository) To indicate the quoter: 17 = Contra Firm 37 = Contra Trader To indicate the RFQ initiator: 36 = Entering Trader 7 = Entering Firm 12 = Executing Trader 1 = Executing Firm	
	Standard Trailer	Y		

5.6 Quote Reponse (AJ)

The Quote Response [AJ] message can be sent by an RFQ initiator to the exchange to accept or decline a quote that was received in a Quote [S] message from the exchange.

A Quote Status Report [AI] will be received back by the initiator to indicate whether the response succeeded (i.e. cancelled, traded or pending approval), or was rejected.

When a quote is successfully traded, Execution Report [8] messages are sent to both the RFQ initiator and RFQ quoter, providing details of the trade.

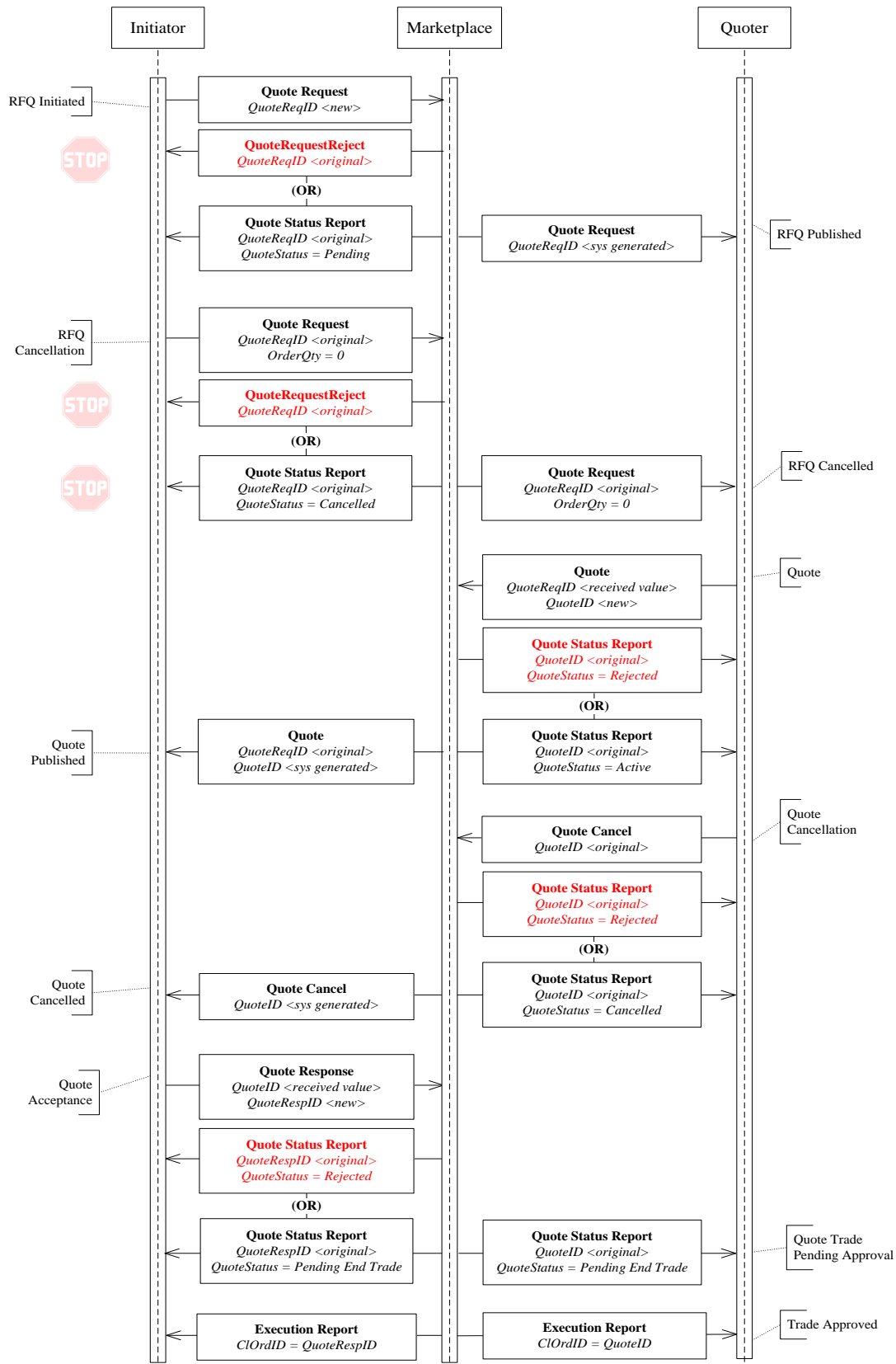
Table 25 – Quote Reponse

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
	StandardHeader	Y	MsgType = AJ SenderSubID (50) is required (RFQ initiator)	
693	QuoteRespID	Y	A unique identifier specified by the initiator.	String
117	QuoteID	Y	Identifier of the quote being accepted.	String
694	QuoteRespType	Y	Type of response this Quote Response is. Valid values: 1 = Hit/Lift (accept) 6 = Pass (decline)	Int
54	Side	Y	Side for the RFQ initiator (always), indicating whether the initiator is buying or selling. Valid values: 1 = Buy 2 = Sell	Char
	Component block <Instrument>	Y	Insert here the set of "Instrument" (symbology) fields. These fields will be ignored by the exchange.	
	Standard Trailer	Y		

5.7 Workflows

5.7.1 Request for Quote

Figure 7 – Request for Quote Workflow



6 Trade Capture Reporting

Trade Capture reports are used for:

- Relaying Confirmed Trades to various parties not directly involved in the execution. Those messages are outbound (from the marketplace).
- Relaying Confirmed Trades to counterparties of the trade. Where Execution Reports may be sufficient for front-office purposes, Trade Capture Reports can serve more demanding back-office processes better. Those messages are outbound (from the marketplace).
- Reporting of privately negotiated trades. Those messages may be inbound or outbound.
- Requesting, confirming and declining a Trade Cancellation

6.1 Trade Capture Messages

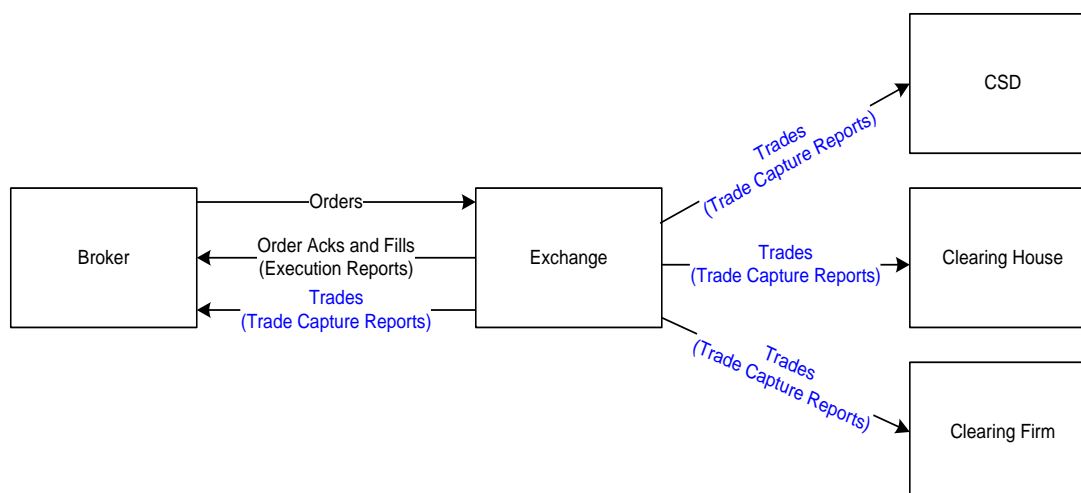
The Trade Capture category of messages consists of the following:

- Trade Capture Report
- Trade Capture Report Ack

6.2 Workflows

6.2.1 Trade Capture Workflow for Multiple Counterparties

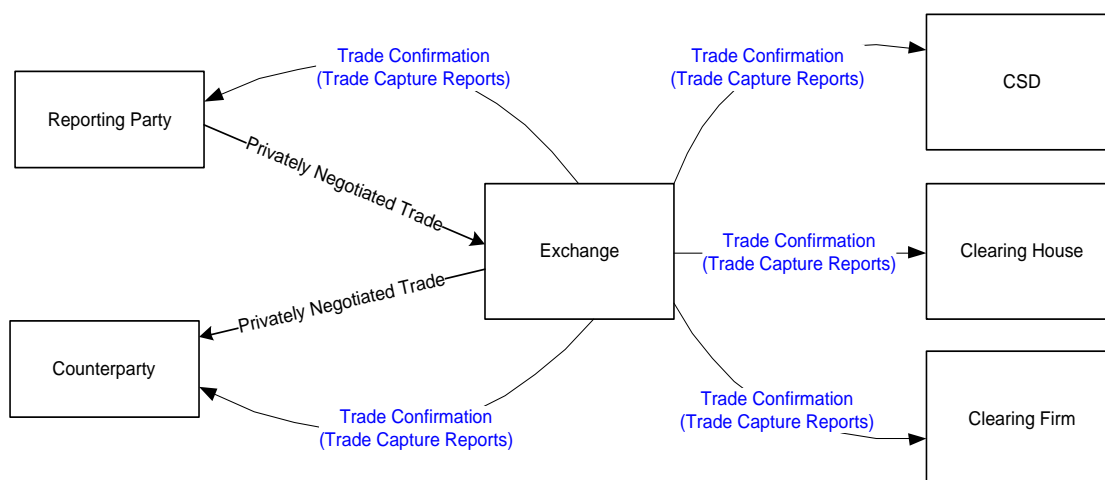
Figure 8 – Trade Capture High Level Workflow



6.2.2 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 9 – Diagram for Privately Negotiated Trade

6.2.3 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 may 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 a TradeCaptureReport (AE) to the counterparty, with a new TradeReportID (571) and a new ExecID (17).
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 2/3.
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 2/3. Once the exchange approves the trade, a further TradeCaptureReport will be sent with 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".

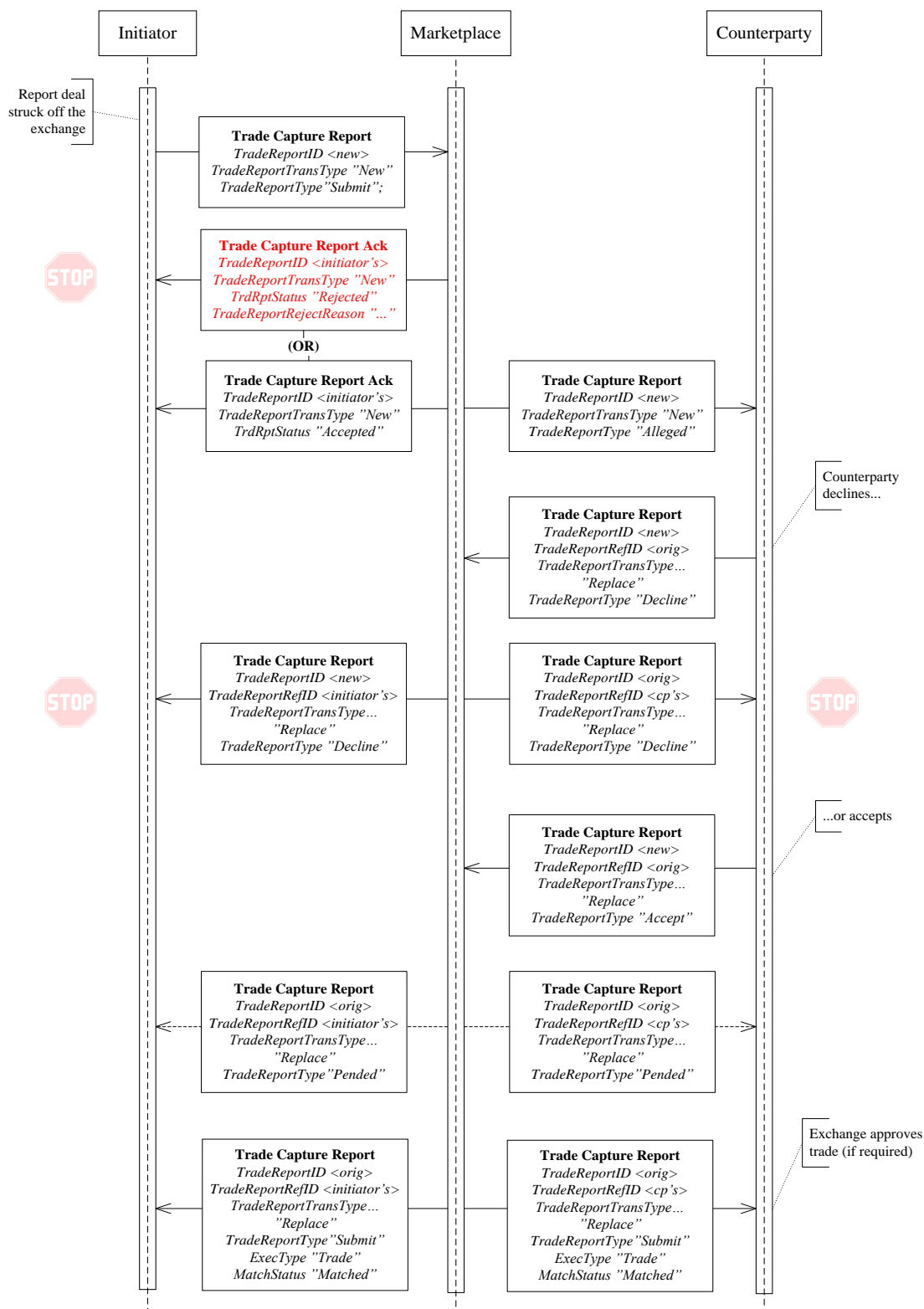
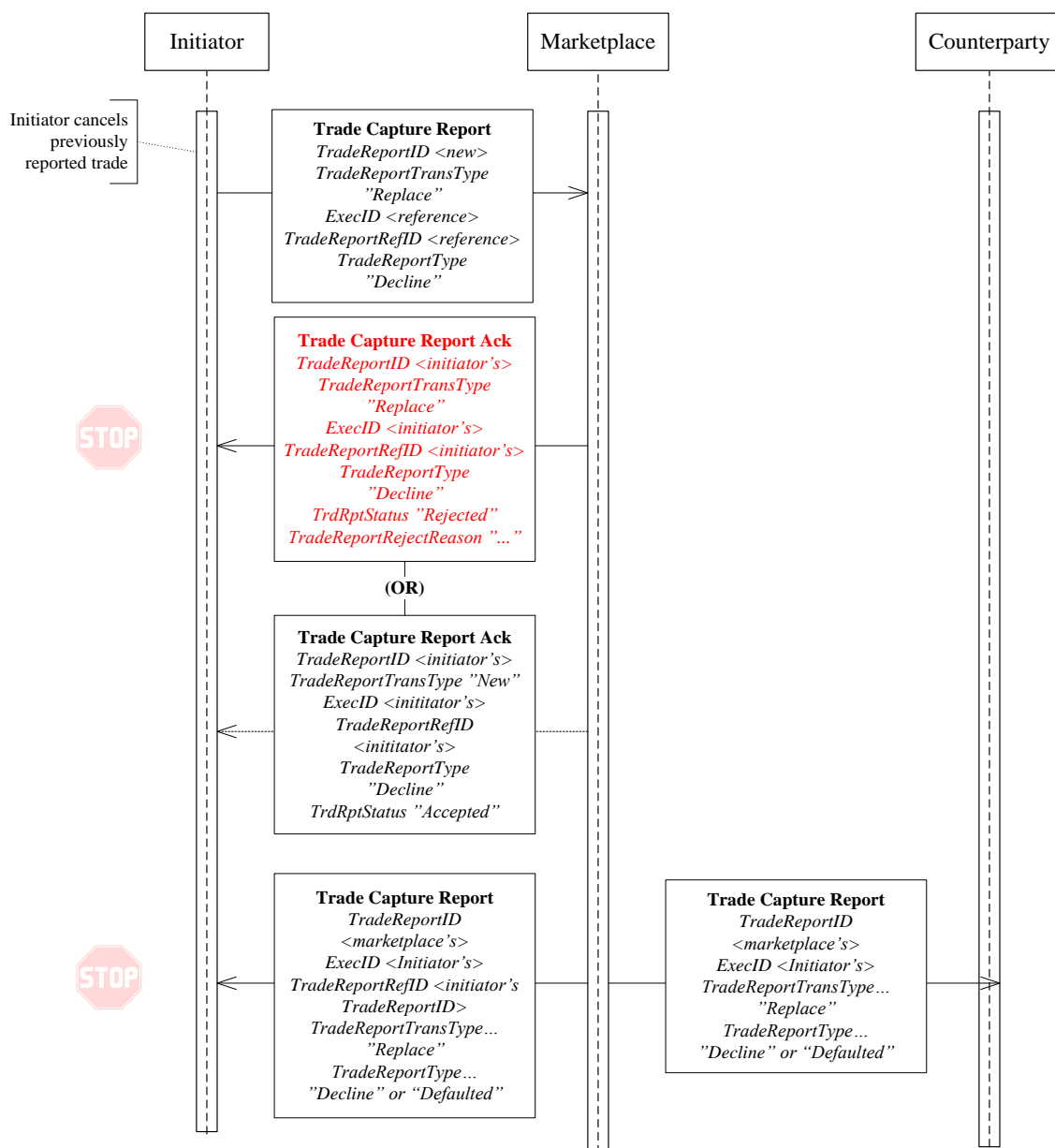
Figure 10 – One-Party Trade Report for Pass-through to Counterparty


Figure 11 – One-Party Trade Report decline (or time out) before counterparty confirms



6.2.4 Trade Cancellation Request Workflow

A Trade Cancellation can be requested, confirmed or declined via FIX according to the following workflow.

1. The initiator sends a TradeCaptureReport (AE) to the exchange with a TrdMatchID (880) that was previously received from the exchange for a trade, and with TradeReportType (856) = 96 'Submit Trade Report Cancel'.
2. The exchange will send to the initiator a TradeCaptureReportAck (AR), either rejecting or accepting their TradeCaptureReport.
3. If accepted, the exchange will also send a TradeCaptureReport (AE) to the counterparty, referencing the same TrdMatchID (880), and with TradeReportType (856) = 14 'Alleged Trade Report Cancel'.

4. The counterparty can decline the trade cancel request by sending a TradeCaptureReport to the exchange referencing the same TrdMatchID (880), and with TradeReportType (856) = 98 'Decline Trade Report Cancel'. The initiator can also withdraw the request in the same manner.

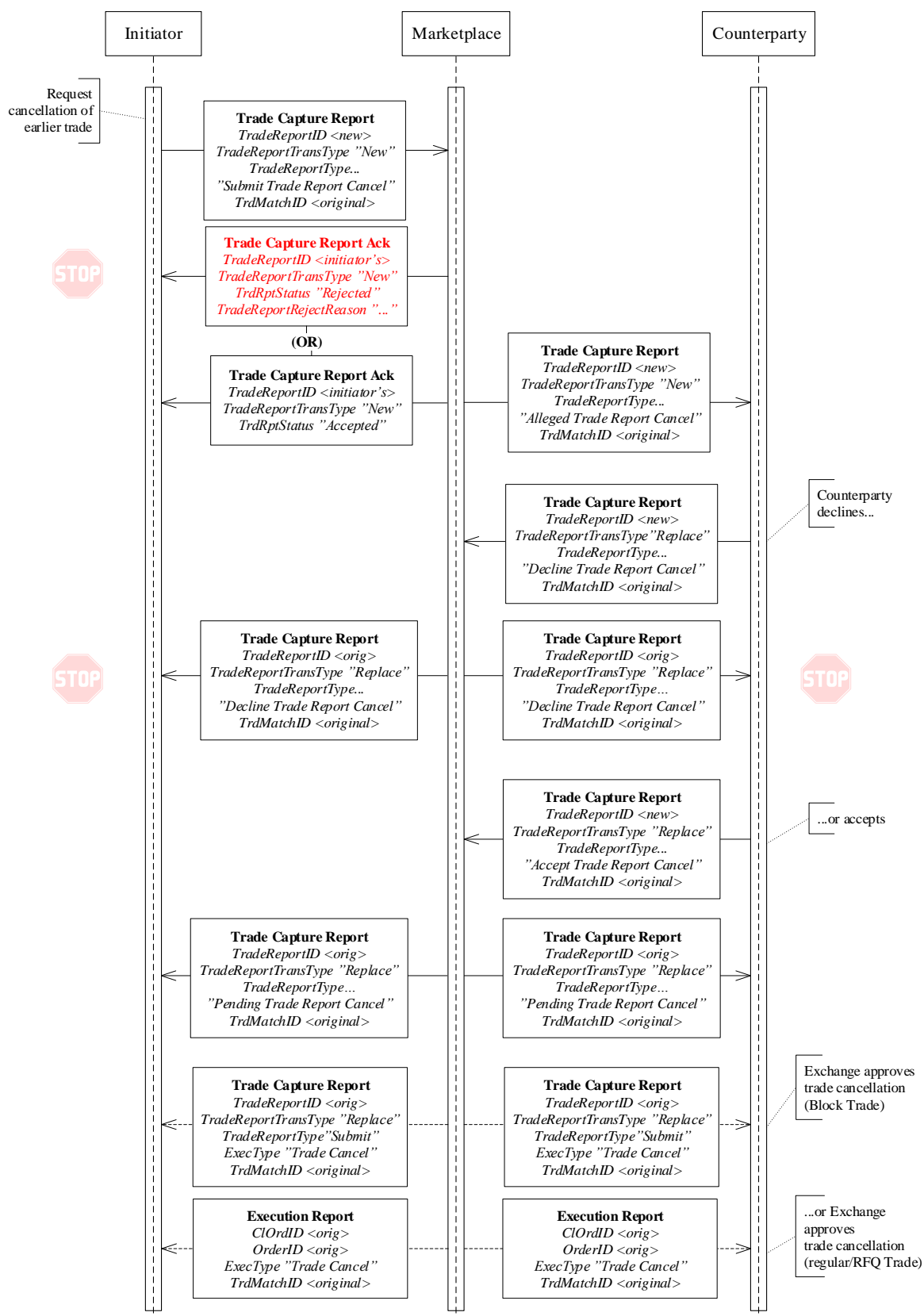
In response, the exchange will send a TradeCaptureReport to both parties, with TradeReportType (856) = 98 'Decline Trade Report Cancel'. This indicates that the trade cancel request has been terminated, and the original trade remains in effect.

5. The counterparty can confirm the trade cancel request by sending a TradeCaptureReport to the exchange referencing the same TrdMatchID (880), and with TradeReportType (856) = 97 'Accept Trade Report Cancel'.

In response, the exchange will send a TradeCaptureReport to both parties, with TradeReportType (856) = 99 'Pending Trade Report Cancel'.

Once the exchange then approves the trade, a further message will be sent to indicate that the trade has been cancelled, according to the type of trade:

- a. If the original trade was a regular trade or RFQ, an ExecutionReport message will be sent with ExecType (150) = H 'Trade Cancel'
- b. If the original trade was a privately negotiated deal (Block Trade), a TradeCaptureReport message will be sent with ExecType (150) = H 'Trade Cancel'

Figure 12 – Trade Cancel Request workflow

6.3 Trade Capture Report (AE)

The Trade Capture Report message can be used for either:

- a) Reporting privately negotiated trades (i.e. Block Trade)
 - Sent by an initiator to the exchange to report a privately negotiated 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
- b) Trade cancellation requests (of all trade types)
 - Sent by a party to a previous trade to request a cancellation
 - Sent by the exchange to parties to inform them that the cancellation has been initiated
 - Sent by a counterparty to a previous trade to confirm or decline a cancellation
 - Sent by the exchange to parties to inform them of the confirmed or declined cancellation

6.3.1 Submitting a Trade Capture Report to the Exchange to initiate a trade

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

Note: a crossing Trade Capture Report does not require confirmation.

Table 26 – Trade Capture Report (inbound to the Exchange) to initiate a trade

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
StandardHeader		Y	MsgType = AE SenderSubID (50) is required (executing trader)	
571	TradeReportID	Y	Unique client generated identifier for this Trade Capture Report message	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 For REPO/TTV/SBB this should be the underlying Equity or Debt security.	
Component block <FinancingDetails>		N	Insert here the set of "FinancingDetails" (symbology) fields Required for a REPO/TTV/SBB trade.	

TAG	FIELDNAME		REQ'D	COMMENTS	FORMAT
Component block <YieldData>			N	Yield percentage – Only Yield (236) should be set in this block. If Yield (236) is specified, LastPx (31) will be ignored.	
Component block <Stipulations>			N	Insert here the set of “Stipulations” fields. Used for Fixed Income REPO/TTV/SBB trades only: <ul style="list-style-type: none">• ‘REPOPRICE’ for REPO buy/sell (required)• ‘CLEANPRICE’ for TTV/SBB buy/sell (required)	
31	LastPx		Y	Trade Price. For REPO/TTV/SBB trades, this will be the repurchase rate. For products traded by clean price this is the clean price. For products traded by dirty price this is the dirty price.	Price
32	LastQty		Y	Trade Quantity, required for all regular and repurchase trades.	Qty
60	TransactTime		N	Time the transaction represented by this Trade Capture Report occurred	UTCTime Stamp
64	SettlDate		N	Specific date of trade settlement (Settlement Date) in YYYYMMDD format	LocalMkt Date
Start of Component block, expanded in line < TrdCapRptSideGrp >					
552	NoSides		Y	Number of sides in this message 1 = For a two party capture report. 2 = For a crossing capture report.	Int
→	54	Side	Y	Side of trade. For a crossing capture report the first Side in this block will be treated as the initiator.	Char
→	Component block <Parties>		N	Insert here the set of “Parties” (firm identification) fields. Valid values: 3 = ClientID (required; must begin with the OTC agreement time in the format HHMMSS, followed by optional free text of max 26 characters) 10 = Settlement Location (Depository, required for Fixed Income) 19 = Sponsoring Firm (optional) 37 = Contra trader (optional; required for two party trade capture report to specify the counterparty)	

TAG	FIELDNAME		REQ'D	COMMENTS	FORMAT
→	1	Account	Y/N	Specifies Investor Account.	String
→	921	StartCash	Y/N	Specifies the total value of the trade. Used for REPO/TTV/SBB trades only: <ul style="list-style-type: none"> Equity & Fixed Income REPO buy/sell (optional, can only be lowered from calculated total) 	Amt
End of Component block, expanded in line < TrdCapRptSideGrp >					
StandardTrailer			Y		

6.3.2 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 has been initiated, confirmed or declined.
- A trade cancellation request has been initiated, confirmed or declined.

Table 27 – Trade Capture Report (outbound from the Exchange)

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
StandardHeader		Y	MsgType = AE TargetSubID (57) identifies the recipient of the report (the initiator or counterparty)	
571	TradeReportID	Y	Unique exchange generated identifier for this Trade Capture Report	String
17	ExecID	N	Exchanged assigned Execution ID	String
487	TradeReportTransType	N	Identifies Trade Report message transaction type. See section 6.2 for more information on the usage of this field. 0 = New 1 = Cancel 2 = Replace	Int

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
856	TradeReportType	N	Type of Trade Report. See section 6.2 for more information on the usage of this field. 0 = Submit 1 = Alleged (confirmed) 3 = Decline 6 = Trade Report Cancel (completed) 8 = Defaulted (timed out) 10 = Pended (trade awaiting approval) 14 = Alleged Trade Report Cancel 98 = Decline Trade Report Cancel 99 = Pending Trade Report Cancel	Int
828	TrdType	N	Type of trade. 22 = Privately Negotiated Trade Only set for a Block Trade.	Int
150	ExecType	N	Type of Execution being reported. F = Trade H = Trade Cancel	Char
572	TradeReportRefID	N	The client TradeReportID being referenced.	String
573	MatchStatus	N	The status of this trade with respect to matching. Set to '0' when matched.	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	
Component block <UnderlyingInstrument>		N	Insert here the set of "UnderlyingInstrument" (symbology) fields	
Component block <FinancingDetails>		N	Insert here the set of "FinancingDetails" (symbology) fields	
Component block <YieldData>		N	Insert here the set of "YieldData" fields	
Component block <Stipulations>		N	Insert here the set of "Stipulations" fields. Used to disseminate the Repo Price for a deal (or the Clean Price for Fixed Income TTV/SBB).	
15	Currency	N	Currency of the trade.	Currency

TAG	FIELDNAME		REQ'D	COMMENTS	FORMAT
31	LastPx		Y	Trade Price. For REPO/TTV/SBB trades, this will be the repurchase rate. For yield based products that allow entry by price or yield, this is the clean price. For products traded by dirty price this is the dirty price.	Price
32	LastQty		Y	Trade Quantity.	Qty
20102	DirtyPrice		N	The dirty price for the deal (only for Fixed Income products traded by clean price). This is a custom tag.	Price
60	TransactTime		N	Time the transaction represented by this Trade Capture Report occurred	UTCTime Stamp
62	ValidUntilTime		N	Indicates expiration time of deal (always expressed in UTC (Universal Time Coordinated, also known as "GMT"))	UTCTime Stamp
63	SettlType		N	Indicates the order settlement period.	String
64	SettlDate		N	Specific date of trade settlement (Settlement Date) in YYYYMMDD format	LocalMkt Date
75	TradeDate		N	Used for report trade date	LocalMkt Date
381	GrossTradeAmt		N	Total amount traded expressed in units of currency. E.g. for equities equal to Price*LastQty	Amt
880	TrdMatchID		N	Identifier assigned by the trading system for a trade	String
829	TrdSubType		N	If specified, this indicates whether the trader is on the buy or sell side of a fill.	Int
824	TradeLegRefID		N	Reference to the TrdMatchID of an associated REPO/TTV/SBB trade at start and return date.	String
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

TAG	FIELDNAME		REQ'D	COMMENTS	FORMAT
→	Component block <Parties>		N	Insert here the set of "Parties" (firm identification) fields Valid values for PartyRole are listed in Appendix B.2 <i>Parties Component Block</i> .	
→	1	Account	Y/N	Specifies Investor Account.	String
→	921	StartCash	N	For REPO/TTV/SBB trades only, specifies the total value of the trade. For crossing capture reports this field will only be set on the first side in the message.	Amt
→	922	EndCash	N	For REPO/TTV/SBB trades only, specifies the return value of the trade. For crossing capture reports this field will only be set on the first side in the message.	Amt
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 to the Exchange for a trade

Counterparty should send a Trade Capture Report (AE) to the exchange to either confirm or decline a pending privately negotiated trade. Fields not listed may be ignored.

Table 28 – Trade Capture Report (inbound to the Exchange) to confirm/decline a trade

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
StandardHeader		Y	MsgType = AE SenderSubID (50) is required (actioning trader)	
571	TradeReportID	Y	Unique client generated identifier for this Trade Capture Report message	String
487	TradeReportTransType	N	Identifies Trade Report message transaction type 2 = Replace	Int

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
572	TradeReportRefID	Y	<p>The TradeReportID or TradeReportRefID that was originally generated by the exchange that is being referenced for some action, such as confirmation or cancellation. Usage as follows:</p> <ul style="list-style-type: none"> For a respondent confirming or declining a report this should be set with the TradeReportID (571) they received in the original Trade Capture Report (AE) message from the exchange. For an initiator withdrawing a report, this field should be set with the TradeReportRefID (572) they received in the original Trade Capture Report Ack (AR) message. 	String (20)
856	TradeReportType	Y*	<p>Type of Trade Report. Required when confirming or withdrawing a trade.</p> <p>2 = Accept</p> <p>3 = Decline</p>	Int
Component block <Instrument>		Y	Insert here the set of "Instrument" (symbology) fields	
Component block <YieldData>		N	Insert here the set of "YieldData" fields	
31	LastPx	Y	Trade Price. Must match existing values when confirming a trade. Ignored by the exchange when declining a trade.	Price
32	LastQty	Y	Trade Quantity. Must match the existing values when confirming a trade. Ignored by the exchange when withdrawing a trade.	Qty
60	TransactTime	N	Time the transaction represented by this Trade Capture Report occurred	UTCTime Stamp
Start of Component block, expanded in line < TrdCapRptSideGrp >				
552	NoSides	Y	Number of sides. Should be 1.	Int
→	54	Side	Y	Side of order
→	Component block <Parties>		N	<p>Insert here the set of "Parties" (firm identification) fields. Valid values:</p> <p>3 = ClientID (required when TradeReportType=Accept; must begin with the OTC agreement time in the format HHMMSS, followed by optional free text of max 26 characters)</p> <p>10 = Settlement Location (Depository, required for Fixed Income)</p> <p>19 = Sponsoring Firm (optional)</p>

TAG	FIELDNAME		REQ'D	COMMENTS	FORMAT
→	1	Account	Y/N	Specifies Investor Account. Ignored by the exchange when withdrawing a trade.	String
End of Component block, expanded in line < TrdCapRptSideGrp >					
StandardTrailer			Y		

6.3.4 Submitting a Trade Capture Report to the Exchange to request a trade cancellation

The Initiator should send a Trade Capture Report (AE) to the exchange to request a trade cancellation.

Note: a crossing Trade Capture Report does not require confirmation.

Table 29 – Trade Capture Report (inbound to the Exchange) to request a trade cancellation

TAG	FIELDNAME		REQ'D	COMMENTS	FORMAT
StandardHeader			Y	MsgType = AE SenderSubID (50) is required (requesting trader)	
571	TradeReportID		Y	Unique client generated identifier for this Trade Capture Report message	String
487	TradeReportTransType		N	Identifies Trade Report message transaction type 0 = New	Int
856	TradeReportType		N	Type of Trade Report 96 = Submit Trade Report Cancel	Int
880	TrdMatchID		N	Identifier originally assigned by the trading system to the trade that is to be cancelled.	String
Component block <Instrument>			Y	Insert here the set of "Instrument" (symbology) fields Ignored by the exchange.	
31	LastPx		Y	Trade Price. Ignored by the exchange.	Price
32	LastQty		Y	Trade Quantity. Ignored by the exchange.	Qty
60	TransactTime		N	Time the transaction represented by this Trade Capture Report occurred	UTCTime Stamp
Start of Component block, expanded in line < TrdCapRptSideGrp >					
552	NoSides		Y	Number of sides. Should be 1.	Int
→	54	Side	Y	Side of trade. Ignored by the exchange.	Char

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

6.3.5 Confirm or withdraw/decline a Trade Capture Report to the Exchange for a trade cancellation

Counterparty should send a Trade Capture Report (AE) to the exchange to either confirm or decline a pending trade cancellation request. Fields not listed may be ignored.

Table 30 – Trade Capture Report (inbound to the Exchange) to confirm/decline a trade cancellation

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
StandardHeader		Y	MsgType = AE SenderSubID (50) is required (actioning trader)	
571	TradeReportID	Y	Unique client generated identifier for this Trade Capture Report message	String
487	TradeReportTransType	N	Identifies Trade Report message transaction type 2 = Replace	Int
856	TradeReportType	Y*	Type of Trade Report. Required when confirming or declining a trade cancellation. 97 = Accept Trade Report Cancel 98 = Decline Trade Report Cancel	Int
880	TrdMatchID	N	Identifier originally assigned by the trading system to the trade that is to be cancelled. Must match the original request.	String
Component block <Instrument>		Y	Insert here the set of "Instrument" (symbology) fields Ignored by the exchange.	
31	LastPx	Y	Trade Price. Ignored by the exchange.	Price
32	LastQty	Y	Trade Quantity. Ignored by the exchange.	Qty
60	TransactTime	N	Time the transaction represented by this Trade Capture Report occurred	UTCTime Stamp
Start of Component block, expanded in line < TrdCapRptSideGrp >				
552	NoSides	Y	Number of sides. Should be 1.	Int
→	54	Side	Y Side of order Ignored by the exchange.	Char

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

6.4 Trade Capture Report Ack (AR)

The Trade Capture Report Ack message can be:

- Sent by the exchange to acknowledge trade capture report messages received from initiators
- Sent by the exchange to reject a trade capture report message received from a counterparty

Table 31 – Trade Capture Report Ack

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
StandardHeader		Y	MsgType = AR TargetSubID (57) identifies the trader	
571	TradeReportID	Y	Echoed back unique identifier for the inbound Trade Capture Report.	String
487	TradeReportTransType	N	Echoed back Trade Report message transaction type	Int
856	TradeReportType	N	Echoed back type of Trade Report	Int
939	TrdRptStatus	N	Trade Report Status 0 = Accepted 1 = Rejected	Int
17	ExecID	N	Exchanged assigned Execution ID	String
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. Valid values: 99 – Other	int
572	TradeReportRefID	N	The TradeReportID being referenced for some action, such as confirmation or cancellation.	String
58	Text	N	If TradeReportRejectReason is set, text of reason	String
StandardTrailer		Y		

7 Market Data

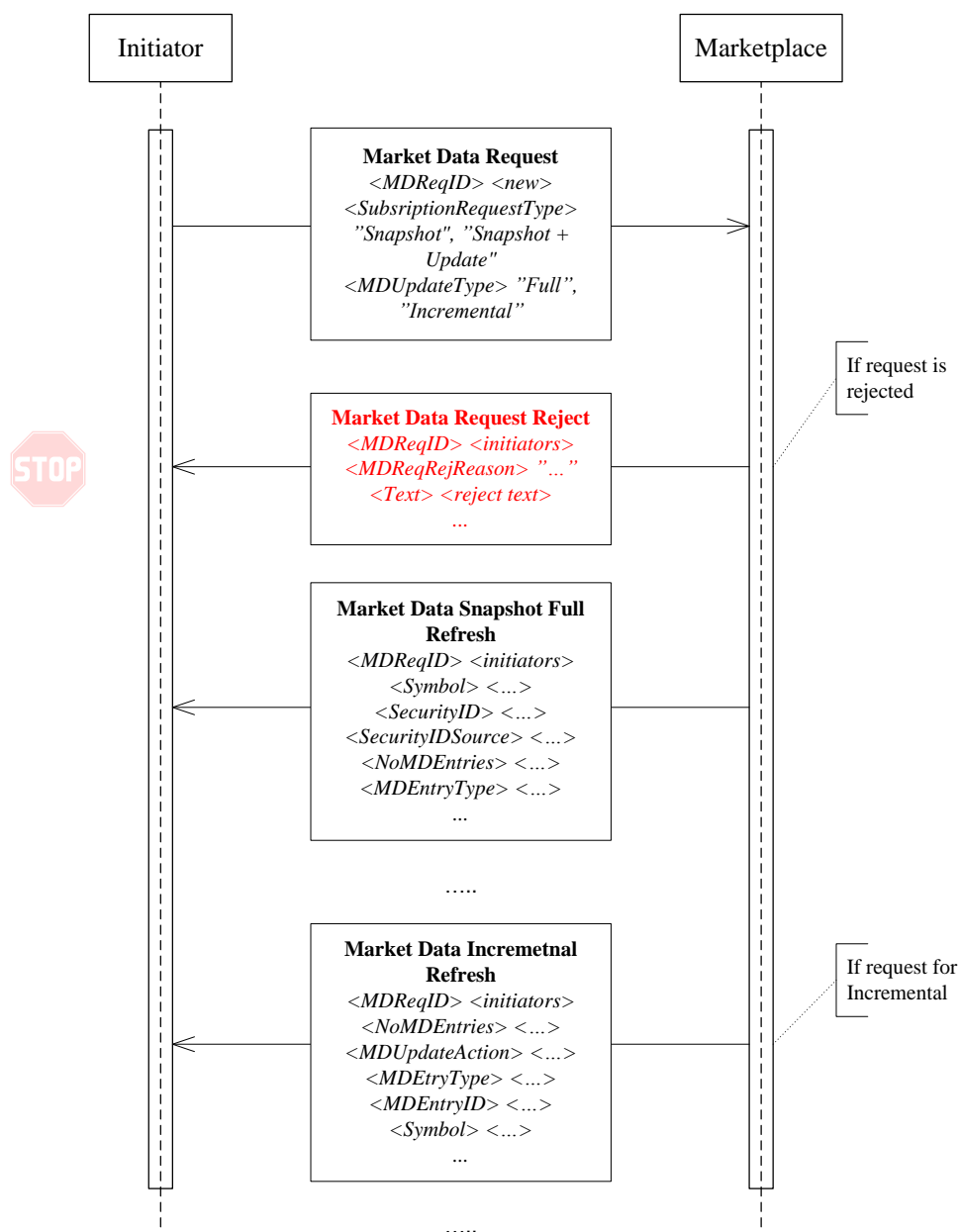
The market category consists of the following messages:

- Market Data Request
- Market Data Request Reject
- Market Data Snapshot/Full Refresh
- Market Data Incremental Refresh
- Trading Session Status Request
- Trading Session Status
- Security List Request
- Security List
- Security List Update Report
- Security Status Request
- Security Status
- News

7.1 Workflows

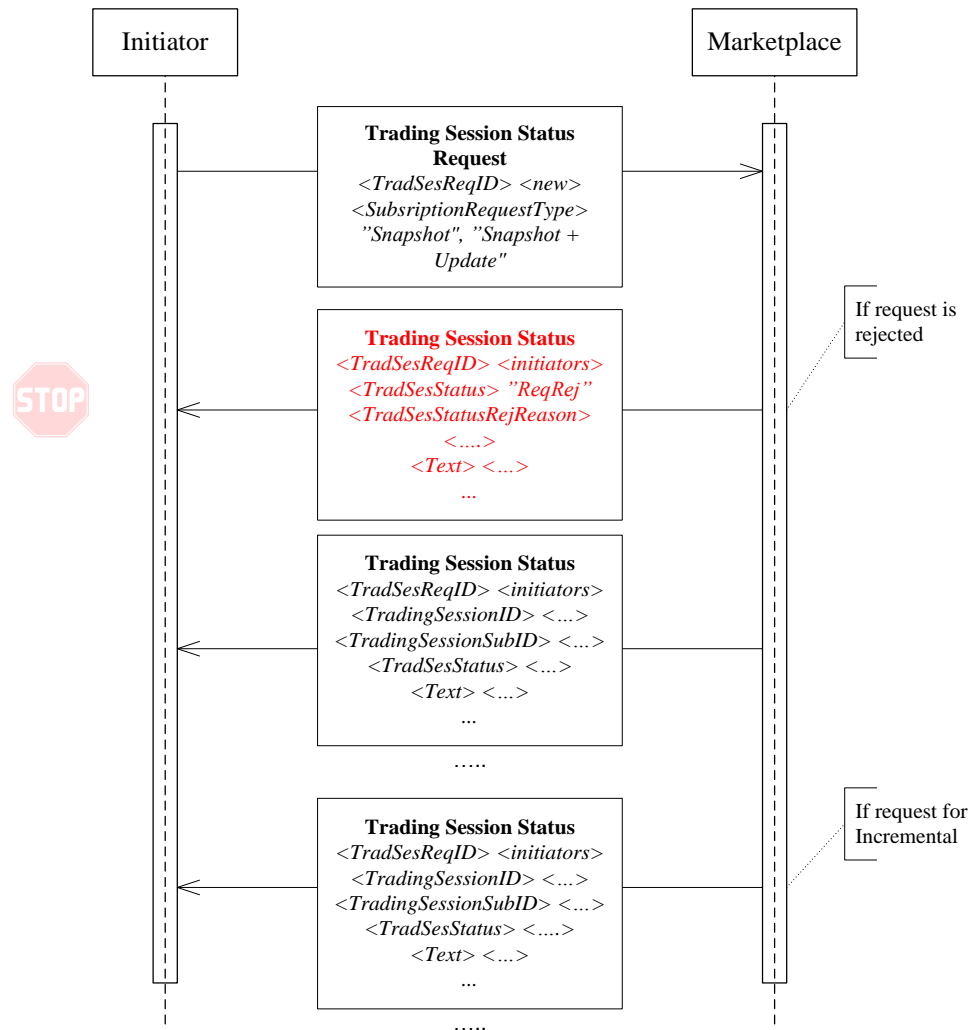
7.1.1 Subscribing to and Receiving Market Data

Figure 13 – Subscribing and Receiving Market Data Workflow



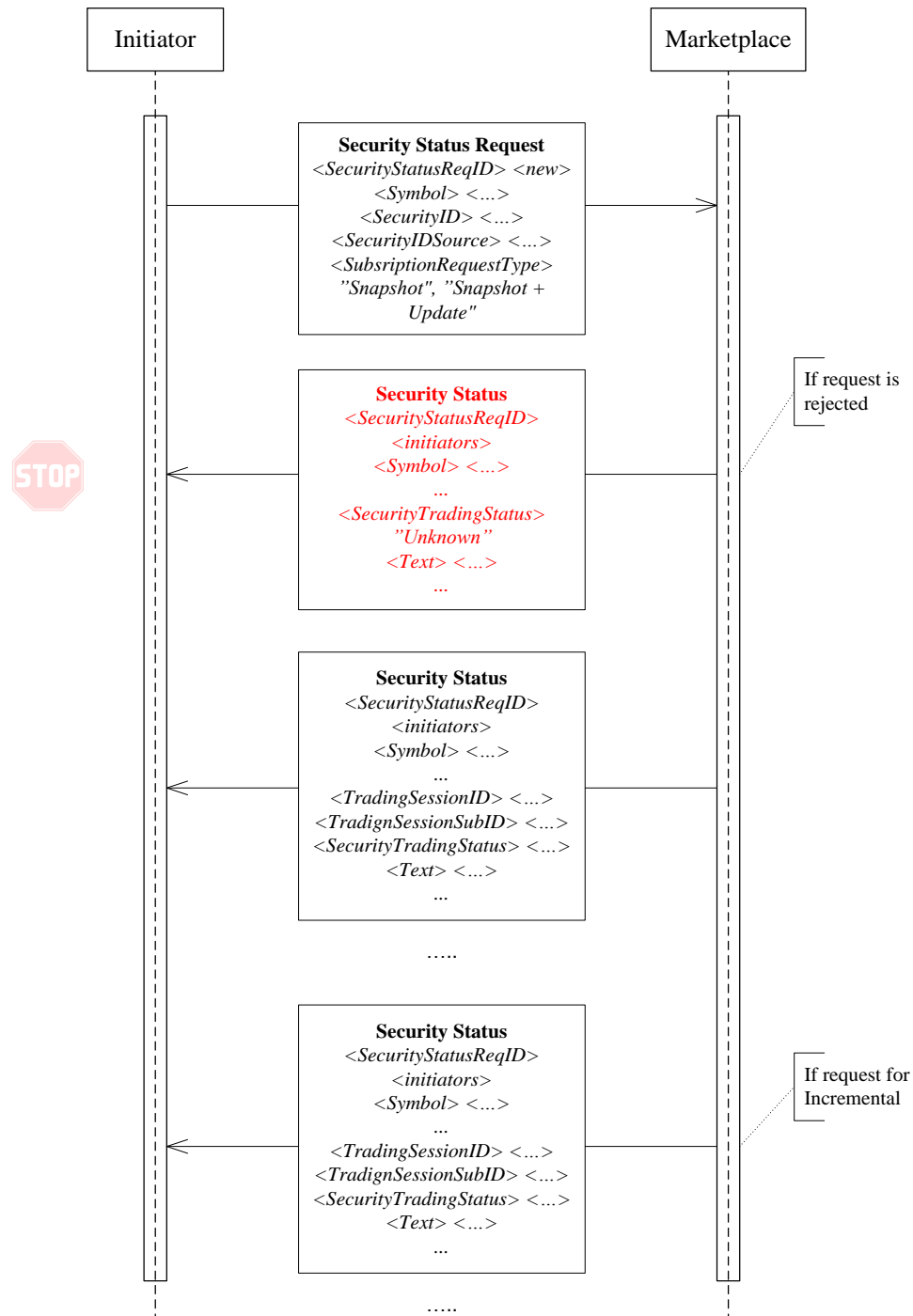
7.1.2 Subscribing to and Receiving Trading Session Status

Figure 14 – Subscribing and Receiving Trading Session Status Workflow



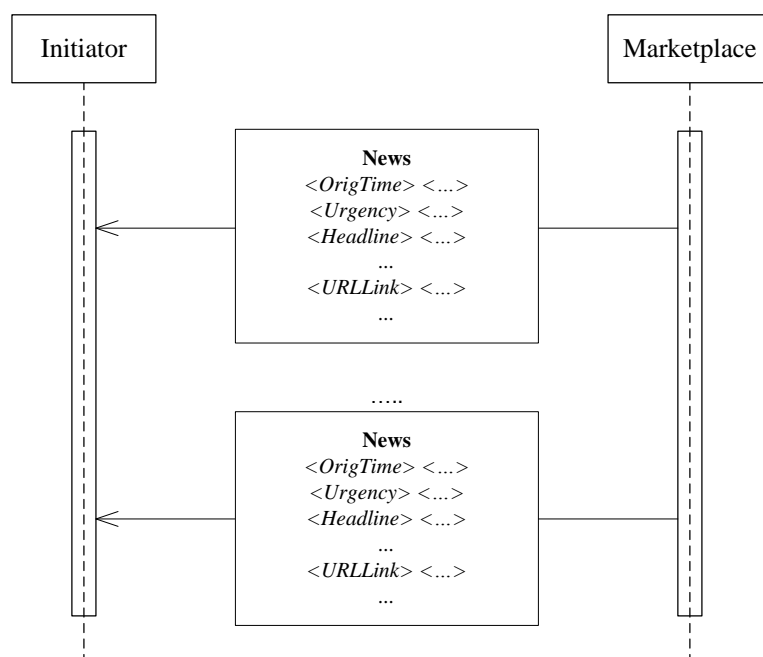
7.1.3 Subscribing to and Receiving Security Status

Figure 15 – Subscribing to and Receiving Security Status Workflow



7.1.4 Receiving News

Figure 16 – Receiving News Workflow



7.2 General Information for Market Data Requests and Responses

7.2.1 Market Data Requests Based on Category

Specific market data requests can be specified for the following major categories using the MDEntryType (269) tag:

- '0' - Order information – requests all order related information in market by order and market by price messages (e.g. bids, offers, etc.).
- '2' - Trade information – returns all market trade information and statistics.
- '3' - Index information – returns all index related information.
- 'a' - Security statistics (NASDAQ specific) – returns security specific market statistics.

7.2.2 Market Data Requests

Specific securities can be specified for requests with a configurable limit on the maximum number of securities. The default limit is configured to five (5) securities. It should be also noted that the greater the number of securities specified the greater bandwidth and latency will be impacted.

7.2.3 Wildcard Security Specification

Wildcard market data requests are supported for REPO/TTV/SBB type boards only.

A market data request can be sent with either:

- 762 (SecuritySubType) set to the desired REPO/TTV/SBB type board identifier, and 55 (Symbol) set to the wildcard '*', which will apply to all symbols from the specified board (maximum 1 board per message, maximum 1 entry type per message, more information below).

- 55 (Symbol) set to a valid symbol, and 762 (SecuritySubType) set to a valid board, which will apply to the symbol on that board only (maximum 5 symbols per message, all should have the same board).
- 55 (Symbol) set to a valid symbol only, which will return the symbol on its main board only (if one exists; maximum 5 symbols per message).

If a valid combination of SecuritySubType and/or Symbol is not found a reject is returned.

If the wildcard Symbol is used ('*') in a market data request, the following rules apply:

- The InstrmtMDReqGrp can contain only the wildcard for a single board (i.e. NoRelatedSym must be set to 1)
- The MDReqGrp can contain only a single entry type (i.e. NoMDEntryTypes must be set to 1)

7.2.4 Response Enumerations Extensions

There are additional NASDAQ specific enumerations which have been added for the MDEntryType (269) tag which apply only to market data request responses (not requests) and are used to identify the type of information being returned. Refer to Appendix C and the tables below for details on enumerated values for MDEntryType (269) for market data responses.

7.2.5 Order Information Request and Responses

The following table applies to Market Data requests for Order Information (MDEntryType = '0') and indicates which tags are used to return the requested information.

Table 32 – Order Information Response Tags

Order Information (Request MDEntryType = '0')			
Returned MDEntryType (269)	Returned Tags	Tag Names	Description
Market By Order			
0,1	236	Yield	Order Yield (Fixed Income only)
0,1	269	MDEntryType	Bid/Offer
0,1	270	MDEntryPx	Order Price. For REPO/TTV/SBB orderbooks this is the repurchase rate. For orderbooks traded by clean price this is the clean price. For orderbooks traded by dirty price this is the dirty price.
0,1	271	MDEntrySize	Order Quantity
0,1	272 & 273	MDEntryDate, MDEntryTime	Date and time the order was last updated
0,1	278	MDEntryID	Order identifier
0,1	290	MDEntryPositionNo	Display position of bid/offer
0,1	520	ContAmtValue	Total Value (optional, REPO/TTV/SBB orderbooks only)

Order Information (Request MDEntryType = '0')			
0,1	1025	FirstPx	For FI REPO orderbooks this is the Repo Price. For FI TTV/SBB orderbooks this is the Clean Price.
0,1	20102	DirtyPrice	Dirty Price (optional, only for Fixed Income orderbooks traded by Clean Price)
0,1	574	MatchType	Indicates a pre-arranged order. Valid values: 99 – Pre-arranged
Market By Price			
0,1	236	Yield	Yield (Fixed Income securities only)
0,1	269	MDEntryType	Bid/Offer
0,1	270	MDEntryPx	Bid/Offer Price. For REPO/TTV/SBB orderbooks this is the repurchase rate. For orderbooks traded by clean price this is the clean price. For orderbooks traded by dirty price this is the dirty price.
0,1	271	MDEntrySize	Total visible quantity of all orders at this price.
0,1	278	MDEntryID	Unique identifier for the price level
0,1	290	MDEntryPositionNo	Display position of bid/offer
0,1	346	NumberOfOrders	Number of orders in the market
0,1	520	ContAmtValue	Total Value (optional, REPO/TTV/SBB orderbooks only)
0,1	20102	DirtyPrice	Dirty Price (optional, only for Fixed Income orderbooks traded by Clean Price)

7.2.6 Trade Information Request and Responses

The following table applies to Market Data requests for Trade Information (MDEntryType = '2') and indicates which tags are used to return the requested information.

By default, only the last 10 trades are reported in a Market Data Snapshot/Full Refresh of trades for a security.

Table 33 – Trade Information Returned Tags

Trade Information (Request MDEntryType = '2')			
Returned MDEntryType (269)	Returned Tags	Tag Names	Description
2	31	LastPx	Value of this trade (price*quantity) For REPO/TTV/SBB trades, this will be the total repurchase value.

Trade Information (Request MDEntryType = '2')			
2	58	Text	Trade status. The possible values are as follows: 'Matched' (default) 'Unconfirmed Cancel Buy' (i.e. waiting for buyer to confirm) 'Unconfirmed Cancel Sell' (i.e. waiting for seller to confirm) 'Unapproved Cancel' 'Withdrawn' (i.e. canceled)
2	64	SettlDate	Settlement date for the trade
2	236	Yield	Yield (Fixed Income securities only)
2	270	MDEntryPx	Trade price. For REPO/TTV/SBB orderbooks this is the repurchase rate. For orderbooks traded by clean price this is the clean price. For orderbooks traded by dirty price this is the dirty price.
2	271	MDEntrySize	Trade quantity
2	272 & 273	MDEntryDate, MDEntryTime	Date and time of the trade
2	278	MDEntryID	Trade identifier
2	279	MDUpdateAction	In IncrementalRefresh MDUpdateAction(279) indicates whether a trade is being matched, cancelled or amended (if applicable). This tag is not provided on Snapshot/FullRefresh as 'cancelled' marketTrades are not disseminated. The possible values are as follows: 0 - New (for trade match) 1 - Change (for trade amend) 2 - Delete (for trade cancel).
2	828	TrdType	Indicates the type of trade (optional, not specified for a regular trade)
2	1025	FirstPx	For FI REPO orderbooks this is the Repo Price. For FI TTV/SBB orderbooks this is the Clean Price.
2	2446	AggressorSide	Which side was the aggressor in the trade (if any)
2	20102	DirtyPrice	Dirty Price (optional, only for Fixed Income orderbooks traded by Clean Price)
2	574	MatchType	Indicates a pre-arranged trade (i.e. both sides were pre-arranged). Valid values: 99 - Pre-arranged

7.2.7 Index Information Request and Responses

The following table applies to Market Data requests for Index Information (MDEntryType = '3') and indicates which tags are used to return the requested information.

Table 34 – Index Information Returned Tags

Index Information (Request MDEntryType = '3')			
Returned MDEntryType (269)	Returned Tags	Tag Names	Description
3	31	LastPx	Last index value
4	270	MDEntryPx	Day opening Index value
5	270	MDEntryPx	Day closing Index value
7	270	MDEntryPx	Day high index value
8	270	MDEntryPx	Day low index value
t	270	MDEntryPx	Delta - Change from reference index value
x	270	MDEntryPx	Reference Index value
v	270	MDEntryPx	Total value of security traded today
w	1020	TradeVolume	Total number of shares traded today
y	332	HighPx	52 week high
y	333	LowPx	52 Week low

*Only returned if Yield values are applicable to fixed income securities/index.

7.2.8 Security Statistics Requests and Responses

The following table applies to Market Data requests for Security Statistics Information (MDEntryType = 'a') and indicates which tags are used to return the requested information.

Table 35 – Security Statistics Returned Tags

Security Statistics (Request MDEntryType = 'a')			
Returned MDEntryType (269)	Returned Tags	Tag Names	Description
4	270	MDEntryPx	Opening Price
5	270	MDEntryPx	Closing Price
6	270	MDEntryPx	Settlement Price
7	270	MDEntryPx	Day high price
8	270	MDEntryPx	Day low price
9	270	MDEntryPx	Weighted Average Price
a	31	LastPx	Last traded price
a	64	SettlDate	Specific date of trade settlement

Security Statistics (Request MDEntryType = 'a')			
a	235, 236	YieldType, Yield	Trading session yields* <u>Valid values for YieldType:</u> 'High', 'Low', 'Last', 'Change', 'WAVg' (Weighted Average), 'OPENAVG' (Open), 'CLOSE', 'Ref' (Reference)
a	271	MDEntrySize	Last traded quantity
a	274	TickDirection	Movement indicator
a	326	SecurityTradingStatus	Status of security
a	451	NetChgPrevDay	Change from previous day (for Market Data Incremental Refresh)
C	271	MDEntrySize	Open Interest
i	270	MDEntryPx	Off-market last traded price
i	271	MDEntrySize	Off-market last traded quantity
k	1020	TradeVolume	Off-market total number of shares traded
l (lower case L)	270	MDEntryPx	Off-market total value of security traded
m	811	PriceDelta	% change from Previous closing price
M	270	MDEntryPx	Previous Settlement Price
n	271	MDEntrySize	Total number of trades today
o	271	MDEntrySize	Opening Quantity
q	270	MDEntryPx	Total value of security traded during regular hours today (i.e. excludes after-market session)
r	1020	TradeVolume	Total number of shares traded during regular hours today (i.e. excludes after-market session)
s	270	MDEntryPx	Unadjusted Previous closing price.
t	270	MDEntryPx	Delta - Change from reference price
u	270	MDEntryPx	Previous closing price
v	270	MDEntryPx	Total value of security traded today
w	1020	TradeVolume	Total number of shares traded today
x	270	MDEntryPx	Reference Price
y	332	HighPx	52 week high price
y	333	LowPx	52 Week low price
z	270	MDEntryPx	Indicative Opening Price
z	271	MDEntrySize	Indicative Opening Quantity

*Only returned if Yield values are applicable to fixed income securities.

7.2.9 The order of Market Data Responses

The order in which the Market Data Responses are disseminated to the subscriber is not guaranteed to be in the same order as the requests.

7.3 Market Data Request (V)

A successful Market Data Request returns one or more Market Data messages containing one or more Market Data Entries. Each Market Data Entry is a Bid, an Offer, a Trade associated with a security, the opening, closing, or settlement price of a security, the buyer or seller, the value of an index, the trading session high price, low price, or VWAP, or the trade volume or open interest in a security. Market Data Entries usually have a price and a quantity associated with them. The market data request message format is as follows.

Table 36 – Market Data Request

TAG	FIELDNAME		REQ'D	COMMENTS	FORMAT
StandardHeader			Y	MsgType = V	
262	MDReqID		Y	Must be unique, or the ID of previous Market Data Request to disable if SubscriptionRequestType = Disable previous Snapshot + Updates Request (2).	String
263	SubscriptionRequestType		Y	SubscriptionRequestType indicates to the other party what type of response is expected. A snapshot request only asks for current information. A subscribe request asks for updates as the status changes. Unsubscribe will cancel any future update messages from the counter party.	Char
264	MarketDepth		Y	Depth of market for Book Snapshot / Incremental updates. The actual depth returned will be determined by the exchange. Ignored if MDEntryType is not 0.	Int
265	MDUpdateType		N	Required if SubscriptionRequestType = Snapshot + Updates (1). Specifies the type of Market Data update.	Int
266	AggregatedBook		N	Specifies whether or not book entries should be aggregated. 'Y' = Market by Price (MBP), 'N' = Market by Order (MBO) – default.	Boolean
Start of Component block, expanded in line < MDReqGrp >					
267	NoMDEntryTypes		Y	Number of MDEntryType fields requested.	NumInGrp
→	269	MDEntryType	Y	Must be first field in repeating group. This is a list of all the category of Market Data Entries that the firm requesting the Market Data is interested in receiving.	Char
End of Component block, expanded in line < MDReqGrp >					
Start of Component block, expanded in line < InstrmtMDReqGrp >					

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
146	NoRelatedSym	Y	Number of securities requested. Must be greater than 0. Maximum is 5. All securities should have the same SecuritySubType.	Int
→	Component block <Instrument>	Y	Insert here the set of "Instrument" (symbology) fields.	
End of Component block, expanded in line < InstrmtMDReqGrp >				
StandardTrailer		Y		

7.4 Market Data Request Reject (Y)

The Market Data Request Reject is used when the Exchange cannot honour the Market Data Request, due to business or technical reasons. The Exchange may choose to limit various parameters, such as the size of requests, whether just the top of book or the entire book may be displayed, and whether Full or Incremental updates must be used.

The market data request reject message format is as follows.

Table 37 – Market Data Request Reject

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
StandardHeader		Y	MsgType = Y	
262	MDReqID	Y	Must refer to the MDReqID of the request.	String
281	MDReqRejReason	N	Reason for the rejection of a Market Data request.	Char
58	Text	N	Free format text string	String
StandardTrailer		Y		

7.5 Market Data Snapshot/Full Refresh (W)

The Market Data messages are used as the response to a Market Data Request message. In all cases, one Market Data message refers only to one Market Data Request. Market Data messages sent as the result of a Market Data Request message will specify the appropriate MDReqID.

There are two types of Market Data Refresh messages, Snapshot/Full and Incremental.

The Market Data message format used for a Snapshot, or a Snapshot + Updates where MDUpdateType = Full Refresh (0) is as follows:

- After a Market Data Request, when a Bid or Offer is added, changed, or deleted, every update to a Market Data Entry results in a new Market Data Snapshot message that contains the entirety of the data requested for that instrument, not just the changed Market Data Entry. In other words, both sides of the market, or just one side in the case of a request of only bids or offers, for the depth requested, must be sent in one FIX Market Data Snapshot message.
- A Market Data Snapshot message may contain several trades, an index value, opening, closing, settlement, high, low, and/or VWAP price for one instrument, as well as the traded volume and open interest, but only for one instrument per message.
- Messages containing bids and/or offers cannot contain trades, index value, opening, closing, settlement, high, low, and/or VWAP prices, trade volume, or open interest.

- Messages containing Price Depth or Order Depth information for instruments traded in yield, the Yield component block will only contain Yield(236). YieldType(235) will not be sent in messages containing Price Depth or Order Depth information.

Table 38 – Market Data Snapshot/Full Refresh

TAG	FIELDNAME		REQ'D	COMMENTS	FORMAT
StandardHeader			Y	MsgType = W	
911	TotNumReports		N	Total number of reports returned in response to a request	Int
1021	MDBookType		N	Describes the type of book for which the feed is intended. Used when multiple feeds are provided over the same connection	Int
264	MarketDepth		N	Can be used to define the current depth of the book. The actual depth returned will be determined by the exchange.	Int
75	TradeDate		N	Used to specify the trading date for which a set of market data applies	LocalMktDate
262	MDReqID		Y/N	Conditionally required if this message is in response to a Market Data Request.	Sting
Component block <Instrument>			Y	Insert here the set of "Instrument" (symbology) fields.	
451	NetChgPrevDay		N	Net change from previous day's closing price.	PriceOffset
Start of Component block, expanded in line < MDFullGrp >					
268	NoMDEntries		Y	Number of entries following.	NuminGroup
→	269	MDEntryType	Y	Must be first field in this repeating group. This is a list of the type of Market Data Entries that the firm is receiving in this market data response message.	Int
→	278	MDEntryID	Y/N	Unique Market Data Entry identifier. Conditionally required when maintaining an order-depth book, that is, when AggregatedBook (266) is "N". This allows subsequent Incremental changes to be applied using MDEntryID.	String
→	270	MDEntryPx	Y/N	Price of the Market Data Entry. Conditionally required depending on MDEntryType. For REPO/TTV/SBB orderbooks this is the repurchase rate.	Price

TAG	FIELDNAME		REQ'D	COMMENTS	FORMAT
→	Component block <YieldData>		N	Insert here the set of YieldData (yield-related) fields	
→	271	MDEntrySize	Y/N	Quantity or volume of the market data entry. Conditionally required if MDEntryType = Trade(2).	Qty
→	272	MDEntryDate	N	Date of Market Data Entry.	UTCDateOnly
→	273	MDEntryTime	N	Time of Market Data Entry.	UTCTimeOnly
→	274	TickDirection	N	Direction of the "tick".	Char
→	520	ContAmtValue	N	Value. Used for REPO/TTV/SBB order.	Amt
→	326	SecurityTradingStatus	N	Identifies the trading status applicable to the transaction.	Int
→	290	MDEntryPositionNo	N	Display position of a bid or offer, numbered from most competitive to least competitive, per market side beginning with 1.	Int
→	346	NumberOfOrders	N	In an Aggregated Book, used to show how many individual orders make up an MDEntry	Int
→	1025	FirstPx	N	For FI REPO orderbooks this is the Repo Price. For FI TTV/SBB orderbooks this is the Clean Price. This is a V5.0 tag value.	Price
→	332	HighPx	N	Highest price paid for the security in the trading session. If MDEntryType = 'y' then this value is the 52 week high price.	Price
→	333	LowPx	N	Lowest price paid for the security in the trading session. If MDEntryType = 'y' then this value is the 52 week low price.	Price
→	31	LastPx	N	Last traded Price for the security. If MDEntryType = '2' this is the trade value. For REPO/TTV/SBB trades, this will be the total repurchase value. (Note: The LastPx field first appears in market data messages in FIX V5.0 SP2 but is included in this specification for completeness).	Price
→	20102	DirtyPrice	N	Used for Fixed Income orderbooks only to indicate the dirty price. This is a custom tag.	Price

TAG	FIELDNAME		REQ'D	COMMENTS	FORMAT
→	1020	TradeVolume	N	Used to report trade volume in association with trade, bid or ask rather than a separate entity. This is a V5.0 tag value.	Int
→	64	SettlDate	N	Specific date of trade settlement (Settlement Date) in YYYYMMDD format.	LocalMktDate
→	811	PriceDelta	N	Price change as a percentage	Float
→	2446	AggressorSide	N	Indicates which side of a trade was the aggressor.	Int
→	828	TrdType	N	Indicates the type of trade (optional, not specified for a regular trade).	Int
→	574	MatchType	N	Indicates a pre-arranged order or trade. Valid values: 99 = Pre-arranged	Int
→	58	Text	N	If MDEntryType = '2' this may provide additional Trade Status information.	String
End of Component block, expanded in line < MDFullGrp >					
StandardTrailer			Y		

7.6 Market Data Incremental Refresh (X)

The second Market Data message format is used for incremental updates. With the incremental message the Exchange has the responsibility to provide all Market Data entries needed by the client user in order to build an order book copy, populate a Trade Ticker, etc.

The Market Data Incremental Refresh message may contain any combination of new, changed, or deleted Market Data Entries, for one or more instruments, with any combination of trades, imbalances, quotes, index values, open, close, settlement, high, low, and VWAP prices, trade volume and open interest so long as the maximum FIX message size is not exceeded.

Market Data Entries may have an MDEntryID unique among all currently active Market Data Entries so they can be referenced for the purposes of deleting and changing them later. When changing a Market Data Entry, it may keep the same MDEntryID, in which case only MDEntryID would be populated, or the MDEntryID may change, in which case MDEntryID will contain the new ID and MDEntryRefID will contain the ID of the Market Data Entry being changed. An MDEntryID can be reused within a day only if it has first been deleted.

7.6.1 Maintaining the Order Book

The following instructions (MDUpdateAction) are used to maintain the order book:

- New – Used to insert a market data entry.
- Delete – Used to remove a market data entry or to remove all entries from the order book

When an order book gets empty during a trading day, a Market Data Incremental Refresh message will be sent to indicate the removal of all entries from the order book as follows:

NoMDEntries(268)=1 MDUpdateAction=2(Delete) and MDEntryType=J(EmptyBook)

- Change – Used to modify a market data entry.

Table 39 – Market Data Incremental Refresh

TAG	FIELDNAME		REQ'D	COMMENTS	FORMAT
StandardHeader			Y	MsgType = X	
262	MDReqID		Y/N	Conditionally required if this message is in response to a Market Data Request.	String
1021	MDBookType		N	Describes the type of book for which the feed is intended. Used when multiple feeds are provided over the same connection	Int
Start of Component block, expanded in line < MDIncGrp >					
268	NoMDEntries		Y	Number of entries following.	NumInGrp
→	279	MDUpdateAction	Y	Type of update action. Must be first field in this repeating group.	Char
→	264	MarketDepth	N	Can be used to define the current depth of the book. The actual depth returned will be determined by the exchange.	Int
→	269	MDEntryType	N	This indicates the type of Market Data Entries that the firm is receiving in this market data response message.	Int
→	278	MDEntryID	Y/N	If specified, must be unique among currently active entries if MDUpdateAction = New (0), must be the same as a previous MDEntryID if MDUpdateAction = Delete (2), and must be the same as a previous MDEntryID if MDUpdateAction = Change (1) and MDEntryRefID is not specified, or must be unique among currently active entries if MDUpdateAction = Change(1) and MDEntryRefID is specified. Conditionally required when maintaining an order-depth book, that is, when AggregatedBook (266) is "N". This allows subsequent Incremental changes to be applied using MDEntryID.	String

TAG	FIELDNAME		REQ'D	COMMENTS	FORMAT
→	Component block <Instrument>		N	Insert here the set of "Instrument" (symbology) fields. Either Symbol (the instrument component block) or MDEntryRefID must be specified if MDUpdateAction = New(0) for the first Market Data Entry in a message. For subsequent Market Data Entries where MDUpdateAction = New(0), the default is the instrument used in the previous Market Data Entry.	
→	270	MDEntryPx	Y/N	Price of the Market Data Entry. Conditionally required when MDUpdateAction = New(0) and MDEntryType is not Trade Volume (B).	Price
→	Component block <YieldData>		N	Contains Yield information.	
→	271	MDEntrySize	Y/N	Quantity or volume represented by the Market Data Entry. Conditionally required when MDUpdateAction = New(0) and MDEntryType Trade(2).	Qty
→	272	MDEntryDate	N	Date of Market Data Entry.	UTCDateOnly
→	273	MDEntryTime	N	Time of Market Data Entry.	UTCTimeOnly
→	274	TickDirection	N	Direction of the "tick".	Char
→	520	ContAmtValue	N	Value. Used for REPO/TTV/SBB order.	Amt
→	326	SecurityTradingStatus	N	Identifies the trading status applicable to the transaction.	Int
→	290	MDEntryPositionNo	N	Display position of a bid or offer, numbered from most competitive to least competitive, per market side, beginning with 1	Int
→	346	NumberOfOrders	N	In an Aggregated Book, used to show how many individual orders make up an MDEntry	Int
→	1025	FirstPx	N	For FI REPO orderbooks this is the Repo Price. For FI TTV/SBB orderbooks this is the Clean Price. This is a V5.0 tag value.	Price
→	332	HighPx	N	Highest price paid for the security in the trading session. If MDEntryType = 'y' then this value is the 52 week high price.	Price

TAG	FIELDNAME		REQ'D	COMMENTS	FORMAT
→	333	LowPx	N	Lowest price paid for the security in the trading session. If MDEntryType = 'y' then this value is the 52 week low price.	Price
→	31	LastPx	N	Last traded Price for the security. If MDEntryType = '2' this is the trade value. For REPO/TTV/SBB trades, this will be the total repurchase value. (Note: The LastPx field first appears in market data messages in FIX V5.0 SP2 but is included in this specification for completeness).	Price
→	20102	DirtyPrice	N	Used for Fixed Income orderbooks only to indicate the dirty price. This is a custom tag.	Price
→	451	NetChgPrevDay	N	Net change from previous day's closing price.	PriceOffset
→	1020	TradeVolume	N	Used to report trade volume in association with trade. This is a V5.0 tag value.	Int
→	64	SettlDate	N	Specific date of trade settlement (Settlement Date) in YYYYMMDD format.	LocalMktDate
→	811	PriceDelta	N	Price change as a percentage	Float
→	2446	AggressorSide	N	Indicates which side of a trade was the aggressor.	Int
→	828	TrdType	N	Indicates the type of trade (optional, not specified for a regular trade).	Int
→	574	MatchType	N	Indicates a pre-arranged order or trade. Valid values: 99 = Pre-arranged	Int
→	58	Text	N	If MDEntryType = '2' this may provide additional Trade Status information.	String
End of Component block, expanded in line < MDIncGrp >					
StandardTrailer			Y		

7.7 Trading Session Status Request (g)

The Trading Session Status Request is used to request information on the status of a market. With the move to multiple sessions occurring for a given trading party (morning and evening sessions for instance) there is a need to be able to provide information on what product is trading on what market.

The Trading Session Status Request message can be used to inquire the trading status of a trading party. The Trading Session Status message can be used to subscribe to updates to the status of a trading session by setting the RequestType field to 1.

Table 40 – Trading Session Status Request

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
StandardHeader		Y	MsgType = g (lowercase)	
335	TradSesReqID	Y	Must be unique, or the ID of previous Trading Session Status Request to disable if SubscriptionRequestType = Disable previous Snapshot+Updates Request (2).	String
263	SubscriptionRequestType	Y	Subscription type request.	Char
StandardTrailer		Y		

7.8 Trading Session Status (h)

The Trading Session Status provides information on the status of a market. For markets multiple trading sessions on multiple-markets occurring (morning and evening sessions for instance), this message is able to provide information on what products are trading on what market during what trading session. It is also used to reject a failed request.

Table 41 – Trading Session Status

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
StandardHeader		Y	MsgType = h (lowercase)	
335	TradSesReqID	N	Provided for a response to a specific Trading Session Status Request message (snapshot).	String
336	TradingSessionID	Y	Identifier for Trading Session. A trading session spans an extended period of time that can also be expressed informally in terms of the trading day. Usage is determined by market or counterparties. In the context of this message the trading session ID will map to the current trading session state (e.g. pre-open, open, closed, etc.)	String
340	TradSesStatus	Y	State of the trading session	Int
567	TradSesStatusRejReason	N	Use with TradSesStatus = "Request Rejected"	Int
341	TradSesStartTime	N	Starting time of the trading session	UTCTimeStamp
58	Text	N	Instrument ID (if one is present)	String
55	Symbol	N	The security for which trading session applies.	String

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
762	SecuritySubType	N	X-stream FIX customization. This field is used to specify board on which security is listed.	String
914	AgreementID	N	X-stream FIX customization. This field is used to specify instrument ID.	String
StandardTrailer		Y		

7.9 Security List Request (x)

The Security List Request message is used to return a list of securities from the Exchange that match criteria provided on the request. The SecurityListRequestType[559] tag specifies the criteria of the request.

Table 42 – Security List Request

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
StandardHeader		Y	MsgType = x (lowercase X)	
320	SecurityReqID	Y	Unique ID for Security List Request	String
559	SecurityListRequestType	Y	Type of Security List Request being made 4 = All securities	Int
263	SubscriptionRequestType	N	Subscribe or unsubscribe for security status to security specified in request. Subscribe or unsubscribe for security status to security specified in request. If not specified, defaults to Snapshot (0)	Char
Standard Trailer		Y		

7.10 Security List (y)

The Security List message is used to return a list of securities that matches the criteria specified in a Security List Request (x). It is also used to reject a failed request.

If the tick rules do not change during the trading day, the TickRules component block will not be included in the message.

Table 43 – Security List

TAG	FIELDNAME	RQD	COMMENTS	FORMAT
StandardHeader		Y	MsgType = y (lowercase Y)	
320	SecurityReqID	Y	Unique Id for the Security List Request.	String
322	SecurityResponseID	Y	Identifier for the Security List message	String
393	TotNoRelatedSym	N	Used to indicate the total number of securities being returned for this request. Used in the event that message fragmentation is required.	Int

TAG	FIELDNAME		RQD	COMMENTS	FORMAT
560	SecurityRequestResult		Y	Result of the Security Request identified by the SecurityReqID.	Int
893	LastFragment		N	Indicates whether this is the last fragment in a sequence of message fragments. When set this indicates the last of the message group.	Boolean
Start of Component block, expanded in line < SecListGrp >					
146	NoRelatedSym		Y/N	Specifies the number of repeating symbols (instruments) specified. Required if SecurityRequestResult (560) = 0 (Valid request)	NumInGrp
→	Component block <Instrument>		Y/N	Insert here the set of "Instrument" (symbology) fields. Required if SecurityRequestResult (560) = 0 (Valid request)	
→	Start of Component block, expanded in line < UndInstrmtGrp >				
→	711	NoUnderlyings	N	Number of underlyings.	NumInGrp
→	→	Component block <Underlying Instrument>	N	Insert here the set of "UnderlyingInstrument" (symbology) fields	
→	End of Component block, expanded in line < UndInstrmtGrp >				
→	Component block <FinancingDetails>		N	Insert here the set of "FinancingDetails" fields for REPO/TTV/SBB securities.	
→	Component block <Stipulations>		N	Insert here the set of "Stipulations" fields. Used to disseminate the Repo Price for a security.	
→	15	Currency	N	Identifies the currency used for price. Absence of this field is interpreted as the default for the security as defined in the reference data.	Currency
→	1306	PriceLimitType	N	Describes the how the price limits are expressed. Valid value: 0 = Price	Int
→	1148	LowLimitPrice	N	Allowable low limit price for the trading day. A key parameter in validating order price. Used as the lower band for validating order prices. Orders submitted with prices below the lower limit will be rejected, or a volatility auction triggered. Please contact the exchange for details.	Price

TAG	FIELDNAME			RQD	COMMENTS	FORMAT
→	1149	HighLimitPrice		N	Allowable high limit price for the trading day. A key parameter in validating order price. Used as the upper band for validating order prices. Orders submitted with prices above the upper limit will be rejected, or a volatility auction triggered. Please contact the exchange for details.	Price
→	1150	TradingReference Price		N	Reference price of the security	Price
→	1305	SecondaryPrice LimitType		N	Describes the how the secondary price limits are expressed. Valid value: 2 = Percentage	Int
→	1221	SecondaryLow LimitPrice		N	Allowable secondary low limit as a percentage from the last marked price. Used as the secondary lower band for validating order prices. Orders submitted with prices below the lower limit will be rejected, or a volatility auction triggered. Please contact the exchange for details.	Price
→	1230	SecondaryHigh LimitPrice		N	Allowable secondary high limit as a percentage from the last marked price. Used as the secondary upper band for validating order prices. Orders submitted with prices above the upper limit will be rejected, or a volatility auction triggered. Please contact the exchange for details.	Price
→	1205	NoTickRules		N	TickRules component block	
→	→	1206	StartTick PriceRange	N	Starting price range for specified tick increment	Price
→	→	1207	EndTick PriceRange	N	Ending price range for the specified tick increment	Price
→	→	1208	Tick Increment	N	Tick increment for stated price range. Specifies the valid price increments at which a security can be quoted and traded	Price
→	562	MinTradeVol		N	The minimum trading volume for a security	Qty
→	1140	MaxTradeVol		N	The maximum order quantity that can be submitted for a security.	Qty
→	Component block <YieldData>			N	Returns Yield information for Fixed Income securities.	
→	20103	MarketPriceType		N	Type of price with which a Fixed Income security is traded. This is a custom tag. Valid values: 0 = Clean Price 1 = Dirty Price	Int
→	2141	StrategyType		N	Strategy identifier	String

TAG	FIELDNAME		RQD	COMMENTS	FORMAT
→	404	LiquidityValue	N	MTP Value (Equities only)	Amt
→	58	Text	N	Free format text string	String
End of Component block, expanded in line < SecListGrp >					
StandardTrailer			Y		

7.11 Security List Update Report (BK)

The Security List Update Report is used for reporting updates for an existing Security List subscription. Updates could be due to Corporate Actions or other business events. Updates may include additions (i.e. new securities) and modifications.

Table 44 – Security List Update Report

TAG	FIELDNAME		REQ'D	COMMENTS	FORMAT
StandardHeader			Y	MsgType = BK	
320	SecurityReqID		Y	Unique Id for the Security List Request.	String
322	SecurityResponseID		N	Identifier for the Security List message	String
393	TotNoRelatedSym		N	Used to indicate the total number of securities being returned for this request. Used in the event that message fragmentation is required.	Int
560	SecurityRequestResult		N	Result of the Security Request identified by the SecurityReqID.	Int
893	LastFragment		N	Indicates whether this is the last fragment in a sequence of message fragments. When set this indicates the last of the message group.	Boolean
292	CorporateAction		N	Identifies the type of Corporate Action that triggered the update (if applicable). Also referred to as 'Basis of Quotation'.	MultipleCharValue
Start of Component block, expanded in line < SecLstUpdRelSymGrp >					
146	NoRelatedSym		Y	Specifies the number of repeating symbols (instruments) specified.	NumInGrp
→	1324	ListUpdateAction	N	If provided then the Instrument occurrence has explicitly changed. This is a V5.0 tag value.	Char
→	Component block <Instrument>		Y	Insert here the set of "Instrument" (symbology) fields.	
→	Start of Component block, expanded in line < UndInstrmtGrp >				

TAG	FIELDNAME		REQ'D	COMMENTS	FORMAT
→	711	NoUnderlyings	N	Number of underlyings.	NumInGrp
→	→	Component block <Underlying Instrument>	N	Insert here the set of "UnderlyingInstrument" (symbology) fields	
→	End of Component block, expanded in line < UndInstrmtGrp >				
→	Component block <FinancingDetails>		N	Contains the X-stream instrument id.	
→	Component block <Stipulations>		N	Insert here the set of "Stipulations" fields. Used to disseminate the Repo Price for a security.	
→	15	Currency	N	Identifies the currency used for price. Absence of this field is interpreted as the default for the security as defined in the reference data.	Currency
→	1306	PriceLimitType	N	Describes the how the price limits are expressed. Valid value: 0 = Price	Int
→	1148	LowLimitPrice	N	Allowable low limit price for the trading day. A key parameter in validating order price. Used as the lower band for validating order prices. Orders submitted with prices below the lower limit will be rejected, or a volatility auction triggered. Please contact the exchange for details.	Price
→	1149	HighLimitPrice	N	Allowable high limit price for the trading day. A key parameter in validating order price. Used as the upper band for validating order prices. Orders submitted with prices above the upper limit will be rejected, or a volatility auction triggered. Please contact the exchange for details.	Price
→	1150	TradingReference Price	N	Reference price of the security	Price
→	1305	SecondaryPrice LimitType	N	Describes the how the secondary price limits are expressed. Valid value: 2 = Percentage	Int

TAG	FIELDNAME		REQ'D	COMMENTS	FORMAT	
→	1221	SecondaryLow LimitPrice		N	Allowable secondary low limit as a percentage from the last marked price. Used as the secondary lower band for validating order prices. Orders submitted with prices below the lower limit will be rejected, or a volatility auction triggered. Please contact the exchange for details.	Price
→	1230	SecondaryHigh LimitPrice		N	Allowable secondary high limit as a percentage from the last marked price. Used as the secondary upper band for validating order prices. Orders submitted with prices above the upper limit will be rejected, or a volatility auction triggered. Please contact the exchange for details.	Price
→	1205	NoTickRules		N	TickRules component block	
→	→	1206	StartTick PriceRange	N	Starting price range for specified tick increment	Price
→	→	1207	EndTick PriceRange	N	Ending price range for the specified tick increment	Price
→	→	1208	Tick Increment	N	Tick increment for stated price range. Specifies the valid price increments at which a security can be quoted and traded	Price
→	562	MinTradeVol		N	The minimum trading volume for a security	Qty
→	1140	MaxTradeVol		N	The maximum order quantity that can be submitted for a security.	Qty
→	Component block <YieldData>			N	Insert here the set of "YieldData" fields.	
→	20103	MarketPriceType		N	Type of price with which a Fixed Income security is traded. This is a custom tag. Valid values: 0 = Clean Price 1 = Dirty Price	Int
→	2141	StrategyType		N	Strategy identifier	String
→	404	LiquidityValue		N	MTP Value (Equities only)	Amt
→	58	Text		N	Free format text string	String
End of Component block, expanded in line < SecLstUpdRelSymGrp >						
StandardTrailer				Y		

7.12 Security Status Request (e)

The Security Status Request message provides for the ability to request the status of a security. One or more Security Status messages are returned as a result of a Security Status Request message.

The Security Status Request message contains a *SubscriptionRequestType* field. This tells the counter party what type of request is being made:

- 0 – indicates that the requestor only wants a snapshot or the current status.
- 1 – indicates that the requestor wants a snapshot (the current status) plus updates as the status changes. This is similar to subscribing for information and can be implemented in applications as a subscription mechanism.
- 2 – indicates that the requestor wishes to cancel any pending snapshots or updates – in essence making this an unsubscribe operation.

The following rules apply for Security Status Requests:

- A request can be sent with either:
 - 762 (SecuritySubType) set to 'ALL' and 55 (Symbol) set to '*', which will return all symbols from all available boards.
 - 762 (SecuritySubType) set to a valid board and 55 (Symbol) set to '*', which will return all symbols from the board.
 - 55 (Symbol) set to a valid symbol, and 762 (SecuritySubType) set to a valid board, which will return the symbol on that board only.
 - 55 (Symbol) set to a valid symbol only, which will return the symbol on its main board only (if one exists).

If a valid combination of SecuritySubType and/or Symbol is not found a reject is returned.

Table 45 – Security Status Request

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
	StandardHeader	Y	MsgType = e (lowercase)	
263	SubscriptionRequestType	Y	SubscriptionRequestType indicates to the other party what type of response is expected. A snapshot request only asks for current information. A subscribe request asks for updates as the status changes. Unsubscribe will cancel any future update messages from the counter party. Subscribe or unsubscribe for security status for security specified in request.	Char
324	SecurityStatusReqID	Y	Must be unique, or the ID of previous Security Status Request to disable if SubscriptionRequestType = Disable previous Snapshot + Updates Request (2).	String
	component block <Instrument>	Y	Insert here the set of "Instrument" (symbology) fields.	
	StandardTrailer	Y		

7.13 Security Status (f)

The Security Status message provides for the ability to report changes in status to a security. The Security Status message is used by the Exchange to report changes in the state of a security. It is also used to reject a failed request.

Table 46 – Security Status

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
StandardHeader		Y	MsgType = f (lowercase)	
324	SecurityStatusReqID	N		String
component block <Instrument>		Y	Insert here the set of "Instrument" (symbology) fields.	
15	Currency	N	Identifies the currency used for price. Absence of this field is interpreted as the default for the security as defined in the reference data.	Currency
31	LastPx	N	Last traded price.	Price
292	CorporateAction	N	Identifies the type of Corporate Action (if applicable). Also referred to as 'Basis of Quotation'.	MultipleCharValue
326	SecurityTradingStatus	Y	Identifies the trading status applicable to the security.	Int
333	LowPx	N	Day low price.	Price
332	HighPx	N	Day high price.	Price
336	TradingSessionID	N	Current trading session	String
58	Text	N	Free format text string	String
60	TransactTime	N	Timestamp when the security status transaction occurred.	UTCTimeStamp
Standard Trailer		Y		

7.14 News (B)

The news message is a general free format message between the participant and Exchange. The message contains flags to identify the news item's urgency and to allow sorting by Subject Company (symbol). The News message can be originated at either the broker or institution side.

Table 47 – News

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
StandardHeader		Y	MsgType = B	
42	OrigTime	N	Time of message origination. Always expressed in UTC time.	UTCTimeStamp
61	Urgency	N	Urgency Flag.	Char
148	Headline	Y	Specifies the headline text	String

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

Appendix A - Standard Header and Trailer

A.1 Standard Header

The standard message header format is as follows.

Table 48 – Standard Message Header

TAG	FIELD NAME	REQ'D	COMMENTS	FORMAT
8	BeginString	Y	FIX.4.4 (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
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	Must be set by a client to identify the trader that is performing the action being sent to the exchange. Pre-assigned to a firm by the exchange.	String
142	SenderLocationID	N	Sender's LocationID (i.e. geographic location and/or desk) (Can be embedded within encrypted data section.). Not supported	String

TAG	FIELD NAME	REQ'D	COMMENTS	FORMAT
57	TargetSubID	N	Set by the exchange to indicate the trader that performed an action.	String
143	TargetLocationID	N	Trading partner LocationID (i.e. geographic location and/or desk) (Can be embedded within encrypted data section.) Not supported	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. Not supported	String
369	LastMsgSeqNumProcessed	N	May be sent by the exchange on a Logout message to indicate the last MsgSeqNum (34) value received from the client and successfully processed.	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 49 – 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 can contain either the following fields:

- Symbol (55)
- SecuritySubType (762)

Or the following two fields:

- SecurityID (48)
- SecurityIDSource (22)

The SecurityStatus and SecurityList responses will return the following tags: 55, 762, 48, 22, 106, 107, 167. SecurityList will return additional tags in the Instrument block according to the security type.

All other messages referencing security information will contain fields 55, 762, 48 and 22 only.

Table 50 – Instrument Component Block

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
55	Symbol	N	Ticker symbol or human readable representation of the security. In X-stream, this is the security code.	String
762	SecuritySubType	N	In X-stream, this field is used to specify board on which SecurityID is listed.	String
48	SecurityID	N	Unique marketplace assigned integer identifier for an order book. This provides a fast lookup for the orderbook.	String
22	SecurityIDSource	N	Valid values: M = Marketplace assigned identifier	Char
106	Issuer	N	Issuer of security	String
107	SecurityDesc	N	Optional textual description of the security.	String

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
167	SecurityType	N	Indicates type of security. Valid values: BUYSELL = Sell/Buy Back Repurchase (SBB) COMM = Commodities CS = Common Stock FUT = Future OPT = Option REPO = Repurchase (cash) SECLOAN = Security Lending Repurchase (TTV) STGY = Strategy TBOND = Fixed Income NONE = No Security Type	String
223	CouponRate	N	For Fixed Income.	Percentage
224	CouponPaymentDate	N	Date interest is to be paid.	LocalMktDate
1946	CouponType	N	Coupon type of the bond. Valid values: 1 = Fixed rate 2 = Floating rate	int
1948	CouponFrequencyPeriod	N	Time unit factor for the frequency of the bond's coupon payment. Note: this will indicate the number of coupons that occur within the CouponFrequencyUnit (1949) specified.	Int
1949	CouponFrequencyUnit	N	Time unit associated with the frequency of the bond's coupon payment. Valid value: 'Yr' = Year	String
1950	CouponDayCount	N	The day count convention used in interest calculations for a bond or an interest bearing security. Absence of this field for a bond or an interest bearing security transaction implies a "flat" trade, i.e. no accrued interest determined at time of the transaction. Valid values: 1 = 30/360 15 = NL365	int
699	BenchmarkSecurityID	N	The identifier of the benchmark security for a bond.	String
225	IssueDate	N	The date when a bond or stock offering is issued.	LocalMktDate
541	MaturityDate	N	Specifies the maturity date or expiry date of a option, or the maturity date of a bond.	LocalMktDate
202	StrikePrice	N	Strike Price for an Option.	Price
231	ContractMultiplier	N	Nominal (qty) multiplier.	Float

TAG	FIELDNAME		REQ'D	COMMENTS	FORMAT
201	PutOrCall		N	Indicates whether an option contract is a put or call Valid values: 0 = Put 1 = Call	Int
228	Factor		N	Circuit Breaker adjustment factor.	Float
1151	SecurityGroup		N	Board Group identifier.	String
1940	AssetType		N	Security Category.	String
1687	ShortSaleRestriction		N	Indicates the current short sell status. Valid values: 0 = No restriction (short selling allowed) 1 = Not shortable (short selling not allowed) 99 = Local only (short selling only allowed by Colombian traders)	Int
Start of Component block, expanded in line < SecAltIDGrp >					
454	NoSecurityAltID		N	Number of SecAltIDGrp fields. Always 1 if specified.	Int
→	455	SecurityAltID	N	Alternate security identifier	String
→	456	SecurityAltID Source	N	Valid value: '4' = ISIN	String
End of Component block, expanded in line < SecAltIDGrp >					

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 (PartyRole = 36)
- EnteringFirm (PartyRole = 7)
- ContraTrader (PartyRole = 37)
- ContraFirm (PartyRole = 17)
- ExecutingTrader (PartyRole = 12)
- ExecutingFirm (PartyRole = 1)
- OrderOriginationTrader (PartyRole = 11)

- ClientID (PartyRole = 3) – free format text string for reference. The maximum length when used in a MassQuote[i] message is 16 characters. When used in any other message the maximum length is 32 characters.
- SettlementLocation (PartyRole = 10) – Depository for Fixed Income only
- SponsoringFirm (PartyRole = 19)
- Transfer to trader (PartyRole = 991)
- Transfer operator (PartyRole = 992)
- Pre-arranged Firm (PartyRole = 999)

Table 51 – Investment Firm 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
 PartyID = “...” – the identifier of the firm
 PartyIDSource = “...” – the type of identifier used
 PartyRole = “1” – Executing Firm
- User
 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
 PartyID = “...” – the identifier of the firm on behalf of the real owner
 PartyIDSource = “...” – the type of identifier used
 PartyRole = “7” – Entering Firm
- User
 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 52 – 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

B.4 TriggeringInstruction Component Block

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

Note: Orders with triggers will not be visible in the order book until the TriggerType event occurs. The OrdStatus (39) field in the Execution Report will return 'X' – Order with trigger in the book but not active, e.g. Order has not been triggered.

Triggered orders when activated (e.g. when the TriggerType occurs) if not immediately traded will cause an unsolicited Execution Report to be sent to the order initiator indicating that the order has become active and available for trading in the order book. If the order is immediately traded (partially or completely) a normal trade execution report will be returned.

Table 53 – TriggeringInstruction Component Block

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
1100	TriggerType	N	Defines when the trigger will hit, i.e. the action specified by the trigger instructions will come into effect. Valid values: 4 = Price Movement	Char
1101	TriggerAction	N	Defines the type of action to take when the trigger hits. Valid values: 1 = Activate	Char
1102	TriggerPrice	Y/N	The price at which the trigger should hit. Mandatory if OrdType is StopLoss or StopLimit.	Price
1107	TriggerPriceType	N	The type of price that the trigger is compared to. Valid values: 2 = Last Trade	Char

B.5 LinesOfTextGrp Component Block

The LinesOfTextGrp component block is used to provide arbitrary text and non-printable information.

Table 54 – LinesOfTextGrp Component Block

TAG	FIELDNAME		REQ'D	COMMENTS	FORMAT
Start of Component block, expanded in line < LinesOfTextGrp >					
33	NoLinesOfText		Y	Specifies the number of repeating lines of text.	NumInGrp
→	58	Text	Y	Free format text string	String
→	354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it. Byte Length of encoded (non-ASCII) characters.	Length
→	355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding.	Data
End of Component block, expanded in line < LinesOfTextGrp >					

B.6 FinancingDetails Component Block

The FinancingDetails component block is used to provide Financial details such as repurchase information.

Table 55 – FinancingDetails Component Block

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
Start of Component block, expanded in line < FinancingDetails >				
914	AgreementID	N	This is the instrument ID in X-stream. Required when submitting a REPO/TTV/SBB order. Valid values are published in the SecurityList message for an underlying Equity or Fixed Income security (see tag 311 in the UnderlyingInstrument Component Block).	String
916	StartDate	N	Settlement date of the beginning of the deal	LocalMktDate
917	EndDate	N	Repayment / repurchase date	LocalMktDate
End of Component block, expanded in line < FinancingDetails >				

B.7 Underlying Instrument Component Block

The Underlying Instrument component block is used to provide underlying security details for Repurchase and Index Securities, and underlying instrument details for securities with Repurchase underlyings.

Table 56 – UnderlyingInstrument Component Block

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
Start of Component block, expanded in line < UnderlyingInstrument >				
311	UnderlyingSymbol	N	Underlying security's Symbol – see tag 55 This indicates the repurchase X-stream instrument when UnderlyingSecurityType (310) is a repurchase type. Values received can be used with the AgreementID (914) tag (see the FinancingAgreement Component Block) when submitting REPO/TTV/SBB orders.	String
763	UnderlyingSecuritySubType	N	Underlying security's SecuritySubType – see tag 762	String

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
310	UnderlyingSecurityType	N	Underlying security's SecurityType – see tag 167 Additional values for repurchase types: 'R' = REPO X-stream instrument 'T' = TTV X-stream instrument 'B' = SBB X-stream instrument	String
879	UnderlyingQty	N	Unit amount of the underlying security	Qty
810	UnderlyingPx	N	Underlying Price	Price
End of Component block, expanded in line < UnderlyingInstrument >				

B.8 Stipulations Component Block

The Stipulations component block is used to disseminate a Repo price or Clean Price, or to submit the Clean Price for Fixed Income orders.

Table 57 – Stipulations Component Block

TAG	FIELDNAME		REQ'D	COMMENTS	FORMAT
Start of Component block, expanded in line < Stipulations >					
232	NoStipulations		N	Specifies the number of repeating stipulations.	NumInGrp
→	233	StipulationType	N	Required if NoStipulations > 0. Valid values: 'REPOPRICE' = Repo Price for a security or order (sent by the exchange only) 'CLEANPRICE' = Clean Price of an order or execution (Fixed Income only, where applicable. Also used where a security is traded by Dirty Price only.)	String
→	234	StipulationValue	N	Value for stipulation.	String
End of Component block, expanded in line < Stipulations >					

Appendix C - Field Enumerations

C.1 Field Enumerations Sorted by Tag Value

Table 58 – Field Enumerations Sorted by Tag Value

TAG	FIELDNAME	COMMENTS	FORMAT
18	ExecInst	Instructions for order handling. Valid values: 'G' – All or None. 'z' – Suspend order 'y' – Release from suspension (mutually exclusive with z) 'o' – Cancel on connection loss (cannot be modified)	MultipleCharValue
39	OrdStatus	Describes the current state of an order. Valid values: 0 – New 1 – Partially filled 2 – Filled 4 – Cancelled 5 – Replaced 8 – Rejected 9 – Suspended (Not Supported) 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	Char

TAG	FIELDNAME	COMMENTS	FORMAT
40	OrdType	<p>Indicates the type of order. Valid values:</p> <p>1 – Market – The Price (44) field is not used, the order executes against the best prices order on the opposite side.</p> <p>2 – Limit – The Price (44) field is specified and the order will execute at this price or better.</p> <p>3 – Stop/Stop Loss – A type of market order that is entered into the book when the defined stop price is reached (i.e. a last trade is at or better than that price). The Price (44) field is not specified, but the TriggerPrice (1102) is. The order will be activated as a Market order when the TriggerPrice is reached.</p> <p>4 – Stop Limit – A type of limit order that is entered into the book when the defined stop price is reached (i.e. a last trade is at or better than that price). Specifies both the Price (44) and the TriggerPrice (1102) field. The order will be activated as a Limit order (using the specified Price as the limit price) when the TriggerPrice is reached.</p> <p>*** NASDAQ Defined ***</p> <p>Z – Market at best - An order that is executed at the best opposite order price limit on the market when it is introduced on the system.</p>	Char
54	Side	<p>Used to indicate the side of the market. Valid values:</p> <p>1 – Buy</p> <p>2 – Sell</p> <p>5 – Short Sell</p>	Char
59	TimeInForce	<p>Indicates time in force techniques that are valid for the specified market segment. Valid values:</p> <p>0 – Day</p> <p>1 – Good till cancelled</p> <p>3 – Immediate or Cancel (IOC)</p> <p>4 – Fill or Kill (FoK)</p> <p>6 – Good till date (GTD)</p> <p>S – Session</p> <p>7 – At the Close</p>	Char
61	Urgency	<p>Urgency Flag. Valid values:</p> <p>0 – Normal</p> <p>1 – Flash</p> <p>2 – Background</p>	Char

TAG	FIELDNAME	COMMENTS	FORMAT
102	CxlRejReason	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
103	OrdRejReason	For optional use with ExecType = 8 (Rejected). Code to identify reason for order rejection. Valid values: 5 = Unknown order 6 – Duplicate order (e.g. duplicate CLOrdID) 99 – Other. Refer to returned Text (58) field for exact reason for rejection.	Int
150	ExecType	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 6 – Pending Cancel (e.g. result of Order Cancel Request) 7 – Stopped 8 – Rejected 9 – Suspended C – Expired F – Trade (partial fill or fill) G – Trade Correct H – Trade Cancel I – Order Status *** NASDAQ Defined *** U – Order is Unplaced o – Transfer offered r – Transfer rejected t – Transferred w – Transfer withdrawn	Char

TAG	FIELDNAME	COMMENTS	FORMAT
235	YieldType	Supported values are: OPENAVG – Open Average Yield CLOSE – Closing Yield TRUE High – Trading session high yield Low – Trading session low yield Last – Last yield WAvg – Weighted Average Change – Change from reference yield.	String
263	SubscriptionRequestType	Used to subscribe for Quote Status Report messages. Subscribe or unsubscribe for security status to security specified in request. Subscription type request. Valid values: 0 – Snapshot 1 – Snapshot+Updates (Subscribe) 2 – Disable previous Snapshot+Update Request (unsubscribe)	Char
264	MarketDepth	Depth of market for Book Snapshot / Incremental updates with market aggregated by price (MBP) or market by order (MBO). Valid values: 0 – full book depth (MBO or MBP) 1 – top of book (MBP only) 5 – 5 price levels per side (MBP only) 10 – 10 orders per side (MBO only)	Int
265	MDUpdateType	Required if SubscriptionRequestType = Snapshot + Updates (1). Specifies the type of Market Data update. Valid values: 1 – Incremental refresh	Int
266	AggregatedBook	Specifies whether or not book entries should be aggregated. Valid values: Y – book entries to be aggregated N – book entries should not be aggregated	Boolean
269	MDEntryType	Must be first field in repeating group. This is a list of all the types of Market Data Entries that the firm requesting the Market Data is interested in receiving. For <u>requests</u> the following are valid values: 0 - Order information – requests all order related information in market by order and market by price messages (e.g. bids, offers, etc.).	Char

TAG	FIELDNAME	COMMENTS	FORMAT
		<p>2 - Trade information – returns all trade information and statistics.</p> <p>3 - Index information – returns all index related information.</p> <p>'a' - Security Statistics (NASDAQ specific) – returns security specific market statistics.</p> <p>For <u>responses to market data requests</u> the following are valid values:</p> <p>0 – Bid</p> <p>1 – Offer</p> <p>2 – Trade</p> <p>3 – Index Value</p> <p>4 – Opening Price</p> <p>5 – Closing Price</p> <p>6 – Settlement Price</p> <p>7 – Trading Session High Price</p> <p>8 – Trading Session Low Price</p> <p>9 – Trading Session VWAP Price</p> <p>C – Open Interest</p> <p>J – Empty Book</p> <p>For 2, 3, 4, 5, 7, 8 the MarketDepth (264) must be set to '1' = 'Top of Book'.</p> <p>*** NASDAQ Extensions ***</p> <p>j – No Trades Exist</p> <p>n – Number of trades</p> <p>q – Total value of security traded during regular hours today (i.e. excludes after-market session)</p> <p>r – Total number of shares traded during regular hours today (i.e. excludes after-market session; returned in tag 1020)</p> <p>s – Unadjusted Previous Closing price</p> <p>t – Delta change from reference price</p> <p>u – Previous closing price</p> <p>v – Total value of security traded today</p> <p>w – Total number of shares traded today (returned in tag 1020)</p> <p>x – Reference Price (returned in tag 270)</p> <p>y – 52 week high and low (returned in tags 332 and 333)</p> <p>z – Indicative opening price and quantity (returned in tags 270, 271)</p>	

TAG	FIELDNAME	COMMENTS	FORMAT
274	TickDirection	Direction of the "tick". Valid values: 0 – Plus Tick 1 – Zero-Plus Tick 2 – Minus Tick 3 – Zero-Minus Tick	Char
279	MDUpdateAction	Must be first field in this repeating group. Valid values: 0 – New 1 – Change 2 – Delete	Char
281	MDReqRejReason	Reason for the rejection of a Market Data request. Valid values: 0 – Unknown symbol 1 – Duplicate MDReqID 2 – Insufficient Bandwidth 3 – Insufficient Permissions 4 – Unsupported Subscription Request Type 5 – Unsupported MarketDepth 6 – Unsupported MDUpdateType 8 – Unsupported MDEntryType 9 – Unsupported TradingSessionID	Char

TAG	FIELDNAME	COMMENTS	FORMAT
292	CorporateAction	<p>Identifies the type of Corporate Action that triggered the update. Also referred to as 'Basis of Quotation'. Valid values:</p> <p>A – Ex-Dividend (XD) C – Ex-Rights (XR) E – Ex-Interest (XI) G – Stock Dividend (CD) H – Stock Split (CS) M – Merger Reorganization (CM) N – Rights Offering (CR)</p> <p>*** NASDAQ Extensions ***</p> <p>a – Cum Bonus (CB) b – Cum Demerge (CE) c – Cum Interest (CI) d – Cum Listing (CL) e – Cum Right of Conversion (CO) f – Call Paid (CP) g – Cum Delisting (CT) h – Offer Closing (OC) i – Unlisted (UL) j – Ex Bonus (XB) k – Ex Demerge (XE) l – Ex Listing (XL) m – Ex Merge (XM) n – Ex Right of Conversion (XO) o – Ex Split (XS) p – Ex Delisting (XT)</p>	MultipleCharValue
297	QuoteStatus	<p>Identifies the status of the quote or mass quote action. Valid values:</p> <p>0 – Accepted 9 – Rejected 16 – Active 17 – Canceled 19 – Pending End Trade</p>	Int
300	QuoteRejectReason	<p>Reason Quote was rejected. Valid values:</p> <p>9 – Not authorized to quote security 99 – Other</p>	Int

TAG	FIELDNAME	COMMENTS	FORMAT
326	SecurityTradingStatus	Identifies the trading status applicable to the transaction. Valid values: 1 – Opening delay 2 – Trading halt 17 – Ready to trade 18 – Not available for trading 19 – Not traded on this market 20 – Unknown or Invalid	Int
336	TradingSessionID	Identifier for Trading Session. A trading session spans an extended 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.)	String
340	TradSesStatus	State of trading session. Valid values: 6 – Request rejected *** NASDAQ Specific *** 100 – Pending – Indicates that trading session has not been started 101 – Triggered – Indicates trading session has either occurred or is the current session 102 – Deleted – This trading session has been removed from the trading schedule.	Int
368	QuoteEntryRejectReason	Reason Quote Entry was rejected. Valid values: 1 – Unknown Symbol (security) 3 – Quote request exceeds limit 7 – Invalid bid/ask spread 8 – Invalid price 9 – Not authorized to quote security 10 – Price exceeds current price band 99 – Other	Int
380	BusinessRejectReason	Reason for rejection. 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

TAG	FIELDNAME	COMMENTS	FORMAT
434	CxlRejResponseTo	Identifies the type of request that a Cancel Reject is in response to. Valid values: 1 – Order Cancel Request 2 – Order Cancel/Replace Request	Char
447	PartyIDSource	Used to identify class source of PartyID value. Required if PartyID is specified. Required if NoPartyIDs > 0. Valid values: C – Participant identifier	Char
452	PartyRole	Identifies the type of PartyID (e.g. Executing Broker). Required if NoPartyIDs > 0. Valid values: 1 – Executing Firm 3 – Client ID 7 – Entering Firm 10 – Settlement Location (Depository) 11 – Order origination trader 12 – Executing Trader 17 – Contra Firm 19 – Sponsoring Firm 36 – Entering trader 37 – Contra trader *** NASDAQ Defined *** 991 – Transfer to trader 992 – Transfer operator (initiator) 999 – Pre-arranged Firm	Int
559	SecurityListRequestType	Specifies the criteria of the request: 4 – All Securities	Int
560	SecurityRequestResult	Result of the Security Request identified by the SecurityReqID. Valid values: 0 – Valid request 1 – Invalid or unsupported request 2 – No instruments found that match selection criteria 3 – Not authorized to retrieve instrument data 4 – Instrument data temporarily unavailable 5 – Request for instrument data not supported	Int
567	TradSesStatusRejReason	Used with TradSesStatus = "Request Rejected". Valid values: 1 – Unknown trading session id 99 – Other	Int

TAG	FIELDNAME	COMMENTS	FORMAT
569	TradeRequestType	Type of Trade Capture Report. Valid values: 0 – All Trades	Int
574	MatchType	The point in the matching process at which this trade was matched. Valid values: 1 - One-Party Trade Report (privately negotiated crossing trade) 2 - Two-Party Trade Report (privately negotiated trade) 99 – Pre-arranged	String
828	TrdType	Type of trade. Valid values: 22 – Privately Negotiated Trade 99 – RFQ Trade	Int
829	TrdSubType	Indicates whether the trader is on the buy or sell side of a fill. Valid values: 98 – Execution on Buy side 99 – Execution on Sell side	Int
911	TotNumReports	Total number of reports returned in response to a request	Int
980	SecurityUpdateAction	Specifies the update action for the security. Valid values: A – Add D – Delete M – Modify	Char
1021	MDBookType	Describes the type of book for which the feed is intended. Used when multiple feeds are provided over the same connection. Valid Values: 1 – Top of Book 2 – Price Depth 3 – Order Depth	Int
1057	AggressorIndicator	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
1167	QuoteEntryStatus	Status of quote entry action. Valid values: 0 = Accepted 5 = Rejected	Int

TAG	FIELDNAME	COMMENTS	FORMAT
1324	ListUpdateAction	Specifies the action for a security list. If provided then the Instrument occurrence has explicitly changed: Valid values: A – Add M – Modify	Char
2446	AggressorSide	Indicates which side of a trade was the aggressor. Valid values: 1 = Buy 2 = Sell	Int
2679	OrderOwnershipIndicator	Change ownership of an order to a specific party. Valid values: 3 = Submit transfer (by initiator) 97 = Reject transfer (by recipient) 98 = Withdraw transfer (by initiator) 99 = Accept transfer (by recipient)	Int

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 "-" 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").</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> <p>The following fields are based on char.</p>	

	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

	<p>special-purpose field is paired with UTCDateOnly 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>
UTCDateOnly	<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>

Appendix E - FIX Message Examples

The following are basic examples of some FIX messages used with X-stream.

E.1 Login

```
← 8=FIX.4.4|9=0087|35=A|49=T01|56=EXC|34=1|52=20150512-01:00:00.000
  |98=0|108=15|553=T01FC001|554=password|10=215|

→ 8=FIX.4.4|9=61|35=A|49=EXC|56=T01|34=1|52=20150512-02:00:05.335
  |98=0|108=15|10=231|
```

E.2 Order Entry

```
← 8=FIX.4.4|9=0154|35=D|49=T01|56=EXC|34=2|50=T01FT001
  |52=20150512-01:00:00.000|11=FIX1-1|55=BCOLOMBIA|762=EQTY|38=2000
  |40=2|44=25000|54=1|60=20150512-01:00:00.000|1=T01AC1|10=235|

→ 8=FIX.4.4|9=305|35=8|49=EXC|56=T01|34=2|57=T01FT001
  |52=20150512-02:00:05.346|11=FIX1-1|17=1|37=20150512000000000001
  |150=0|453=3|448=T01FC001|447=C|452=11|448=T01FT001|447=C|452=12
  |448=T01|447=C|452=1|55=BCOLOMBIA|762=EQTY|48=2012|22=M
  |14=0|38=2000|39=0|40=2|44=25000.00|54=1|59=0
  |60=20150512-02:00:05.346|151=2000|1=T01AC1|10=246|
```

E.3 Order Modification

```
← 8=FIX.4.4|9=0155|35=G|49=T01|56=EXC|34=3|50=T01FT001
  |52=20150512-01:00:00.000|11=FIX1-2|41=FIX1-1|55=ECOH16F|762=FUEQY
  |38=300|40=2|44=13|54=1|60=20150512-01:00:00.000|59=0|10=218|

→ 8=FIX.4.4|9=333|35=8|49=EXC|56=T01|34=3|57=T01FT001
  |52=20150512-02:00:05.507|11=FIX1-2|17=2|37=20150512000000000002
  |41=FIX1-1|150=5|198=20150512000000000001|453=3
  |448=T01FC001|447=C|452=11|448=T01FT001|447=C|452=12
  |448=T01|447=C|452=1|55=ECOH16F|762=FUEQY
  |48=2337|22=M|14=0|38=300|39=0|40=2|44=13.000|54=1|59=0
  |60=20150512-02:00:05.507|151=300|1=T01AC1|10=064|
```

E.4 Order Cancellation

```
← 8=FIX.4.4|9=0111|35=F|49=T01|56=EXC|34=4|50=T01FT001
  |52=20150512-01:00:00.000|41=FIX1-2|11=FIX1-3|54=2
  |60=20150512-01:00:00.000|10=000|

→ 8=FIX.4.4|9=308|35=8|49=EXC|56=T01|34=4|57=T01FT001
  |52=20150512-02:00:06.514|11=FIX1-3|17=3|37=20150512000000000002
  |41=FIX1-2|150=4|453=3|448=T01FC001|447=C|452=11
  |448=T01FT001|447=C|452=12|448=T01|447=C|452=1|55=ECOH16F
  |762=FUEQY|48=2337|22=M|14=0|38=300|39=4|40=2|44=13.000|54=1
  |59=0|60=20150512-02:00:06.514|151=0|1=T01AC1|10=147|
```

E.5 Order Execution (Fixed Income)

- ← 8=FIX.4.4|9=0195|35=D|49=T02|56=EXC|34=2|50=FITRADER02
|52=20150512-01:00:00.000|11=FIX1-9
|453=1|448=DEP|447=C|452=10
|55=TFIT16240724|762=ASPT|38=500|40=2|44=99.95|54=2
|60=20150512-01:00:00.000|10=053|
- 8=FIX.4.4|9=365|35=8|49=EXC|56=T02|34=2|57=FITRADER02
|52=20150512-15:00:15.153|11=FIX1-9|17=1|37=201505120000000006|150=0
|453=4|448=T02FC001|447=C|452=11|448=FITRADER02|447=C|452=12
|448=DEP|447=C|452=10|448=T02|447=C|452=1
|55=TFIT16240724|762=ASPT|48=2201|22=M|14=0|38=500
|39=0|40=2|44=99.950|20102=101.213|54=2|59=0
|60=20150512-15:00:15.153|151=500|236=16.084|10=215|
- 8=FIX.4.4|9=481|35=8|49=EXC|56=T02|34=3|57=FITRADER02
|52=20150512-15:00:15.153|11=FIX1-9|17=2|37=201505120000000006|150=F
|453=4|448=T02FC001|447=C|452=11|448=FITRADER02|447=C|452=12
|448=DEP|447=C|452=10|448=T02|447=C|452=1
|55=TFIT16240724|762=ASPT|48=2201|22=M|6=121.160
|14=500|31=121.160|32=500|38=500|39=2|40=2|44=99.950|20102=101.213
|54=2|59=0|60=20150512-15:00:15.153|75=20150512
|64=20150512|151=0|236=16.084
|1623=3.000|381=625.000|880=201505120000000001
|1057=Y|10=085|

E.6 Negotiated Deal Execution (Fixed Income)

Negotiated Deal entry:

- ← 8=FIX.4.4|9=0274|35=AE|49=T01|56=EXC|34=2|50=FITRADER01
|52=20150512-01:00:00.000|571=FIX1-1|487=0|856=0|55=TFIT16240724
|762=RTES|31=99.95|32=2000|60=20150512-01:00:00.000|552=1|54=1
|453=3|448=FITRADER02|447=C|452=37|448=DEP|447=C|452=10
|448=130100|447=C|452=3|10=188|
- 8=FIX.4.4|9=124|35=AR|49=EXC|56=T01|34=2|57=FITRADER01
|52=20150512-15:00:05.592|571=FIX1-1|487=0|856=0|939=0
|17=1|60=20150512-15:00:05.592|10=248|
- 8=FIX.4.4|9=450|35=AE|49=EXC|56=T02|34=2|57=FITRADER02
|52=20150512-15:00:05.592|571=20150512-000000000002|487=0|856=0
|17=1|828=22|15=COP|20102=101.213|55=TFIT16240724|762=RTES
|48=2209|22=M|236=16.084|31=99.950|32=2000|381=2077.000
|60=20150512-15:00:05.592|64=20150512|552=2|54=2|453=3
|448=FITRADER02|447=C|452=12|448=T02|447=C|452=1
|448=FITRADER02|447=C|452=36|54=1|453=2|448=FITRADER01
|447=C|452=37|448=T01|447=C|452=17|10=046|

Negotiated Deal confirmation and execution:

- ← 8=FIX.4.4|9=0266|35=AE|49=T02|56=EXC|34=2|50=FITRADER02
 |52=20150512-01:00:00.000|571=FIX2-2|487=2|856=2
 |572=20150512-0000000000002|55=TFIT16240724|762=RTES|31=99.95
 |32=2000|60=20150512-01:00:00.000|552=1|54=2|453=3
 |448=FITRADER01|447=C|452=37|448=DEP|447=C|452=10
 |448=130100 free text|447=C|452=3|10=059|
- 8=FIX.4.4|9=543|35=AE|49=EXC|56=T01|34=3|57=FITRADER01
 |52=20150512-15:00:07.189|571=20150512-0000000000001|487=2
 |856=0|572=FIX1-1|17=2|828=22|150=F|573=0|574=2|15=COP
 |880=2015051200000000001|75=20150512|20102=101.213|55=TFIT16240724
 |762=RTES|48=2209|22=M|236=16.084|31=99.950|32=2000|381=2077.000
 |60=20150512-15:00:07.189|64=20150512|552=2|54=1|453=5
 |448=FITRADER01|447=C|452=12|448=T01|447=C|452=1|448=T01FC001
 |447=C|452=36|448=DEP|447=C|452=10|448=130100|447=C|452=3
 |54=2|453=2|448=FITRADER02|447=C|452=37
 |448=T02|447=C|452=17|10=146|
- 8=FIX.4.4|9=542|35=AE|49=EXC|56=T02|34=3|57=FITRADER02
 |52=20150512-15:00:07.189|571=20150512-0000000000002|487=2|856=0
 |572=FIX2-2|17=2|828=22|150=F|573=0|574=2|15=COP
 |880=2015051200000000001|75=20150512|20102=101.213|55=TFIT16240724
 |762=RTES|48=2209|22=M|236=16.084|31=99.950|32=2000
 |381=2077.000|60=20150512-15:00:07.189|64=20150512|552=2|54=2|453=5
 |448=FITRADER02|447=C|452=12|448=T02|447=C|452=1|448=T02FC001
 |447=C|452=36|448=DEP|447=C|452=10|448=130100 free text|447=C|452=3
 |54=1|453=2|448=FITRADER01|447=C|452=37|448=T01|447=C|452=17|10=152|

E.7 RFQ Execution (Fixed Income)*RFQ entry:*

- ← 8=FIX.4.4|9=0235|35=R|49=T01|56=EXC|34=2|50=FI_DEALER01
 |52=20150512-01:00:00.000|131=FIX1-1|146=1|55=TFIT16240724
 |762=RFQF|54=1|38=1000|62=20150513-01:00:00|453=3
 |448=FITRADER02|447=C|452=37|448=T03|447=C|452=17
 |448=DEP|447=C|452=10|10=134|
- 8=FIX.4.4|9=325|35=AI|49=EXC|56=T01|34=2|57=FITRADER01
 |52=20150512-15:00:05.162|131=FIX1-1|453=6|448=DEP|447=C
 |452=10|448=FITRADER01|447=C|452=12|448=T01|447=C|452=1
 |448=T01FC001|447=C|452=36|448=FITRADER02|447=C|452=37|448=T03
 |447=C|452=17|55=TFIT16240724|762=RFQF|48=2209|22=M|54=1
 |38=1000|62=20150513-01:00:00|297=10|10=122|
- 8=FIX.4.4|9=262|35=R|49=EXC|56=T02|34=2|57=FITRADER02
 |52=20150512-15:00:05.162|131=20150512-0000000000001.00|146=1
 |55=TFIT16240724|762=RFQF|48=2209|22=M|54=1|38=1000
 |62=20150513-01:00:00|453=3|448=FITRADER01|447=C|452=12|448=T01
 |447=C|452=1|448=FITRADER02|447=C|452=37|10=218|
- 8=FIX.4.4|9=239|35=R|49=EXC|56=T03|34=2|52=20150512-15:00:05.162
 |131=20150512-0000000000001.01|146=1|55=TFIT16240724|762=RFQF
 |48=2209|22=M|54=1|38=1000|62=20150513-01:00:00|453=3
 |448=FITRADER01|447=C|452=12|448=T01|447=C|452=1|448=T03|447=C
 |452=17|10=241|

Quote entry by respondent:

- ← 8=FIX.4.4|9=0212|35=S|49=T02|56=EXC|34=2|50=FITRADER02
|52=20150512-01:00:00.000|131=20150512-000000000001.00
|117=FIX2-1|55=TFIT16240724|762=RFQF|453=1|448=DEP
|447=C|452=10|634=3.1|135=1000|62=20150513-00:00:00|10=106|
- 8=FIX.4.4|9=331|35=S|49=EXC|56=T01|34=3|57=FITRADER01
|52=20150512-15:00:06.575|131=FIX1-1|117=20150512-000000000001.00
|55=TFIT16240724|762=RFQF|48=2209|22=M|453=5
|448=DEP|447=C|452=10|448=FITRADER01|447=C|452=12
|448=T01|447=C|452=1|448=T01FC001|447=C|452=36|448=FITRADER02
|447=C|452=37|634=3.100|135=1000|62=20150513-00:00:00|10=151|
- 8=FIX.4.4|9=318|35=AI|49=EXC|56=T02|34=3|57=FITRADER02
|52=20150512-15:00:06.575|131=20150512-000000000001.00
|117=FIX2-1|453=4|448=DEP|447=C|452=10|448=FITRADER01
|447=C|452=12|448=T01|447=C|452=1|448=FITRADER02|447=C|452=37
|55=TFIT16240724|762=RFQF|48=2209|22=M|54=1|634=3.100
|135=1000|62=20150513-00:00:00|297=16|10=229|

Quote acceptance by initiator, and execution:

- ← 8=FIX.4.4|9=0148|35=AJ|49=T01|56=EXC|34=3|50=FITRADER01
|52=20150512-01:00:00.000|693=FIX1-2
|117=20150512-000000000001.00|694=1|54=1|55=TFIT16240724
|762=RFQF|10=087|
- 8=FIX.4.4|9=522|35=8|49=EXC|56=T01|34=4|57=FITRADER01
|52=20150512-15:00:08.028|11=FIX1-2|17=1|37=0|150=F
|1080=20150512-000000000001.00|1081=5|453=6|448=FITRADER01
|447=C|452=12|448=T01|447=C|452=1|448=FITRADER02|447=C|452=37
|448=T02|447=C|452=17|448=T01FC001|447=C|452=36|448=DEP
|447=C|452=10|55=TFIT16240724|762=RFQF|48=2209|22=M|6=120.970
|14=1000|31=120.970|32=1000|38=1000|39=2|20102=122.537|54=1
|60=20150512-15:00:08.028|75=20150512|63=1|64=20150512|151=0|1623=3.100
|381=1249.000|880=201505120000000001|10=176|
- 8=FIX.4.4|9=522|35=8|49=EXC|56=T02|34=4|57=FITRADER02
|52=20150512-15:00:08.028|11=FIX2-1|17=1|37=0|150=F
|1080=20150512-000000000001.00|1081=6|453=6|448=FITRADER02|447=C
|452=12|448=T02|447=C|452=1|448=FITRADER01|447=C|452=37|448=T01
|447=C|452=17|448=T02FC001|447=C|452=36|448=DEP|447=C
|452=10|55=TFIT16240724|762=RFQF|48=2209|22=M|6=120.970
|14=1000|31=120.970|32=1000|38=1000|39=2|20102=122.537|54=2
|60=20150512-15:00:08.028|75=20150512|63=1|64=20150512|151=0|1623=3.100
|381=1249.000|880=201505120000000001|10=102|
- 8=FIX.4.4|9=215|35=R|49=EXC|56=T03|34=3|52=20150512-15:00:08.028
|131=20150512-000000000001.01|146=1|55=TFIT16240724|762=RFQF
|48=2209|22=M|54=1|38=0|453=3|448=FITRADER01|447=C|452=12
|448=T01|447=C|452=1|448=T03|447=C|452=17|10=102|

E.8 Repo Order Execution (Fixed Income)

First repo order entry (new repo security TFIT16240724_S0E167C_A is generated):

```

← 8=FIX.4.4|9=0250|35=D|49=T01|56=EXC|34=2|50=FITRADER01
  |52=20150512-01:00:00.000|11=FIX1-1|453=1
  |448=DEP|447=C|452=10|55=TFIT16240724|762=ASPT
  |38=1|40=2|44=5|54=1|60=20150512-01:00:00.000|59=S
  |914=RPT16240724|916=20150512|917=20151026|921=100000000000|10=212|

→ 8=FIX.4.4|9=463|35=8|49=EXC|56=T01|34=2|57=FITRADER01
  |52=20150512-15:00:05.300|11=FIX1-1|17=1|37=201505120000000001
  |150=0|453=4|448=T01FC001|447=C|452=11|448=FITRADER01|447=C|452=12
  |448=DEP|447=C|452=10|448=T01|447=C|452=1
  |55=TFIT16240724_S0E167C_A|762=ARPO|48=3304|22=M|14=0|39=0
  |40=2|44=5.000|20102=122.537|54=1|59=S|60=20150512-15:00:05.300|151=0
  |914=RPT16240724|916=20150512|917=20151026|311=TFIT16240724
  |763=ASPT|232=1|233=REPOPRICE|234=38682.000
  |921=99999977601.000|922=102319421526.000|10=243|

```

Second repo order entry to match with first order:

```

← 8=FIX.4.4|9=0248|35=D|49=T02|56=EXC|34=2|50=FITRADER02
  |52=20150512-01:00:00.000|11=FIX2-1|453=1
  |448=DEP|447=C|452=10|55=TFIT16240724_S0E167C_A
  |762=ARPO|38=2645143|40=2|44=5|54=2|60=20150512-01:00:00.000
  |59=S|914=RPT16240724|916=20150512|917=20151026|10=106|

→ 8=FIX.4.4|9=479|35=8|49=EXC|56=T02|34=2|57=FITRADER02
  |52=20150512-15:00:06.308|11=FIX2-1|17=1|37=201505120000000002
  |150=0|453=4|448=T02FC001|447=C|452=11|448=FITRADER02|447=C|452=12
  |448=DEP|447=C|452=10|448=T02|447=C|452=1
  |55=TFIT16240724_S0E167C_A|762=ARPO|48=3304|22=M|14=0
  |38=2645143|39=0|40=2|44=5.000|54=2|59=S|60=20150512-15:00:06.308
  |151=2645143|914=RPT16240724|916=20150512|917=20151026
  |311=TFIT16240724|763=ASPT|232=1|233=REPOPRICE|234=38682.000
  |921=99999977601.000|922=102319421526.000|10=121|

```

Executions for both sides (two repo leg trades produced):

- ➔ 8=FIX.4.4|9=638|35=8|49=EXC|56=T01|34=3|57=FITRADER01
 |52=20150512-15:00:06.308|11=FIX1-1|17=2|37=201505120000000001
 |150=F|453=6|448=T01FC001|447=C|452=11|448=FITRADER01|447=C|452=12
 |448=DEP|447=C|452=10|448=T01|447=C|452=1
 |448=FITRADER01|447=C|452=37|448=T01|447=C|452=17
 |55=TFIT16240724_S0E167C_A|762=ARPO|48=3304|22=M|6=5.000
 |14=0|31=5.000|32=2645143|39=C|40=2|44=5.000|54=1|59=S
 |60=20150512-15:00:06.308|75=20150512|64=20150512|151=0
 |914=RPT16240724|916=20150512|917=20151026|311=TFIT16240724
 |763=ASPT|232=1|233=REPOPRICE|234=38682.000|921=99999977601.000
 |922=102319421526.000|381=99999977601.000|880=201505120000000001
 |824=201505120000000002|1057=N|10= 202|
- ➔ 8=FIX.4.4|9=639|35=8|49=EXC|56=T01|34=4|57=FITRADER01
 |52=20150512-15:00:06.308|11=FIX1-1|17=3|37=201505120000000001
 |150=F|453=6|448=T01FC001|447=C|452=11|448=FITRADER01|447=C|452=12
 |448=DEP|447=C|452=10|448=T01|447=C|452=1
 |448=FITRADER02|447=C|452=37|448=T02|447=C|452=17
 |55=TFIT16240724_S0E167C_A|762=ARPO|48=3304|22=M|6=5.000
 |14=0|31=5.000|32=2645143|39=C|40=2|44=5.000|54=1|59=S
 |60=20150512-15:00:06.308|75=20150512|64=20151026|151=0
 |914=RPT16240724|916=20150512|917=20151026|311=TFIT16240724
 |763=ASPT|232=1|233=REPOPRICE|234=38682.000|921=99999977601.000
 |922=102319421526.000|381=102319421526.000|880=201505120000000002
 |824=201505120000000001|1057=Y|10= 228|
- ➔ 8=FIX.4.4|9=654|35=8|49=EXC|56=T02|34=3|57=FITRADER02
 |52=20150512-15:00:06.308|11=FIX2-1|17=2|37=201505120000000002
 |150=F|453=6|448=T02FC001|447=C|452=11|448=FITRADER02|447=C|452=12
 |448=DEP|447=C|452=10|448=T02|447=C|452=1
 |448=FITRADER02|447=C|452=37|448=T02|447=C|452=17
 |55=TFIT16240724_S0E167C_A|762=ARPO|48=3304|22=M|6=5.000
 |14=2645143|31=5.000|32=2645143|38=2645143|39=2|40=2|44=5.000|54=2
 |59=S|60=20150512-15:00:06.308|75=20150512|64=20150512|151=0
 |914=RPT16240724|916=20150512|917=20151026|311=TFIT16240724
 |763=ASPT|232=1|233=REPOPRICE|234=38682.000|921=99999977601.000
 |922=102319421526.000|381=99999977601.000|880=201505120000000001
 |824=201505120000000002|1057=Y|10=049|
- ➔ 8=FIX.4.4|9=655|35=8|49=EXC|56=T02|34=4|57=FITRADER02
 |52=20150512-15:00:06.308|11=FIX2-1|17=3|37=201505120000000002
 |150=F|453=6|448=T02FC001|447=C|452=11|448=FITRADER02|447=C|452=12
 |448=DEP|447=C|452=10|448=T02|447=C|452=1
 |448=FITRADER01|447=C|452=37|448=T01|447=C|452=17
 |55=TFIT16240724_S0E167C_A|762=ARPO|48=3304|22=M|6=5.000
 |14=2645143|31=5.000|32=2645143|38=2645143|39=2|40=2|44=5.000|54=2
 |59=S|60=20150512-15:00:06.308|75=20150512|64=20151026|151=0
 |914=RPT16240724|916=20150512|917=20151026|311=TFIT16240724
 |763=ASPT|232=1|233=REPOPRICE|234=38682.000|921=99999977601.000
 |922=102319421526.000|381=102319421526.000|880=201505120000000002
 |824=201505120000000001|1057=N|10=049|

E.9 Repo Negotiated Deal Execution (Fixed Income)

Repo Negotiated Deal entry (new repo security TFIT16240724_S0E10C_A is generated):

```

← 8=FIX.4.4|9=0278|35=AE|49=T01|56=EXC|34=2|50=FITRADER01
  |52=20150512-01:00:00.000|571=FIX1-1|487=0|856=0|55=TFIT16240724
  |762=ASPT|914=RPT16240724|916=20150512|917=20150522|31=5
  |32=3294802|60=20150512-01:00:00.000|552=1|54=2|453=2
  |448=FITRADER02|447=C|452=37|448=DEP|447=C|452=10|10=014|

→ 8=FIX.4.4|9=124|35=AR|49=EXC|56=T01|34=2|57=FITRADER01
  |52=20150512-15:00:04.959|571=FIX1-1|487=0|856=0|939=0
  |17=1|60=20150512-15:00:04.959|10=004|

→ 8=FIX.4.4|9=512|35=AE|49=EXC|56=T02|34=2|57=FITRADER02
  |52=20150512-15:00:04.959|571=20150512-000000000000|487=0|856=0
  |17=1|828=22|311=TFIT16240724|763=ASPT|15=COP
  |55=TFIT16240724_S0E10C_A|762=ARPO|48=3304|22=M|914=RPT16240724
  |916=20150512|917=20150522|232=1|233=REPOPRICE|234=38682.000
  |31=5.000|32=3294802|60=20150512-15:00:04.959|64=20150522|552=2|54=1
  |453=3|448=FITRADER02|447=C|452=12|448=T02|447=C|452=1
  |448=FITRADER02|447=C|452=36|921=127272763237.000|922=127449530964.000
  |54=2|453=2|448=FITRADER01|447=C
  |452=37|448=T01|447=C|452=17|10=195|

```

Repo Negotiated Deal confirmation and execution (two repo leg trades produced):

```

← 8=FIX.4.4|9=0241|35=AE|49=T02|56=EXC|34=2|50=FITRADER02
  |52=20150512-01:00:00.000|571=FIX2-2|487=2|856=2
  |572=20150512-000000000000|55=TFIT16240724_S0E10C_A|762=ARPO
  |31=5|32=3294802|60=20150512-01:00:00.000|552=1
  |54=1|453=1|448=DEP|447=C|452=10|10=139|

→ 8=FIX.4.4|9=628|35=AE|49=EXC|56=T01|34=3|57=FITRADER01
  |52=20150512-15:00:05.966|571=20150512-000000000000|487=2
  |856=0|572=FIX1-1|17=2|828=22|150=F|311=TFIT16240724
  |763=ASPT|573=0|574=2|15=COP|880=2015051200000000001
  |824=2015051200000000002|75=20150512|55=TFIT16240724_S0E10C_A
  |762=ARPO|48=3304|22=M|914=RPT16240724|916=20150512
  |917=20150522|232=1|233=REPOPRICE|234=38682.000|31=5.000
  |32=3294802|381=127272763237.000|60=20150512-15:00:05.966
  |64=20150512|552=2|54=2|453=4|448=FITRADER01|447=C|452=12|448=T01
  |447=C|452=1|448=T01FC001|447=C|452=36|448=DEP
  |447=C|452=10|921=127272763237.000|922=127449530964.000
  |54=1|453=2|448=FITRADER02|447=C|452=37
  |448=T02|447=C|452=17|10=061|

```

- ➔ 8=FIX.4.4|9=629|35=AE|49=EXC|56=T01|34=4|57=FITRADER01
 |52=20150512-15:00:05.966|571=20150512-0000000000001X|487=2
 |856=0|572=FIX1-1|17=3|828=22|150=F|311=TFIT16240724
 |763=ASPT|573=0|574=2|15=COP|880=2015051200000000002
 |824=2015051200000000001|75=20150512|55=TFIT16240724_S0E10C_A
 |762=ARPO|48=3304|22=M|914=RPT16240724|916=20150512
 |917=20150522|232=1|233=REPOPRICE|234=38682.000|31=5.000
 |32=3294802|381=127449530964.000|60=20150512-15:00:05.966
 |64=20150522|552=2|54=2|453=4|448=FITRADER01|447=C|452=12|448=T01
 |447=C|452=1|448=T01FC001|447=C|452=36|448=DEP
 |447=C|452=10|921=127272763237.000|922=127449530964.000
 |54=1|453=2|448=FITRADER02|447=C|452=37
 |448=T02|447=C|452=17|10=158|
- ➔ 8=FIX.4.4|9=628|35=AE|49=EXC|56=T02|34=3|57=FITRADER02
 |52=20150512-15:00:05.966|571=20150512-0000000000002|487=2
 |856=0|572=FIX2-2|17=2|828=22|150=F|311=TFIT16240724
 |763=ASPT|573=0|574=2|15=COP|880=2015051200000000001
 |824=2015051200000000002|75=20150512|55=TFIT16240724_S0E10C_A
 |762=ARPO|48=3304|22=M|914=RPT16240724|916=20150512
 |917=20150522|232=1|233=REPOPRICE|234=38682.000|31=5.000
 |32=3294802|381=127272763237.000|60=20150512-15:00:05.966
 |64=20150512|552=2|54=1|453=4|448=FITRADER02|447=C|452=12|448=T02
 |447=C|452=1|448=T02FC001|447=C|452=36|448=DEP
 |447=C|452=10|921=127272763237.000|922=127449530964.000
 |54=2|453=2|448=FITRADER01|447=C|452=37
 |448=T01|447=C|452=17|10=067|
- ➔ 8=FIX.4.4|9=629|35=AE|49=EXC|56=T02|34=4|57=FITRADER02
 |52=20150512-15:00:05.966|571=20150512-0000000000002X
 |487=2|856=0|572=FIX2-2|17=3|828=22|150=F|311=TFIT16240724
 |763=ASPT|573=0|574=2|15=COP|880=2015051200000000002
 |824=2015051200000000001|75=20150512|55=TFIT16240724_S0E10C_A
 |762=ARPO|48=3304|22=M|914=RPT16240724|916=20150512
 |917=20150522|232=1|233=REPOPRICE|234=38682.000|31=5.000
 |32=3294802|381=127449530964.000|60=20150512-15:00:05.966
 |64=20150522|552=2|54=1|453=4|448=FITRADER02|447=C|452=12|448=T02
 |447=C|452=1|448=T02FC001|447=C|452=36|448=DEP
 |447=C|452=10|921=127272763237.000|922=127449530964.000
 |54=2|453=2|448=FITRADER01|447=C|452=37
 |448=T01|447=C|452=17|10=164|