

Bolsa Institucional
de Valores

BIVA

SPECIFICATION

BIVA X-STREAM EXTERNAL FIX SPECIFICATION

Version	1.02
Date	10 March 2017
File	BIVA X-Stream External FIX Specification v1.02

Copyright © 2016 Central de Corretajes (CENCOR), S.A. de C.V.

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 2016 Central de Corretajes, S.A. de C.V.

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 OMX Group, Inc, its subsidiaries or employees for any direct or indirect loss or damage caused by omissions from or inaccuracies in this document.

OMX 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 NASDAQ OMX Group, Inc.

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

Document History

VERSION	DATE	AUTHOR	SUMMARY OF CHANGES
0.8	2015-12-30	Carlos Arvizu	Version for Member Firms – Reference version 1.01
0.9	2016-03-07	Carlos Arvizu	Version for Member Firms – Reference version 1.03
1.0	2016-08-23	Carlos Arvizu	Version for Member Firms – Reference version 1.06
1.01	2016-10-19	Juan J. Martínez	Version for Member Firms – Reference version 1.06
1.02	2017-03-10	Juan J. Martínez	Version for Member Firms – Reference version 1.07, add 'o' to Execlnst to indicate the order should be withdrawn on logoff or connection loss.

Contents

1	Introduction to X-stream INET FIX	6
2	Session Information	7
2.1	FIX Admin and Infrastructure Messages Supported	7
2.2	SenderCompID and TargetCompID	7
3	FIX Application Messages	9
4	FIX Message Definitions	10
4.1	Session	10
4.1.1	Logon (A)	10
4.1.2	Logout (5)	10
4.1.3	Reject (3)	10
4.1.4	Resend Request (2)	11
4.1.5	Sequence Reset (Gap Fill) (4)	12
4.1.6	Test Request (1)	13
4.1.7	Heartbeat (0)	13
4.2	Infrastructure	13
4.2.1	Business Message Reject (j)	13
4.3	Application	14
4.3.1	New Order Single (D)	14
4.3.2	New Order Cross [s]	15
4.3.3	Order Cancel Request (F)	16
4.3.4	Order Cancel/Replace Request (G)	17
4.3.5	Execution Report (8)	19
4.3.6	Order Cancel Reject (9)	22
4.3.7	Trade Capture Report [AE]	23
4.3.8	Trade Capture Report Ack [AR]	25
4.3.9	Trade Capture Report Work Flow for IPO Cross	25
5	Order Management	27
5.1	Unique ClOrdID (11)	27
5.2	Order Identification	27
5.3	Order Modification via Order Cancel/Replace Request	27
5.4	Order Cancellation	27
5.5	On-Behalf Order Management	27

Appendix A - Component Blocks 29

A.1 Instrument (symbology) Component Block.....29

A.2 Parties Component Block29

A.3 Triggering Instruction Component Block30

A.4 Peg Instruction Component Block30

A.5 SideCrossOrdModGrp Component Block31

A.6 TrdCapRptSideGrp Component Block31

A.7 StrategyParameters Component Block.....32

Appendix B - Field Enumerations Sorted By Tag Name 33

Appendix C - FIX Data Types..... 38

Figures

Figure 1 – Message Flow for Trade Capture Report for an IPO Security 25

Tables

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

Table 2 – FIX Infrastructure Messages Supported 7

Table 3 – FIX Client to X-stream FIX Server 7

Table 4 – X-stream FIX server to FIX Client 8

Table 5 - FIX Inbound Application Messages..... 9

Table 6 – FIX Outbound Application Messages 9

Table 7 – Logon..... 10

Table 8 – Logout..... 10

Table 9 – Reject 11

Table 10 – Resend Request..... 12

Table 11 – Sequence Reset 12

Table 12 – Test Request 13

Table 13 – Heartbeat 13

Table 14 – Business Message Reject 13

Table 15 – New Order Single..... 14

Table 16 – New Order Cross..... 15

Table 17 – Order Cancel Request..... 16

Table 18 – Order Cancel/Replace Request.....	17
Table 19 – Execution Report	20
Table 20 – Order Cancel Reject	22
Table 21 – Trade Capture Report	23
Table 22 – Trade Capture Report Ack	25
Table 23 – Instrument Component with Block Symbol and SecuritySubType	29
Table 24 – Instrument Component with X-stream Orderbook Identifier.....	29
Table 25 – Parties Component Block	30
Table 26 – TriggerInstruction Component Block.....	30
Table 27 – Peg Instruction Component Block.....	30
Table 28 – SideCrossOrdModGrp Component Block	31
Table 29 – TrdCapRptSideGrp Component Block	31
Table 30 – StrategyParametersGrp Component Block	32
Table 31 – Field Enumerations Sorted By Tag Name.....	33

1 Introduction to X-stream INET FIX

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

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

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

2 Session Information

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

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

X-stream FIX does not support encryption or compression.

2.1 FIX Admin and Infrastructure Messages Supported

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

Table 1 – FIX Admin Messages Supported

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

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

Table 2 – FIX Infrastructure Messages Supported

HEADING	MSGTYPE
Business Message Reject	j

2.2 SenderCompID and TargetCompID

FIX clients should send these tags in the message header.

Table 3 – FIX Client to X-stream FIX Server

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

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

Table 4 – X-stream FIX server to FIX Client

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

3 FIX Application Messages

X-stream FIX supports the following FIX protocol application messages for Order Management.

Table 5 - FIX Inbound Application Messages

MESSAGE NAME	MSGTYPE	COMMENTS
New Order Single	D	Used by participants to submit orders for execution.
Order Cancel Request	F	Request to cancel a live order.
Order Cancel / Replace Request	G	Request to amend or cancel a live order.
New Order Cross	s	Used by participants to submit cross orders for execution.
Trade Capture Report	AE	Used in IPO security report workflow.

Table 6 - FIX Outbound Application Messages

MESSAGE NAME	MSGTYPE	COMMENTS
Execution Report	8	Accept or reject for message D, F or G, order expiry, trade or restatement of overnight orders - if GTD or GTC orders supported.
Order Cancel Reject	9	Failure of message F or G.
Trade Capture Report	AE	Used in IPO security report workflow.
Trade Capture Report Ack	AR	Used to as an explicit acknowledgment of the corresponding Trade Capture Report message.

4 FIX Message Definitions

4.1 Session

4.1.1 Logon (A)

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

Table 7 – Logon

TAG	FIELD NAME	REQ'D	COMMENTS	FORMAT
Standard Header		Y	MsgType = A	
98	EncryptMethod	Y	(Always unencrypted)	Int
108	HeartBtInt	Y	Note same value used by both sides	Int
141	ResetSeqNumFlag	N	Ignored	Boolean
553	Username	Y	The FIX connector username	String
554	Password	Y	The FIX connector password. No security exists without transport level encryption.	String
1137	DefaultAppVerID	Y	Specifies the service pack release being applied by default to the message at the session level. The only valid value is '8' = FIX50SP1.	String
Standard Trailer		Y		

If client HeartBtInt is out of this range, the server will reply the default value (30) if it is the first logon of the day.

4.1.2 Logout (5)

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

The logout format is as follows.

Table 8 – Logout

TAG	FIELD NAME	REQ'D	COMMENTS	FORMAT
Standard Header		Y	MsgType = 5	
58	Text	N	Free format text string	String
Standard Trailer		Y		

4.1.3 Reject (3)

The reject message should be issued when a message is received but cannot be properly processed due to a session-level rule violation. An example of when a reject may be appropriate would be the receipt of a message with invalid basic data (e.g. MsgType=&) which successfully passes de-encryption,

Checksum and BodyLength checks. As a rule, messages should be forwarded to the trading application for business level rejections whenever possible.

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

The reject format is as follows.

Table 9 – Reject

TAG	FIELD NAME	REQ'D	COMMENTS	FORMAT
Standard Header		Y	MsgType = 3	
45	RefSeqNum	Y	MsgSeqNum of rejected message	SeqNum
371	RefTagID	N	The tag number of the FIX field being referenced.	Int
372	RefMsgType	N	The MsgType of the FIX message being referenced.	String
373	SessionRejectReason	N	Code to identify reason for a session-level Reject message. 0 – Invalid tag number 1 – Required tag missing 2 – Tag not defined for this message type 3 – Undefined tag 4 – Tag specified without a value 5 – Value is incorrect (out of range) for this tag 6 – Incorrect data format for value 9 – CompID problem 10 – SendingTime accuracy problem 11 – Invalid MsgType 13 – Tag appears more than once 14 – Tag specified out of required order 15 – Repeating group fields out of order 16 – Incorrect NumInGroup count for repeating group 18 – Invalid/unsupported application version 99 – Other. Refer to returned Text (58) field for exact reason for rejection	Int
58	Text	N	Free format text string	String
Standard Trailer		Y		

4.1.4 Resend Request (2)

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

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

The resend request format is as follows.

Table 10 – Resend Request

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

4.1.5 Sequence Reset (Gap Fill) (4)

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

Gap Fill mode

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

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

Reset mode

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

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

Table 11 – Sequence Reset

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

4.1.6 Test Request (1)

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

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

Table 12 – Test Request

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

4.1.7 Heartbeat (0)

Table 13 – Heartbeat

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

4.2 Infrastructure

4.2.1 Business Message Reject (j)

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

Table 14 – Business Message Reject

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
StandardHeader		Y	MsgType = j (lowercase)	
45	RefSeqNum	N	MsgSeqNum of rejected message	SeqNum
372	RefMsgType	Y	The MsgType of the FIX message being referenced.	String

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
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.	Int
58	Text	Y	Free format text string	String
Standard Trailer		Y		

4.3 Application

4.3.1 New Order Single (D)

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

Table 15 – New Order Single

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
StandardHeader		Y	MsgType = D	
11	ClOrdID	Y	Unique identifier for Order as assigned by the buy-side (institution, broker, intermediary etc.). Uniqueness must be guaranteed within a single trading day. Firms, particularly those which electronically submit multi-day orders, trade globally or throughout market close periods, should ensure uniqueness across days, for example by embedding a date within the ClOrdID field. Maximum length 20 characters.	String
Component <Instrument> block		Y	Insert here the set of "Instrument" (symbology) fields.	
1	Account	Y	Trading account type identifier associated with the order.	String
21	HandlInst	N	Used to indicate DMA. If unset, indicates that the order is not DMA.	Int
18	ExecInst	N	Instructions for order handling.	MultiChar
38	OrderQty	Y	Quantity ordered. This value represents the number of shares for equities or par, face or nominal value for Fixed Income instruments.	Qty
40	OrdType	Y	Indicates the type of order.	Char
44	Price	Y/N	Required for all limit order types – not required for Market orders.	Price

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
54	Side	Y	Side of the market	Char
60	TransactTime	Y	Time of order creation by Trader. This field is not processed by the Exchange nor is it used as a mechanism to place an order at a future time.	UTCTimeStamp
110	MinQty	Y/N	Specifies the minimum fill quantity. Note, for Whole or None orders this field must not be set.	Qty
59	TimeInForce	N	Indicates time in force techniques that are valid for the specified market segment. Absence of this field indicates default duration (5 mins for OPEL-B and 'day' for other boards).	Char
432	ExpireDate	Y/N	Conditionally required if TimeInForce = 'Good till Date/Time'. Not to be specified with ExpireTime.	LocalMktDate
126	ExpireTime	Y/N	Conditionally required if TimeInForce = 'Good till Date/Time'. Not to be specified with ExpireDate.	UTCTimeStamp
1138	DisplayQty	N	Specifies the disclosed volume on hidden/iceberg orders.	Qty
58	Text	N	Free Text. Maximum length 30 characters.	String
Component block <TriggeringInstruction>		N	Insert here the set of "TriggeringInstruction" fields.	
Component block <PegInstructions>		N	Insert here the set of "PegInstructions" fields.	
847	TargetStrategy	Y/N	Set to indicate whether Block order should be disseminated to the market.	Int
Standard Trailer		Y		

4.3.2 New Order Cross [s]

The New Order Cross type is used to submit a cross order into a market. The cross order contains two order sides (a buy and a sell). The cross order is identified by its CrossID (548).

Table 16 – New Order Cross

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
StandardHeader		Y	MsgType = s	
548	CrossID	Y	Identifier for a cross order. Must be unique during a given trading day. The maximum size is 20 characters.	String
549	CrossType	Y	Type of cross being submitted to a market.	Int
550	CrossPrioritization	Y	Indicates if one side or the other of a cross order should be prioritized.	Int

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
			0 = None	
	Component block <Instrument>	Y	Insert here the set of "Instrument" fields	
	Component block <SideCrossOrdModGrp>	Y	Insert here the set of "SideCrossOrderModGrp" fields	
40	OrdType	Y	Indicates the type of order. Must be 2 (Limit).	Char
44	Price	Y	Price of the cross.	Price
60	TransactTime	Y	Time of order creation by Trader. This field is not processed by the Exchange nor is it used as a mechanism to place an order at a future time.	UTCTimeStamp
847	TargetStrategy	N	The target strategy of the order	Int
	Component block <StrategyParametersGrp>	Y/N	Insert here the set of "StrategyParametersGroup" fields required if TargetStrategy = 1 VWAP OR 1000 TWAP	
	Standard Trailer	Y		

4.3.3 Order Cancel Request (F)

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

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

The format of the cancel request message is:

Table 17 – Order Cancel Request

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
	StandardHeader	Y	MsgType = F	
11	ClOrdID	Y	Unique identifier for Order as assigned by the buy-side (institution, broker, intermediary etc.) 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	String

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
			within the ClOrdID field. Maximum length 20 characters.	
37	OrderID	Y/N	Unique order identifier as assigned by X-stream that identifies the Order to be changed. Maximum length 18 characters.	String (All X-stream generated OrderIDs are integers. A non-integer value will result in a session-level reject due to invalid data format.)
41	OrigClOrdID	Y/N	ClOrdID(11) of the previous non-rejected order (NOT the initial order of the day) when cancelling or replacing an order. Required when referring to orders that were electronically submitted over FIX or otherwise assigned a ClOrdID. Maximum length 20 characters. Mandatory if OrderID (37) is not set.	String
Component <Instrument>		block Y	Insert here the set of "Instrument" (symbology) fields.	
54	Side	Y	Side of the market.	Char
60	TransactTime	Y	Time this order request was initiated. This field is not processed by the Exchange nor is it used as a mechanism to cancel an order at a future time.	UTCTimeStamp
Standard Trailer		Y		

4.3.4 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 18 – Order Cancel/Replace Request

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
StandardHeader		Y	MsgType = G	
11	ClOrdID	Y	Unique identifier for Order as assigned by the buy-side (institution, broker, intermediary etc.) (identified by SenderCompID (49)). Uniqueness must be guaranteed within a single trading day. Note that this identifier	String

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
			will be used in ClOrdID field of the Cancel Reject message if the replacement request is rejected. Maximum length 20 characters.	
37	OrderID	Y/N	Unique identifier of most recent order as assigned by the Exchange. Maximum length 18 characters.	String (All X-stream generated OrderIDs are integers. A non-integer value will result in a session-level reject due to invalid data format.)
41	OrigClOrdID	Y/N	ClOrdID(11) of the previous non-rejected order (NOT the initial order of the day) when cancelling or replacing an order. Required when referring to orders that were electronically submitted over FIX or otherwise assigned a ClOrdID. Maximum length 20 characters. Mandatory if OrderID (37) is not set.	String
Component block <Instrument>		Y	Insert here the set of "Instrument" (symbology) fields. Must match original order	
1	Account	Y	Trading account type identifier associated with the replacement order.	String
18	ExecInst	N	Instructions for order handling.	MultiChar
38	OrderQty	Y	Quantity ordered. This value represents the number of shares for equities or par, face or nominal value for Fixed Income instruments.	Qty
40	OrdType	Y	Indicates the type of order to change to (must follow rules of the Exchange).	Char
44	Price	Y/N	Required for all limit order types.	Price
54	Side	Y	Side of the market.	Char
60	TransactTime	Y	Time of execution/order creation. This field is not processed by the Exchange nor is it used as a mechanism to amend an order at a future time.	UTCTimeStamp
110	MinQty	N	Specifies the minimum fill quantity.	Qty

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
			Note, for Whole or None orders this field must not be set.	
59	TimeInForce	N	Indicates time in force techniques that are valid for the specified market segment. Absence of this field indicates default duration (5 mins for OPEL-B and 'day' for other boards).	Char
432	ExpireDate	Y/N	Conditionally required if TimeInForce = 'Good till Date/Time'. Not to be specified with ExpireTime.	LocalMktDate
126	ExpireTime	Y/N	Conditionally required if TimeInForce = 'Good till Date/Time'. Not to be specified with ExpireDate.	UTCTimeStamp
1138	DisplayQty	N	Specifies the disclosed volume on hidden/iceberg orders.	Qty
58	Text	N	Free Text. Maximum length 30 characters.	String
Component <TriggeringInstruction>	block	N	Insert here the set of "TriggeringInstruction" fields.	
Component <PegInstructions>	block	N	Insert here the set of "PegInstructions" fields.	
Standard Trailer		Y		

4.3.5 Execution Report (8)

The execution report message is used to:

1. Confirm the receipt of an order
2. Confirm changes to an existing order (i.e. accept cancel and replace requests)
3. Report order status information
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 cancelled.

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 19 – Execution Report

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
StandardHeader		Y	MsgType = 8	
11	CIOrdID	Y/N	Unique identifier for Order as assigned by the buy-side (institution, broker, intermediary etc.). 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 CIOrdID field. Required when referring to orders that were electronically submitted over FIX or otherwise assigned a CIOrdID(11).	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.	MultiChar
37	OrderID	Y	OrderID is required to be unique for each chain of orders.	String (All X-stream generated OrderIDs are integers.)
41	OrigCIOrdID	Y/N	Conditionally required for response to a Cancel or Cancel/Replace request	String
548	CrossID	N	Identifier for a cross order. Must be unique during a given trading day. The maximum size is 20 characters.	String (20)
150	ExecType	Y	Type of Execution being reported. Describes the specific ExecutionRpt (i.e. Pending Cancel) while OrdStatus (39) will always identify the current order status (i.e. Partially Filled) and Trade Amend so TradeCorrect = G	Char
Component block <Parties>		N	Insert here the set of "Parties" (firm identification) fields.	
Component block <Instrument>		Y	Insert here the set of "Instrument" (symbology) fields.	
1	Account	Y	Trading account type identifier.	String

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
14	CumQty	Y	Total matched quantity.	Qty
31	LastPx	N	Price of this fill.	Price
32	LastQty	N	Quantity (e.g. shares) bought/sold on this fill.	Qty
38	OrderQty	N	Quantity ordered.	Qty
110	MinQty	N	Minimum fill quantity.	Qty
39	OrdStatus	Y	Describes the current state of an order.	Char
40	OrdType	N	Order type	Char
44	Price	N	Price on order.	Price
54	Side	Y	Side of order.	Char
59	TimeInForce	N	Indicates time in force techniques that are valid for the specified market segment. Absence of this field indicates a 'day' order.	Char
60	TransactTime	Y	Time of execution/order creation (expressed in Universal Time Coordinated (UTC)).	UTCTimeStamp
75	TradeDate	N	Indicates date of trade referenced in this message in YYYYMMDD format.	LocalMktDate
432	ExpireDate	Y/N	Conditionally required if TimeInForce = 'Good till Date/Time'. Not to be specified with ExpireTime.	LocalMktDate
126	ExpireTime	Y/N	Conditionally required if TimeInForce = GTD or GTT. Not to be specified with ExpireDate.	UTCTimestamp
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
847	TargetStrategy	N	The target strategy of the order.	Int

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
	Component block <StrategyParametersGrp>	Y/N	Insert here the set of "StrategyParametersGroup" fields required if TargetStrategy = 1 VWAP	
880	TradeMatchID	N	Identifier assigned by the trading system for a trade. This is the X-stream trade id.	String
1057	AggressorIndicator	N	Used to identify whether the order initiator is an aggressor or not in the trade. Valid during continuous trading only.	Boolean
1138	DisplayQty	N	Replaces 'MaxFloor' and specifies the disclosed volume on hidden/iceberg orders. This field is always returned as part of a fill or partial fill for all order types. For non-hidden/iceberg orders this field will contain the same value as LeavesQty (151).	Qty
58	Text	N	Free Text. On an error condition, this will specify the X-stream generated error message.	String
	Component block <TriggeringInstruction>	N	Insert here the set of "TriggeringInstruction" fields.	
	Component Block <PegInstructions>	N	Insert here the set of "PegInstruction" fields.	
797	CopyMsgIndicator	N	Drop Copy	Boolean
	Standard Trailer	Y		

4.3.6 Order Cancel Reject (9)

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

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

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

The order cancel reject message format is as follows.

Table 20 – Order Cancel Reject

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
	StandardHeader	Y	MsgType = 9	

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
11	CIOrdID	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	OrigCIOrdID	Y/N	CIOrdID(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 CIOrdID.	String
60	TransactTime	Y	Time of order cancellation request rejection by the Exchange.	UTCTimeStamp
102	CxlRejReason	Y	Code to identify reason for cancel rejection. 1 – Unknown order 6 – Duplicate order (e.g. duplicate CIOrdID) 99 – Other. Refer to 'text' (58) for exact reason for rejection.	Int
434	CxlRejResponseTo	Y	Identifies the type of request that a Cancel Reject is in response to.	Char
58	Text	N	Specify X-stream generated error message.	String
Standard Trailer		Y		

4.3.7 Trade Capture Report [AE]

The Trade Capture report will be used to facilitate the IPO process.

All IPO securities will require a confirmation from the Lead Underwriter. The TradeCaptureReport [AE] message will be used as part of this IPO workflow.

Table 21 – Trade Capture Report

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
StandardHeader		Y	MsgType = AE	
571	TradeReportID	Y	Exchange: Unique identifier generated by X-stream for this Trade Capture Report chain. Subsequent messages in the chain refer to the original TradeReportID via the TradeReportRefID.	String

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
			Underwriter: Identifier for the inbound Trade Capture report. The TradeCaptureReportAck will return the value of this tag.	
17	ExecID	N	Exchange assigned Execution ID	String
487	TradeReportTransType	N	Identifies Trade Report message transaction type 0 = New 1 = Cancel	Int
856	TradeReportType	N	Type of Trade Report 2 = Accept 3 = Decline	Int
572	TradeReportRefID	Y/N	The TradeReportID that is being referenced for some action, such as confirmation or cancellation.	String
880	TrdMatchID	N	Identifier assigned by the trading system for a trade.	String
Component block <Instrument>		Y	Insert here the set of "Instrument" (symbology) fields.	
31	LastPx	Y	Trade Price.	Price
32	LastQty	Y	Trade Quantity	Qty
60	TransactTime	Y/N	Time the transaction represented by this Trade Capture Report occurred	UTCTime Stamp
Component block <TrdCapRptSideGrp>		Y	Insert here the set of "TrdCapRptSideGrp" fields.	
1011	MessageEventSource	N	Used to identify the IPO type that gave rise to this message. Valid values are: IP – IPO Cross BO – Buy Offering SB – Subscription GS – Greenshoe FO – Follow On CG – Greenshoe Confirmation	String
797	CopyMsgIndicator	N	Drop Copy	Boolean
StandardTrailer		Y		

4.3.8 Trade Capture Report Ack [AR]

The Trade Capture Report Ack will be used by the Exchange to acknowledge or reject a Trade Capture Report from a counterparty.

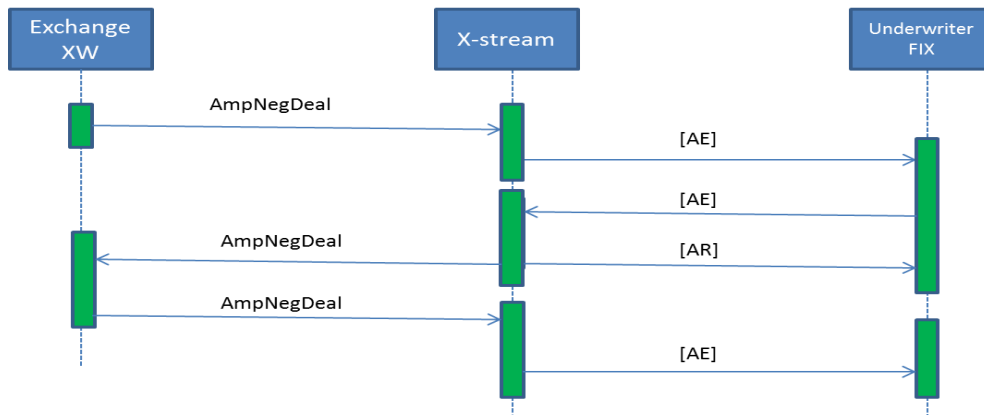
Table 22 – Trade Capture Report Ack

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
	StandardHeader	Y	MsgType = AR	
571	TradeReportID	Y	Identifier for the Trade Capture Report	String
487	TradeReportTransType	N	Identifies Trade Report message transaction type 0 = New 1 = Cancel	Int
856	TradeReportType	N	2 = Accept 3 = Decline	Int
939	TrdRptStatus	N	0 = Accepted 1 = Rejected	Int
17	ExecID	N	Exchanged assigned Execution ID (Trade Identifier)	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 99 – Other. Refer to returned Text (58) field for exact reason for rejection.	Int
572	TradeReportRefID	N	The TradeReportID that is being referenced for some action, such as confirmation or cancellation	String
58	Text	N	If TradeReportRejectReason is set, text of reason	String
	StandardTrailer	Y		

4.3.9 Trade Capture Report Work Flow for IPO Cross

The reporting and confirmation of an IPO cross will have the following message flow where the underwriter participant uses FIX.

Figure 1 – Message Flow for Trade Capture Report for an IPO Security



The process is initialised via the Exchange XW. The Underwriter will receive a TradeCaptureReport [AE] message to indicate a new IPO cross requires confirmation.

A TradeCaptureReport [AE] is sent by the FIX participant to the Exchange to accept or reject the IPO cross. A TradeCaptureReportAck will be received to indicate receipt of the inbound TradeCaptureReport.

The final confirmation of an IPO cross, using the Exchange XW, then results in the last TradeCaptureReport as confirmation the acceptance of IPO. This will contain the unique trade identifier in the TradeMatchID (880) tag.

5 Order Management

5.1 Unique ClOrdID (11)

X-stream will only check for uniqueness of ClOrdID(11) on New Order Single, Order Cancel/Replace Request and Order Cancel Request messages for open or traded orders. If a firm has multiple FIX connections, then ClOrdID(11) should be unique across all FIX connections for that firm.

5.2 Order Identification

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

OrderID(37) should be used to identify an order between FIX connections, even if they belong to the same firm. The X-stream (Exchange) OrderID is guaranteed to be unique for all order durations including over-night orders.

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

Note: OrderID (37) will likely change after order amendment.

5.3 Order Modification via Order Cancel/Replace Request

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

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

5.4 Order Cancellation

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

5.5 On-Behalf Order Management

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

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

Generally X-stream INET FIX connections operate in 'on-behalf of' mode.

A FIX order message with SenderSubID (50) will send two usernames to the X-stream backend - OperatorId and UserId. X-stream first checks that the OperatorId, the owner of the FIX session, has

permission to enter messages 'on-behalf' of the UserId from the SenderSubID (50). The transaction is then processed with the permissions of UserId.

Appendix A - Component Blocks

A.1 Instrument (symbology) Component Block

The Instrument component block contains all the fields commonly used to describe a security or instrument. The Instrument component block can be used to describe any asset type supported by FIX.

Values to populate the Instrument component should be sourced from the ITCH Total View reference data spin.

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

- Symbol (55)
- SecuritySubType (762)

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

Table 23 – Instrument Component with Block Symbol and SecuritySubType

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
55	Symbol	Y	Marketplace identifier for a security.	String
762	SecuritySubType	Y/N	In X-stream, this field is used to specify board on which Symbol is listed. Default is OPEL-E.	String

Table 24 – Instrument Component with X-stream Orderbook Identifier

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
48	SecurityID	Y	Unique marketplace assigned integer identifier for an order book. This provides a fast lookup for the orderbook.	String
22	SecurityIDSource	Y	M = Marketplace assigned identifier	Char

A.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:

Table 25 – 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

A.3 Triggering Instruction Component Block

The TriggeringInstruction component block specifies the conditions under which an order will be triggered by related market events as well as the behavior of the order in the market once it is triggered. Only market and limit orders are supported with a Triggering Instruction Component.

Table 26 – TriggerInstruction Component Block

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
1100	TriggerType	Y	Defines when the trigger will hit, i.e. the action specified by the trigger instructions will come into effect. 1 = Partial Execution	Char
1102	TriggerPrice	Y	The price at which to activate the trigger order.	Price
1107	TriggerPriceType	Y	The type of price that the trigger is compared to. 2 = Last Trade	Char

A.4 Peg Instruction Component Block

The PegInstruction component block specifies the behaviour of a pegged order. ExecInst (18) is used to indicate whether the pegged order is to be passive or active.

Table 27 – Peg Instruction Component Block

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
211	PegOffsetValue	Y	Amount added to the peg for a pegged order in the context of the PegOffsetType	PriceOffset
836	PegOffsetType	Y	Type of Peg Offset Value 0 = Price	Int

A.5 SideCrossOrdModGrp Component Block

The SideCrossOrdModGrp component block gives the two sides for the NewOrderCross message.

Table 28 – SideCrossOrdModGrp Component Block

TAG	FIELDNAME		REQ'D	COMMENTS	FORMAT
552	NoSides		Y	Must be set to 2	NumInGrp
→	54	Side	Y	Side of the market	Char
→	11	CIOrdID	Y	Unique identifier for Order as assigned by the buy-side (institution, broker, intermediary etc.). 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 CIOrdID field. Maximum length 20 characters.	String
→	Component block <Parties>		N	Insert here the set of "Parties" (symbology) fields.	
→	1	Account	Y	Trading account type identifier associated with the order.	String
→	38	OrderQty	Y	Quantity ordered. This value represents the number of shares for equities or par, face or nominal value for Fixed Income instruments.	Qty
→	58	Text	N	Free text field. Maximum length 30 characters.	String

A.6 TrdCapRptSideGrp Component Block

The TrdCapRptSideGrp component block contains two Parties block for each side.

Table 29 – TrdCapRptSideGrp Component Block

TAG	FIELDNAME		REQ'D	COMMENTS	FORMAT
552	NoSides		Y	Number of sides. Should be 2.	NumInGrp
→	54	Side	Y	Side of order.	Char
→	Component block <Parties>		Y	Insert here the set of "Instrument" (symbology) fields.	

A.7 StrategyParameters Component Block

The StrategyParametersGrp component block contains two StrategyParameter blocks representing the start and end time of the Exceptional Cross order type VWAP calculation.

Table 30 – StrategyParametersGrp Component Block

TAG	FIELDNAME		REQ'D	COMMENTS	FORMAT
957	NoStrategyParameters		Y	Number of strategy parameters. Should be 2.	NumInGrp
→	958	StrategyParameterName	Y	Name of parameter. "S"tart of VWAP calculation strategy parameter "E"nd of VWAP calculation strategy parameter	String
→	959	StrategyParameterType	Y	19 = UTCTimestamp	Int
→	960	StrategyParameterValue	Y	The UTCTimestamp VWAP parameter	String

Appendix B - Field Enumerations Sorted By Tag Name

Table 31 – Field Enumerations Sorted By Tag Name

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
1	Account	Y	'C' – Client 'H' – House 'O' – Other 'Y' – Strategy 'M' – Market Maker 'S' – Stabilisation	String
1057	AggressorIndicator	N	Used to identify whether the order initiator is an aggressor or not in the trade. Valid values: Y – Order initiator is aggressor N – Order initiator is passive	Boolean
380	BusinessRejectReason	Y	Valid values: 0 – Other 1 – Unknown ID 2 – Unknown Security 3 – Unknown Message Type 4 – Application not available 5 – Conditionally required field missing 6 – Not Authorized	Int
550	CrossPrioritization	Y	Indicates if one side or the other of a cross order should be prioritized. 0 – None	Int
549	CrossType	Y	Used to indicate behaviour of a NewOrderCross. 1 - Cross AON – cross trade is executed completely or not. 4 – Cross Same Price – cross trade is executed with existing orders at the same price.	Int
102	CxlRejReason	N	Identifies the reason for the cancel rejection. Valid values: 1 – Unknown order 6 – Duplicate order (e.g. duplicate ClOrdID) 99 – Other. Refer to returned Text (58) field for exact reason for rejection.	Int
434	CxlRejResponseTo	Y	Identifies the type of request that a Cancel Reject is in response to. Valid values are: 1 – Order Cancel Request 2 – Order Cancel/Replace Request	Char
98	EncryptMethod	Y	0 – None/Other	Int

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
18	ExecInst	N	Instructions for order handling. Valid values: 'o' – Withdraw on logoff or connection loss. 'A' – No cross, used for passive pegged order 'B' – OK to Cross, used for active pegged order	MultiChar
150	ExecType	Y	Type of Execution being reported. Describes the specific ExecutionRpt (i.e. Pending Cancel) while OrdStatus (39) will always identify the current order status (i.e. Partially Filled) Valid values: 0 – New 3 – Done for day 4 – Cancelled 5 – Replaced 6 – Pending Cancel (e.g. result of Order Cancel Request) 7 – Stopped 8 – Rejected C – Expired F – Trade (partial fill or fill) H – Trade Cancel	Char
21	HandlInst	N	1 – Automated execution order, no Broker intervention. Used to indicate DMA. 2 – Automated execution order, Broker intervention OK.	Int
1011	MessageEventSource	N	Used to indicate the IPO type when reporting the IPO cross. Valid values are: IP – IPO Cross BO – Buy Offering SB – Subscription GS – Greenshoe FO – Follow On CG – Greenshoe Confirmation	String
103	OrdRejReason	N	For optional use with ExecType = 8 (Rejected). Code to identify reason for order rejection. Valid values are: 1 – Unknown symbol 5 – Unknown order 6 – Duplicate order (e.g. duplicate CLOrdID) 11 – Unsupported order characteristic 15 – Unknown account(s) 99 – Other. Refer to returned Text (58) field for exact reason for rejection.	Int
39	OrdStatus	Y	Describes the current state of an order. Valid values are: 0 – New 1 – Partially filled	Char

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
			2 – Filled 4 – Cancelled 5 – Replaced 8 – Rejected C – Expired *** NASDAQ OMX Defined *** X – Order with trigger in the book but not active (e.g. Order has not been triggered).	
40	OrdType	Y	Indicates the type of order. Valid values are: 1 – Market – The Price (44) field is not used, the order executes against the best prices order on the opposite side. 2 – Limit – The Price (44) field is specified and the order will execute at this price or better. P – Pegged	Char
447	PartyIDSource	N	Used to identify class source of PartyID value. Required if PartyID is specified. Required if NoPartyIDs > 0. Valid values are: C – Participant identifier	Char
452	PartyRole	N	Identifies the type of PartyID (e.g. Executing Broker). Required if NoPartyIDs > 0. Valid values are: 1 – Executing Firm 3 – Client ID 4 – Clearing Firm 5 – Investor ID (Deposit Account) 6 – Introducing Firm 7 – Entering Firm 11 – Order origination trader 12 – Executing Trader 13 – Order origination firm 17 – Contra Firm 36 – Entering trader 37 – Contra trader 44 – Order Entry Operator ID	Int
836	PegOffsetType	N	Type of Peg Offset value. 0 – Price	Int
22	SecurityIDSource	N	Identifies class or source of the SecurityID value. Required if SecurityID is specified. M – Marketplace assigned identifier	Char
373	SessionRejectReason	N	Code to identify reason for a session-level Reject message.	Int

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
			0 – Invalid tag number 1 – Required tag missing 2 – Tag not defined for this message type 3 – Undefined tag 4 – Tag specified without a value 5 – Value is incorrect (out of range) for this tag 6 – Incorrect data format for value 9 – CompID problem 10 – SendingTime accuracy problem 11 – Invalid MsgType 13 – Tag appears more than once 14 – Tag specified out of required order 15 – Repeating group fields out of order 16 – Incorrect NumInGroup count for repeating group 18 – Invalid/unsupported application version 99 – Other. Refer to returned Text (58) field for exact reason for rejection	
54	Side	Y	Optional qualifier used to indicate the side of the market. Valid values are: 1 – Buy 2 – Sell 5 – Short Sell	Char
847	TargetStrategy	N	1 – VWAP Set to indicate an Exceptional Cross order type. The StrategyParameterGrp must be set to indicate the start and end times for the VWAP calculation. 2 – Participate Set to indicate whether an order on the Block book should participate in the calculation of market statistics. The default on the Block board is not to participate so must be set explicitly. 1000 – TWAP Set to indicate an Exceptional Cross order type. The StrategyParameterGrp must be set to indicate the start and end times for the TWAP calculation.	Int
59	TimeInForce	N	Indicates time in force techniques that are valid for the specified market segment. Valid values are: 0 – Day 1 – Good till cancelled 3 – Immediate or Cancel (IOC)	Char

TAG	FIELDNAME	REQ'D	COMMENTS	FORMAT
			4 – Fill or Kill (FoK) 6 – Good till date (GTD) *** NASDAQ OMX Defined *** S – Session	
751	TradeReportRejectReason	N	Reason for rejection of Trade Report. 99 – Other. Refer to returned Text (58) field for exact reason for rejection.	Int
487	TradeReportTransType	N	Identifies Trade Report message transaction type. 0 – New 1 – Cancel	Int
856	TradeReportType	N	Type of Trade Report. 2 – Accept 3 – Decline	Int
939	TrdRptStatus	N	Status of Trade Report. 0 – Accepted 1 – Rejected	Int
1107	TriggerPriceType	Y	The type of price that the trigger is compared to. 2 – Last Trade	Char
1100	TriggerType	Y	Defines when the trigger will hit, i.e. the action specified by the trigger instructions will come into effect. 1 – Partial Execution	Char

Appendix C - FIX Data Types

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

int	<p>Sequence of digits without commas or decimals and optional sign character (ASCII characters "-", "0" - "9"). The sign character utilizes one byte (i.e. positive int is "99999" while negative int is "-99999"). Note that int values may contain leading zeros (e.g. "00023" = "23").</p> <p>Examples:</p> <p>723 in field 21 would be mapped int as 21=723 .</p> <p>-723 in field 12 would be mapped int as 12=-723 </p> <p>The following data types are based on int.</p>	
	Length	int field representing the length in bytes. Value must be positive.
	TagNum	int field representing a field's tag number when using FIX "Tag=Value" syntax. Value must be positive and may not contain leading zeros.
	SeqNum	int field representing a message sequence number. Value must be positive.
	NumInGrp	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	Single character value, can include any alphanumeric character or punctuation except the delimiter. All char fields are case sensitive (i.e. m != M).	

	The following fields are based on char.	
	Boolean	char field containing one of two values: 'Y' = True/Yes 'N' = False/No
String	Alpha-numeric free format strings, can include any character or punctuation except the delimiter. All String fields are case sensitive (i.e. morstatt != Morstatt).	
	MultipleCharValue	string field containing one or more space delimited single character values (e.g. 18=2 A F).
	MultipleStringValue	string field containing one or more space delimited multiple character values (e.g. 277=AV AN A).
	Country	string field representing a country using ISO 3166 Country code (2 character) values (see FIX Specifications Volume 6 - Appendix 6-B).
	Currency	string field representing a currency type using ISO 4217 Currency code (3 character) values (see FIX Specifications Volume 6 - Appendix 6-A).
	Exchange	string field representing a market or exchange using ISO 10383 Market Identifier Code (MIC) values (see FIX Specifications Volume 6 - Appendix 6-C).
	MonthYear	string field representing month of a year. An optional day of the month can be appended or an optional week code. Valid formats: YYYYMM YYYYMMDD YYYYMMWW Valid values: YYYY = 0000-9999; MM = 01-12; DD = 01-31; WW = w1, w2, w3, w4, w5.
UTCTimestamp	string field representing Time/date combination represented in UTC (Universal Time Coordinated, also known as "GMT") in either YYYYMMDD-HH:MM:SS (whole seconds), YYYYMMDD-HH:MM:SS.sss (milliseconds), or YYYYMMDD-HH:MM:SS.ssssss (microseconds) 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). YYYY = 0000-9999, MM = 01-12, DD = 01-31, HH = 00-23, MM = 00-59, SS = 00-60 (60 only if UTC leap second), ssssss=000000-999999 (indicating microseconds). 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	

	<p>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)</p>
UTCTimeOnly	<p>string field representing Time-only represented in UTC (Universal Time Coordinated, also known as "GMT") in either HH:MM:SS (whole seconds), HH:MM:SS.sss (milliseconds), or HH:MM:SS.ssssss (microseconds) format. Colons and period required. This 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> <p>HH = 00-23, MM = 00-59, SS = 00-60 (60 only if UTC leap second), ssssss=000000-999999 (indicating microseconds).</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>