

# CHI-X EUROPE

## Chi-X FIX 4.2 Application Notes

March 11<sup>th</sup>, 2011  
Version: 1.27

©2011 Chi-X Europe Ltd ("Chi-X"). All rights reserved. Chi-X is a registered mark. Approved for distribution in Europe by Chi-X Europe Limited which is authorised and regulated by the Financial Services Authority.

This information is provided for informational purposes only. It does not take into account the particular investment objectives, financial situation, or needs of any individual or entity. Under no circumstances is it to be used or considered as an offer to purchase or sell any security, or as a solicitation or recommendation of the purchase, sale, or offer to purchase or sell any security. While the information has been obtained from sources deemed reliable, neither Chi-X nor its licensors, nor any other party through whom the user obtains any such information: (i) makes any guarantees that it is accurate, complete, timely, or contains correct sequencing of information; (ii) makes any warranties with regard to the results obtained from its use; or (iii) shall have any liability for any claims, losses, or damages arising from or occasioned by any inaccuracy, error, delay, or omission, or from the use of the information or actions taken in reliance on the information. Reproduction or redistribution of this information is prohibited except with written permission from Chi-X.

System response times may vary for a number of reasons including market conditions, trading volumes and system performance.

## Table of Contents

<b>1</b>	<b>INTRODUCTION</b>	<b>4</b>
1.1	DOCUMENTATION METHODOLOGY	5
1.2	VERSION HISTORY	5
<b>2</b>	<b>CHI-X'S FIX INTERFACE</b>	<b>7</b>
<b>3</b>	<b>CONFIGURATION INFORMATION</b>	<b>8</b>
3.1	CLIENT	8
3.1.1	<i>SenderCompID</i>	8
3.1.2	<i>TargetCompID</i>	8
3.1.3	<i>SenderSubID (50) / OnBehalfOfCompId (115)</i>	8
3.1.4	<i>Encryption</i>	8
3.2	CHI-X CONFIGURATION	8
3.2.1	<i>SenderCompID</i>	8
3.2.2	<i>TargetCompID</i>	8
3.2.3	<i>IP Addresses</i>	8
<b>4</b>	<b>SESSION MANAGEMENT</b>	<b>9</b>
4.1	MESSAGE HEADER FORMAT TO CHI-X	9
4.2	MESSAGE HEADER FORMAT TO CLIENT	9
4.3	MESSAGE TRAILER FORMAT	10
4.4	LOGON	10
4.4.1	<i>Client logon</i>	10
4.4.2	<i>Chi-X logon</i>	10
4.5	ADMINISTRATIVE MESSAGES	11
4.6	LOGOUT	11
4.6.1	<i>Client</i>	11
4.6.2	<i>Chi-X Nightly Maintenance</i>	11
4.7	REJECT MESSAGES	11
4.8	RECOVERY	11
<b>5</b>	<b>APPLICATION MESSAGES</b>	<b>13</b>
5.1	SYMBOLOGIES	13
5.1.1	<i>Client</i>	13
5.2	ORDER ENTRY	14
5.2.1	<i>Client</i>	14
5.2.1.1	<i>New Order Single</i>	14
5.2.1.2	<i>Order Cancel Request</i>	17
5.2.1.3	<i>Order Cancel/Replace Request</i>	17
5.2.2	<i>Chi-X Order Entry Messages</i>	18
5.2.2.1	<i>New Order Single Response</i>	18
5.2.2.2	<i>Order Cancel Acknowledgment</i>	20
5.2.2.3	<i>Order Cancel Reject</i>	21
5.2.2.4	<i>Cancel/Replace Response</i>	21
5.2.2.5	<i>Trade Report</i>	22
5.2.2.6	<i>Unsupported FIX Messages</i>	25
<b>6</b>	<b>APPENDIX A – EXAMPLE ORDER FLOW MATRICES</b>	<b>26</b>
<b>7</b>	<b>APPENDIX B – CHI-X PEG ORDER DEFINITIONS</b>	<b>34</b>
	PRIMARY (PRIM) PEG TYPE	34
	<i>Basic Primary Peg</i>	34
	<i>Primary Peg with Increments</i>	34
	MID (MID) PEG TYPE	35

<i>Basic Mid Peg</i> .....	35
<i>Mid Peg with Increments</i> .....	35
MARKET (MKT) PEG TYPE .....	36
<i>Basic Market Peg</i> .....	36
<i>Market Peg with Increments</i> .....	36
<i>Summary of the Effects of PegDifference on Order Prices.</i> .....	37
<b>8 APPENDIX C: HIDDEN ORDER PARAMETERS</b> .....	<b>38</b>
8.1 FUNCTIONALITY MATRIX .....	38
8.2 SUPPORTED NON-DISPLAYED ORDER TYPES/ATTRIBUTES.....	39
<b>9 APPENDIX D: CHI-VISION FUNCTIONALITY</b> .....	<b>40</b>
9.1 SUPPORTED ROUTING STRATEGIES .....	40
9.2 SUPPORTED ORDER TYPES/ATTRIBUTES.....	40
9.3 MARKET DATA AND USE OF TRADE LIQUIDITY INDICATOR .....	40

## 1 INTRODUCTION

This document describes specifically what functionality is available through Chi-X's version FIX 4.2-based interface. It also describes how Chi-X uses the FIX protocol to achieve this functionality.

This document is intended to supplement the FIX Protocol Specification ([www.fixprotocol.org](http://www.fixprotocol.org)), by describing:

- Where there are multiple ways to achieve a desired outcome with the protocol, this document describes which one(s) Chi-X supports.
- Where the protocol does not define the exact meaning or content of various fields, this document provides as much detail as possible to describe Chi-X's chosen implementation.
- Where there are possible alternative interpretations, this document describes which interpretation Chi-X has selected.

### **What is FIX?**

FIX Connectivity enables the easy integration of the Chi-X trading system into your workflow and makes it easier for you to trade by reducing your keystrokes. FIX stands for the Financial Information eXchange Protocol. The FIX protocol is a 'language' created by a group of institutional clients and brokers to standardize the delivery of relevant pre-trade and trade information. It is a public-domain specification owned and maintained by the FIX Protocol Organization. Chi-X offers many options for you to easily integrate Chi-X into your workflow using a FIX connection.

### **Send orders to Chi-X directly from your trade blotter**

With a FIX connection between your system and Chi-X, you can easily send orders to Chi-X automatically from your trade blotter, providing instant access to European trading opportunities and liquidity.

## 1.1 Documentation Methodology

FIX messages and fields are reprinted in this document, only if their content or usage differs from the original FIX specification ([www.fixprotocol.org](http://www.fixprotocol.org)). **FIX messages** are quoted in **boldface** (e.g. **Resend Request, Logon**); *FIX fields* are quoted in *italics* (e.g. *PossResend, SenderSubID*).

## 1.2 Version History

Date	Version	Changes
06/12/2006	1.0	Initial draft.
26/01/2007	1.01	Clarification to message recovery.
13/03/2007	1.02	Addition of account field.
18/07/2007	1.03	Removed draft status, removed reference to short sells, added "Principal" capacity.
13/11/2007	1.04	Added description of new system behaviour when sessions disconnect.
13/12/2007	1.05	Removed references to auctions and trade corrections.
11/01/2008	1.06	Clarified use of <i>ExecInst</i> tag in New Order Messages.
27/02/2008	1.07	Removed reference to list IDs and clarified comments against <i>PossDup</i> tag.
06/03/2008	1.08	Added <i>TransactTime</i> to New Order Single Response.
26/03/2008	1.09	Removed <i>OrdRejReason</i> from Section 5.2.2.1.
16/06/2008	1.10	Added <i>OrigClOrdID</i> to Section 5.2.2.2. Amended description of <i>ClOrdID</i> in Section 5.2.2.2 and Section 5.2.2.5. Added <i>LastCapacity</i> to Section 5.2.2.5.
25/07/2008	1.11	Added <i>ClientID</i> to Section 5.2.2.1 and Section 5.2.2.5.
14/08/2008	1.12	Added description for hidden order behaviour.
14/10/2008	1.13	Clarified use of <i>OrigClOrdID</i> in Section 5.2.2.2.
12/11/2008	1.14	Removed SEDOL support from document.
25/11/2008	1.15	Correct Appendix A use case 16.
02/03/2009	1.16	Added <i>Text</i> tag to section 5.2.2.1. Removal of Reference to <i>ExecTransType=2</i> in Section 5.2.2.5. and also in use case 16 in Appendix A.
06/03/2009	1.17	Removed reference to <i>CxlRejReason</i> in use cases. Corrected values in use cases 5, 6
15/04/2009	1.18	Added References to UMTF code. Added <i>TargetSubID</i> to Section 4.1. Added <i>MaxFloor</i> to sections 5.2.2.1, 5.2.2.2, 5.2.2.3, 5.2.2.4 and 5.2.2.5. Added new values for <i>TradeLiquidityIndicator</i> . Added <i>MinQty</i> tag to document and added Appendix C.
20/04/2009	1.19	Added final scenario in Section 8.1. Amended use case 4.
06/05/2009	1.20	Amended Section 8.1 to reflect changes to non-pegged hidden orders

		and added scenarios.
19/05/2009	1.21	Amended Section 8.1 and 8.2 to remove reference to Market and Mid Pegs.
08/06/2009	1.22	Added Appendix D.
24/11/2009	1.23	Amended Appendix D to reflect correct values of <i>TradeLiquidityIndicator</i> .
22/07/2010	1.24	Added character length limitation to Section 3.1.3.
18/10/2010	1.25	Changed Description of <i>OrderQty</i> 5.2.1.3 to reflect new support for revise up of open quantity in the integrated orderbook (effective 22 <sup>nd</sup> November 2010).
13/12/2010	1.26	Added new values to <i>TradeLiquidityIndicator</i> in Section 5.2.2.5.
11/03/2011	1.27	Amended description of <i>Contrabroker</i> in Section 5.2.2.5.

## 2 CHI-X'S FIX INTERFACE

Since Chi-X is not a multi-day trading system, it does not support multi-day orders. Prior to the system being brought down for nightly maintenance, all orders residing on the system will be cancelled, and FIX sequence numbers will be reset. When the system comes back up, it is considered the next trading day.

The following is the schedule for nightly maintenance:

- Monday through Friday 17:50 UK time. After this time the Chi-X FIX application will not be listening for connections again until 06:00 the following day.

Please also note that nightly maintenance will occur during Holidays.

Upon request, our Chi-X Support staff will provide you with access to the FIX test server. An appointment is required for certification purposes.

The support staff can be reached at +44 (0) 207 131 3300, Monday through Friday, from 6.30am to 6.00pm UK time.

## 3 CONFIGURATION INFORMATION

### 3.1 Client

#### 3.1.1 SenderCompID

Clients must identify the session in the *SenderCompID* (49) field. Chi-X must approve this value. Chi-X's software is case sensitive and the maximum size of the *SenderCompID* (49) field is **32** characters.

#### 3.1.2 TargetCompID

*TargetCompID* must identify Chi-X as the receiving firm and must begin with a prefix of "CHIX".

#### 3.1.3 SenderSubID (50) / OnBehalfOfCompId (115)

*SenderSubID* (50) or *OnBehalfOfCompId* (115) can be used to identify the trader who entered the order. The client may link a trader identifier to a specific Chi-X account (for billing purposes) by having Chi-X pre-configure one of these fields *SenderSubID* (50) for this purpose. Only one of *SenderSubId* and *OnBehalfOfCompId* can be used, if both are sent on an order, the order will be rejected. The maximum size of the *SenderSubID* (50) and *OnBehalfOfCompId* (115) field is **32** characters.

#### 3.1.4 Encryption

Chi-X does not support encryption of FIX messages.

### 3.2 Chi-X Configuration

#### 3.2.1 SenderCompID

The value the client will receive in the *SenderCompID* field from Chi-X will be the value originally supplied to Chi-X in the *TargetCompID* field in the logon message.

#### 3.2.2 TargetCompID

The value the client will receive in the *TargetCompID* field from Chi-X will be the value originally supplied to Chi-X in the *SenderCompID* field in the logon message.

#### 3.2.3 IP Addresses

Chi-X will provide clients with the following:

- Two production IP-addresses; and
- One or more test IP-addresses.

## 4 SESSION MANAGEMENT

This section describes session-level FIX messages sent between Chi-X and the client.

The production servers consist of a primary and a standby server. If the session to the primary server fails, a client should retry this connection after 30 seconds. If failover to the standby server is required, clients will be contacted by Chi-X support. Once a session is reestablished, Chi-X will immediately begin sending execution reports that were not successfully delivered in the previous FIX session, for the same trade date.

### 4.1 Message header format to Chi-X

Chi-X processes only the following fields in the message header and ignores all others:

Tag	Field Name	Req'd	Comments
8	BeginString	Y	FIX.4.2
9	BodyLength	Y	Must be the second field in the message.
34	MsgSeqNum	Y	See standard FIX explanation.
35	MsgType	Y	Must be the third field in the message.
43	PossDupFlag	N	Always required for retransmissions.
49	SenderCompID	Y	The value used must be recognized and agreed to by Chi-X.
50	SenderSubID	N	Identity of the trader who entered the order. This can be used to indicate a specific Chi-X account for billing purposes. Any SenderSubID values must be pre-configured by Chi-X. Any unrecognized SenderSubID value will be allocated against a common pre-assigned account for that client.
52	SendingTime	Y	Indicates the time the message was sent by the client.
57	TargetSubID	N	See Appendix D.
56	TargetCompID	Y	<b>Must always have a prefix of "CHIX"</b> . Example, "CHIX_01", "CHIX_02"
97	PossResend	N	Required when message may be duplicate of another message sent under a different sequence number.

### 4.2 Message header format to Client

Chi-X processes only the following fields in the message header and ignores all others:

Tag	Field Name	Req'd	Comments
8	BeginString	Y	FIX.4.2
9	BodyLength	Y	Will be the second field in the message.
34	MsgSeqNum	Y	See standard FIX explanation.
35	MsgType	Y	Will be the third field in the message.
43	PossDupFlag	N	Always required for retransmissions.
49	SenderCompID	Y	The value originally supplied to Chi-X in the <i>TargetCompID</i> field in the logon message from the Client.

Tag	Field Name	Req'd	Comments
50	SenderSubID	N	The value originally supplied to Chi-X in the <i>TargetSubID</i> field in related application messages from the Client, if specified..
52	SendingTime	Y	Indicates the time the message was sent by Chi-X.
56	TargetCompID	Y	The value originally supplied to Chi-X in the <i>SenderCompID</i> field in the logon message from the Client.
57	TargetSubID	N	The value originally supplied to Chi-X in the <i>SenderSubID</i> field in related application messages from the Client, if specified.
97	PossResend	N	Required when message may be duplicate of another message sent under a different sequence number.

### 4.3 Message trailer format

Chi-X processes only the following fields in the message trailer and ignores all others:

Tag	Field Name	Req'd	Comments
10	Checksum	Y	(Always unencrypted, always last field in message)

## 4.4 Logon

### 4.4.1 Client logon

The first expected message Chi-X will receive from a client is a **Logon** message. The following are the logon parameters:

- The sequence number, on the initial logon for each trading day, must be set to “1”.
- The heartbeat interval must be greater than zero.
- The *SenderCompID* (49) must be recognized by Chi-X (see section titled [Configuration Information](#)).
- The client must set the *TargetCompID* (56) with a prefix of “CHIX”.
- **If a client receives a sequence number less than expected, the client must terminate their session immediately, and should then contact Chi-X to correct the problem, as per the FIX protocol.**

### 4.4.2 Chi-X logon

Once Chi-X receives a **Logon** request, it will validate the *SenderCompID* and perform a recovery process (see section titled *Recovery*). **No** messages should be sent to Chi-X until a **Logon** message is received in reply from Chi-X.

In some cases, some time will elapse before a response is sent from Chi-X. Once the positive response is returned, the client’s heartbeat interval timer should begin. The session is signed on and both parties can begin exchanging messages. The negative response to a **Logon** request is a **Logout** message with the reason for rejection.

## 4.5 Administrative messages

This section describes the minimum requirements to keep the session alive and synchronized.

Chi-X must receive a message from the client at least **once** in the heartbeat interval defined in the logon.

Chi-X will send a message at least once in the heartbeat interval. In addition, Chi-X handles the following session level messages: **Resend Request**, **Sequence Reset** and **Test Request** messages and ignores the *OrigSendingTime (122)* in all messages types.

## 4.6 Logout

This section concerns normal and abnormal termination of a session by either party.

### 4.6.1 Client

A Client's FIX session should remain established throughout the trading day. Chi-X will logout client sessions before the start of the maintenance window. Abnormal session termination is treated as though the client had logged out from Chi-X. The following are considered abnormal session terminations:

- Network level disconnection
- Failure to send a message after two heartbeat intervals (see section titled "[Administrative Messages](#)")

### 4.6.2 Chi-X Nightly Maintenance

When Chi-X shuts down for nightly maintenance, any live sessions will be terminated. All remaining open orders will be cancelled and a new trade date will begin. No **Execution Reports** will be sent for orders sent on previous trading days.

## 4.7 Reject messages

**Reject** messages sent by Chi-X will include the sequence number of the rejected message and an explanation of the nature of the error, in the *text* field, whenever possible.

If Chi-X receives a message with a sequence number *less* than expected during normal session processing, and it does not contain the *PossDupFlag* field, the message is discarded and a **Reject** message is sent to the client.

## 4.8 Recovery

When a client reconnects after a break in the session during the same trading day, Chi-X begins the following recovery sequence:

- **If Chi-X receives a sequence number less than expected the session will be terminated immediately without sending a logoff. The client should contact Chi-X to correct the problem.**
- Chi-X will transmit any unsent execution reports on receipt of a **Resend Request** from the client for the missing sequence numbers. If trades occur while the FIX session is down, Chi-X's outgoing sequence number will be higher than expected by the client.

- When a FIX session is terminated, the system's default behaviour is to cancel all open orders. If a client wishes not to have this default applied to their sessions they must make this clear to their connectivity representative during the on-boarding process.

The client is responsible for detecting message gaps for messages transmitted by Chi-X that may have been lost in the previous session, as per the FIX protocol. Chi-X will retransmit those messages when requested to do so by the client.

## 5 APPLICATION MESSAGES

This section discusses the application-level FIX messages sent and accepted by Chi-X.

### 5.1 Symbologies

Chi-X supports only the following stock naming identifiers in FIX messages (in order of preference):

1. Chi-X symbol
2. RIC
3. ISIN

#### 5.1.1 Client

When Chi-X receives a FIX application-level client message, it processes the symbol definition fields in the message in the following order to yield a valid stock symbol:

- If the client elects to use ISIN to identify stocks, the client must:
  1. Set the *IDSource* (22) field to “ISIN”;
  2. Put the ISIN value into the *SecurityID* (48) field; and
  3. Use the *SecurityExchange* (207) field to identify the listing market of the ISIN.
  4. Use the *Currency* (15) field to identify the currency the stock is listed in.
- If the *IDSource* (22) field is not set, Chi-X will **expect** the Chi-X symbol (or UMTF code) to exist in the *Symbol* (55) field.

Please note that Chi-X ignores the *Currency* (15) field in a client message if symbologies other than ISINs are used, since this information is derived from Chi-X’s internal Symbol Database.

## 5.2 Order Entry

### 5.2.1 Client

Chi-X currently supports the **New Order Single, Order Cancel Request, Order/Cancel Replace Request** FIX messages.

#### 5.2.1.1 New Order Single

In addition to the standard header, trailer, and Chi-X-accepted symbol definition fields, Chi-X processes only the following fields in a **New Order Single** message, and ignores all others:

Tag	Field Name	Req'd	Comments
1	Account	N	<p>Values supported by Chi-X and supplied to CCP/clearing house.</p> <p>H = House</p> <p>C = Client</p> <p>This field may also include the account in the format i.e. C:0123456 where 0123456 is the account id.</p> <p>If this field is absent, Chi-X will assume a value of H.</p> <p>If this field contains a value that is not prefixed by H or C i.e. 01234, Chi-X will assume a value of H:01234</p>
11	ClOrdID	Y	Must be unique for each order throughout the trading day, across all stocks and sides from the same FIX Session ID.
15	Currency	N	Conditionally required if ISIN is used. (See Section 5.1.1)
18	ExecInst	N	<p>Values supported by Chi-X:</p> <p>Pegging options (mutually exclusive and used only with Pegged <i>OrderType</i> )</p> <p>P = Market Peg</p> <p>R = Primary Peg.</p> <p>M = Mid-price Peg.</p>
21	HandlInst	Y	<p>Instructions for order handling on Chi-X's trading system.</p> <p>Value supported by Chi-X:</p> <p>1 = Automated execution order, private, no Broker intervention.</p> <p>.</p> <p><b>NOTE:</b> Values other than 1 will cause the order to be rejected.</p>

Tag	Field Name	Req'd	Comments
22	IDSource	N	Value supported by Chi-X: 4 = ISIN Number 5 = RIC
38	OrderQty	Y	Quantity of order.
40	OrdType	Y	Values supported by Chi-X: 1 = Market 2 = Limit P = Pegged (requires ExecInst = M, P or R)
44	Price	N	Required for limit orders. If included in a Market order, the order will be rejected.
47	Rule80A	N	Values supported by Chi-X: A = Agency single order P = Principal <b>NOTE:</b> If absent , the order will be accepted as Principal
48	SecurityID	N	RIC or ISIN code depending upon the value of the <i>IDSource</i> (22).
54	Side	Y	Values supported by Chi-X: 1 = Buy 2 = Sell
55	Symbol	Y	Chi-X Symbol (or UMTF Code) Only
59	TimeInForce	N	Absence of this field indicates a day order. Values supported by Chi-X: 0 = Day 3 = Immediate or Cancel - as much of the order as possible must be executed immediately. Any part of the order that is not executed immediately gets canceled. 4 = Fill Or Kill 6 = Good Till Date (GTD) – Date must be today's trading date. This field must be used in conjunction with field <i>ExpireTime</i> (Field ID 126). Day orders are in effect until the client cancels the order, or until the Chi-X trading system is shut down for Chi-X Nightly Maintenance.
60	TransactTime	Y	Time this order request was initiated by client.

Tag	Field Name	Req'd	Comments
110	MinQty	N	<p>Used to indicate the minimum quantity of an order to be executed. This attribute is only supported on hidden orders to Chi-X. If this attribute is added to non hidden orders then the order will be rejected.</p> <p>See Appendix C for a summary of Hidden Order functionality available on Chi-X.</p> <p>For more information please see the Chi-X Exchange Guide.</p>
111	MaxFloor	N	<p>Used to indicate display quantity. If greater than zero then the value must be equal to or multiple of RLS (Round Lot Size).</p> <p>If zero is used then this indicates order is fully hidden.</p> <p>If a fully hidden order has no Peg type included, then to be accepted it must adhere to the MIFID ADT rules for Large In Scale orders.</p> <p>See Appendix C for a summary of Hidden Order functionality available on Chi-X.</p> <p>For more information please see the Chi-X Exchange Guide.</p>
126	ExpireTime	N	<p>Time/Date of order expiration in GMT. Only valid when <i>TimeInForce</i> = 6 (Good Till Date). Date must be today's Trading Date.</p> <p>Incorrect data results in a rejected order.</p>
207	SecurityExchange	N	<p>Required when the <i>IDSource</i> (Field ID 22) equals ISIN (4)</p> <p><b>Note:</b> Chi-X uses the Reuters exchange mnemonic to specify the Market for the ISIN number. (see Appendix C of the FIX 4.2 protocol specification for appropriate values).</p>
211	PegDifference	N	<p>This is an optional field that can be used by the client to indicate an increment for Pegged orders. The increment is in the same denomination as the stock's currency <i>Price</i> (44).</p>

### 5.2.1.2 Order Cancel Request

Chi-X processes a Cancel Request quantity as the full remaining quantity. Chi-X does not support partial cancels.

In addition to the standard header, trailer, and Chi-X-accepted symbol definition fields, Chi-X processes only the following fields in an **Order Cancel Request** message, and ignores all others:

Tag	Field Name	Req'd	Comments
11	ClOrdID	Y	Unique ID of cancel request assigned by the client.
38	OrderQty	Y	Required by the FIX protocol, ignored by Chi-X. All cancel requests are for full remaining quantity.
41	OrigClOrdID	Y	Unique ID of original order as assigned by client.
54	Side	Y	
55	Symbol	Y	
60	TransactTime	Y	Time this cancel request was initiated by client.

### 5.2.1.3 Order Cancel/Replace Request

Cancel/Replace requests will be handled as per the FIX protocol. Cancel/Replace requests that cannot be processed will be rejected using the **Cancel Reject** message; If Chi-X rejects the Cancel/Replace request, the *ClOrdID* of the replacement order will be inserted in the *ClOrdID* field of the Cancel Reject message for identification purposes.

In addition to the standard header, trailer, and Chi-X-accepted symbol definition fields, Chi-X processes only the following fields in an **Order Cancel/Replace Request** message, and ignores all others

Tag	Field Name	Req'd	Comments
11	ClOrdID	Y	Unique ID of replacement order as assigned by the client.
18	ExecInst	N	For a replacement order, this field must be populated anew (i.e. original order values will not be brought forward to replacement order unless redefined within this message).
21	HandlInst	Y	
38	OrdQty	Y	<p><b>Note:</b> The quantity in the Cancel/Replace message is the total order quantity, as defined by the FIX protocol and total order quantity semantics.</p> <p>Order quantity can be revised up or down for visible orders and Chi-Delta orders.</p> <p>Please refer to Appendix A for more information.</p>

Tag	Field Name	Req'd	Comments
40	OrdType	Y	New Order Type for the request. Values supported: 1 = Market 2 = Limit P = Pegged (must include value in <i>ExecInst</i> )
41	OrigClOrdID	Y	Unique ID of order to be replaced as assigned by client.
44	Price	N	Required for Limit orders
47	Rule80A	N	Must match original order.
54	Side	Y	
55	Symbol	Y	
59	TimeInForce	N	Absence indicates a Day order
60	TransactTime	Y	Time this order request was initiated by client.
110	MinQty	N	See New Order Single section for details.
111	MaxFloor	N	See New Order Single section for details. If a Participant wishes to remove an order from the Chi-X visible order book and place it on the Chi-Delta destination (or vice versa) then this should be achieved by cancelling the original order and the entering of a new order. Cancel/replace messages attempting to do this will be rejected.
126	ExpireTime	N	If the client attempts to revise the expiration time to a time prior than the current time, the Cancel/Replace Request will be rejected.
211	PegDifference	N	This is an optional field that can be used by the client to indicate an increment for Pegged orders. The increment is in the same denomination as the stock's currency <i>Price (44)</i> .

## 5.2.2 Chi-X Order Entry Messages

### 5.2.2.1 New Order Single Response

In addition to the standard header, trailer and Chi-X-accepted symbol definition fields, Chi-X will provide the following fields in an **Execution Report** message in response to a New Order Single request.

Tag	Field Name	Req'd	Comments
1	Account	N	If present in New Order Single message from client.
6	AvgPx	Y	The average price of all shares traded. Defaulted to 0 for order acknowledgements.

Tag	Field Name	Req'd	Comments
11	ClOrdID	N	Chi-X will always populate this field with the original value assigned by client in the <b>New Order Single</b> message.
14	CumQty	Y	The total shares traded against the original order quantity. Defaulted to 0 for order acknowledgements.
15	Currency	N	Currency associated with symbol in <i>Symbol (55)</i> field
17	ExecID	Y	A unique identifier of execution message as assigned by Chi-X.
20	ExecTransType	Y	0 = New
31	LastPx	Y	Defaulted to 0 for order acknowledgements.
32	LastShares	Y	Defaulted to 0 for order acknowledgements.
35	MsgType	Y	8 = Execution Report
37	OrderID	Y	Chi-X order reference number.
38	OrderQty	Y	Quantity of original order.
39	OrdStatus	Y	0 = New 8 = Rejected
40	OrdType	N	Values supported by Chi-X: 1 = Market 2 = Limit P = Pegged (requires <i>ExecInst</i> of M or R)
44	Price	N	Limit Price Required for limit orders only.
47	Rule80A	N	If present on order.
54	Side	Y	Values supported by Chi-X: 1 = Buy 2 = Sell
55	Symbol	Y	Chi-X Symbol (or UMTF Code) Only
58	Text	N	Free form text field to convey error text
59	TimeInForce	N	Will be returned if supplied in original New Order Single message.
60	TransactTime	Y	Time the order acknowledgement was generated.
109	ClientID	N	Internal client identifier as assigned by Chi-X
110	MinQty	N	Will be returned if supplied with a value of >0 in the original New Order Single message.
111	MaxFloor	N	Will be returned if supplied with a value of >0 in the original New Order Single message.

Tag	Field Name	Req'd	Comments
150	ExecType	Y	0 = New 8 = Rejected
151	LeavesQty	Y	Amount of shares open for further execution.

### 5.2.2.2 Order Cancel Acknowledgment

In addition to the standard header, trailer and Chi-X-accepted symbol definition fields, Chi-X will provide the following fields in an **Execution Report** message in response to an order cancel request that is accepted by Chi-X.

Tag	Field Name	Req'd	Comments
6	AvgPx	Y	The average price of all shares traded.
11	ClOrdID	N	Chi-X will always populate this field with the original value assigned by client in the <b>Order Cancel Request</b> message.
14	CumQty	Y	The total shares traded against the original order quantity.
15	Currency	N	Currency associated with symbol in <i>Symbol (55)</i> field.
17	ExecID	Y	A unique identifier of execution message as assigned by Chi-X.
20	ExecTransType	Y	0 = New.
31	LastPx	Y	Defaulted to 0 for order cancel acknowledgements.
32	LastShares	Y	Defaulted to 0 for order cancel acknowledgements.
35	MsgType	Y	8 = Execution Report.
37	OrderID	Y	Chi-X order reference number.
38	OrderQty	Y	Quantity of original order.
39	OrdStatus	Y	4 = Cancelled.
40	OrdType	N	Values supported by Chi-X: 1 = Market 2 = Limit P = Pegged (requires <i>ExecInst</i> of M or R)
41	OrigClOrdID	Y	Unique ID of original order as assigned by client. (Note: this is not sent if message is an unsolicited cancellation from Chi-X to client).
44	Price	N	Limit Price Required for limit orders only.

Tag	Field Name	Req'd	Comments
54	Side	Y	Values supported by Chi-X: 1 = Buy 2 = Sell
55	Symbol	Y	Chi-X Symbol (or UMTF code).
59	TimeInForce	N	Will be returned if supplied in original New Order Single message.
110	MinQty	N	Will be returned if supplied with a value of >0 in the original New Order Single message.
111	MaxFloor	N	Will be returned if supplied with a value of >0 in the original New Order Single message.
150	ExecType	Y	4 = Cancelled.
151	LeavesQty	Y	0

### 5.2.2.3 Order Cancel Reject

In addition to the standard header, trailer and Chi-X-accepted symbol definition fields, Chi-X will provide the following fields in a Cancel Reject message in response to an order cancel request that is rejected.

Tag	Field Name	Req'd	Comments
11	ClOrdID	Y	Chi-X will populate this field with the value assigned by the client in the corresponding <b>Order Cancel Request</b> message.
35	MsgType	Y	9 = Order Cancel Reject.
37	OrderID	Y	Chi-X order reference number.
39	OrdStatus	Y	<i>OrdStatus</i> value after this cancel reject is applied.
41	OrigClOrdID	Y	<i>ClOrdID</i> that could not be canceled.
58	Text	N	The reason for the cancel reject.
434	CxlRejResponseTo	Y	Type of request to which this is a response: 1=Order Cancel Request 2=Order Cancel/Replace request

### 5.2.2.4 Cancel/Replace Response

Chi-X responds to an Order Cancel/Replace request as per the FIX protocol.

The message structure for a response to a successful request is detailed in the section “New Order Single Response”. The message structure for a rejection of a Cancel/Replace request is detailed in the “Order Cancel Reject” section.

### 5.2.2.5 Trade Report

Chi-X sends your trade reports via the **Execution Report** message. The trade report provides status information as orders trade, including: average price of shares traded; total shares traded against the original order quantity; transaction time; and trade date. Please note that this does not include settlement information such as commission or tax information.

**NOTE:** The *ExecID* is considered to be the unique identifier of an execution message by Chi-X, as per the FIX protocol. It is the client's responsibility to detect and appropriately process possible duplicate ExecIDs, regardless of whether the *PossResend* flag has been set on the message or not.

In addition to the standard header, trailer, and Chi-X-accepted symbol definition fields, Chi-X provides only the following fields in an **Execution Report**:

Tag	Field Name	Req'd	Comments
1	Account	N	If present in New Order Single message from client.
6	AvgPx	Y	Average price of shares traded.
11	ClOrdID	N	Chi-X will populate this field with the <i>ClOrdID</i> from the current state of the order.
14	CumQty	Y	Total shares traded against the original order quantity.
15	Currency	N	Currency in which the security is traded.
17	ExecID	Y	A unique identifier of execution message as assigned by Chi-X.
19	ExecRefID	N	Used in a trade cancel message (i.e. <i>ExecTransType</i> = 1) to refer to the previous <b>Execution Report</b> to which the current message applies.
20	ExecTransType	Y	Values supported by Chi-X: 0 = New to indicate a trade report 1 = Cancel to indicate a trade cancel
22	IDSource	N	Value supported by Chi-X: 4 = ISIN Number 5 = RIC
29	LastCapacity	N	This is an optional field but will be turned on by default for new clients from July 2008. Supported values are below: 1 = Agent 2 = Cross as agent 3 = Cross as principal 4 = Principal Values of 2 or 3 essentially indicate that the client has executed against themselves.

Tag	Field Name	Req'd	Comments
31	LastPx	Y	Price of shares bought or sold on this fill.
32	LastShares	Y	Quantity of shares bought or sold on this fill.
37	OrderID	Y	Chi-X order reference number.
38	OrderQty	Y	Quantity of original order.
39	OrdStatus	Y	1 = Partially filled 2 = Filled
40	OrdType	N	Values supported by Chi-X: 1 = Market 2 = Limit P = Pegged (requires <i>ExecInst</i> of M or R)
44	Price	N	Limit Price Required for limit orders only.
47	Rule80A	N	If present on order.
48	SecurityID	N	Security ID of traded security when <i>IDSource</i> is set.
54	Side	Y	Values supported by Chi-X: 1 = Buy 2 = Sell
55	Symbol	Y	Returns the Chi-X Symbol (or UMTF code) of the security traded.
60	TransactTime	N	Time and date of execution (expressed as GMT).
75	TradeDate	N	Date of trade in YYYYMMDD format.
109	ClientID	N	Internal client identifier as assigned by Chi-X
110	MinQty	N	Will be returned if supplied with a value of >0 in the original New Order Single message.
111	MaxFloor	N	Will be returned if supplied with a value of >0 in the original New Order Single message.
150	ExecType	Y	1 = Partially filled 2 = Filled
151	LeavesQty	Y	Amount of shares open for further execution.
382	NoContraBrokers	N	Only provided in an Execution Report where <i>OrdStatus</i> is Partially Filled (39=1) or Filled (39=2) and <i>ExecTransType</i> is New (20=0) and <i>LastShares</i> >0. If present will have value of 1.

Tag	Field Name	Req'd	Comments
375	ContraBroker	Y	Counterparty Identifier. In the case of a Central Counterparty trade it will be the identifier of the CCP, in the event of the trade being a suppressed Self Trade then the value would be set to "SELF" .
9730	TradeLiquidityIndicator	N	Values supported by Chi-X: A = Order added liquidity R = Order removed liquidity D = Dark Trade X = Onward routed trade T = IOC Dark (Delta) L = Non IOC LIS in Lit book

### **5.2.2.6 Unsupported FIX Messages**

Chi-X does not support the following FIX message types:

**Allocation and Allocation ACK messages**

**Quote Request and Quote messages**

**Advertisement, Email and News messages**

**Order List, Order Status**

**List Cancel, List Execute and List Status**

## 6 APPENDIX A – EXAMPLE ORDER FLOW MATRICES

### 1 - Filled order

<u>Time</u>	<u>Message Received</u> (ClOrdID, OrigClOrdID)	<u>Message Sent</u> (ClOrdID, OrigClOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Comment</u>
1	New Order(X)					10000				
2		Execution(X)	Rejected	Rejected	New	10000	0	0	0	If order is rejected
2		Execution(X)	New	New	New	10000	0	10000	0	
3		Execution(X)	Partial Fill	Partially Filled	New	10000	2000	8000	2000	Execution of 2000
4		Execution(X)	Partial Fill	Partially Filled	New	10000	3000	7000	1000	Execution of 1000
5		Execution(X)	Fill	Filled	New	10000	10000	0	7000	Execution of 7000

### 2 – Cancel request issued for a zero-filled order

<u>Time</u>	<u>Message Received</u> (ClOrdID, OrigClOrdID)	<u>Message Sent</u> (ClOrdID, OrigClOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Comment</u>
1	New Order(X)					10000				
2		Execution(X)	Rejected	Rejected	New	10000	0	0	0	If order is rejected
2		Execution(X)	New	New	New	10000	0	10000	0	
3	Cancel Request(Y,X)					10000				
4		Cancel Reject (Y,X)		New		10000				If rejected
4		Execution (Y,X)	Canceled	Canceled	New	10000	0	0	0	

## 3 – Cancel request issued for a part-filled order – executions occur whilst cancel request is active

<u>Time</u>	<u>Message Received</u> (ClOrdID, OrigClOrdID)	<u>Message Sent</u> (ClOrdID, OrigClOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Comment</u>
1	New Order(X)					10000				
2		Execution(X)	Rejected	Rejected	New	10000	0	0	0	If order is rejected
2		Execution(X)	New	New	New	10000	0	10000	0	
3		Execution(X)	Partial Fill	Partially Filled	New	10000	2000	8000	2000	Execution for 2000
4	Cancel Request(Y,X)					10000				
4		Execution(X)	Partial Fill	Partially Filled	New	10000	5000	5000	3000	Execution for 3000. This execution passes the cancel request on the connection
5		Cancel Reject (Y,X)		Partially Filled		10000				If request is rejected
5		Execution(X)	Partial Fill	Partially Filled	New	10000	6000	4000	1000	Execution for 1000 whilst order is pending cancel
6		Execution (Y,X)	Canceled	Canceled	New	10000	6000	0	0	'Canceled' order status takes precedence over 'partially filled' order status

## 4 – Cancel request issued for an order that becomes filled before cancel request can be accepted

<u>Time</u>	<u>Message Received</u> (ClOrdID, OrigClOrdID)	<u>Message Sent</u> (ClOrdID, OrigClOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Comment</u>
1	New Order(X)					10000				
2		Execution(X)	Rejected	Rejected	New	10000	0	0	0	If order is rejected
2		Execution(X)	New	New	New	10000	0	10000	0	
3		Execution(X)	Partial Fill	Partially Filled	New	10000	2000	8000	2000	Execution for 2000
4	Cancel Request(Y,X)					10000				
5		Execution(X)	Fill	Filled	New	10000	10000	0	8000	Execution for 8000. This execution passes the cancel request on the connection
6		Cancel Reject (Y,X)		Filled		10000				Cancel reject reason sent in Tag 58 Text

## 5 – Zero-filled order, cancel/replace request issued to decrease order qty

<u>Time</u>	<u>Message Received</u> (ClOrdID, OrigClOrdID)	<u>Message Sent</u> (ClOrdID, OrigClOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Comment</u>
1	New Order(X)					10000				
2		Execution(X)	Rejected	Rejected	New	10000	0	0	0	If order is rejected
2		Execution(X)	New	New	New	10000	0	10000	0	
3	Replace Request(Y,X)					9000				Request to decrease order qty to 9000
4		Cancel Reject (Y,X)		New		10000				If rejected by trader
4		Execution (Y,X)	Replace	Replaced	New	9000	0	9000	0	'Replaced' order status takes precedence over 'new' order status
5		Execution (Y)	Partial Fill	Partially Filled	New	9000	1000	8000	1000	Execution for 1000
6		Execution (Y)	Partial Fill	Partially Filled	New	9000	3000	6000	2000	Execution for 2000

6 – Part-filled order, followed by cancel/replace request to decrease order qty, execution occurs whilst order is pending replace

<u>Time</u>	<u>Message Received</u> (ClOrdID, OrigClOrdID)	<u>Message Sent</u> (ClOrdID, OrigClOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Comment</u>
1	New Order(X)					10000				
2		Execution(X)	Rejected	Rejected	New	10000	0	0	0	If order is rejected
2		Execution(X)	New	New	New	10000	0	10000	0	
3		Execution(X)	Partial Fill	Partially Filled	New	10000	1000	9000	1000	Execution for 1000
4	Replace Request(Y,X)					8000				Request decrease in order quantity to 8000
5		Cancel Reject (Y,X)		Partially Filled		10000				If request is rejected
6		Execution(X)	Partial Fill	Partial Fill	New	10000	1100	8900	100	Execution for 100 before cancel/replace request is responded to
7		Cancel Reject (Y,X)		Partially Filled		10000				If request is rejected
7		Execution (Y,X)	Replace	Partially Filled	New	8000	1100	6900	0	'Partially filled'' order status takes precedence over 'replaced' order status
8		Execution(Y)	Fill	Filled	New	8000	8000	0	6900	Execution for 6900

7 – Cancel/replace request (not for quantity change) is rejected as a fill has occurred

<u>Time</u>	<u>Message Received</u> (ClOrdID, OrigClOrdID)	<u>Message Sent</u> (ClOrdID, OrigClOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Comment</u>
1	New Order(X)					10000				
2		Execution(X)	Rejected	Rejected	New	10000	0	0	0	If order is rejected
2		Execution(X)	New	New	New	10000	0	10000	0	
3		Execution(X)	Partial Fill	Partially Filled	New	10000	1000	9000	1000	Execution for 1000
4	Replace Request(Y,X)					10000				
4		Execution (X)	Fill	Filled	New	10000	10000	0	9000	Execution for 9000 – the replace request message and this execution report pass each other on the connection
5		Cancel Reject (Y,X)		Filled		10000				Cancel reject reason sent in Tag 58 Text

8 – Cancel/replace request sent whilst execution is being reported – the requested order qty exceeds the cum qty. Order is replaced then filled

<u>Time</u>	<u>Message Received</u> (ClOrdID, OrigClOrdID)	<u>Message Sent</u> (ClOrdID, OrigClOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Comment</u>
1	New Order(X)					10000				
2		Execution(X)	Rejected	Rejected	New	10000	0	0	0	If order is rejected
2		Execution(X)	New	New	New	10000	0	10000	0	
3		Execution(X)	Partial Fill	Partially Filled	New	10000	1000	9000	1000	Execution for 1000
4	Replace Request(Y,X)					8000				Request a decrease order quantity to 8000 (leaving 7000 open)
4		Execution(X)	Partial Fill	Partially Filled	New	10000	1500	8500	500	Execution for 500 sent. Replace request and this execution report pass each other on the connection
5		Execution(X)	Partial Fill	Partially Filled	New	10000	1600	8400	100	Execution for 100 occurs before cancel/replace request is accepted
6		Cancel Reject (Y,X)		Partially Filled		10000				If request is rejected
6		Execution (Y,X)	Replace	Partially Filled	New	8000	1600	6400	0	'Partially filled' order status takes precedence over 'replaced' order status. Replace is accepted as requested order qty exceeds cum qty
7		Execution (Y)	Fill	Filled	New	8000	8000	0	6400	Execution for 6400.

9 – Cancel/replace request sent whilst execution is being reported – the requested order qty equals the cum qty – order qty is amended to cum qty

<u>Time</u>	<u>Message Received</u> (CLOrdID, OrigCLOrdID)	<u>Message Sent</u> (CLOrdID, OrigCLOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Comment</u>
1	New Order(X)					10000				
2		Execution(X)	Rejected	Rejected	New	10000	0	0	0	If order is rejected
2		Execution(X)	New	New	New	10000	0	10000	0	
3	Replace Request(Y,X)					7000				Client wishes to amend order qty to 7000 shares
3		Execution(X)	Partial Fill	Partially Filled	New	10000	7000	3000	7000	Execution for 7000 - the replace message and this execution report pass each other on the connection
4		Execution (Y,X)	Replace	Filled	New	7000	7000	0	0	The replace request is interpreted as requiring the balance of the order to be canceled – the ‘filled’ order status takes precedence over ‘canceled’ or ‘replaced’

10 – Cancel/replace request sent whilst execution is being reported – the requested order qty is below cum qty – order qty is amended to cum qty

<u>Time</u>	<u>Message Received</u> (CLOrdID, OrigCLOrdID)	<u>Message Sent</u> (CLOrdID, OrigCLOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Comment</u>
1	New Order(X)					10000				
2		Execution(X)	Rejected	Rejected	New	10000	0	0	0	If order is rejected
2		Execution(X)	New	New	New	10000	0	10000	0	
3	Replace Request(Y,X)					7000				Client wishes to amend order qty to 7000 shares
3		Execution(X)	Partial Fill	Partially Filled	New	10000	8000	2000	8000	Execution for 8000 - the replace message and this execution report pass each other on the connection
4		Execution (Y,X)	Replace	Filled	New	8000	8000	0	0	The replace request is interpreted as requiring the balance of the order to be canceled – the ‘filled’ order status takes precedence over ‘canceled’ or ‘replaced’

11 – One cancel/replace request is issued which is accepted – another one is issued which is also accepted

<u>Time</u>	<u>Message Received</u> (CLOrdID, OrigCLOrdID)	<u>Message Sent</u> (CLOrdID, OrigCLOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Comment</u>
1	New Order(X)					10000				
2		Execution(X)	Rejected	Rejected	New	10000	0	0	0	If order is rejected
2		Execution(X)	New	New	New	10000	0	10000	0	
3		Execution(X)	Partial Fill	Partially Filled	New	10000	1000	9000	1000	Execution for 1000

4	Replace Request(Y,X)					8000				Request decrease in order quantity to 8000, leaving 7000 open
5		Execution(X)	Partial Fill	Partially Filled	New	10000	1500	8500	500	Execution for 500
6		Execution (Y,X)	Replace	Partially Filled	New	8000	1500	6500	0	'Partially filled' order status takes precedence over 'replaced' order status
7		Execution (Y)	Partial Fill	Partially Filled	New	8000	3500	4500	2000	Execution for 2000
8	Replace Request(Z,Y)					6000				Request decrease in order quantity to 6000, leaving 2500 open
9		Execution (Z,Y)	Replace	Partially Filled	New	6000	3500	2500	0	'Partially filled' order status takes precedence over 'replaced' order status
10		Execution(Z)	Fill	Filled	New	6000	6000	0	2500	Execution for 2500

12 – Unsolicited cancel of a part-filled order

<u>Time</u>	<u>Message Received</u> (CLOrdID, OrigCLOrdID)	<u>Message Sent</u> (CLOrdID, OrigCLOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Comment</u>
1	New Order(X)					10000				
2		Execution(X)	Rejected	Rejected	New	10000	0	0	0	If order is rejected
2		Execution(X)	New	New	New	10000	0	10000	0	
3		Execution(X)	Partial Fill	Partially Filled	New	10000	1000	9000	1000	Execution for 1000
4										Chi-X verbally agrees to cancel order
5		Execution(X)	Canceled	Canceled	New	10000	1000	0	0	Unsolicited cancel of open order

13– Order rejected due to duplicate CLOrdID

<u>Time</u>	<u>Message Received</u> (CLOrdID, OrigCLOrdID)	<u>Message Sent</u> (CLOrdID, OrigCLOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Comment</u>
1	New Order(X)					10000				
2		Execution(X)	New	New	New	10000	0	10000	0	

3		Execution(X)	Partial Fill	Partially Filled	New	10000	1000	9000	1000	Execution for 1000
4	New Order(X)					10000				Order submitted with the same order id
5		Execution(X)	Rejected	Partially Filled	New	10000	1000	9000	0	

14 - Poss resend order

<u>Time</u>	<u>Message Received</u> (ClOrdID, OrigClOrdID)	<u>Message Sent</u> (ClOrdID, OrigClOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Comment</u>
1	New Order(X)					10000				
2		Execution(X)	New	New	New	10000	0	10000	0	
3	New Order(X)					10000				PossResend=Y
4		Execution(X)	New	New	Status	10000	0	10000		Because order X has already been received, confirm back the current state of the order. Last shares not required when ExecTransType = Status
5	New Order(Y)					15000				PossResend=Y
6		Execution(Y)	New	New	New	15000	0	15000	0	Because order Y has not been received before, confirm back as a new order.

15 – Immediate or Cancel order that cannot be immediately hit

<u>Time</u>	<u>Message Received</u> (ClOrdID, OrigClOrdID)	<u>Message Sent</u> (ClOrdID, OrigClOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Comment</u>
1	New Order(X)					10000				Order is IOC
2		Execution(X)	Rejected	Rejected	New	10000	0	0	0	If order is rejected
2		Execution(X)	New	New	New	10000	0	10000	0	
3		Execution(X)	Partial Fill	Partially Filled	New	10000	1000	9000	1000	Execution for 1000
4		Execution(X)	Canceled	Canceled	New	10000	1000	0	0	If order cannot be immediately hit

## 16 – Fully filled order (1 trade), followed by cancellation of execution.

<u>Time</u>	<u>Message Received</u> (ClOrdID, OrigClOrdID)	<u>Message Sent</u> (ClOrdID, OrigClOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>AvgPx</u>	<u>Last Shares</u>	<u>Last Px</u>	<u>ExecId (ExecRefID)</u>	<u>Comment</u>
1	New Order(X)					10000							
2		Execution(X)	Rejected	Rejected	New	10000	0	0		0		A	If order is rejected
2		Execution(X)	New	New	New	10000	0	10000	0	0		B	
3		Execution(X)	Fill	Filled	New	10000	10000	0	100	10000	100	C	Execution for 10000 @ 100
4		Execution(X)	Fill	Filled	Cancel	10000	0	0	0	0	0	D (C)	Cancel execution for 10000

## 17 – Fully filled order (&gt;1 trade), followed by cancellation of 1 execution.

<u>Time</u>	<u>Message Received</u> (ClOrdID, OrigClOrdID)	<u>Message Sent</u> (ClOrdID, OrigClOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>AvgPx</u>	<u>Last Shares</u>	<u>Last Px</u>	<u>ExecId (ExecRefID)</u>	<u>Comment</u>
1	New Order(X)					10000							
2		Execution(X)	Rejected	Rejected	New	10000	0	0		0		A	If order is rejected
2		Execution(X)	New	New	New	10000	0	10000	0	0		B	
3		Execution(X)	Partial Fill	Partially Filled	New	10000	8000	2000	100	8000	100	C	Execution for 8000 @ 100
4		Execution(X)	Fill	Filled	New	10000	10000	0	100	2000	100	D	Execution for 2000 @ 100
5		Execution(X)	Fill	Filled	Cancel	10000	8000	0	100	2000	0	E (D)	Cancel execution for 2000

## 18 – Partially filled order (1 trade), followed by cancellation of execution and remaining quantity trades

<u>Time</u>	<u>Message Received</u> (ClOrdID, OrigClOrdID)	<u>Message Sent</u> (ClOrdID, OrigClOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>AvgPx</u>	<u>Last Shares</u>	<u>Last Px</u>	<u>ExecId (ExecRefID)</u>	<u>Comment</u>
1	New Order(X)					10000							
2		Execution(X)	Rejected	Rejected	New	10000	0	0		0		A	If order is rejected
2		Execution(X)	New	New	New	10000	0	10000	0	0		B	
3		Execution(X)	Partial Fill	Partially Filled	New	10000	8000	2000	100	8000	100	C	Execution for 8000 @ 100
4		Execution(X)	Partial Fill	New	Cancel	10000	0	2000	0	2000	0	D (C)	Cancel execution for 8000
5		Execution(X)	Fill	Filled	New	10000	2000	0	100	2000	100	E	Execution for 2000 @ 100

## 7 APPENDIX B – CHI-X PEG ORDER DEFINITIONS

### Primary (PRIM) Peg Type

This section will provide examples of Basic and Incremental PRIM Pegged orders.

#### Basic Primary Peg

Basic PRIM pegged orders are pegged to the same side of the stock's primary market quote. The order does not have any discretion, and therefore, will only trade with other orders at its then displayed price. The order will float with the market up to the limit price.

Peg Order Type	Does the Display Float?	Does the order have Discretion?	Does the Discretion Float?	Side Pegged
Basic Primary (PRIM)	Y	N	N/A	Same side of primary mkt quote

In the following example, the order is to *Buy 3000 at 10.20 pegged to the PRIM*:

ExecInst (18)	OrderQty (38)	OrdType (40)	Price <sup>1</sup> (44)	Side (54)
R	3000	P	10.20	1

The Primary Market best bid and offer is 10.10 – 10.16. The order will be initially displayed at 10.10 and will float with the market but never beyond the limit price of 10.20. It will only trade with other orders at the then displayed price, since the order does not have any discretion.

#### Primary Peg with Increments

Chi-X also enables clients to float PRIM Pegged orders a user defined increment away from the best quote on the same side of the stock's primary market. The display price will float with the market to the limit price and only trade with other orders at the then displayed price.

Peg Order Type	Does the Display Float?	Does the order have Discretion?	Does the Discretion Float?	Side Pegged
Primary with Increment (PRIM + increment)	Y	N	N/A	Same side of primary mkt.

In the following example, the order is to *Buy 3000 at 10.20 pegged to the PRIM -0.01*:

ExecInst (18)	OrderQty (38)	OrdType (40)	Price (44)	Side (54)	PegDifference (211)
R	3000	P	10.20	1	-0.01

<sup>1</sup> If no limit price (*Price*) is entered for Primary Pegged orders then the order will float until fully executed. This applies to all Primary Pegged orders.

The primary market's best bid and offer is 10.00 – 10.05. Therefore, the order would initially be displayed at 10.00-0.01, or 9.99, and will float with the market 0.01 away from the best bid, but never beyond the limit of 10.20. The order will trade with other orders at the then displayed price.

## Mid (MID) Peg Type

This section will provide examples of Basic and Incremental MID Pegged orders.

### Basic Mid Peg

Basic MID pegged orders are pegged to the middle of the primary market best bid and offer. The displayed price does not have any discretion, and therefore, will only trade with other orders at its then displayed price. The order will float with the market up to the limit price.

Please note that a MID order is calculated between the Bid and the Ask, and is **rounded passively** to the minimum price variance of that stock.

Peg Order Type	Does the Display Float?	Does the order have Discretion?	Does the Discretion Float?	Side Pegged
Basic MID	Y	N	N/A	Middle Primary Mkt best bid/offer.

In the following example, the order is to *Buy 3000 at 10.20 pegged to the MID*:

ExecInst (18)	OrderQty (38)	OrdType (40)	Price <sup>2</sup> (44)	Side (54)
M	3000	P	10.20	1

The primary market best bid and offer is 10.00 – 10.06. Therefore, the order will initially be displayed at 10.03, and will float with the market, but never beyond the limit of 10.20. The order will trade with other orders at the displayed price.

### Mid Peg with Increments

Chi-X also enables clients to float MID Pegged orders a user defined increment away from the middle of the primary market best bid and offer. The displayed price will float with the market to the limit price and only trade with other orders at the displayed price.

Peg Order Type	Does the Display Float?	Does the order have Discretion?	Does the Discretion Float?	Side Pegged
MID with Increment (MID + increment)	Y	N	N/A	Middle of primary market best bid and offer

In the following example, the order is to *Buy 3000 at 10.20 pegged to the MID -0.01*:

<sup>2</sup> If no limit price (*Price*) is entered for Mid Pegged orders then the order will float until fully executed. This applies to all Mid Peg orders.

ExecInst (18)	OrderQty (38)	OrdType (40)	Price (44)	Side (54)	PegDifference (211)
M	3000	P	10.20	1	-0.01

The primary market best bid and offer is 10.00 – 10.06. Therefore, the order will initially be displayed at 10.03 - 0.01, or 10.02, and will float with the market, but never beyond the limit of 10.20. The order will trade with other orders at the displayed price.

## Market (MKT) Peg Type

This section will provide examples of Basic and Incremental Market (MKT) Pegged orders.

### Basic Market Peg

Basic Market (MKT) pegged orders are pegged to the contra-side of the Primary Market. The order will float with the market up to the limit price.

Peg Order Type	Does the Display Float?	Does the order have Discretion?	Does the Discretion Float?	Side Pegged
Basic Market (MKT)	Y	N	N/A	Contra side of Primary Market

In the following example, the order is to *Buy 3000 at 10.20 pegged to the MKT*:

ExecInst (18)	OrderQty (38)	OrdType (40)	Price <sup>3</sup> (44)	Side (54)
P	3000	P	10.20	1

The Primary Market best bid and offer is 10.01 – 10.06. Therefore, the order will initially be displayed at 10.06, and will float with the market, but never beyond the limit of 10.20.

### Market Peg with Increments

Chi-X also enables clients to float Market Pegged orders a user defined increment away from the contra-side of the Primary Market. The displayed price will float with the market to the limit price and only trade with other orders at the displayed price.

Peg Order Type	Does the Display Float?	Does the order have Discretion?	Does the Discretion Float?	Side Pegged
Market with Increment [MKT + increment]	Y	N	N/A	Contra side of primary market.

In the following example, the order is to *Buy 3000 at 10.20 pegged to the MKT +0.01*:

ExecInst	OrderQty	OrdType	Price <sup>4</sup>	Side	PegDifference
----------	----------	---------	--------------------	------	---------------

<sup>3</sup> If no limit price (*Price*) is entered for Market Peg orders then the order will float until fully executed. This applies to all Market Peg orders.

(18)	(38)	(40)	(44)	(54)	(211)
P	3000	P	10.20	1	+0.01

The Primary Market best bid and offer is 10.01 – 10.06. The order will be displayed at 10.07, and will float with the market, but never beyond the limit of 10.20.

### **Summary of the Effects of PegDifference on Order Prices.**

If a positive value for PegDifference is used on a buy order then the bid increases.  
If a negative value for PegDifference is used on a buy order then the bid decreases.  
If a positive value for PegDifference is used on a sell order then the offer decreases.  
If a negative value for PegDifference is used on a sell order then the offer increases.

---

<sup>4</sup> If no limit price (*Price*) is entered for Market Peg orders then the order will float until fully executed. This applies to all Market Peg orders.

## 8 APPENDIX C: HIDDEN ORDER PARAMETERS

### 8.1 Functionality Matrix

This section discusses the hidden order functionality available to Chi-X Participants routing orders to Chi-Delta.

MaxFloor = 0 Present?	Mid Peg Type Present?	MinQty Present?	LIS or above?	Result
No	Yes	No	No	Routed to visible orderbook. (Any order not having MaxFloor set to zero will be treated as being destined for the visible book.)
No	No	Yes	Yes	Rejected due to MinQty not being supported in visible orderbook.
Yes	Yes	No	Yes	Routed as non-displayed to visible order book.
Yes	No	No	No	Routed to Chi-Delta but order will have a Mid Peg attribute attached.
Yes	No	Yes	Yes	Routed to Chi-Delta but order will have a Mid Peg attribute attached.
Yes	No	Yes	No	Routed to Chi-Delta but order will have a Mid Peg attribute attached.
Yes	Yes	Yes	Yes	Routed to Chi-Delta.
Yes	Yes	Yes	No	Routed to Chi-

				Delta
Yes	Yes	No	No	Routed to Chi-Delta
Yes	No	No	Yes	1. IOC/FOK orders routed to Chi-Delta but order will have a Mid Peg attribute attached. 2. Non IOC/FOK orders routed to visible book.

## 8.2 Supported Non-Displayed Order Types/Attributes

Chi-X Delta will support the following Order Types/Attributes (all must be flagged as fully hidden via use of MaxFloor = 0 in the FIX message):

- Mid-Pegged Orders (optionally can have a limit price). Note: peg offsets are not supported to Chi-Delta.
- Immediate or Cancel (IOC)
- Fill Or Kill (FOK)
- Good till date (date must be date of order entry – used to set expiry time)
- Minimum Quantity
- Limit Orders (no Peg type) will result in Chi-X attaching a Mid Peg to the order.

## 9 APPENDIX D: CHI-VISION FUNCTIONALITY

### 9.1 Supported Routing Strategies

Chi-Vision Supports order routing strategies based on value supplied in FIX Tag 57 *TargetSubID* (as described in Section 4.1).

The current order routing strategy is outlined below.

Strategy Value	Routing Destinations	Posting Destination
CXEP	<ol style="list-style-type: none"> <li>1. Chi-Delta</li> <li>2. Chi-X visible order-book</li> <li>3. External Liquidity Providers (ELPs)</li> </ol>	Chi-X visible order-book if no <i>MaxFloor=0</i> Tag present, otherwise Chi-Delta if <i>MaxFloor=0</i> Tag is present.

Participants can elect to not route to specific ELPs by prior arrangement. In addition if clients cannot use *TargetSubID* then all orders on a given FIX session can be defaulted to a given strategy.

### 9.2 Supported Order Types/Attributes

Orders flagged with any of the supported Chi-Vision strategies can support only the following subset of order types/attributes:

- Time in Force: Immediate or Cancel (IOC)
- Time in Force: Day
- Order type: Limit
- Fully hidden (via use of *MaxFloor = 0*)

### 9.3 Market Data and Use of Trade Liquidity Indicator

Trades executed via the Chi-Vision service will be reported either;

- on the CHIXMD feed (if the trade is conducted on the Chi-Delta or the Chi-X visible order-book)

- or via the CHIXTDM service (if the trade is an off book trade) and will be flagged appropriately.

The *TradeLiquidityIndicator* (Tag 9730) will return the value of X for trades executing “off book on MTF”. For trades occurring on the Chi-X books then the normal values of A, R and D will be utilised.