

T7™

**Direct Trading Interface (DTI)
Programming Manual**

Version: 10.0.4

Date: February 20, 2015

Abstract

This document provides information on business descriptions, programming interfaces and protocols for connecting client applications to the ISE T7™ system.

This version is applicable for ISE T7 Release 10.0.4

Please note that some functionality described herein may not be available.

T7 is a trademark of Deutsche Börse Group. To find out more, please visit <http://deutsche-boerse.com/7>.

Copyright © 2015. International Securities Exchange, LLC

While reasonable care has been taken in the preparation of this publication to provide details that are accurate and not misleading at the time of publication, this publication is distributed to you solely on an “as is” basis. No representations or warranties are made regarding the information contained herein, whether express or implied, including without limitation any implied warranty of merchantability or fitness for a particular purpose or any warranty with respect to the accuracy, correctness, quality, completeness or timeliness of such information. ISE and its affiliates shall have no liability of any kind whatsoever to any third-parties in connection with this publication and the information contained herein. This publication is published for general informational purposes only, and it may be used solely for your internal business purposes in connection with ISE. You may not redistribute it to anyone outside of your organization. Much of the information contained herein is subject to detailed exchange trading rules, which are subject to change. Some of the functionality described in this publication may be subject to approval by the U.S. Securities and Exchange Commission. All descriptions, examples, and calculations contained in this publication are for illustrative purposes only. Unauthorized reproduction or use of any content of this publication or the subject matter thereof, including, but not limited to, trademarks, text, and pictures, is strictly prohibited.

Table of Contents

1. About This Document	8
1.1 ISE T7.....	8
1.2 Intended Audience	8
1.3 ISE Related Documents	8
1.4 Formatting Notes	9
1.5 Acronyms	9
2. Introduction.....	11
2.1 Background	11
2.2 Auction Market	11
2.3 Membership Types.....	11
2.4 Connecting to the ISE DTI	12
2.5 Developer Support	15
3. Session Information	16
3.1 FIX Semantics	16
3.2 Length Tag Value (LTV) Encoding.....	16
3.3 XML Interface Description	16
3.4 Message Headers	19
3.5 Establishing a DTI Session.....	21
3.6 Terminating a DTI Session	26
3.7 Maintaining a Session.....	27
3.8 Sending Application Messages	28
3.9 Response Messages.....	29
3.10 Broadcast Messages.....	31
4. ISE Market Description.....	40
4.1 Reference Data.....	40
4.2 Market Participant Structure.....	41
4.3 Risk Protection & Management.....	41
4.4 Pro-Rata Sharing Algorithm	42
4.5 Basic Order Functions.....	43
4.6 Deals and Trades	47
4.7 Special Order Types.....	54
4.8 Cross Orders.....	58

4.9	Auctions	59
4.10	Quote Related Functions	68
4.11	Complex Orders and Quotes	73
5.	DTI Message Specifications	82
5.1	Attributes Common to All DTI Messages	82
5.2	New Order Single Request.....	85
5.3	Alter Order Single Request	89
5.4	Delete Order Request.....	92
5.5	Delete All Orders Request	93
5.6	New Order Multileg Request	95
5.7	Alter Order Multileg Request	98
5.8	New Order Cross Request	101
5.9	Mass Quote Request	103
5.10	Delete All Quotes Request.....	107
5.11	Quote Action Request	109
5.12	Change Instrument State Request.....	111
5.13	Order Lock Request	113
5.14	Order Release Request.....	114
5.15	Add Complex Instrument Request.....	115
5.16	MM Parameter Definition Request	117
5.17	Modify Deal Request	119
5.18	Member Kill Switch Request.....	121
5.19	Private Order Broadcast	123
5.20	Application Sequence Number Reset Broadcast.....	130
5.21	Private Quote Broadcast	131
5.22	Product Reset Broadcast	134
5.23	Deal Item Broadcast	135
5.24	Trade Item Broadcast	138
5.25	Locked Order Broadcast	142
5.26	Business End Of Stream Broadcast	143
5.27	Business End Of Day Broadcast	143
5.28	End Of Stream Broadcast.....	144
5.29	Trading Session Marker Message Broadcast.....	145
5.30	User Notification Broadcast.....	146

5.31	Invalid Data Broadcast.....	147
5.32	News Broadcast.....	147
5.33	Heartbeat.....	148
List of Appendices.....		149
Appendix A: LTV Encoding		150
	Logical Organization.....	150
	LTV Container Layout.....	150
	Byte Ordering	150
	Length and Tag Field Encoding.....	150
	Special Length/Tag Field Encoding for the DTI Message	152
	Value Field Encoding.....	152
	Example LTV Encoded Message	155
Appendix B: ISE Parameters.....		158
Appendix C: Glossary		160
Appendix D: DTI Conformance Test.....		164
	Testing all DTI Applications	164
	Testing Order Routing Applications	164
	Testing MM Applications	164
	Testing PMM Applications	165
Appendix E: DTI Application Messages.....		166
	DTI Application Messages, by name.....	166
	DTI Application Messages, by numeric ID	167
Appendix F: DTI Groups		168
	DTI Groups, by name	168
	DTI Groups, by numeric ID	169
Appendix G: DTI Field Descriptions (Alphabetical)		170
Appendix H: DTI Field Descriptions (Numerical)		187
Appendix I: Security Trading Event Codes		204
Appendix J: Event Reason Codes		205
Appendix K: Throttles		210
	Transaction Limit	210
	Outstanding Message Limit.....	211

Reject/Disconnect Limit	211
Summary	212
Appendix L: Locked Order Handling	213
Processing a Locked Order	213
Executing the Locked Order	213
Releasing the Locked Order	213
Auto-Release of Locked Quantity	214
Appendix M: Standard Strategies	215
Strategy Examples.....	216
Unconventional Strategies	216
Appendix N: Away Market Codes.....	217
Appendix O: Unicast IP Addresses.....	218
Additional Information	220
Member Website	220
Contact Information	220
Document Revision Table	221

Tables

Table 1: ISE-Related Documentation	8
Table 2: Acronyms.....	9
Table 3: Membership Types.....	12
Table 4: Request Message Header.....	20
Table 5: Response Message Header	20
Table 6: Broadcast Message Header.....	21
Table 7: Logon Request (MsgID=10000)	22
Table 8: Response to Logon Request (MsgID=10001)	23
Table 9: Gateway Request (MsgID=10018).....	24
Table 10: Response to Gateway Request (MsgID=10019).....	25
Table 11: Logout Request (MsgID=10002).....	26
Table 12: Response to Logout Request (MsgID=10003).....	27
Table 13: Reject Message Format (MsgID=10010)	30
Table 14: Pending Response (MsgID=10014)	30
Table 15: Invalid Response (MsgID=10017).....	31

Table 16: Broadcast Types	32
Table 17: Subscribe Broadcast Request (MsgID=10004)	32
Table 18: Subscribe Response (MsgID=10005)	33
Table 19: Unsubscribe Request (MsgID=10006)	34
Table 20: Unsubscribe Broadcast Response (MsgID=10007)	35
Table 21: Retransmission Request (MsgID=10008)	36
Table 22: Retransmission Response (MsgID=10009)	37
Table 23: Order/Clearing Capacity Values	47
Table 24: Liquidity Indicator Values	49
Table 25: Mapping status broadcasts to deal/trade broadcasts (simple instruments)	53
Table 26: Mapping status broadcasts to deal/trade broadcasts (complex instruments)	53
Table 27: Cross Order Types	59
Table 28: Message Summary for One-Sided Auctions	61
Table 29: Message Summary for Two-Sided Auctions	63
Table 30: Auction Types	67
Table 31: New Order Single Request (MsgID=10100)	86
Table 32: Order Acknowledge Response (MsgID=10103)	88
Table 33: Order Cancel Replace Request (MsgID=10101)	89
Table 34: Order Cancel Request (MsgID=10102)	92
Table 35: Delete All Orders Request (MsgID=10105)	93
Table 36: Delete All Orders Response (MsgID=10106)	95
Table 37: New Order Multileg Request (MsgID=10115)	96
Table 38: Multileg Order Cancel Replace Request (MsgID=10116)	99
Table 39: New Order Cross Request (MsgID=10114)	102
Table 40: Mass Quote Request (MsgID=10203)	105
Table 41: Mass Quote Acknowledgement (MsgID=10204)	106
Table 42: Quote Reject Reasons	107
Table 43: Delete All Quotes Request (MsgID=10205)	108
Table 44: Delete All Quotes Response (MsgID=10206)	108
Table 45: Quote Action Request (MsgID=10201)	109
Table 46: Quote Action Response (MsgID=10202)	110
Table 47: Change Instrument State Request (MsgID=10300)	112
Table 48: Change Instrument State Response (MsgID=10301)	112
Table 49: Order Lock Request (MsgID=10108)	113
Table 50: Order Lock Response (MsgID=10110)	113
Table 51: Order Release Request (MsgID=10109)	114
Table 52: Order Release Response (MsgID=10111)	114
Table 53: Add Complex Instrument Request (MsgID=10306)	115

Table 54: Add Complex Instrument Response (MsgID=10307)	116
Table 55: MM Parameter Definition Request (MsgID=10308)	118
Table 56: MM Parameter Definition Request Acknowledge (MsgID=10309)	119
Table 57: Deal Item Maintenance Instruction Request (MsgID=10402)	120
Table 58: Deal Item Maintenance Instruction Response (MsgID=10403)	121
Table 59: Member Kill Switch Request (MsgID=10021)	122
Table 60: Member Kill Switch Response (MsgID=10022)	123
Table 61: Sequence Number Tracking Example.....	123
Table 62: Private Order Broadcast (MsgID=10104)	124
Table 63: Application Sequence Number Reset (MsgID=10020)	131
Table 64: Private Quote Broadcast (MsgID=10200)	132
Table 65: Product Reset Broadcast (MsgID=10406)	134
Table 66: Deal Item Broadcast (MsgID=10400)	136
Table 67: Trade Item Broadcast (MsgID=10401)	139
Table 68: Private MM Execution Report Broadcast (MsgID=10305)	142
Table 69: Business End of Stream Broadcast (MsgID=10404)	143
Table 70: Business End of Day Broadcast (MsgID=10405).....	144
Table 71: End of Stream Broadcast (MsgID=10015)	144
Table 72: Trading Session Marker Message Broadcast (MsgID=10407).....	145
Table 73: User Notification Broadcast (MsgID=10012)	146
Table 74: Invalid Data Broadcast (MsgID=10016).....	147
Table 75: News Broadcast (MsgID=10013)	148
Table 76: Heartbeat (MsgID=10011).....	148
Table 77: LTV Data Element	153
Table 78: Generic LTV Structure	154
Table 79: <i>RequestHeader</i> LTV structure encoding.....	154
Table 80: Generic LTV Array.....	155
Table 81: <i>SessionInstGrp</i> LTV array encoding	155
Table 82: <i>LogonReq</i> LTV message encoding	157
Table 83: Standard Options Strategies	215
Table 84: Standard Stock-Combo Strategies.....	215
Table 85: Away Market Codes	217
Table 86: Production Unicast IP Addresses and Ports	218

1. About This Document

This document is a programmer's reference guide intended to aid in the development and integration of trading applications to the ISE T7 trading system. It covers the general business behavior of the ISE markets and the technology standards and techniques employed to access the T7 system. The most recent version is available at <https://members.ise.com>.

1.1 ISE T7

ISE T7 is a high-throughput, low-latency trading platform that operates on the Linux platform using Intel-based servers. Primary and backup servers are located at ISE's primary data center with a disaster recovery system available at ISE's backup data center. Market participants are required to have network connectivity to the primary data center. It is also recommended having additional connections to the backup data center.

T7 offers four interfaces:

1. Direct Trading Interface (DTI): This is the binary trading interface to the core trading system. Members and third party software vendors may develop trading applications that communicate directly to the exchange system.
2. FIX Interface (IORS): This is an industry standard trading interface for sending orders and receiving execution reports in standard FIX ver. 4.2 format.
3. PrecISE Trade®: ISE's proprietary trader workstation that displays the ISE market prices and provides the full suite of trading functions available from ISE's exchanges.
4. Market Data Interface (MDI): High volume market data are distributed using Multicast (UDP), over high volume data lines or via cross connects at data centers.

NOTE: The ISE MDI does not include OPRA redistribution.

This document describes the Direct Trading Interface (DTI).

1.2 Intended Audience

This document is for programmers, analysts, and IT managers who are developing trading applications to connect directly to the ISE T7 trading system.

1.3 ISE Related Documents

The following are documents related to T7.

Table 1: ISE-Related Documentation

Document	Description	Location
Market Data Interface (MDI) Programming Manual	Guide to developing applications to consume ISE market and reference data.	https://members.ise.com

Document	Description	Location
ISE FIX Order Routing (IORS) Manual	Guide to developing trading applications to connect to the ISE FIX interface.	https://members.ise.com
Member Connectivity Guide	Technical guide for connecting to ISE data centers	https://members.ise.com
Member Simulation Guide	Information about ISE's simulation environment.	https://members.ise.com

1.4 Formatting Notes

Application message names referenced within the text are formatted in ***Bold Italic***. For example:

New Order Single

Deal Item Broadcast

Message field names referenced within the text are formatted in *italic*. For example:

MsgSeqNum

ClOrdID

Application message and field names within tables are unformatted.

Table rows describing repeating group fields are shaded to indicate nesting.

If there are discrepancies between this document and the supplied XML template file, please conform to the template file.

1.5 Acronyms

The following table lists the acronyms used in this document (descriptions and additional acronyms can be found in **Appendix C: Glossary**, on page 160):

Table 2: Acronyms

Acronym	Definition
ABBO	Away Market Best Bid and Offer
ACK	Acknowledgement
ALO	Add Liquidity Only Order
AMB	Away Market Better
AON	All Or None
BBO	Best Bid and Offer
BE	Big Endian
BPMM	Backup Primary Market Maker
BU	Business Unit
CMTA	Clearing Member Transfer Agreement
CMM	Competitive Market Maker
DIB	Deal Item Broadcast
DTI	Direct Trading Interface

Acronym	Definition
DMM	Directed Market Maker
DNR	Do Not Route
EAM	Electronic Access Member
ETF	Exchange-Traded Fund
FARMM	Far (non-ISE) Market Maker
FIX	Financial Information eXchange
FOK	Fill Or Kill
FX	Foreign Exchange
GDO	Guaranteed Directed Order
GTC	Good Til Canceled
GTD	Good Til Date
IBBO	ISE Best Bid and Offer
IOC	Immediate Or Cancel
ISE	International Securities Exchange
ISO	Intermarket Sweep Order
LE	Little Endian
LTV	Length Tag Value
MDI	Market Data Interface
ME	Matching Engine
MM	Market Maker
NBBO	National Best Bid and Offer
NTT	No Trade Through
OCC	Options Clearing Corp.
OPRA	Options Price Reporting Authority
PIM	Price Improvement Mechanism
PMM	Primary Market Maker
POB	Private Order Broadcast
PQB	Private Quote Broadcast
TCP/IP	Transmission Control Protocol/Internet Protocol
TIB	Trade Item Broadcast
TIF	Time-In-Force
TM	Trade Manager
TMS	Technology Member Services
UPC	Underlying Price Contingency
XML	eXtensible Markup Language

2. Introduction

2.1 Background

The International Securities Exchange (ISE) operates two equity options exchanges offering options trading on over 2,000 underlying equity, ETF, index, and FX products. As the first all-electronic options exchange in the U.S., ISE transformed the options industry by creating efficient markets through innovative market structure and technology. ISE continues to expand its marketplace through the ongoing development of enhanced trading functionality and new product introductions.

ISE is a member of the Options Price Reporting Authority (OPRA) and is an owner of the Options Clearing Corporation (OCC). ISE sends its quotes and trades to OPRA, and the trades are cleared by the OCC. Options contracts traded on the ISE exchanges are fungible with contracts traded on other U.S. options exchanges.

ISE combines electronic trading with auction market principles. Primary features of the ISE markets include:

- Market participants trade anonymously
- Customer orders have priority over non-customer orders
- Non-customers receive pro-rata priority in the order book

2.2 Auction Market

The ISE markets are auction markets for stock, ETF, and index options. All trading for both markets is centralized in the ISE T7 trading system.

2.2.1 Market Participant Anonymity

When a market participant enters an order or quote, the ISE system identifies the entity by its member code. The member code is not generally visible to other ISE market participants. The ISE system informs the parties involved that they have traded. Real-time counterparty information is provided via trade reports.

2.2.2 Market Continuity

The design of the ISE system facilitates depth and liquidity at the Best Bid and Offer (BBO) as well as at price levels below the best bid and above the best offer. A Market Maker (MM) has the ability to pre-set parameters for the number of contracts they are willing to execute at different price levels. When an MM's quote trades out, the system automatically regenerates the quote according to the MM parameters.

2.3 Membership Types

In order to participate in the ISE marketplace, a broker-dealer must be an ISE member. A new member must enter into an access agreement before connecting to the T7 trading system or using ISE software and equipment. There are three types of ISE members:

Table 3: Membership Types

Type	Description	Restrictions	Notes
Electronic Access Member (EAM)	An EAM is a broker-dealer that posts agency and proprietary orders on the exchange.	An EAM cannot enter quotes or perform any market-making activities	<ul style="list-style-type: none"> An EAM does not purchase a membership. After an EAM is approved as an ISE member, the EAM pays a monthly access fee to place orders in all of the options traded on the exchange.
Primary Market Maker (PMM)	An MM with significant responsibilities: <ul style="list-style-type: none"> Oversees the opening. Provides continuous quotes in all of their options classes, within an assigned bin. The quote size must be initially set to 10 contracts (minimum). Maintains tight markets. Resolves trading issues. 	A PMM cannot represent agency orders.	<ul style="list-style-type: none"> A PMM purchases or leases a PMM membership. One PMM is assigned to each bin.
Competitive Market Maker (CMM)	<ul style="list-style-type: none"> An MM that provides continuous quotes in at least 60% of the options classes to which it is appointed (90% for preferred MMs). The quote size must be initially set to 10 contracts (minimum). Each CMM quotes independently and maintains tight markets. 	A CMM cannot represent agency orders	<ul style="list-style-type: none"> A CMM purchases or leases CMM trading rights. The 1st trading right permits a CMM to quote in options classes that account for 20% of the volume on the exchange. Each additional trading right permits quoting in options classes that account for an additional 10% of volume.

NOTE: A member firm can obtain more than one membership. It is possible to be a PMM for one bin, obtain several CMM trading rights to provide markets in other bins, and have an EAM membership to enter agency and proprietary orders in all bins. Member firms that are both MMs and EAMs must conduct those activities separately. However, a member cannot be a PMM and CMM within the same bin.

2.4 Connecting to the ISE DTI

Before connecting to the ISE DTI, participants are required to:

- Access the ISE DTI member simulation system
- Complete the ISE conformance test in member simulation
- Complete the ISE connectivity test to the Production System

- Complete the ISE market control and certification test

Market participants must use the member simulation system in order to perform ISE conformance testing. Although the member simulation and production systems are functionally similar, there are differences between the two systems. These differences include (but are not limited to):

- The network connection to the member simulation system has different performance characteristics
- The member simulation system has less CPU and memory capacity
- The daily market schedule for the member simulation system has extended hours

2.4.1 Accessing the ISE DTI

To access the DTI in the production system, the market participant must:

- Enter into the developer's agreement with ISE.
- Submit a written request for access to the ISE member simulation system. The description of the monthly access fee is in the developer's agreement.
- Install a Virtual Private Network (VPN) or private line and purchase a router to connect to the member simulation system.
- Complete the ISE conformance test in the ISE member simulation environment.
- Complete the ISE connectivity test when connecting to the production system.

NOTE: On the member simulation system, all developers share the same central testing system. The ISE configures each developer with several user IDs, and allocates two or more stock options.

2.4.2 Completing ISE Conformance Test

ISE conducts conformance testing to document the major functions of the application, and to test how the application interacts with the trading system. This includes:

- Whether your application is being used for retrieving information, order routing, or MM quoting
- The transactions and broadcasts that your application uses

Contact ISE Technology Member Services (TMS) to review the application and complete the conformance test.

For more information on conformance testing, please refer to **Appendix D: DTI Conformance Test**, on page 164. If the application satisfies the ISE requirements, you will receive a letter from the ISE authorizing you to connect your application to the production system.

NOTE: Only ISE-approved applications can connect to the ISE. Approval is subject to the limitations and conditions specified in the access agreement.

2.4.3 Completing the ISE Connectivity Test to the Production System

The ISE connectivity test must be performed before your software application is installed in production at each site. Completion of this test is done after trading hours and at the market participant's site and performed on each installation of the market participant's application.

After receiving written ISE approval to connect to the production system, participants must:

- Contact ISE market control to advise them that they will be using their application in the ISE production system from the specified site
- Contact ISE TMS to schedule the connectivity test

Participants must demonstrate that their application can successfully:

- Log on to the production system using the assigned user ID and password
- Subscribe to the MDI
- Receive broadcast messages

The ISE monitors the performance of the market participant's application over several days. Market participants must provide a technical contact in the event that the application affects the production system.

If the application adversely affects the production system:

- ISE will notify the participant's contact to disconnect the software application from the system
- The application must be disconnected immediately upon request by ISE
- ISE may request more tests on the application when considering approval

2.4.4 Completing the ISE Market Control and Certification Test

The ISE market control and certification test certifies that new ISE MMs can use their applications to meet market-making responsibilities. Market participants must demonstrate that their traders can successfully perform market-making activities. ISE authorizes the traders to connect and trade in production after they have satisfied the testing requirements.

To perform the ISE market control and certification test:

- If you have coded your own application for the DTI, complete and pass the DTI conformance test
- If you use approved third-party software to trade at the ISE, the ISE recommends that a technology contact from the third-party firm is available during the test. Please contact tms@ise.com to schedule a test.

The test is performed on the ISE member simulation system with:

- ISE TMS
- ISE market control
- Other ISE users who trade as CMMs or PMMs in production
- Technology or developer contact to address technical questions

NOTE: The ISE reserves the right to refuse access to the production system if the application affects the production system. A trading functionality test is not performed when the application connects to the production system.

2.5 Developer Support

Support is available to developers who have completed the developer's agreement. Send technical questions via e-mail to: tms@ise.com. Answers to these technical questions may be distributed to other developers.

3. Session Information

Client applications connect to the trading system through the DTI. This section describes how the client application establishes a session with the system and sends/receives messages through the DTI.

The DTI is a socket-based interface that allows the client to:

- Establish a session with the trading system
- Send messages and receive responses
- Receive broadcast messages that provide status of orders, quotes, and trades

Subjects covered in this section include:

- FIX Semantics
- Length Tag Value (LTV) Encoding
- XML Interface Description
- Message Headers
- Establishing a DTI Session
- Application Messages
- Request and Response Messages
- Throughput Limits
- Endian Issues
- String Formats
- Development Tips
- Subscribing for Broadcasts
- Re-transmissions

3.1 FIX Semantics

All application messages between the client and the DTI follow (but are not explicitly compliant with) FIX V5.0 SP2 semantics. That is, each message is composed of a stream of <tag>=<value> fields. Some changes have been made for performance reasons. Message formats are described using FIX terminology wherever possible.

3.2 Length Tag Value (LTV) Encoding

All messages between the client and the DTI are encoded using LTV Encoding, a self-describing and self-delimiting algorithm for encoding and decoding binary application messages. Each LTV data element is encoded with a length description, which describes the length of the entire encoded element (*including* the length field itself); a tag identifier, which identifies the data; and the actual data. For a full description of LTV encoding, please refer to **Appendix A: LTV Encoding**, on page 150.

3.3 XML Interface Description

In addition to this document, a machine-readable description of the functional message layouts and data types is provided in Extensible Markup Language (XML) format. To automate DTI access, users may generate an access layer using the XML interface description.

3.3.1 XML Templates

The XML interface description is defined using three XML templates, provided by ISE in the template file, `isefix.xml`. The three templates are:

- **ApplicationMessages**, which defines the messages, field names, tags, and data types for all application messages used between the client and the DTI.
- **DataTypes**, which defines the field data types and, for some fields, the allowed values or codes.
- **Structures**, which provides a structured view of the application messages. (This template is not required for processing.)

NOTE: If there are discrepancies between this document and the `isefix.xml` template file, please conform to the template file.

3.3.2 ApplicationMessages Template

This template has three elements, or nodes: **ApplicationMessage**, **Member**, and **Group**.

3.3.2.1 ApplicationMessage Node

DTI application messages — *NewOrderSingle*, *MassQuote*, etc. — are defined by the **ApplicationMessage** node. Messages are referenced using the *name* attribute (“MassQuote”) or the *numericID* attribute (“10203”); however, messages across the interface use **only** the *numericID*.

(The *type* attribute can be used to cross-reference the Structures template.)

Application messages are comprised of **Member** and **Group** child-nodes, which define the fields and groups that make up the message.

3.3.2.2 Member Node

Discrete data elements (fields) — *Price*, *Quantity*, *Side*, etc. — are defined by the **Member** node. Fields are referenced using the *name* attribute (“Price”) or the *numericID* attribute (“44”); however, fields across the interface use **only** the *numericID* (tag).

- The *type* attribute defines a field’s data type and can be used to cross-reference the DataTypes template.
- The *usage* attribute defines the appearance of a field within a message and may be “mandatory” or “optional.” “Mandatory” fields must appear. “Optional” fields may appear, or may be conditionally required, depending on context.
- The *cardinality* attribute defines how many times the field may appear within the message or group context. All fields (elements defined by a **Member** node) have a *cardinality* of “1” — the field may only appear once in context.

3.3.2.3 Group Node

Groups — *Parties*, *LegOrdGrp*, etc. — are defined by the **Group** node, and contain all the same attributes as a **Member** node. Groups are referenced using the *name* attribute or the *numericID* attribute; however, groups across the interface use **only** the *numericID*.

(The *type* attribute can be used to cross-reference the Structures template.)

- Groups are comprised of **Member** and/or **Group** child-nodes, which define the fields and sub-groups that make up the group.
- The *usage* attribute defines the appearance of a group within a message and may be “mandatory” or “optional.” “Mandatory” groups must appear. “Optional” groups may appear, or may be conditionally required, depending on context.
- The *cardinality* attribute defines how many times a group may appear. A group *cardinality* of “unbounded” means that the group may appear any number of times, or not at all. A group *cardinality* of “1” means that the group may appear only once.
- LTV encoding and decoding rules for groups are slightly different depending on *cardinality*. Please see **Appendix A: LTV Encoding**, on page 150, for details.

NOTE: An “optional” group may contain a “mandatory” field. If the group is present in the message, then the field *must* also be present. If the group is not present in the message, then the field is not present.

3.3.3 DataTypes Template

This template has two elements, or nodes: **DataType** and **ValidValue**. In addition, this template uses four elemental data types described below: String, char, int, and float.

3.3.3.1 DataType Node

Message field *type* attributes — *MsgType*, *MsgSeqNum*, etc. — are defined by the **DataType** node, and are referenced using the *name* attribute. That is:

ApplicationMessage.Member.type == DataTypes.DataType.name.

The actual data type for the given *name* is defined by the *type* attribute.

All data types are based on elemental data types. An elemental data type is defined as a **DataType** where *name* == *type*. For example:

'DataType name="int" type="int".'

Depending on the field data type, it may be necessary to iterate through the DataTypes template multiple times before the elemental data type is discovered. For example:

ApplicationMessage template:

Member.type = “BidPx” →

DataTypes template:

DataType.name “BidPx” = DataType.type “PriceType” →

DataType.name "PriceType" = **DataType.type** "float" →

DataType.name "float" = **DataType.type** "float" — Elemental type

Data types may be comprised of **ValidValue** child-nodes, which enumerate the allowed values for that data type. Please note that not all values are valid for all fields of a given data type, or valid for a given field of that type in a given context.

3.3.3.2 ValidValue Node

Field values *may be* defined by the **ValidValue** node, and are referenced using the *name* attribute or the *value* attribute. Field values in messages across the interface, whose values are enumerated in this template, use **only** the *value*.

Field values enumerated in this template follow the FIX specification — they are standard values for the field types. Values that have not been enumerated are ISE specific and may be found in this manual, where appropriate.

3.3.3.3 Elemental Data Types

Data types described by the DataTypes template are based on four (4) elemental data types:

- **String:** Alphanumeric, free-format strings that can include any character or punctuation. All string fields are case sensitive (for example, capital ≠ Capital). Please note: trailing spaces in strings are ignored (for example "this string " ≈ "this string")
- **char:** Single character value that can include any alphanumeric character or punctuation. All char fields are case sensitive (for example, m ≠ M).
- **int:** Binary, signed 64-bit integer values. Minimum value = -9,223,372,036,854,775,808. Maximum value = 9,223,372,036,854,775,807.
- **float:** Binary, signed 64-bit integer values with an implicit precision of eight digits. Minimum value = -92,233,720,368.54775808. Maximum value = 92,233,720,368.54775807. For example, the decimal value 1.00 is sent as integer 100,000,000.

3.3.4 Structures Template

The Structures template simply provides a structured view of all application messages. It can be used, for example, to create static typedef structures for an application program. It is not necessary for processing.

3.4 Message Headers

All messages sent and received through the DTI have common message headers. There are three headers:

- **Request Header** (client to DTI)
- **Response Header** (Response back to client)
- **Broadcast Header** (Unsolicited, DTI to client; and *Heartbeats*)

3.4.1 Request Header

All messages sent to the DTI, except the **Heartbeat**, must include the Request Header:

Table 4: Request Message Header

Tag	Field	Req	Value(s)	Comments
11000	RequestHeader	Y		
35	> MsgType	Y		Request message type
34	> MsgSeqNum	Y		Ascending sequence number
50	> SenderSubID	N		Identification of PreciSE user

The client must supply a strictly monotonically increasing sequence number on each message sent on the session, starting with one (1). The DTI checks the sequence number on each message and sends a reject message if it is incorrect.

NOTE: There is no application message sequence number recovery mechanism in the DTI; the session *is disconnected* in the event of a sequence number reject.

3.4.2 Response Header

Every message sent to the DTI (except the **HeartBeat**) receives a response. All response messages include the Response Header:

Table 5: Response Message Header

Tag	Field	Req	Value(s)	Comments
11002	ResponseHeader	Y		
35	> MsgType	Y		Message Type
34	> MsgSeqNum	Y		Echoed from request msg.
52	> SendingTime	Y		When the DTI sent the response.
5979	> RequestTime	Y		When the request was received.
6228	> CompletionIndicator	N	Y=Last message	For multi-part responses.
1686	> ThrottleCountIndicator	N	0=outstanding request unchanged 1=Outstanding Requests reduced (default)	If not present, or set to one, the open window count is reduced. If set to zero, the open window size is unchanged.
1609	> ThrottleStatus	N	0=not delayed (default) 1=Response delayed as throttle exceeded	If not present, or set to zero, the response is not delayed. If set to one, the response has been delayed.

The *MsgSeqNum* in the response header is the same *MsgSeqNum* as the request message to which the response corresponds. Do not use *MsgSeqNum* as a sequence number to check for

missed messages. Responses may return in a different sequence than the sequence in which the messages were sent.

3.4.3 Broadcast Header

All Broadcast messages include the Broadcast Header:

Table 6: Broadcast Message Header

Tag	Field	Req	Value(s)	Comments
11006	BroadcastHeader	Y		
35	> MsgType	Y		Message Type
49	> SenderCompID	N		ApplResponseID value from subscription response Or "RXMIT" for retransmits Or "HB" for heartbeats
52	> SendingTime	Y		Outgoing timestamp of DTI (always present) Optional for client heartbeats to the DTI.

The Broadcast Header does not contain a sequence number. Individual broadcast types may have their own sequence number to use for re-transmissions. That sequence number appears in the body of the broadcast message.

SenderCompID represents the unique Application Response ID for that broadcast subscription.

NOTE: It is possible to set up more than one subscription to the same broadcast.

3.5 Establishing a DTI Session

The client application must first initiate a TCP/IP connection to the IP/port number provided by the exchange. Once the TCP/IP connection is established, the client sends a logon request message to establish the DTI session. The logon request must contain the business unit (BU) name, the user name, and the password provided by the exchange.

A two-step, direct-connect gateway connection option is also available. See **Section 3.5.3: Premium Access Connectivity**, on page 24, for more information.

3.5.1 Logon Request

The format of the **Logon Req** message is shown below:

Table 7: Logon Request (MsgID=10000)

Tag	Field	Req	Value(s)	Comments
10000		Y		Logon Req
<Request header>		Y	MsgType=A MsgSeqNum=1	Logon
108	HeartBtInt	N	$h \geq 100$	Milliseconds. If not present then The DTI will set default.
5364	MemberName	Y		BU name. For example, ABC01E
553	Username	Y		For example, 1
554	Password	Y		It is not possible to change the password through the logon request.
14001	SessionInstGrp	N	0-1	Actions to take if the session is disconnected.
8502	> SessionInstType	Y	0=None (default) 1=Cancel	Action to take
8503	> SessionInstScope	Y	1=All Orders 2=Non-persistent Orders	1 and 2 are mutually exclusive.

3.5.1.1 Heartbeat Interval

Use the heartbeat interval to specify the period in which the DTI sends heartbeats to the client and the interval the DTI checks for request messages from the application.

A heartbeat interval of zero (0) may be used on test systems, *only*, and indicates that DTI does not take any action for missed heartbeats. The minimum value is 100 milliseconds in the production system. The only upper limit is the size of a four-byte integer. If the field is not supplied, then the DTI uses the default value.

3.5.1.2 Session Instructions

By default, when a session is disconnected, quotes entered by that session *are deleted* and orders *remain active*.

Session instructions may specify a different action to take, for orders only, upon a session disconnect.

Instructions to delete orders can be specified for all orders (persistent and non-persistent), or non-persistent orders.

(Session instructions may also be optionally specified on the **Logout Req** message.)

NOTE: If session instructions are not provided, orders *remain active* if the session is disconnected. Quotes are always deleted.

3.5.2 Logon Response

The response message indicates that the logon was successful and contains information about the session including the heartbeat interval, system partitions, and throttle limits.

Table 8: Response to Logon Request (MsgID=10001)

Tag	Field	Req	Value(s)	Comments
10001				Logon Resp
<Response header>		Y	MsgType=A	Logon
108	HeartBtInt	Y		The heartbeat interval. DTI overrides out-of-range value or sets to default if logon omits the field.
14003	PartitionGrp	Y	1 or more	The List of known partitions
5948	> PartitionID	Y		Partition ID
1301	> MarketID	N	XISX (default) GMNI MCRY	ISE Exchange ISE Gemini Exchange ISE Mercury Exchange
14030	ThrottleParamsGrp	N	3	Description of throttles
1611	> ThrottleAction	Y	1=Queue Outbound 2=Reject 3=Disconnect	
1612	> ThrottleType	Y	0=Inbound Rate 1=Outstanding Requests	ThrottleType 0 (inbound rate) only possible for ThrottleAction 1 (queue outbound).
1613	> ThrottleNoMsgs	Y		Limit for ThrottleAction 3 (disconnect) refers to number of rejected msgs. after ThrottleAction 2 (reject) was triggered.
1614	> ThrottleTimeInterval	N		Conditionally required for ThrottleType = 0 (Inbound Rate).
1615	> ThrottleTimeUnit	N	0=Seconds (default) 1=Tenths Sec 2=Hundredths Sec 3=milliseconds	Conditionally required for ThrottleType = 0 (Inbound Rate).
1409	SessionStatus	N	9=Session Takeover	Indicates that another session of the same user existed and was terminated.
1408	DefaultCstmApplVerID	Y		isefix.xml template version
58	Text	Y		Session ID assigned by DTI.

Partitions: The list of partitions must be saved after logging on to the DTI. They are used when requesting re-transmissions of broadcasts. There are approximately 15 partitions, with a maximum of 99.

Heartbeat Interval: The DTI overrides the specified value on the logon message if it is omitted or out of range. The minimum value is 100 milliseconds.

Throttle Parameter Group: Provides the values for the throttle parameters as described in **Appendix K: Throttles**, on page 210.

3.5.3 Premium Access Connectivity

ISE offers members a low-latency, direct-connect gateway option. Connectivity via this option is accomplished using a two-step login process.

NOTE: Whether or not this connection option is implemented, members should maintain their ability to connect via the normal, one-step login process.

The client application first initiates a TCP/IP connection to the *Connection Gateway* — the IP/port numbers are provided by the exchange. Once that connection is established, the client sends a **Gateway Request** message. The **Gateway Request** must contain the BU name, the user name, and the password provided by the exchange. The Connection Gateway can be used for no other purpose than to send the **Gateway Request** message.

The **Gateway Response** contains the IP/Port of a direct-connect gateway where the member application can establish an active DTI session. Once the **Gateway Response** message is received, the member application must drop the Connection Gateway connection, reestablish the connection at the assigned DTI gateway, and send the **Logon Req** message (see **Section 3.5.1, Logon Request**, on page 22). The direct-connect session must be established within a certain time interval (e.g., 10 seconds) after receipt of the gateway assignment or else the **Logon Req** is rejected, and the member application must resend the **Gateway Request**.

If the active, direct-connect DTI session is ever disconnected, for any reason, the member application must reconnect to the Connection Gateway, resend the **Gateway Request** message, and receive a new direct-connect DTI gateway assignment.

3.5.3.1 Gateway Request

The format of the **Gateway Request** message is shown below:

Table 9: Gateway Request (MsgID=10018)

Tag	Field	Req	Value(s)	Comments
10018		Y		Gateway Req
<Request header>		Y	MsgType=A MsgSeqNum=1	Logon
5364	MemberName	Y		BU name ("ABC01E")
553	Username	Y		User name ("1")
554	Password	Y		

3.5.3.2 Gateway Response

The response message indicates that the **Gateway Req** was processed successfully, and provides an IP address and port number to which the application should connect.

Table 10: Response to Gateway Request (MsgID=10019)

Tag	Field	Req	Value(s)	Comments
10019				Gateway Resp
	<Response header>	Y	MsgType=5	Logout
14035	GatewayGrp	Y	> 0	At least one gateway IP addresses/port number
8644	> GatewayID	Y		IP Address
8645	> GatewaySubID	Y		Port Number
1409	SessionStatus	Y	4=Session logged out	

3.5.4 Suggested Logon Sequence

An example of a logon sequence:

Connect to the market data feeds to get reference data and market data:

- Connect to the Reference Data feed, as described in the *T7 Market Data Interface (MDI) Programming Manual*, to obtain a complete description of all products and instruments in the market.
- Connect to the Top Quote feed and/or Depth feed to get the market data.
- Connect to the Reference Data incremental to receive intraday additions of instruments.

Logon to the DTI to establish a trading session:

- Open a TCP/IP socket to the specified IP address and port.
- (Optional: Send a **Gateway Request** message for DTI gateway assignment.)
- (Optional: Open a new TCP/IP socket to the specified IP address and port.)
- Send a logon message and wait for the response.
- If successful, process the response to get your session information including the request limit and throttle limits.
- Send messages to subscribe for the required broadcast messages. This includes order status, quote status, trades, and deals. It is possible to receive broadcasts on a separate session to that supplying the quotes/orders.

The client is now ready to send application messages to the trading system.

3.5.5 Managing Unsuccessful Logons

The following suggestions are to avoid a situation where the client goes into a loop trying to logon to the DTI:

- Raise an alert each time the logon is unsuccessful
- Ensure there have not been two established sessions with the same user ID that are continuously logging each other out
- Do not perform an automatic retry if the password is incorrect
- Wait 10 seconds before attempting to log on again
- Do not perform an automatic retry more than three times

3.6 Terminating a DTI Session

The client should send a logout message to perform a controlled disconnect from the DTI. As with the **Logon Req**, the client can optionally specify instructions to be taken against orders submitted by this session. Instructions provided in this message override instructions submitted with the **Logon Req** message. If the client does not specify anything in the logout message, the instructions given with the **Logon Req** message continue to be valid and are applied as part of the controlled disconnect. The DTI attempts a forced logout of a session if the TCP/IP session is dropped or if it fails to receive a message for three consecutive heartbeat intervals. After a successful logout, the client should shut down the connection and close the socket.

3.6.1 Logout Request

The **Logout Req** message is used to gracefully log out of, and disconnect from the DTI. Refer to the **Logon Request**, on page 21, for information on the session instructions. The format of the logout request is shown below:

Table 11: Logout Request (MsgID=10002)

Tag	Field	Req	Value(s)	Comments
10002				Logout Req
<Request Header>		Y	MsgType=5	Logout
14001	SessionInstGrp	N	0-1 elements	
8502	> SessionInstType	Y	0=None 1=Cancel	
8503	> SessionInstScope	Y	1=All Orders 2=Non-persistent Orders	

3.6.2 Response to the Logout Request

The response to a logout request indicates that the session has been terminated successfully. The port is closed one second after the response message is sent:

Table 12: Response to Logout Request (MsgID=10003)

Tag	Field	Req	Value(s)	Comments
10003				Logout Resp
<Response header>		Y	MsgType=5	Logout

3.7 Maintaining a Session

3.7.1 Message Sequence Numbers

The message sequence number (*MsgSeqNum*) in the request header must increment with each message, starting with the logon message, sequence number 1. If there are any unexpected sequence numbers, sequence number gaps, or duplicate sequence numbers, the request message is rejected with a sequence number error and **the session is disconnected**.

There is no application message sequence number recovery mechanism in the DTI. All client connections (including reconnection after disconnect) are considered “new,” and all logon requests must contain message sequence number 1.

3.7.2 Heartbeats

Heartbeat broadcast messages are transmitted between the client application and the DTI as a keep-alive mechanism across an otherwise idle connection. Heartbeats are not sent as a response to the peer’s **Heartbeat** message, but are independently timed events.

Heartbeats from the DTI are sent automatically, there is no subscription request.

The **Heartbeat** message is described in **Section 5.33, Heartbeat**, on page 148.

3.7.2.1 Heartbeats: Client to DTI

The client application must maintain an active session by sending at least one message, either a request message or a **Heartbeat** broadcast message, within every heartbeat interval to maintain the session. The DTI forcibly disconnects a session if the client is inactive for three-times the heartbeat interval specified in the logon response.

NOTE: You can send a **Heartbeat** even if you have exceeded your Outstanding Request Limit. Heartbeats do not count against, nor do they reduce, any of the throttle counters.

3.7.2.2 Heartbeats: DTI to Client

A client application can check on the state of the exchange trading system. If there are no broadcast messages sent, a **Heartbeat** broadcast is sent by the DTI to the client every heartbeat interval. If a heartbeat is not received by the client in two intervals, the client should initiate a recovery action. The recovery action can include:

- Raise an alarm/call ISE Technical Support

- Close the existing TCP/IP session, wait 5 seconds then attempt to start a new session. (However, do not attempt more than three times in case there are two sessions with same password continuously logging each other out.)

3.7.3 Daily Logout

Client applications must disconnect from the DTI each day after trading. The system performs a forced logout overnight after which time the client may log back in.

3.8 Sending Application Messages

The client application can begin to send application messages as soon as the logon response has been received. This section contains some guidelines to assist developers in connecting to the DTI.

3.8.1 Outstanding Request Limit

The client may not have more open requests (messages waiting for a response) than the request limit. An attempt to exceed the request limit causes further requests to be rejected. If too many requests are rejected due to exceeding the request limit, the session is terminated.

- Heartbeat messages do not count as request messages. If the client is waiting for a response, it must still send a heartbeat message every heartbeat interval to maintain the session.
- Some request messages can receive multiple responses. In this case, only the first response is counted. The Response Header contains the *ThrottleCountIndicator* field. If the *ThrottleCountIndicator* is set to "N," the response does not count in the reduction of the open message count.
- Broadcast messages do not reduce the count of open requests.

For a full description of throttle parameters, including the outstanding request limit, please see **Appendix K: Throttles**, on page 210.

3.8.2 Transaction Limit

A DTI session is limited to a specified throughput limit that is specified in the logon response. If this limit is exceeded, the DTI delays response messages to the client in order to limit the session throughput.

The DTI counts the number of transactions sent in each interval, as specified in the logon response. If the number of transactions exceeds the value in the transaction limit, responses are delayed.

Required heartbeats do not count against the transaction limit.

NOTE: The transaction limit does not guarantee throughput rates. As load fluctuates in the exchange system, throughput rates may vary.

For a full description of throttle parameters, including the transaction limit, please see **Appendix K: Throttles**, on page 210.

3.8.3 Endianness and Byte Swapping

Binary numbers sent to and received from the DTI are always in little-endian format. Depending on your system architecture, it may be necessary to byte-swap numbers before sending or processing received messages.

3.8.4 String Formats

String fields used in the DTI messages have variable lengths. The field descriptions contained herein specify the maximum lengths of each field. LTV encoding specifies the *transmitted* length of each field. String fields should not be space padded, or null terminated, nor should they contain leading spaces. Trailing spaces in string fields are ignored, and may be stripped. (For example, “this string” ≈ “this string”.)

3.8.5 Repeating Groups

The FIX “Number of...” fields are not used by the DTI. Instead, the DTI uses custom group names and tags which, via LTV encoding, encapsulate the repeating groups by size rather than number.

Because repeating groups are not bound by “Number of...” fields, the actual number of repeating groups (on broadcasts, for example) is not known until the message is parsed and decoded.

3.9 Response Messages

Every request message receives a response, either acknowledging or rejecting the request. A **Pending** response may be sent for a request which cannot immediately be processed, or if the final status of the request is not yet determined. A pending response is followed by a final acknowledgement or reject.

3.9.1 Acknowledgment Responses

Acknowledgements (ACKs) are specific to the request. Most order-related ACKs carry minimal business information and are used primarily for flow-control. For MMs, most quote- or instrument-related ACKs provide feedback that the requested action did or did not occur. The individual ACK messages are described in detail in **Section 5, DTI Message Specifications**, beginning on page 82.

3.9.2 Reject Response

If a request message is rejected, the application receives a **Reject** message. The reject message contains an error code and error text.

The format of the **Reject** message is shown below:

Table 13: Reject Message Format (MsgID=10010)

Tag	Field	Req	Value(s)	Comments
10010		Y		Reject
	<Response header>	Y	MsgType=U3	Reject
373	SessionRejectReason	Y		
58	Text	Y	String (unbounded)	
1409	SessionStatus	N		

3.9.2.1 Notable Reject Messages

This section describes reject messages that should be noted by members. Special action may be indicated upon receipt of one of these.

- **Transaction Lost In Core** — The GW has timed-out waiting for a response from the core. Your transaction *may be processed*.
Recommended action: Stop sending new transactions. Call Market Operations to determine the actual status of your transaction.
- **Transaction Timeout** — The GW has timed-out trying to forward your transaction to the core. Your transaction *is not processed*.
Recommended action: This is usually a temporary condition and no special action is required.

NOTE: A **Reject** message may *not* contain the *MsgSeqNum* field in the response header. This can occur if the request transaction is not parse-able, or prior to the Gateway terminating the connection for some session violation.

3.9.3 Pending Response

A **Pending Response** is sent when a request cannot be immediately, fully processed or validated, and indicates that the final status of the request is not yet known. For example, a cancel request against a frozen or locked order.

The format of the **Pending Response** message is shown below:

Table 14: Pending Response (MsgID=10014)

Tag	Field	Req	Value(s)	Comments
10014		Y		Pending Response
	<Response header>	Y	MsgType=U2	Pending
60	TransactTime	N		
5946	PendingReason	Y		
58	Text	N		

3.9.4 Invalid Response

An **Invalid Response** indicates that the DTI could not correctly format the proper response message, but is nevertheless providing a response for the request.

This response indicates an internal error — please call technical support.

The format of the **Invalid Response** message is shown below:

Table 15: Invalid Response (MsgID=10017)

Tag	Field	Req	Value(s)	Comments
10017		Y		Invalid Response
<Response header>		Y	MsgType=U4	InvalidResponse

3.10 Broadcast Messages

The ISE system advises market participants of certain events by sending broadcast messages. Public market data, such as the ISE BBO (IBBO), depth, and trades (ticker) are sent as multicast data feeds through the MDI; these public market data are not available through the DTI.

Private trading messages, such as trades and status information related to specific quotes and orders, are sent to each user of the owning BU over the DTI. These unsolicited trading messages sent over the DTI are called broadcasts, distinguishing them from requests and responses. This section describes how broadcast messages are sent to users over the DTI.

Broadcast messages work as follows:

- Once logged on, the session is not automatically subscribed for any broadcasts except heartbeats (see **Section 2.7.2, Heartbeats**, on page 27). The client software must subscribe for the broadcast message types it wants to receive in the session.
- A client application can subscribe or unsubscribe to the available broadcast message types using subscribe and unsubscribe request messages. Once subscribed, the specified broadcast messages are sent to the client from that time forward.
- Some broadcast types are recoverable: they are saved by the system and the client can request a re-transmission of specific messages. Other broadcast types are not recoverable.
- When there are multiple clients logged on for the BU, a client can specify if it wants to receive the status broadcast (quote status or order status) for just its own user, or all broadcasts for the BU.
- A subscription request specifies the broadcast type and indicates whether it is for the BU or just for the client's own session (applies to quote status or order status broadcasts only).
- Only recoverable broadcasts contain a sequence number and a partition ID. The sequence number is ascending within each partition ID for the BU. Recoverable broadcasts can be re-transmitted by requesting a range of sequence numbers and the partition ID. For example, if a user in BU ABC01M subscribes for the Trade broadcast, the first message from each partition contains message sequence number one.
- Broadcast messages do not affect any of the throttle counters.

The complete list of available broadcast types is shown below:

Table 16: Broadcast Types

ID	Name	Mode
O	Order status	Recoverable
Q	Quote status	Unrecoverable
M	MM information	Recoverable
T	Trades	Recoverable
D	Deals	Recoverable
N	Market message	Recoverable
S	Service Availability	Unrecoverable

3.10.1 Subscribing to Broadcasts

A client must subscribe to broadcast messages that it wants to receive by sending in a subscription request message.

- The **Subscribe Req** message is used to request broadcast subscriptions. This message uses the FIX “Application Message Request” message type.
- Each broadcast type requires a separate subscription request.
- Only one subscription is allowed per request.
- All subscriptions are market-wide by default, but a specific partition, or partitions, may be optionally specified using the *Partition* group. The *Partition* group is not applicable for the Market Messages (News) broadcast.
- For market-scope subscriptions, if *MarketID* is not specified, the default value (“XISX”) is used.
- The *RefApplID* field specifies the message broadcast type that is being requested.
- The *ApplReqScope* field specifies if the DTI should send all messages for the BU or just those that relate to the user ID. This feature to filter broadcasts by user ID is available only for the order and quote status broadcasts.

If the request is for Service Availability (*RefApplID* = “S”), the Matching Engine (ME) or Trade Manager (TM) service must also be specified in the *ServiceID* field. Please see **Section 5.30, User Notification Broadcast**, on page 146, for a complete description of this subscription.

The format of the **Subscribe Req** message is shown below:

Table 17: Subscribe Broadcast Request (MsgID=10004)

Tag	Field	Req	Value(s)	Comments
10004		Y		Subscribe Req
	<Request header>	Y	MsgType=BW	ApplicationMessageRequest
1347	ApplReqType	Y	1=Subscription	Type of request

Tag	Field	Req	Value(s)	Comments
8569	ApplReqScope	N	0=All messages (default) 1=Own messages	Filter option (default is entire BU). "Own messages" only available for quote status and order status broadcasts
14029	ApplIDRequestGrp	Y	1	Only one subscription per request
1355	> RefApplID	Y	O=Order status Q=Quote status M=MM information T=Trade item D=Deal item N=Mkt. Messages (News) S=Service Availability	Broadcast type
1301	> MarketID	N	XISX (default) GMNI	Required if RefApplID = N
7930	> ServiceID	N	ME=Matching Engine TM=Trade Manager	Required if RefApplID = S
14003	> PartitionGrp	N	0 or more	List of partitions for which broadcast subscription is requested. N/A for News.
5948	>> PartitionID	Y		

3.10.2 Response to the Subscribe Request

Subscription requests generate responses.

- A mal-formed or otherwise incorrect subscription request is rejected.
- The **Subscribe Resp** message indicates that the request was successful.
- The *ApplResponseID* field provides a unique identifier for each subscription. This ID is provided on the subscribed broadcast message in the *SenderCompID* field in the broadcast header.

NOTE: It is possible to create duplicate subscriptions (on purpose, or accidentally) — this is not an error. Each subscription has a unique *ApplResponseID*.

The format of the **Subscribe Resp** message is shown below:

Table 18: Subscribe Response (MsgID=10005)

Tag	Field	Req	Value(s)	Comments
10005		Y		Subscribe Resp
<Response header>		Y	MsgType=BX	ApplicationMessageRequestAck
1347	ApplReqType	Y	1=Subscription	

Tag	Field	Req	Value(s)	Comments
1353	ApplResponseID	Y		Subscription ID
14032	ApplIDRequestAckGrp	N	0 – 1	Present on “Service Availability” response
1355	> RefApplID	Y	S=Service Availability	Broadcast type
7930	> ServiceID	Y	ME=Matching Engine TM=Trade Manager	
14003	> PartitionGrp	N	0 or more	One entry per partition
5948	>> PartitionID	Y		
5969	>> PartitionStatus	N	0=Unavailable 1=Available	Current Service ID status

3.10.3 Unsubscribe Request

A client can unsubscribe to a broadcast type, so that broadcast messages are no longer sent to that session.

- The **Unsubscribe Req** message is used for this purpose.
- The unsubscribe request *must* be formatted as per the subscription request. In addition, the *ApplResponseID* from the subscription response must be specified in the *RefApplResponseID* field.
- One request message is sent for each broadcast to be unsubscribed.

The format of the **Unsubscribe Req** message is shown below:

Table 19: Unsubscribe Request (MsgID=10006)

Tag	Field	Req	Value(s)	Comments
10006		Y		Unsubscribe Req
<Request header>		Y	MsgType=BW	ApplicationMessage Request
1347	ApplReqType	Y	4=Unsubscribe	Type of request
8569	ApplReqScope	N	0=All messages (default) 1=Own messages	Scope of request
14029	ApplIDRequestGrp	Y	1	One per request
1355	> RefApplID	Y	O=Order status Q=Quote status M=MM information T=Trade item D=Deal item N=Market Messages (News) S=Service Availability	
1301	> MarketID	N	XISX (default) GMNI	Required if RefApplID = N
8514	> RefApplSeqMode	N	0=Unrecoverable 1=Recoverable	Deprecated

Tag	Field	Req	Value(s)	Comments
8515	> RefApplResponseID	Y		ApplResponseID from subscription request response.
7930	> ServiceID	N	ME=Matching Engine TM=Trade Manager	Required if RefApplID = S
14003	> PartitionGrp	N	0 or more	List of partitions for which broadcast subscription was requested (empty list = all partitions)
5948	>> PartitionID	Y		

3.10.4 Response to the Unsubscribe Request

Unsubscribe requests generate responses.

- A mal-formed, or otherwise incorrect unsubscribe request is rejected.
- The **Unsubscribe Resp** message indicates that the subscription request was successful.

The format of the **Unsubscribe Resp** message is show below:

Table 20: Unsubscribe Broadcast Response (MsgID=10007)

Tag	Field	Req	Value(s)	Comments
10007		Y		Unsubscribe Resp
<Response header>		Y	MsgType=BX	ApplicationMessageRequestAck
1347	ApplReqType	Y	4=Unsubscribe	Request and broadcast types implicitly defined by the request MsgSeqNum in the response header

3.10.5 Retransmissions

Clients can request the retransmission of the recoverable broadcast messages, as defined in **Table 16: Broadcast Types**, on page 32. The retransmission request specifies the broadcast type, the partition ID, and the sequence number range. The response indicates the range of messages that will be re-transmitted. This might only be a subset of the range requested, or no messages. The re-transmitted messages, if any, are the next messages sent on that session by the DTI.

- Only one broadcast type and partition is allowed per request. The partition IDs are provided in the logon response message and must be saved for this purpose. (Product to partition mappings are available in the Reference Data feed in the MDI.)
- If the response indicates only a subset of the requested range, the client must keep requesting retransmission, adjusting the sequence numbers as appropriate, until all missing messages have been recovered for that partition.

3.10.5.1 Retransmission Request Message

- The **Retransmit Req** message is used to request a retransmission of recoverable broadcasts.
- The *RefApplID* field specifies the broadcast type to be re-transmitted.
- The *PartitionID* field, within the *Partition* group, specifies the partition for the retransmitted broadcasts. The *Partition* group is not applicable for News.
- Optionally, the *ApplBegSeqNum* and *ApplEndSeqNum* fields provide the broadcast sequence number range. If a starting sequence number is not specified, it is assumed to be one. If an ending sequence number is absent, it is assumed to be infinity (“all available messages”).

Note: MAX_INT64, or some other sufficiently large integer, may be used to represent infinity.

The format of the **Retransmit Req** message is show below:

Table 21: Retransmission Request (MsgID=10008)

Tag	Field	Req	Value(s)	Comments
10008		Y		Retransmit Req
<Request Header>		Y	MsgType=BW	ApplicationMessageRequest
1347	ApplReqType	Y	0=Retransmission	Type of request
1301	MarketID	N	XISX (default) GMNI	Recommended if RefApplID = N
14029	ApplIDRequestGrp	Y	1	one per request
1355	> RefApplID	Y	O=Order status M=MM information T=Trade item D=Deal item N=Market Messages (News)	Broadcast type
14003	> PartitionGrp	N	0 or 1	Req'd except for News request
5948	>> PartitionID	Y		
1182	> ApplBegSeqNum	N	Default=1	First sequence number to be retransmitted.
1183	> ApplEndSeqNum	N	Default=infinity	Last sequence number to be retransmitted.

3.10.5.2 Retransmission Response Format

Retransmission requests generate responses.

- A mal-formed or otherwise incorrect retransmission request is rejected. A request for “out-of-range” sequence numbers is not incorrect.
- The **Retransmit Resp** message indicates that the retransmission request was successful.

- The *RefApplLastSeqNum* field contains the last sequence number of that broadcast type for that partition as per the request. If no broadcasts have been created (for example, at the start of day), this field contains the value zero (0).
- The response *ApplEndSeqNum* **may be greater** than the requested *ApplEndSeqNum* if the requested *ApplEndSeqNum* is inside a sequence number gap (order status and MM information broadcasts, only).
- The response *ApplBegSeqNum* and *ApplEndSeqNum* fields indicate the range of broadcasts to be retransmitted. If there are no broadcasts to be retransmitted (for example, at the start of day, or because the requested sequence numbers are “out-of-range”), these fields are not present. Because the DTI limits the number of broadcasts to be retransmitted, *ApplEndSeqNum* may be less than *RefApplLastSeqNum*.
- The re-transmitted broadcast messages immediately follow the response message and contain the text “RXMIT” in the *SenderCompID* field of the broadcast header.

The format of the **Retransmit Resp** message is shown below:

Table 22: Retransmission Response (MsgID=10009)

Tag	Field	Req	Value(s)	Comments
10009		Y		Retransmit Resp
<Response header>		Y	MsgType=BX	ApplicationMessageRequestAck
1347	ApplReqType	Y	0=Retransmission	
14032	ApplIDRequestAckGrp	Y		Only one entry possible
1355	> RefApplID	Y	O=Order status M=MM information T=Trade item D=Deal item N=Market Messages (News)	Broadcast type
14003	> PartitionGrp	N	0 or 1	Present except for News response
5948	>> PartitionID	Y		
1182	> ApplBegSeqNum	N		First sequence number to be re-transmitted
1183	> ApplEndSeqNum	N		Last sequence number to be re-transmitted (\leq RefApplLastSeqNum)
1357	> RefApplLastSeqNum	Y		Current last sequence number for requested b'cast type

3.10.5.3 Retransmission Request Examples

The following examples are provided to clarify retransmission request behavior.

Example 1 — Start of day, no broadcast messages:

Request —

ApplBegSeqNum = 1 (or not present)
ApplEndSeqNum = MAX_INT (or not present)

Response —

RefApplLastSeqNum = 0

No messages to retransmit.

Done.

Example 2 — 20 messages numbered 1 - 20, retrieve all:

Request —

ApplBegSeqNum = 1 (or not present)
ApplEndSeqNum = MAX_INT (or not present)

Response —

ApplBegSeqNum = 1
ApplEndSeqNum = 10
RefApplLastSeqNum = 20

Messages 1 through 10 are retransmitted.

Second Request —

ApplBegSeqNum = 11
ApplEndSeqNum = MAX_INT (or not present)

Response —

ApplBegSeqNum = 11
ApplEndSeqNum = 20
RefApplLastSeqNum = 20

Messages 11 through 20 are retransmitted.

Done.

Example 3 — 20 messages numbered 1 - 20, retrieve some:

Request —

ApplBegSeqNum = 3
ApplEndSeqNum = 15

Response —

ApplBegSeqNum = 3
ApplEndSeqNum = 12
RefApplLastSeqNum = 20

Messages 3 through 12 are retransmitted.

Second Request —

ApplBegSeqNum = 13

ApplEndSeqNum = 15

Response —

ApplBegSeqNum = 13

ApplEndSeqNum = 15

RefApplLastSeqNum = 20

Messages 13 through 15 are retransmitted.

Done.

Example 4 — 20 messages numbered 1 - 10, GAP, 16 - 25, retrieve some:

Request —

ApplBegSeqNum = 3

ApplEndSeqNum = 15

Response —

ApplBegSeqNum = 3

ApplEndSeqNum = 16 — **Greater than original request**

RefApplLastSeqNum = 25

Messages 3 through 10 are retransmitted.

Message 16, an **Application Sequence Number Reset Broadcast**, is retransmitted.

Done.

Example 5 — 20 messages numbered 1 - 20, request out of range:

Request —

ApplBegSeqNum = 30

ApplEndSeqNum = MAX_INT (or not present)

Response —

RefApplLastSeqNum = 20

No messages to retransmit.

Done.

4. ISE Market Description

This section describes how the markets appear through the DTI and how the various messages are used to perform the different trading functions. Subjects covered in this section are:

- Reference Data
- Market Participant Structure
- Risk Protection & Management
- Pro-Rata Sharing Algorithm
- Basic Order Functions
- Deals and Trades
- Special Order Types
- Cross Orders
- Auctions
- Quote Related Functions
- Complex Orders and Quotes

4.1 Reference Data

NOTE: Reference data are not available through the DTI.

- T7 provides reference data as a continuous market data feed. T7 can host multiple exchanges, each with many products, and within each product, many instruments. For more information about the reference data feed, please refer to the *Market Data Interface (MDI) Programming Manual*.
- The reference data feed disseminates a snapshot of a product followed by a snapshot of each instrument within that product. The feed sends out snapshots for each product and instrument at regular intervals. When there are multiple exchanges involved, each exchange is sent on a separate feed. If a product or instrument is added intra-day, an incremental message is sent out immediately and the new product/instrument appears in the next complete snapshot cycle.
- An exchange (market) is identified using a unique string identifier called the market ID. ISE identifies its markets using the ISO 10383 Market Identifier Code (MIC).
- An instrument is identified with unique 8-byte binary identifier, called the instrument ID. If the same instrument is traded in multiple markets, it has a different instrument ID in each market. The instrument snapshot record fully describes the instrument including OSI name, its owning product ID, the ISE display name, and contract description. If the instrument is a complex instrument (for example spread, straddle, etc.), it also contains a list of legs. Each leg contains the instrument ID, the side, and the ratio.
- A product is a group of Instruments. A product is identified with a unique 4-byte binary identifier, called the product ID. If the same product is traded in multiple markets, it has a different product ID in each market. The product snapshot record fully describes the product, including the exchange and market segment to which the product belongs (for example,

XISX/Primary Market/Bin 4). It also specifies various trading parameters such as tick interval, the minimum size for a block auction, the PMM, matching model (pro-rata or price/time), and the market data feed IP addresses and ports.

- DTI applications **must** get the product and instrument IDs each day from the reference data feed in order to send and receive messages with the DTI. The Reference Data feed can be processed prior to logging in to the DTI. (A static reference data file is also available daily via FTP.)

4.2 Market Participant Structure

A market participant must become a member of the exchange and have a letter of clearing authorization from a clearing firm before being permitted to access T7 for trading purposes.

4.2.1 Participants, Business Units and Users

The trading system recognizes three entities in a member organization when connected to T7:

- A member firm is a participant. A participant may have multiple “Business Units” that conduct trading operations independently of each other.
- A Business Unit (BU) is considered the owner of the orders and trades. A BU connects to system through the DTI by logging on with a BU name, User ID, and password.
- The User ID identifies the client to the DTI. The User ID and password are validated by the system when establishing a session. The User ID identifies the owning BU. Each User ID has a role to identify which functions may be performed by that user on the system. The User ID also has specified throttle parameters that are configured by the ISE.

A market participant may have multiple BUs to partition trading activity. One BU cannot see another BU’s orders and trades, even when they belong to the same market participant. However, all users belonging to the same BU can see and maintain all trades, orders, and quotes that belong to that BU, even when entered by another user.

A BU also can have multiple users logged into the system at the same time. Multiple user sessions may be required for various reasons including:

- Capacity
- Client application partitioning
- Backup system configurations

Market participants may designate specific users with special privileges to block new incoming order transactions for one or more of their BUs, or for a specific user (or users) of a BU. In addition to disabling new order flow, all open orders for that BU/user are deleted.

4.3 Risk Protection & Management

T7 provides a number of both passive and active mechanisms to help members manage risk.

- Automatic quote deletion — On disconnect, all quotes for the disconnecting session are automatically deleted.

- Automatic order deletion — By default, orders are not deleted on disconnect; however members may specify on their logon request that orders entered by that session should be deleted upon disconnect. This is described in detail in **Section 3.5, Establishing a DTI Session**, on page 21.
- Quote curtailment — MMs must specify quoting parameters so that their quotes are “inactivated” (taken out of the market) if executed quantity exceeds a configured threshold within a configured period of time. This is described in detail in **Section 4.10.4, Market Maker Protection (Curtailment/Speed Bump)**, on page 70.
- Quote inactivation — MMs can inactivate and delete their quotes at any time, either across the entire market, or for specific products, by sending a **Quote Action Request** message. This is described in detail in **Section 4.10.3, Quote Inactivation**, on page 69.
- Member Kill Switch — As described in **Section 4.2, Market Participant Structure**, on page 41, a specially designated user may be used to actively disable order entry for another BU or session. In addition to disabling new order entry, all open orders for the disabled BU/session are deleted. The kill switch transaction is described in **Section 5.18, Member Kill Switch Request**, on page 121.

4.4 Pro-Rata Sharing Algorithm

The ISE options markets use a pro-rata sharing algorithm when trading against the order book for simple instruments, not combination instruments. These rules apply when an order or quote is trading against the book to specify how the orders and quotes on the book share the incoming order/quote.

The incoming order/quote trades against the book as follows:

- One price level at a time is traded out
- Only orders and quotes at the best price level are considered before moving onto the next price level
- The execution occurs at the book price, even when the incoming order’s limit price overlaps at the books price

4.4.1 Trading At Each Price Level

Trading at each price levels follows these rules:

- An order trades with the opposing side of the book if it is a market order, or limit order where the limit price matches or crosses with the best price on the opposing side.
- Matching of the incoming order against the book occur one price level at a time. As the incoming order trades with the book, the trade price is set to the price of the book orders.

4.4.2 Trading Against the Book

When trading against the book, the incoming order is shared among the orders on the book according to the following rules:

- Priority Customer orders on the book at the best price level trade in time priority.

- If the order has traded through one or more price levels already in this match event, the quotes that ticked worse to this price level within this match event are traded in time priority (using time of entry of the quote), up to the quantity that was traded out before being ticked worse.
- If the original size of the incoming order is less than six and this is the first price level in this match event, and the PMM is at the BBO, the PMM's quote is traded next, before other non-customer interest, unless the order is preferenced to another MM.
- If this is the first price level and the PMM is at the BBO, and the order is not preferenced to another MM, the PMM receives an enhanced share of the trade. The PMM share percentage depends on the number of quotes and non-customer orders that are with the PMM at the BBO. If there are more than two others at this price level, the PMM gets 30% of the original size of the order. If there are only two others, the PMM gets 40%, and if there is only one other, the PMM gets 60% of the original size of the order. If the PMM would get a greater allocation using its pro-rata share of the balance, the PMM gets the pro-rata share. If this is not the first price level, or if the order is preferenced, and the preferred MM is at the BBO, the PMM only gets its pro-rata share.
- If the order is preferenced to an MM other than the PMM and the preferred MM is present at the BBO, and this is the first price level, the preferred MM receives 40% if there are two or more other quotes or orders. If there is only one other quote or order, the preferred MM receives 60% of the original size of the order. If the preferred MM was not at the first price level, it only gets its pro-rata share.
- The balance is shared amongst the remaining MM quotes and non-Priority Customer orders on a pro-rata basis, going from largest to smallest. Each allocation is rounded up to the next full contract.
- If there is any remaining quantity of the original incoming order to be executed at this price level, it trades where possible against contingent orders, such as the reserve quantity of reserve orders, minimum execution quantity orders, and all-or-none orders.

4.5 Basic Order Functions

This section describes how orders are entered and maintained in T7. The messages used to enter and maintain orders are based on the FIX 5.0 SP2 standards.

4.5.1 Order Entry Overview

A client system can send an order to the trading system using the **New Order Single** message. This message must include the product ID, instrument ID, quantity, price, and side. The client may optionally specify give-up, CMTA, sub-account, and client account using the *Parties* group.

- The client can specify its own order reference (*ClOrdID*), which is used to track the order in subsequent messages. If *ClOrdID* is not provided, the client must use the exchange's *OrderID* (provided in the response) as the order reference.

- After submitting the order, it is validated and possibly rejected. Some reasons may include that the Instrument ID does not belong to the specified product, or that the submitting user is not authorized to trade in that product. If the order is rejected, a reject message is returned with a reason code and reason text. The reject message contains the message sequence number of the message being rejected.
- If the order is accepted then it is processed and an ACK is returned.
- An MM may not place orders on the book in bins in which they are MMs.

4.5.2 Private Order Broadcast

The **Private Order Broadcast (POB)** is sent following a message that enters, alters, or deletes an order. This broadcast describes the order, events that occurred in processing the order, and the final state of the order. The client must subscribe for this broadcast type if it wants to receive these messages. A user can subscribe for its own broadcasts or for broadcasts for the entire BU.

Any subsequent events affecting the order results in additional **POBs**. **POBs** are sent if the order is altered, canceled, or traded.

4.5.3 Alter/Cancel Order

A client application can alter an order using the **Order Cancel Replace Request** message, and cancel an order using the **Order Cancel Request**. The client references the original order using either *CIOrdID* or *OrderID*. Clients can also mass-delete orders using the **Delete All Orders Request**.

A client can:

- Alter or cancel any order for any user within the BU
- Mass delete all orders in an instrument, a product, a list of products, or all products within the BU

The client receives an acknowledgment of the request. The client also receives a **POB** (if subscribed) for each order that is deleted. Any change to an order results in a **POB** to advise the client of the change.

4.5.4 Trades

An order may execute upon entry or after it is resting on the book. When an order is executed, a **POB** is sent to the client to advise it of the trade and the status of the remaining quantity if any remains.

The ME sends the execution information to the TM. The TM enriches the trade with clearing information and sends out a **Deal Item Broadcast (DIB)** and a **Trade Item Broadcast (TIB)**.

The client application can only subscribe to deal and trade messages for the entire BU.

4.5.5 Market Data Feeds

The highest disclosed bid and the lowest disclosed offer in the order book determine the BBO. The total size of the BBO is updated after each transaction. Every time the BBO is updated, it is disseminated on the Top Quote feed, the Depth feed and the OPRA feed at the same time. There are some small holdback delays in the calculation of the BBO to control the volume of market data disseminated.

- The BBO is only updated when an instrument is in the regular trading state. Outside of the state, the BBO is not provided on the Top Quote feed, the Depth feed or to OPRA.
- The Pre-Open feed is available only to MMs and shows the BBO for each series before going into a trading state.
- The Order feed is available to all participants. Any order that rests on the book is sent on the order feed. All trades are sent on the Trade feed and the Top Quote feed. All trade updates are also sent to OPRA.

For more information about all data feeds, please refer to the *Market Data Interface (MDI) Programming Manual*.

4.5.6 Session Disconnect

Instructions sent with the **Logon Req** message determine if orders are to be removed at session disconnect. It is possible to specify whether all orders, or only non-persistent orders, are removed by the system when the user logs off or is disconnected.

4.5.7 Market Orders

A market order trades against the book until fully traded. A market order locks to the acting PMM if one of the following conditions exists:

- the order would trade more than three price levels
- the order is submitted when there is no quantity on the book
- the order would exhaust the book with quantity remaining

Order locks are described below.

4.5.8 Order Locks

Orders for simple instruments may be locked by the system based on various market conditions:

- **Away Market Better (AMB) Lock** – A limit or market order that would execute through the away market BBO (ABBO) is **Flashed** (see **Section 4.9.1.3, Flash Auction**, on page 62). Any remaining quantity locks to the linkage handler with an AMB lock. To opt out of away market routing, orders can be marked Do Not Route (h) on the *ExecInst* field (tag 18).
The AMB lock occurs before, and takes precedence over a three-tick or market order lock.
- **Three-Tick Lock** – If a limit or market order would execute at multiple price levels, and those price levels are separated by three or more price steps, the order trades at the first two levels

and then locks to the PMM with the three-tick lock.

For example: The ISE is offering at \$1.35, \$1.45, and \$1.50, with a price step of \$0.05. An incoming order to Buy @ \$1.60 executes at \$1.35 and \$1.45. Any remainder locks to the PMM with the three-tick lock because \$1.50 is three steps away from \$1.35. (For a non-customer order, any remainder is canceled.)

- **Market Order Lock** – As indicated in **Section 4.5.7, Market Orders**, on page 45, a market order that would exhaust the ISE book with quantity remaining, or that is submitted when there is no quantity on the book, locks to the PMM with a market order lock.

Depending on the order type and market conditions, the same order can lock with any or all of these locks. The **POB** indicates the lock, the locked quantity, and the reason for the lock.

Orders for complex instruments do not lock.

4.5.9 Order Field Descriptions

The fields on the **New Order Single** that control behavior follow the FIX 5.0 SP 2 standard. Some of the relevant fields are described below. The full list of values for each field is provided in **Appendix G: DTI Field Descriptions (Alphabetical)**, on page 170.

Order Type: *OrdType* specifies the type of price: market, limit, stop market, or stop limit.

Time In Force: *TimeInForce* specifies how long the order is to stay in the system if it is not fully traded. It can be Day, Good Til Canceled (GTC), Good Til Date (GTD), Immediate Or Cancel (IOC), and Fill Or Kill (FOK).

Execution Instruction: *ExecInst* allows special functions to be specified. For example:

- The order is to be recovered after a system failure (persisted)
- The order is an Intermarket Sweep Order (ISO)
- The order is an All-or-None (AON) order

Note: GTC and GTD orders *must* be marked “persistent” or they will be rejected. Orders marked persistent carry over to the next trading day, and are reinstated after an intraday trading system restart.

Clearing Capacity: *ClearingCapacity* identifies the clearing account type. When entering a simple order, a participant must specify the type of clearing account as Customer (1), Firm (2), or Market Maker (3).

Order Capacity: *OrderCapacity* specifies the role of the participant submitting the order. There are restrictions on *ClearingCapacity/OrderCapacity* combinations.

Order Capacity types, with allowed clearing, are shown below:

Table 23: Order/Clearing Capacity Values

Order Capacity	Code	Description	Clearing Code
Customer	C	EAM trading for an entity that is not a registered broker dealer. Must clear Customer.	1
Customer-Professional	D	EAM trading for an entity that is not a registered broker dealer but who agrees, in writing, to be treated as such. Can clear Customer or Firm	1 2
Proprietary	G	EAM who is trading for its own account. Can clear Customer or Firm.	1 2
Broker Dealer	B	EAM trading as Agent (not its own account) for a registered broker dealer. Can clear Customer or Firm.	1 2
ISE Market Maker	M	ISE Market Maker. Must clear Market Maker	3
Away market maker	N	Options market maker from another exchange. Must clear Market Maker	3

4.6 Deals and Trades

This section describes functionality relating to deals and trades including:

- System Overview
- Deal and Trade Broadcasts
- Clearing Information
- Liquidity (Maker/Taker) Codes
- Post Trade Substitution Rules
- Subscribing to Deal and Trade Broadcasts
- Deal/Trade Adjustments
- Identifying and cross-referencing trade-related broadcasts.

4.6.1 System Overview

The system holds and matches orders and quotes in the ME process. When an order is matched, an order status or quote status message is sent to the parties involved by the ME. The ME also sends a message to the TM to apply clearing information and to advise the executing broker and the clearing firm of the trade.

Example:

- An executing broker has a BU — ABC01E — which is configured to clear account 998. Clearing account 998 is for clearing firm XYZ.
- Clearing firm XYZ has a BU — XYZ01C — that is linked to account 998.

- When executing broker ABC01E creates a trade, the trade broadcast is sent to the BU ABC01E, which clears 998, which is linked to BU XYZ01C. The trade message is also sent to XYZ01C stating that the executing broker is ABC01E.

4.6.2 Deal and Trade Broadcasts

When an order or quote is traded, the ME sends a status broadcast to all users of the BU that have subscribed to receive status messages. The details of the match are sent to the TM, which determines the clearing information and creates deal and trade broadcast messages.

- A deal broadcast (**DIB**) advises the BU that its order or quote has traded. If the incoming order trades over multiple price levels, then there is at least one deal for each price level. The TM splits up the deal into trade pairs to send to OCC. There can be many trades relating to each deal.

NOTE: There can be *multiple* deals at a single price level.

- A BU can also subscribe to the trade messages (**TIB**), which equate to the trade pairs that are sent to OCC. Clearing firms may prefer to subscribe to trade messages rather than deal messages, as they match the trades coming from OCC, and have unique reference numbers that can be used to match each trade message from ISE to the trades at OCC.
- A BU sees only its side of the deal (**DIB**). However, the counterparty information is disclosed on the trade (**TIB**).

Example:

- BU ABC01E submits an order to buy 100 @ Mkt. The order trades 60 at \$0.87 and another 40 at \$0.88. ABC01E gets a deal broadcast "BOUGHT 60 @ \$0.87," and a deal broadcast "BOUGHT 40 @ \$0.88."
- The order happened to trade against two quotes at \$0.87, for quantities 50 and 10. The trading system sends three matched trade pairs to the OCC: 50 @ \$0.87, 10 @ \$0.87, and 40 @ \$0.88. The BU ABC01E and its clearing firm XYZ01C can subscribe to the trade messages so that they also get drop copies of the one-sided trades as they were sent to OCC. In this case, they would get three trade broadcasts: "BOUGHT 50 @ \$0.87," "BOUGHT 10 @ \$0.87," "BOUGHT 10 @ \$0.88."

4.6.3 Clearing Information

An EAM or MM BU is configured with a default clearing account and can also have a default sub-account. A trade, by default, has the clearing account and sub-account as specified in the configuration of the BU. Deal and trade messages are sent to the BU that entered the order (Executing Broker), and to the default clearing firm.

An order can specify the following alternate clearing information:

- The clearing account (give-up).

- Clearing Member Transfer Agreement (CMTA) account.
- The sub-account

If any of these are present on an order, they override the default information.

When a CMTA is specified, both the CMTA BU, and the clearing firm BU are copied on the trade. If a clearing account or CMTA is specified, even if it equals the default account, the default sub-account is not used.

4.6.4 Liquidity (Maker/Taker) Codes

The *SideLiquidityInd* field on the **TIB** informs members of the pricing strategy applied to their side of the trade. The following table defines the liquidity indicator values:

Table 24: Liquidity Indicator Values

Code	Description
1	Maker
2	Taker
4	Response
5	Hidden
6	Opening Rotation
7	Cross
8	Flashed Order
9	Flash Response
10	Routed Out
11	Trade Report
12	Combo Maker Against Combo
13	Combo Taker Against Combo
14	Combo Response Against Combo
15	Combo Hidden Against Combo
16	Combo Opening Rotation
17	Combo Cross
18	Combo Taker Against Regular
19	Regular Maker Against Combo
20	Combo Taker Against IO
21	Regular (incl. PIM) Taker Against IO
22	IO Maker Against Combo
23	IO Maker Against Regular
24	Regular Maker Against IO Participant
25	IO Participant Taker Against Regular
26	Broken Price Improvement
27	Broken Facilitation

Code	Description
28	Broken Solicitation
29	Combo Broken Price Improvement
30	Combo Broken Facilitation
31	Combo Broken Solicitation
32	Block
33	Block Response
34	Directed Response
35	Facilitation
36	Facilitation Response
37	Price Improvement
38	Price improvement Response
39	Solicitation
40	Solicitation Response
41	Qualified Contingent Cross
42	Customer to Customer
43	Combo Facilitation
44	Combo Facilitation Response
45	Combo Price Improvement
46	Combo Price Improvement Response
47	Combo Solicitation
48	Combo Solicitation Response
49	Combo Qualified Contingent Cross
50	Combo Customer to Customer
51	Sweep Routed Out
52	Sweep Trade Report

4.6.5 Post Trade Substitution Rules

It is possible to have additional rules configured in the system so clearing fields can be changed based on different information in the trade. The clearing account, CMTA, and sub-account can have override rules configured based on things like username, product, free-text etc.

NOTE: If the clearing account is changed, the default sub-account is ignored.

4.6.6 Subscribing to Deal and Trade Broadcasts

A user can subscribe to deal messages and trade messages. Deal and trade messages are recoverable broadcasts. The user can request a re-transmission for a range of sequence numbers.

When connecting, it is possible to obtain all deals/trades by requesting a re-transmission of all messages starting at message one.

The trading system sends deal/trade broadcasts to the executing BU. It also sends a copy of these broadcasts to the clearing account and to the CMTA if specified. If a give-up is provided, the give-up clearing account gets the drop copy instead of the default account. The clearing account and CMTA account BUs subscribe to the broadcasts by asking for its own deals/trades, not those of the executing broker.

4.6.7 Deal/Trade Adjustments

An executing broker can adjust — or alter — certain information on a **deal**, including clearing information and any of the free text fields. Members cannot alter individual trades, only deals. Members cannot bust deals or trades, or adjust deal/trade price or quantity; only ISE Market Operations can perform those functions.

A deal is altered using the Modify Deal Request (***Deal Item Maintenance Instruction Request***) message. The deal to be altered is specified using the *TradeReportID* field. The deal can also be reallocated among multiple accounts as long as the total quantity adds up to the size of the original deal.

ISE Market Operations can also alter deals and trades, and can adjust previous day's trades.

- When a deal is altered, the original deal and all associated trades are busted with *TradeReportTransType* = 4 (Reverse). Replacement deal (or deals) and trades are issued with *TradeReportTransType* = 2 (Replace) and new Trade Report IDs.
 - If a single deal is reallocated among several accounts, the aggregate quantity of all “replace” messages is equal to the quantity of the “reverse” message.
- The **DIB** and **TIB** field, *MessageEventSource*, identifies the reason for, and source of the alteration.
- The **DIB** and **TIB** field, *TradeReportRefID*, identifies the original deal/trade being altered.
- When a deal is altered, only the affected side clearing firms receive the Reverse/Replace **DIBs**; however *all* parties to the trades receive the Reverse/Replace **TIBs**.

Market Operations can rectify **deals** and **trades**. That is, bust trades, adjust price and quantity, or manually add trades.

- If a **deal** is busted, the original deal and all associated trades are busted with *TradeReportTransType* = 4 (Reverse).
- If a **trade** is busted, the original deal and specific trade are busted with *TradeReportTransType* = 4 (Reverse). If that trade-bust reduces the total quantity of the original deal, a replacement deal is issued with *TradeReportTransType* = 2 (Replace).
- If the **deal price** or **quantity** is adjusted (rectified), the original deal and all associated trades are busted with *TradeReportTransType* = 4 (Reverse). New deal (or deals) and trades are issued with *TradeReportTransType* = 0 (New) and new trade report IDs.
- *All* parties to the affected deals/trades receive these messages.

- The **DIB** and **TIB** field, *MessageEventSource* identifies the reason for, and source of the rectification.
- The **DIB** and **TIB** field, *TradeReportRefID*, identifies the original deal/trade being rectified.

4.6.8 Identifying and Cross-referencing trade-related broadcasts

Many members use the status broadcasts (**POB** and **Private Quote Broadcast [PQB]**), which are usually the fastest broadcasts, to capture trades, and then reconcile them with the slower trade broadcasts (**DIB** and **TIB**).

Various fields on the status broadcasts and trade broadcasts can be used to identify trades, or to cross-reference the various broadcasts with one another. A single field MAY NOT be sufficient to uniquely identify a trade, or to cross-reference another broadcast. The following fields and guidelines may be used:

- The *ExecID* field is carried on all status and trade-related broadcasts. It is unique per product (*MarketSegmentID*) per business date. It is assigned by the ME to a *transaction*, and is carried on ALL broadcast messages related to that specific transaction.

For example, an MM submits a quote containing five items – the ME assigns an *ExecID* to that quote transaction. Two of those quotes, for different instruments, execute on entry. All trade-related broadcasts – even though they represent two different instruments – carry that same *ExecID*.

- The **DIB**, *TradeReportID* field is unique per product per date and *Side*, and uniquely identifies a deal.
- The **TIB**, *TradeReportID* field is unique per product per date, and uniquely identifies the trade. The trade includes both sides; therefore, to uniquely identify one side of the trade, the *Side* field should be considered along with the *TradeReportID*.
- The **TIB**, *TradeLinkID* field associates the **TIB** to its parent **DIB**.

The dissemination of trade information may be thought of hierarchically:

1. Trade information in a status broadcast represents a trade at the transaction (*ExecID*) level. For example, Sold 100 @ \$2.50. This is also called a “match event.”

A single match event is made up of one or more deals.

2. The **DIB** represents the quantity bought/sold at a single price level. For example, Sold 100 @ \$2.50. However, a single transaction (match event), depending on the actual components of that event, can result in *multiple* deals at the same price level – instead of a single deal, Sold 100 @ \$2.50, there may be multiple deals: Sold 40 @ \$2.50 and Sold 60 @ \$2.50.

A single deal is made up of one or more trades.

3. The **TIB** represents the component trades that make up the deal. For example, the deal, Sold 100 @ \$2.50, may be a single trade, Sold 100 @ \$2.50; or it may be multiple trades, all at \$2.50: Sold 10, Sold 10, Sold 50, and Sold 30.

The following tables identify the fields, in suggested order of precedence, used for cross-referencing between status and deal/trade broadcasts. Remember that the event quantity on the status broadcasts is the sum of the associated deal quantities, and the deal quantity is the sum of the associated trade quantities.

Table 25: Mapping status broadcasts to deal/trade broadcasts (simple instruments)

Status Broadcast		Deal/Trade Broadcast	
Private Ord. B'cast	Private Quote B'cast	Deal Item	Trade Item
ExecID		ExecID	
SecurityID		SecurityID	
OrdEventPx	QuoteEventPx	LastPx	
OrdEventQty (sum of deals)	QuoteEventQty (sum of deals)	LastQty (sum of trades)	LastQty
Side	QuoteEventSide	TrdCptRptSideGrp.Side	
ClOrdID	QuoteMsgID	TrdCptRptSideGrp.ClOrdID	
OrderID		TrdCptRptSideGrp.OrderID	
		TradeReportID	TradeLinkID

Table 26: Mapping status broadcasts to deal/trade broadcasts (complex instruments)

Status Broadcast		Deal/Trade Broadcast	
Private Ord. B'cast	Private Quote B'cast	Deal Item	Trade Item
ExecID		ExecID	
SecurityID (complex instr. ID)		(…).RelatedSecurityID (complex instr. ID)	
(…).OrdEventLegLastPx	(…).QuoteEventLegLastPx	LastPx	
(…).OrdEventLegQty (sum of deals)	(…).QuoteEventLegQty (sum of deals)	LastQty (sum of trades)	LastQty
ClOrdID	QuoteMsgID	TrdCptRptSideGrp.ClOrdID	
OrderID		TrdCptRptSideGrp.OrderID	
		TradeReportID	TradeLinkID
Note:			
Not present. Must be inferred from complex instr. definition		SecurityID (leg instr. ID)	
Order side is as per message, but leg side must be inferred from complex instr. definition. For example, order side may be Buy, but leg side is Sell.		TrdCptRptSideGrp.Side (leg side)	

4.7 Special Order Types

There are a number of special order types that are entered using **New Order Single** or **New Order Multileg** message:

- Minimum Quantity Orders
- Stop Orders
- Preferenced Orders
- Directed Orders
- Reserve Orders
- Add Liquidity Only Orders (ALO)

4.7.1 Minimum Quantity Orders

In general, minimum quantity orders are orders that must execute a specified minimum quantity. Depending on other order attributes, orders that cannot execute the required minimum quantity are posted (hidden) to the order book or canceled.

ISE supports three minimum quantity order types:

- All Or None (AON)
- Fill Or Kill (FOK)
- Minimum Execution Quantity (MEQ)

4.7.1.1 All Or None (AON)

An AON order must be filled completely in one execution or it does not execute at all.

The order attempts to match with the order book upon entry. If the order is not marketable, or if it is marketable but the order book does not contain sufficient quantity to execute the entire order, the order is placed on the book.

- An AON order is not displayed and does not form part of the BBO.
- An AON order has no priority – it is executed on a best-efforts basis.
- An order is marked AON by setting *ExecInst* = “G” (AON).

4.7.1.2 Fill Or Kill (FOK)

An FOK order must be filled completely upon entry or the entire order is immediately canceled.

The order attempts to match with the order book upon entry. If the order is not marketable, or if it is marketable but the order book does not contain sufficient quantity to execute the entire order, the order is canceled.

- An order is marked FOK by setting *TimeInForce* = 4 (FOK).

4.7.1.3 Minimum Execution Quantity (MEQ)

An MEQ order must execute *at least* the quantity indicated in the *MinQty* field.

The order attempts to match with the order book upon entry. If the order is not marketable, or if it is marketable but the order book does not contain sufficient quantity to execute the required minimum quantity, then, depending on other order attributes, the order is placed on the book, or canceled.

If *TimeInForce* = 3 (IOC), any quantity left unexecuted at order entry is canceled.

If *TimeInForce* = 0 (Day), 1 (GTC), or 6 (GTD), any quantity left unexecuted at order entry is posted to the order book.

If *TimeInForce* is not IOC, the *MinQtyType* field is used to indicate whether successive executions must also be for the required minimum quantity.

If *MinQtyType* = "R" (Repeat), all executions must be for at least the minimum quantity. If remaining quantity is less than *MinQty*, then the remaining quantity must be executed AON.

- A "repeating" MEQ order is not displayed and does not form part of the BBO.

If *MinQtyType* = "O" (Once), only the first execution must be for at least the minimum quantity. Any remaining quantity is displayed on the order book, and may be executed in any quantity.

- Until the first execution, a "once" MEQ order is not displayed and does not form part of the BBO.
- After the first execution, a "once" MEQ order *is* displayed and *can* form part of the BBO.

4.7.2 Stop Orders

Stop orders are hidden orders with a trigger price that are placed in the order book. If the BBO or last sale of the instrument should reach the trigger price, the order is activated and processed.

There are two types of stop orders:

- A standard stop order becomes a market order when activated.
- A stop limit order becomes a limit order when activated.

An order is marked as either a stop order, or a stop limit order using the *OrdType* field.

If an order is marked as a stop or stop limit order, the *StopPx* field must be provided.

4.7.3 Preferred Orders

An EAM may specify an MM to get a preferred allocation on an order.

- If an order specifies a preferred MM, and the order trades with the book at entry, and the preferred MM is present at the BBO, the MM receives an enhanced share. If the preferred MM is not quoting at ISE's BBO, then preferencing has no effect even if the order trades with that MM at farther price levels.
- Specifying a preferred MM does not change priority rules or away market protection for the order being entered.

- Crossing and auction orders, except complex exposure auctions, cannot be preferenced.
- Preferenced orders cannot be directed (see **Section 4.7.4, Directed Orders**, on page 56).
- The preferred MM is specified in the *TargetParties* group on the **New Order Single** and **New Order Multileg** message types.
- PMMs and CMMs do not need to make changes to their applications to receive the enhanced allocation.

4.7.4 Directed Orders

An EAM can specify a Directed Market Maker (DMM) on a customer order and the order is routed to the specified MM for handling. By specifying a DMM on an order, the EAM is transferring the opportunity to facilitate the order to the DMM.

- An MM must be configured to receive directed orders. Once configured, the MM must be willing to accept all directed orders, regardless of underlying symbol and entering EAM. If the specified MM is not configured to accept directed orders, the order is processed as a regular order.
- For an order to be directed to an MM, the series status must be regular or fast. A directed order received prior to the exchange opening is not directed.
- Upon receipt of a directed order, the DMM can either initiate a Price Improvement Mechanism (PIM) auction or release the order to the ISE order book for processing.
- It is possible for an EAM to alter or delete a directed order if the DMM has not yet acted on the order. If the DMM releases the order to the order book, any portion of the order that is not immediately executed may also be altered or deleted. It is not possible to alter or delete a directed order once the DMM enters the order into a PIM.
- EAMs specify the DMM in the *TargetParties* group on the **New Order Single** message.

A directed order follows this processing cycle:

- An EAM enters a directed order, specifying the DMM in the *TargetParties* group. If the specified DMM does not accept directed orders, the order is not directed.
- A locked order broadcast (**Private MM Execution Broadcast**) is sent to the DMM. A DMM session must subscribe to the MM information broadcast type if the locked order broadcast is to be received on that session.

Upon receipt of the directed order, the DMM can:

- Initiate a PIM using the **New Order Single** message, specifying *AuctionType* = 15 (Directed Order PIM).
- Release the order back to the market using the **Order Release Request** message with *ExecInst* = "q" (Release).
- Do nothing – after one second, the order is automatically released.

If the directed order is released (actively or passively):

- A copy of the DMM's quote is taken at the time the order was routed to the DMM.

- The system executes the order against the IBBO if it equals the NBBO, but not against the DMM's quote.
- The system initiates a directed order auction, requesting additional liquidity for this order.
- Once the auction expires, the order is executed against the book and responses, up to the away market. The DMM's quote is executed last at each price level. If the DMM's quote should fade then the copy of the original quote is used.
- The balance is either placed in the order book or locked to the linkage handler.

4.7.5 Reserve Orders

Reserve orders do not show their full size in the order book. The *OrderQty* field is the total order size. The *DisplayQty* field is the initial displayed size. When the order trades, it automatically refreshes itself from its reserve quantity, which is the order quantity less the traded quantity. Only the displayed quantity has priority against other orders, the reserve quantity has no priority at the same price level, but it does have priority over orders at worse price levels.

Reserve orders are entered using the (depending on instrument type) **New Order Single** or **New Order Multileg** message. The presence of the *DisplayQty* field indicates that the order is a reserve order.

There are two methods for refreshing the quantity:

- Refresh the display quantity whenever it is fully traded out (*DisplayWhen* = 2 [Exhaust]).
- Refresh it after every trade (*DisplayWhen* = 1 [Immediate]).

There are also two choices when refreshing the quantity:

- Refresh the initial display quantity (*DisplayMethod* = "1" [Initial]).
- Randomize the displayed quantity between a minimum and a maximum supplied value (*DisplayMethod* = "3" [Random]).

4.7.6 Add Liquidity Only Orders (ALO)

An ALO is an order that is guaranteed to add liquidity ("Maker") to the order book. If the order cannot be added to the book, it is canceled, or optionally, may be re-priced to allow it to rest on the book.

For example, if an incoming ALO would lock or cross either or both of the IBBO or ABBO, a re-price ALO would be re-priced by the exchange *better*, in order to uncross the BBO(s) and rest on the book, while a cancel ALO would simply be canceled.

An order is marked ALO by setting *ExecInst* = "6" (participate don't initiate). If the ALO should be allowed to be re-priced, then the *ExecInst* value "1" (not held) must also be set: *ExecInst* = "6 1" (or "1 6"). An ALO is only re-priced once.

Additional information:

- ALOs can be entered for simple instruments, only.

- ALOs can only be entered when the instrument is in a continuous trading state (“Regular”). If the instrument moves out of “Regular,” any resting ALOs for that instrument are canceled.
- The only allowed validity time is Day (*TimeInForce* = 0).
- The only allowed order type is Limit (*OrdType* = 2).
- All client categories are allowed.
- An ALO does not flash, it is either re-priced or canceled.
- ALOs can also be marked “ISO.”
- ALOs may be preferenced.
- ALOs can be canceled and modified. In addition, a non-ALO can be modified to be an ALO, and an ALO can be modified to be a non-ALO.
- Auction and cross orders cannot be marked ALO.

4.7.7 Sweep Orders

A sweep order is designed to access top of book liquidity only at various market centers. A sweep order is split and sent to multiple exchanges simultaneously, executing against only the protected (top) quotes to the limit price. If a better or equal price is available at the ISE, a sweep order will trade with the ISE book before routing away. Sweep orders are entered using the **New Order Single** message.

- Sweep orders can be entered for simple instruments, only.
- Sweep orders can only be entered when the instrument is in a continuous trading state (“Regular” or “Fast”).
- An order is marked “Sweep” by setting “*RoutingInst*” (9303) = 1 (Sweep)
- A sweep order will not flash. *AuctionInst* must be set to “Do not Flash” – DNF . (*AuctionInst* = 1)
- The only allowed validity time is IOC (*TimeInForce* = 3).
- The only allowed order type is Limit (*OrdType* = 2 - Limit).
- All client categories are allowed.
- A sweep order cannot be Cancel Replaced.
- A sweep order cannot be marked “ISO” or “IAM”
- A sweep order cannot be marked “Do-Not-Route”.
- A sweep order cannot be MEQ, AON, UPC, Directed, Stop, Reserve, or ALO.

4.8 Cross Orders

This order type is used to cross two orders with each other where the orders are not exposed in an auction. This request uses the **New Order Cross** message type.

- This cross request is permitted for Qualified Contingent Cross (QCC) and Customer-to-Customer Cross (CCC) crossing types.
- The message contains details of both the buy side and the sell side orders.
- The same message is used for simple and complex instrument types. For complex instruments, the *ProductComplex* field is required to indicate the type of instrument.
- If the instrument type is complex, the clearing information must be entered on each leg.

- The *AgreementDesc* is a free text field that can be used by the entering trader to record the time at which the cross was agreed between the parties.

The following table enumerates the allowable values for the cross type field (*CrossType*):

Table 27: Cross Order Types

Code	Name
Simple Instruments:	
1	Customer-To-Customer Cross (CCC)
4	Tied-To-Stock (QCC)
Complex Instruments:	
2	Customer-To-Customer Cross (CCC)
5	Tied-To-Stock (QCC)
Complex w/Stock Instruments:	
3	Customer-To-Customer Cross (CCC)

4.9 Auctions

An auction is a process whereby an order is exposed to the market for a small amount of time, called the exposure period (see **Appendix B: ISE Parameters**, on page 158, for the current auction exposure times). During the exposure period, market participants can respond to the auction to provide liquidity to the order being exposed. At the end of the exposure period, the order being exposed is executed against the responses and against the order book. There are many different types of auctions, each with different rules.

NOTE: All auction announcements are broadcast over the Order feed(s) on the MDI. **There are no auction announcements through the DTI.**

Types of auctions include:

- One-sided Auctions
 - Block Auctions
 - Complex Exposure Auctions
 - Flash Auctions
 - Directed Order Auctions
- Two-sided Auctions
 - Facilitation Auctions
 - Price Improvement (PIM) Auctions
 - Directed Order PIM
 - Solicitation Auctions
- Two-sided auctions are auctions where the submitting broker supplies its response and is seeking to cross the orders.

- Two-sided auctions are not exposed to the market until the originating EAM submits its own response.
- All auctions are initiated using the **New Order Single** or **New Order Multileg** message, depending on instrument type. The *AuctionType* field specifies the type of auction order being submitted.
- *TimeInForce* is not required, but if provided, Auction orders/responses may have a time validity of DAY, GTC, or GTD.
- Auction responses are entered using the **New Order Single** or **New Order Multileg** message, as appropriate.
 - The *OrdType* field, set to "Q" (Counter Order Selection), indicates that this is an auction response.
 - The *Auction Type* field indicates the specific auction type.
 - The *RefOrderID* field identifies the auction order to which you are responding.
 - *Side* must be set to the opposite side of the auction order.
- Multiple responses are allowed, however each response replaces (overwrites) the previous response. Depending on the auction, responses may only be allowed to improve the previous response.
- Multiple auctions of any type can occur simultaneously for the same instrument, but only one PIM auction for an instrument can occur at a time. In other words, two facilitation auctions for the same instrument can occur at the same time, or a facilitation auction and a PIM auction for the same instrument, but two PIM auctions for the same instrument cannot occur at the same time.

NOTE: Flash and Directed Order auctions are created by the system because of market conditions. They are not self-initiated by order entry.

4.9.1 One-Sided Auctions

There are four types of one-sided auctions:

- Block
- Complex Exposure
- Flash
- Directed Order

All market participants can respond to an auction.

Block and Complex Exposure auctions are initiated by the system upon receipt of a **New Order Single** or **New Order Multileg** message with the corresponding auction type set. Flash and Directed Order auctions are initiated by the system based on specific market conditions — there is no *AuctionType* code to initiate a Flash or Directed Order auction.

The transaction messages associated with one-sided auctions are shown below:

Table 28: Message Summary for One-Sided Auctions

Function	Message	Description
Enter order	New Order (Single/ Multileg)	Enter an order to be exposed
Expose order	Order On Book	A message is broadcast on the MDI Order feed announcing the start of the auction.
Send response	New Order(Single/Multileg)	Send an order in response to the auction
Maintain order or response	Order Cancel Request	Cancel the order / response
	New Order (Single/Multileg)	Replace (overwrite) existing response
Trade Reports	Order status Broadcast Deal/Trade Broadcast	Advise participants of their participation in the auction, and of their deals/trades, including drop copies
	Market Data Incremental	Report trade on ticker, Top-Of-Book feed and OPRA

4.9.1.1 Block Auction

EAMs, CMMs, and PMMs can use the block auction to anonymously solicit liquidity for a customer or a firm order. A block order is an auctioned order with a minimum of 50 contracts. Block auctions are only available for simple instruments.

NOTE: A block auction for a mini option requires a minimum of 500 contracts.

The block auction processing cycle follows these steps:

- An EAM enters a block order (***New Order Single***) for a minimum number of contracts at a specified limit price. The auction type (*AuctionType*) is set to 1 (Block). The EAM can specify which fields on the order they wish to disclose, including size, price, and side, using the *DisclosureInstGrp* group.
 - By default, size, price, and size are always disclosed. The *DisclosureInstGrp* group can be used to *hide* any or all of those data.
- An auction is initiated and a message is broadcast via the MDI Order feed announcing the block auction and describing the order being exposed.
- Market participants can respond to the auction during the exposure period. The responses are hidden.
- Price improvement must be in full-tick increments.
- At the expiration of the exposure period, the order is executed against the responses and the order book. Any responses that are not executed are canceled.
- At termination, the execution price is determined as the better of:
 - The price at which the whole order can be filled
 - The limit price of the block order
- All responses that are better than the execution price are executed first in price/time priority; then customer orders at the execution price, then responses and other non-customer interest are executed at the execution price according to the pro-rata algorithm.
- The participant can delete an active block order before the block timer expires. Responses may also be canceled.

4.9.1.2 Exposure Auction

The exposure auction is used to obtain price improvement for orders in standard combination and stock combination instruments. Exposure auctions are not available in simple instruments. All members can participate in the auction by entering responses to the auction order.

The exposure auction processing cycle follows these steps:

- An EAM enters a **New Order Multileg** order —
 - AuctionType = 4 (Exposure Order Complex) or 5 (Exposure Order Complex w/ Stock)
 - TimeInForce must be DAY, GTC, or GTD
 - OrdType must be Limit or Market
 - The order may be preferenced
 - If ExecInst = z, any remaining quantity at the end of the exposure period is canceled
 - Each exposure order starts its own auction process
- An auction message is broadcast via the MDI stating that the auction has started and describing the order being exposed.
- Market participants can respond to the auction during the exposure period. The responses are exposed, showing the aggregate quantity at the best price.
- Price improvement may be in penny increments.
- Exposure orders can be canceled, which terminates the auction, or modified which also terminates the auction, but starts a new auction.
- Responses can be canceled.
- When the auction period ends, the exposure order may trade with the following:
 - One or more responses
 - Resting orders/quotes in the complex order book
 - Orders/quotes in the regular (simple) order book
- Any remaining quantity is canceled, as instructed, or posted to the order book.

4.9.1.3 Flash Auctions

A regular order cannot trade through the ABBO unless marked ISO. If any regular, marketable order, which is not an IOC or FOK and is not marked ISO, can receive a better execution price at a competing exchange, the ISE exposes that order to the market in an attempt to obtain a price that matches the away market. The flash auction is similar to the block auction.

If a better price is not obtained during the flash auction, the order (or any remaining quantity) will be routed to the linkage handler for processing by default. An order can be marked Do Not Route (h) via the *ExecInst* field (tag 18) to prevent away market routing.

The flash auction processing cycle follows these steps:

- An EAM enters a marketable order (**New Order Single**) which is subject to away market price protection. An order can be canceled or modified while it is being exposed.
- An auction is initiated. An auction message is broadcast via the MDI stating that a flash auction has started, describing the order being exposed.

- Market participants can respond to the auction during the exposure period. Responses are hidden.
- Price improvement must be in full-tick increments.
- At the end of the exposure period, interest at ISE (responses, and orders and quotes on the book) that is at or better than the NBBO trades with the flashed order in price priority. Untraded and partially traded responses are canceled. (Please refer to *International Securities Exchange Rules*, Rule 803, for specific behavior.)
- Any unexecuted quantity that remains on an order (customer or non-customer) after the Flash auction ends is routed to the linkage handler for subsequent processing if the away market is still better unless the order is explicitly marked Do Not Route (ExecInst=h).
- Any unexecuted quantity that remains on an order marked as Do Not Route is canceled.

4.9.1.4 Directed Order Auction

A Directed Order auction is initiated by the system under certain market conditions when a Directed Order is released without going through a Directed Order PIM auction. Once initiated, the auction proceeds similarly to a regular PIM auction, as described above.

For specifics on Directed Orders, please see **Section 4.7.4, Directed Orders**, on page 56.

4.9.2 Two-Sided Auctions

There are four types of two-sided auctions:

- Facilitation
- Price Improvement Mechanism (PIM)
- Directed Order PIM
- Solicitation

Only EAMs can submit an order for these auction types. All market participants can respond to an auction.

Two-sided auctions — except Directed Order PIM — are initiated in the same way as single-sided auctions: by sending a **New Order Single** or **New Order Multileg** message specifying the appropriate auction type. However, the system does not expose the order until the submitting EAM enters its own, initial response. Once the initial response is received, the order is exposed to all market participants.

The transaction messages that are associated with two-sided auctions are shown below:

Table 29: Message Summary for Two-Sided Auctions

Function	Message	Description
Enter order	New Order (Single/Multileg)	Enter an order to be exposed
Initial response	New Order (Single/Multileg)	Two-sided auctions do not commence until the submitting EAM sends the initial response

Function	Message	Description
Expose order	Order On Book	A message is sent as a broadcast on the MDI Order On Book feed announcing the start of the auction
Send response	New Order (Single/Multileg)	Send an order in response to the auction
Maintain order or response	Order Cancel Request	Cancel the order / response
	New Order (Single/Multileg)	Replace (overwrite) existing response
Trade reports	Order status Broadcast Deal/Trade Broadcast	Advise participants of their participation in the auction, and of their deals/trades, including drop copies
	Market Data Incremental	Report trade on ticker, Top-Of-Book feed and OPRA

4.9.2.1 Facilitation Auction

A market participant can use the facilitation auction to enter block-size agency orders — 50 contracts for regular options, 500 contracts for mini options — and trade against those orders as principal. The EAM that enters the order must be willing to execute the entire size of the order. Facilitation orders can be entered for all instrument types.

The facilitation auction processing cycle follows these steps:

- An EAM enters a block-sized order at a specified limit price.
- The EAM must also enter an initial Auction Response order:
 - Auction type set to 6, 7, or 8 (simple, complex, or complex w/stock facilitation)
 - OrdType = "Q" (Counter Order Selection)
 - Side must be the opposite side of the auction order
 - Price must be the same as, or better than, the auction order price
 - Quantity must be the same as the auction order quantity
 - RefClOrdID references the original order's ClOrdID
- An auction is initiated. An auction message is broadcast via the MDI stating that a facilitation auction has started, describing the order being exposed.
- Market participants can respond to the auction during the exposure period. The responses are hidden.
- Price improvement for simple instruments may be in ½-tick increments (but not less than a penny). Price improvement for complex instruments may be in penny increments.
- The submitting EAM cannot alter the auction order, but it can be canceled prior to the termination of the auction. Responses can also be canceled.

The order is executed at the end of the exposure period according to the following rules:

- The facilitation price is calculated as the better of
 - a) the price that can fully execute the order, and
 - b) the limit price on the customer order being exposed.
- All responses, orders, and quotes at a *better* price than the facilitation price are executed first.
- Customer orders trade at the facilitation price.
- Non-customer orders trade at the price on their response, order or quote.

- The submitting EAM receives 40% of the original size of the facilitation order, or the remainder of the order if more than 40% is already executed.
- The EAM can specify that they want less than 40% allocation of the order using the *AuctionAllocationPct* field.
- Non-customer orders, quotes, and the remaining responses are executed according to the pro-rata sharing algorithm.
- The submitting EAM is assigned any remaining balance.

4.9.2.2 Price Improvement Mechanism (PIM)

The PIM is used to conduct an Auction and trade regardless if it is conducted in 1¢/5¢/10¢ increments. An EAM can use the PIM to trade against the customer order as principal. PIMs can be entered for all instrument types. There is no minimum quantity for a PIM, but The EAM must be willing to execute the entire size of the customer order.

Other market participants can compete for execution by entering response orders that improve upon the previously entered responses. The customer order is executed when the exposure period ends.

The PIM order processing cycle follows these steps:

- The EAM enters a PIM order, specifying the size and limit price. The price can be specified in one-cent increments.
- A PIM order can be marked ISO.
- The EAM must also enter an initial Auction Response order:
 - Auction type set to 12, 13, or 14 (simple, complex, complex w/stock PIM)
 - OrdType = "Q" (Counter Order Selection)
 - Side must be the opposite side of the auction order
 - Price must be the same as, or better than, the auction order price
 - Quantity must be the same as the auction order quantity
 - RefCLOrdID references the original order's CLOrdID
- An auction is initiated. An auction message is broadcast via the MDI Order feed stating that a PIM auction has started, describing the order being exposed.
- Market participants can respond to the PIM order within the response period. The responses are hidden.
- Price improvement may be in penny increments.
- PIM orders cannot be canceled or modified.
- Response orders cannot be canceled but can be replaced (overwritten) by a subsequent response that improves the price and/or increases the quantity.
- The auction is terminated after the response period. It can terminate prematurely when unrelated orders are received in the same series that can cause the PIM order to lose priority.

The order is executed at the end of the exposure period according to the following rules:

- All trading interest at a better price than the PIM execution price

- The customer orders at the PIM price
- The EAM receives 40% of the original size of the PIM order, or the remainder of the order if more than 40% is already executed
- The EAM can specify that they want less than 40% allocation of the order using the *AuctionAllocationPct* field.
- Non-customer orders, quotes and the remaining responses are executed according to the pro-rata sharing algorithm
- The EAM is assigned any remaining balance

4.9.2.3 Directed Order PIM

A Directed Order PIM is initiated by the DMM from the Directed Order. Once initiated, the auction proceeds similarly to a regular PIM auction, as described above.

For specifics on Directed Orders, please see **Section 4.7.4, Directed Orders**, on page 56.

4.9.2.4 Solicitation Auction

A Solicitation auction is an auction mechanism by which an EAM can execute orders of at least 500 contracts — 5,000 contracts for mini options — by soliciting contra-orders. Solicitation auctions can be entered for all instrument types.

NOTE: Solicitation orders are not included in the IBBO.

The solicitation auction processing cycle follows these steps:

- An EAM enters a solicitation order at a specified limit price.
- The EAM must also enter an initial Auction Response order:
 - Auction type set to 9, 10, or 11 (simple, complex, complex w/stock solicitation)
 - OrdType = "Q" (Counter Order Selection)
 - Side must be the opposite side of the auction order
 - Price must be the same as the auction order price
 - Quantity must be the same as the auction order quantity
 - RefCLOrdID references the original order's CLOrdID
- An auction is initiated. An auction message is broadcast via the MDI stating that a solicitation auction has started, describing the order being exposed.
- Market participants can respond to the auction during the exposure period. The responses are hidden.
- Price improvement for simple instruments may be in ½-tick increments (but not less than a penny). Price improvement for complex instruments may be in penny increments.
- The submitting EAM cannot alter the auction order, but the order can be canceled prior to the termination of the auction. Responses can also be canceled.
- The order is only executed against the book when there is a Priority Customer resting at the price. The away market is considered when an unsolicited order is executed.

- If the unsolicited order can be executed in its entirety at a better price, the solicited order is not executed at all. Solicited orders can trade before non-customer interest on the book at the solicitation price.
- The system cancels a solicitation order and its responses when the execution price is outside of the IBBO, or if the execution price is on the same side as the unsolicited order and a customer order is also at that price.

4.9.3 Auction Step-Up Price

Auction step-up is a mechanism that allows the EAM the ability to silently match, up to the step-up price, any responses that improve the auction price. The EAM may also indicate a “market price” step-up, which matches all improving responses.

A step-up price is entered by setting *Price* on the initial response better than *Price* on the auction order. For example, if the auction order is buying at \$1.00, the initial response may sell at \$0.95. Or, if the auction order is selling at \$1.00, the initial response may buy at \$1.05.

To indicate a “market price” step-up, do not send the *Price* field.

Step-up only matches the quantity required. For example, if the auction is to buy 10 @ \$1.00 with a step-up price of \$0.95, and the only other response is 1 @ \$0.98, the EAM will get 1 @ \$0.98, the response will get 1 @ \$0.98, and the EAM will get the remaining 8 @ \$1.00.

Step-up pricing may be used on Facilitation and PIM auctions, only.

4.9.4 Auction Types

The following table enumerates the allowable values for the auction type field (*AuctionType*):

Table 30: Auction Types

Code	Name
Simple Instruments:	
1	Block Order
2	Directed Order
3	Flash Order (entered only on response order)
6	Facilitation Order
9	Solicitation Order
12	PIM
15	Directed Order PIM (DMM's initial response)
Complex Instruments:	
4	Exposure Order
7	Facilitation Order
10	Solicitation Order
13	PIM
Complex w/Stock Instruments:	

Code	Name
5	Exposure Order
8	Facilitation Order
11	Solicitation Order
14	PIM

4.10 Quote Related Functions

- The products within the ISE market are divided into ten groups called bins. This is done for the purpose of assigning PMM rights. Among other obligations, PMMs must provide quotes in all products within their assigned bins, and provide quotes according to specific spread (width) requirements.
- CMMs, based on Trading Rights, may select products to quote from across the market. CMMs must not quote products that are not part of their selection. For each product quoted, CMMs must continuously quote 60% of the series (or 90%, if they are a preferred MM) and provide quotes according to specific spread (width) requirements.
- The system does not validate the quotes upon entry to ensure that the quote selections, minimum quote obligations, or spread requirements are being met. That function is performed by the ISE Surveillance group.

NOTE: Please contact the ISE Business Development group for complete information on MM quoting requirements and obligations.

4.10.1 Maintaining Quotes

Quotes are used by MMs to provide liquidity to many instruments at once. An MM can only have one quote in each instrument at a time. A quote replaces a previous quote from the same MM in the same instrument if one exists. Quotes are available for immediate execution and update the BBO.

- The **Mass Quote** message allows two-sided quotes to be entered for all instruments in a product. All quotes in a **Mass Quote** message must relate to the same product. One instrument item can be used to enter a single quote for the bid or offer, or both sides at the same time. However, there cannot be more than one item for the same instrument in a **Mass Quote** message. If a BU has multiple users, their quotes overwrite each other. Quotes are “owned” by the BU, it doesn’t matter which user enters them.
- Quotes can be deleted by entering a quote with zero quantity at no price — set both the *Size* and *TotalSize* field to zero, and do not set the price field. Quotes can be updated and other quotes deleted within the same quote transaction.
- An MM must be approved to provide quotes in a product. If the MM BU is not approved, the **Mass Quote** is rejected.
- If the quote *message* is rejected, a **Reject** message is returned as the response, containing an error number and error text.

- If one or more quote *items* are rejected, the **Mass Quote Acknowledgement** contains a list of the items that were rejected. Each rejected item contains the Instrument ID and error number.
- When new quotes overlap the previous quotes, the side that is crossed is removed from the book before the new quote is applied.

Example:

The current quote is 12.00 x 12.25. A new quote of 12.25 x 12.50 does not cause a trade, as the new bid causes the old offer to be removed before the new offer is applied.

4.10.2 Tick-Worse

When a quote is traded out, it can be automatically reinstated into the book at a worse price. This is known as the tick-worse function. Tick-worse parameters are set using the **MM Parameters Definition Request** message, specifying how many price ticks back, and for what size, the quote is to be reinstated. A **PQB** is sent to the BU when a quote is ticked worse.

NOTE: Setting tick-worse parameters is optional, and is not required for quoting.

- A quote is only reinstated once. If the tick-worse quantity trades out, the MM is out of the market.
- The system treats the tick-worse quantity as a modification of the original quote quantity, not as a new quote. In other words, if the original quote quantity was 10, and the quote ticked worse for 10 more, the system would treat this as if the original quote was increased to 20, and there is now 10 remaining. This is relevant when factoring the various speed bump counters (see **Section 4.10.4, Market Maker Protection (Curtailed/Speed Bump)**, on page 70).
- There are separate parameters for simple instruments, complex instruments, and complex instruments w/stock.

4.10.3 Quote Inactivation

Quote inactivation instructs the system to hide all quotes in a product from the BBO calculation and from trading. While the quotes are inactive, the MM may continue to update the quotes and the updated values are stored on the book. The MM can send a separate message to activate or inactivate the quotes. Quotes can be inactivated/activated in a product, a list of products, or for all products market wide.

- The inactivation indicator is held at the product level, so that quotes can be inactivated in a product before any quotes are submitted. The quotes can be entered and deleted while the inactivation flag is set.
- Quote activation/inactivation affects the *entire* BU. In other words, if two different sessions (1 and 2) are quoting instruments in the same product, and session 1 inactivates quotes in that product, session 2's quotes are *also* inactivated. Similarly, if sessions 1 and 2 are quoting

different products, and session 1 inactivates quotes across the market, session 2's quotes are also inactivated.

- Quotes are inactivated and reactivated using the **Quote Action Request** message, field *MassActionType* (tag 1373), values 1 and 2. A list of products can be specified on the message. There is no restriction on how many products are specified. If the list of products has zero items, then all products that the BU is entitled to quote across the market are inactivated.
- Quotes are both inactivated and deleted in a single step using the *Quote Action Request* field *MassActionType* (tag 1373), value 100.
- The **Quote Action Report** enumerates the products captured by the request and provides positive or negative confirmation that each of the specified products is now active/inactive.
- Quote inactivation/reactivation does *not* reset an MM's curtailment counters (see **Section 4.10.4, Market Maker Protection (Curtailment/Speed Bump)**, on page 70).

NOTE: The **Quote Action Request**, whether used to inactivate or inactivate *and* delete quotes, is not subject to throttling – it is not counted against the transaction limit or outstanding request limit, nor does the response reduce the outstanding counter. It is a “free” transaction.

4.10.4 Market Maker Protection (Curtailment/Speed Bump)

Curtailment (or, the speed bump) assists MMs by limiting their total exposure in a product. After every trade against a quote, the MM's speed bump is checked for that product. If the specified limit is *exceeded* within the specified time interval then all quotes for that BU in that product are *inactivated*. A **PQB** is sent to advise the MM that the quotes were inactivated because the speed bump was triggered.

- Quotes in that product remain inactive until the MM reactivates the quotes with the **Quote Action Request** message.
- Speed bump parameters are set using the **MM Parameters Definition Request** message. Four discrete counters must be set, as well as the time interval over which the count is to be checked. The four counters are:
 - Cumulative Count — Total number of contracts traded
 - Percentage Count — Traded quantity as a percentage of total quantity
 - Delta — Long vs. Short, evaluated as:
Absolute Value(Bought Call + Sold Put – Sold Call – Bought Put)
 - Vega — Bought vs. Sold, evaluated as:
Absolute Value(Bought Call + Bought Put – Sold Call – Sold Put)
- All four speed-bump counters *must* be set; any one counter triggers curtailment.
- The time interval is a *rolling* interval and must be specified in milliseconds from 500 ms to 30,000 ms (30 seconds).
- There are separate counters for simple instruments and complex instruments; there are NO counters for complex instruments w/stock.

- Trades that occur when the quote is entered are counted. If quote items remain to be processed in the quote transaction after curtailment occurs, those quote items are processed, but as inactive quotes.
- The speed bump only counts trades made against the MM's quotes. It does not count IOC orders or auction responses entered by the MM.
- A new quote does *not* reset the curtailment counters. Only quote reactivation after a curtailment event resets the curtailment counters. Manual inactivation/reactivation by the MM does *not* reset curtailment.

4.10.4.1 Curtailment Examples

The following examples are provided to help clarify curtailment behavior. The time interval is not specified — it is assumed that all executions occur within whatever interval may be set.

Example 1 — Cumulative Count

CumQty = 100

MM Quotes:

Series XYZ 1: quantity 50

Series XYZ 2: quantity 50

Series XYZ 3: quantity 50

XYZ 1 executes 50; cumulative quantity = 50 — less than threshold, no curtailment

XYZ 2 executes 50; cumulative quantity = 100 — equals threshold, no curtailment

XYZ 3 executes 50; cumulative quantity = 150 — greater than threshold, curtailment triggered.

Example 2 — Percentage Count

PctCount = 150 (150%)

MM Quotes:

Series XYZ 1: quantity 50

Series XYZ 2: quantity 50

XYZ 1 executes 50 (100%); percentage count = 100% — less than threshold, no curtailment

XYZ 2 executes 25 (50%); percentage count = 150% — equals threshold, no curtailment

XYZ 2 executes 25 (50%); percentage count = 200% — greater than threshold, curtailment triggered.

Example 3 — Percentage Count with Tick-Worse

Tick-worse *modifies* the original quote quantity.

PctCount = 150 (150%)

NewQuoteSize = 20

MM Quotes:

Series XYZ 1: quantity 100
 Series XYZ 2: quantity 100
 XYZ 1 executes 100 (100%); percentage count = 100% — less than threshold, no curtailment
 System re-quotes 20 of XYZ 1 (modifying original quote qty UP to 120)
 XYZ 1 executes 10 (8% of 120); percentage count = 108% — less than threshold
 XYZ 2 executes 50 (50%); percentage count = 158% — greater than threshold, curtailment triggered.

4.10.5 Market Wide Speed Bump (MWSB)

MWSB assists a market maker by limiting his total exposure across a market when there are multiple product curtailments. Similarly, Cross Market Wide Speed Bump (CMWSB) limits total exposure across all ISE markets. MWSB counts the number of curtailments across all products per market, while CMWSB counts the number of curtailments across all products and across all ISE markets. All quotes for a specific Business Unit (BU) are inactivated if the specified number of curtailment events is *exceeded* within the specified time interval. A market maker may set either MWSB or CMWSB, but not both.

A Private Quote Broadcast (**PQB**) is sent to advise the market maker that the quotes were inactivated because the MWSB or CMWSB was triggered.

- Defining either MWSB or CMWSB parameters is mandatory for all market makers.
- MWSB/CMWSB parameters cannot be set via **DTI** message. The ISE Market Operations set these parameters as defined by the market maker.
- There are no defaults for MWSB/CMWSB. Each market maker must set the following two parameters for each Business Unit (BU):
 - Number of Curtailments.
 - Time window in seconds.
- MWSB for ISE, ISE Gemini and ISE Mercury exchanges can be set independently or combined by setting CMWSB.
- Curtailments in simple and complex instruments are counted as two separate curtailments.
- Any new quotes sent by market makers after MWSB/CMWSB has been triggered will remain inactive.
- Quotes in all products for all instrument types remain inactive and locked until the market maker instructs ISE Market Operations to unlock them.
- A market maker cannot reactivate quotes using **Quote Action Request** when quotes are in locked state. Once ISE Market Operations unlocks a BU, the market maker can reactivate their quotes using **Quote Action Request**.

4.10.6 Session Disconnect

By default, all quotes, active and inactive, entered by a user *are deleted* when that user either actively or passively disconnects.

4.10.7 Specifying Market Maker Parameters

Curtailment (speed bump) parameters are **required**. Tick-worse, short-sale indicator, and auto-rotate parameters are **optional**. The system does not define default MM parameters. If curtailment parameters are not explicitly set for a product and instrument type, quote transactions for that product/instrument type are rejected.

- Parameters are entered using the **MM Parameter Definition Request** message and, regardless of the user submitting the request, are set for the entire BU. A single request can be submitted, specifying the parameters for the entire market; or multiple requests can be sent, specifying parameters for individual products or groups of products.
- Parameters can be changed by sending a new request with new values. In this way, a default set of parameters can be issued for the entire market, followed by additional requests to redefine parameters for specific products.
- When MM parameters are updated, they take effect immediately. The next quote to trade out is ticked worse according to the new parameter. The speed bump limits are updated immediately. If the next trade exceeds those limits then the quotes are inactivated. If, based on the new parameters, the limit is already exceeded, then quotes are not inactivated until the next trade.

4.11 Complex Orders and Quotes

Complex orders are traded at the ISE by buying or selling complex instruments. Complex instruments are dynamically created by market participants to buy or sell multi-leg strategies. Once an instrument is created, market participants can enter orders or two-sided quotes for these instruments in the same way as for simple instruments (single leg).

The ISE supports the following types of complex instruments:

- **Standard Combinations:** Includes two to eight options legs within the same product
- **Stock Combinations:** Includes one underlying stock and one to eight options legs within the same product

4.11.1 Life Cycle of a Complex Instrument

Complex instruments are dynamically created by the system upon request by participants. Once created, they are available to trade for at least the remainder of the day. The creation and deletion of complex instrument is disseminated on the reference data feed. Complex instrument creation is also disseminated via the Top Quote and Depth feeds using the snapshot message.

At the end of the day, complex instruments with no remaining orders are automatically deleted. Instruments with unfilled GTC orders are carried forward to the next business day and are distributed on the reference data feed at the start of new business day.

4.11.2 Creating Complex Instruments

Before entering a multi-leg strategy order or quote, a market participant should confirm, via the Reference Data feed, if an instrument for that strategy already exists. If the instrument does not exist, a participant must first send a request to create the instrument using the **Add Complex Instrument Request** message. The Instrument ID for the complex instrument is returned in the **Add Complex Instrument Response** message.

If a request is entered for an existing complex instrument, the system returns the existing Instrument ID; otherwise, a new, unique Instrument ID is assigned to the newly created complex instrument.

The definition of the newly created instrument is broadcast via the MDI over the Reference Data, Top Quote, and Depth feeds. For the rest of the day, market participants must use this complex instrument ID to submit orders or quotes for that particular strategy.

NOTE: FIX field *LegRatioQty* is defined as data type float in FIX but is defined as data type int in the DTI. Only positive, whole numbers are allowed (1, 2, 3, etc.).

To create a complex instrument, the following information must be provided for each leg:

- Instrument ID (or Underlying ID for stock leg)
- Ratio
- Side (buy or sell)

The following restrictions apply when creating a complex instrument:

- All component leg instruments must be for the same product.
- There must be at least two legs – single leg complex instruments are not allowed.
- A maximum of eight options legs and one stock leg are allowed.
- Ratios must be in lowest common denominator. If the lowest common denominator is not entered, the system automatically adjusts the ratios. For example, a standard combo ratio of 6:4 is changed to 3:2 by the system. A stock-combo ratio of 100:2 is changed to 50:1. In such cases, the user must adjust the quantity on their order depending upon the final instrument created by ISE.
- The ratio between largest and smallest option leg cannot exceed 3:1. For example, an instrument with ratio 17:6 is allowed, but an instrument with ratio 16:5 is not allowed.
- The deliverable (contract size) for each component leg must be equal. For example, it is not possible to create a complex instrument with one leg delivering 100 shares and another leg delivering 150 shares.

The legs on the defined complex instrument may be reordered from the original request, and are sorted as follows:

- Stock leg (for stock combinations only)
- Options legs
- Sort by calls then puts
- Within call/put sort by ascending expiration date
- Within same expiration date sort by ascending strike prices

Example

A request to 1) buy a Call, 2) sell a Put, and 3) sell the Stock would be reordered Stock / Call / Put.

4.11.2.1 Synthetic Best Bid and Offer (Synthetic BBO)

The synthetic BBO for a complex instrument is calculated as follows:

- Synthetic bid = (the best bid price of each buy leg times its ratio) – (the best offer price of each sell leg times its ratio)
- Synthetic offer = (the best offer price of each buy leg times its ratio) – (the best bid price of each sell leg times its ratio)

For stock combinations, if the stock is being bought, the synthetic BBO is calculated as:

- Synthetic bid of stock combinations = Synthetic bid of options legs + national best bid of stock leg * Ratio on stock leg / size of option contract
- Synthetic offer of stock combinations = Synthetic offer of options legs + national best offer of stock leg * Ratio on stock leg / size of option contract

If the stock is being sold, the synthetic BBO is calculated as:

- Synthetic bid of stock combinations = Synthetic bid of options legs – national best offer of stock leg * Ratio on stock leg / size of option contract
- Synthetic offer of stock combinations = Synthetic offer of options legs – national best bid of stock leg * Ratio on stock leg / size of option contract

4.11.2.2 Signature Inversion

To facilitate the trading of a complex instrument with positive prices throughout the day, the system may invert the signature of the requested complex instrument.

Example

A vertical spread to sell May 45 call and buy May 50 call is always created as buy May 45 call and sell May 50 call. Participants must pay attention to the *LegSide* fields on the Add **Complex Instrument Response** to check if the instrument was inverted or not. If inverted, a participant who wishes to sell May 45 call and buy May 50 call must enter a sell order for the created instrument.

The inversion rules are as follows:

- If a strategy is a pre-defined debit strategy, it is created as specified. If the requested instrument results into a credit trade for these pre-defined strategies, the instrument signature is inverted.
- If all legs of a requested complex instrument are sell legs, they are inverted to buy legs.
- If a strategy is not a pre-defined strategy, a synthetic price based on simple instruments is calculated. If the synthetic price is negative, the instrument signature is inverted.
- Otherwise, if the simple instruments have no BBO, the complex instrument is created as is.

For a complete list of the pre-defined standard strategies, please see **Appendix M: Standard Strategies**, on page 215.

4.11.2.3 Limited Number of Instruments

For efficient management of ISE systems capacity, each BU is only allowed to create a limited number of complex instruments at a time in a single product. In addition, only a limited number of instruments can exist for each product at a time. Once these limits are reached, all further requests to create new complex instruments are rejected. Only ISE market operations can delete complex instruments. If a participant inadvertently creates the wrong complex instruments, they can request ISE market operations to delete their instruments. Market participants cannot delete complex instruments.

4.11.3 Entering Complex Orders

Once an instrument is created or available on the reference data feed, market participants can enter buy and sell orders using the **New Order Multileg** message. Order handling and attributes for complex instruments are similar to simple instruments with some exceptions:

- The limit (net) price of a complex order can be positive, negative, or zero (0).
- Orders for standard combinations can be priced in pennies regardless of the minimum price variation of the component legs. Orders for stock combinations can be priced in increments of 100ths of a penny (\$0.0001).
- Directed complex orders are not supported.
- Stop complex orders are not supported.
- Complex orders can trade with opposite complex quotes and orders, or with orders and quotes on individual legs in the regular order book ("legging-in"), with the following exceptions:
 - An order with more than a system-configured number of legs will *not* leg-in to the regular order book. See **Appendix B: ISE Parameters**, on page 158.
 - An order with an unconventional strategy will not leg into the simple order book. See **Appendix M**: on page 215.
- The limit price of a complex order cannot exceed the synthetic BBO price based on the individual legs by more than \$1.00 or 1%

- The three-tick rule does not apply to complex orders.

NOTE: If a complex order legs into the simple instruments, a *three*-tick check is done on each simple instrument.

4.11.3.1 Net Limit Price

The limit price of a complex order is referred to as the “net price.” The net limit price of a standard combination order is calculated as (add Buy leg, subtract Sell leg):

$$\text{Premium} = \pm(\text{Leg_1.RatioQty} * \text{Instrument 1 price}) \pm (\text{Leg_2.RatioQty} * \text{Instrument 2 price})$$

[± ... up to 8 legs]

For the stock leg, the ratio is adjusted by dividing the leg ratio with size of contract (add Buy leg, subtract Sell leg):

$$\text{Premium} = \pm(\text{Stk_leg.RatioQty} * \text{stock price}) / \text{Instrument Contract Size}$$

$$\pm(\text{Simple_Instr_leg.Multiplier} * \text{Instrument price}) [\pm \dots \text{up to 8 simple instrument legs}]$$

4.11.3.2 Clearing Information:

For all options legs, the values for clearing and give-up must be same. Give-up value can be different on the stock leg. Position effect, clearing sub account and give-up are not required for the stock leg. The MPID of the give-up party must be specified on the stock leg in the give-up field.

The following information is provided for each leg:

- Position effect: Open or Close (required, options legs only)
- Clearing account (CMTA) for options legs only (optional)
- Clearing sub account for options legs only (optional)
- Give-up for options legs only (optional)
- Give-up for stock leg (optional)

NOTE: There are no leg reference ID fields (*LegRefID*) defined in the DTI. The legs on the **New Order Multileg** must be specified in the order they are defined in reference data, or the **Add Complex Instrument Response** message.

4.11.3.3 Short Sale Indicator

For stock combination orders, if the stock leg is being sold, the short sale indicator can be provided.

4.11.4 Quoting in Complex Instruments

All MMs are allowed to quote *all* complex instruments regardless of whether they are assigned trading rights to that product or not. Quotes for complex instruments are not subject to any quotation requirements that are applicable to MM quotes in the regular market for individual options series or classes. There are no PMMs or backup PMMs for complex instruments.

NOTE: The list of products eligible for complex quoting is available from ISE Market Operations, or by emailing helpdesk@ise.com.

- The net price of complex quotes can be positive, negative, or zero (0).
- The **Mass Quote** request is used to add, modify, or delete multiple quotes. **Delete All Quotes Request** can be used to delete all quotes within a product.
- Complex quotes can only trade with inverse complex orders or quotes. Complex quotes, unlike complex orders, do not leg-in — they do not trade — with orders and quotes on individual legs in the regular book.
- Complex quotes can be inactivated or reactivated using the **Quote Activation Request**.
- Speed bump parameters must be set in order to quote standard complex instruments. **There is no speed bump for stock combination instruments.** Speed bump parameters are independently maintained for simple and standard combination instruments.
- Tick-worse capability is available for complex quotes. Tick-worse parameters are independently maintained for simple, standard combination, and stock combination instruments.
- The Short Sale indicator for quoting stock combination instruments is set for the product and BU using the *LegSide* field on the **MM Parameter Definition Request** message.

4.11.5 Price Protections

Extra price protections are provided while trading vertical spreads. A vertical spread is defined as:

- Buy Low Strike Call, Sell High Strike Call, Same Expiry, 1:1 Ratio
- Buy High Strike Put, Sell Low Strike Put, Same Expiry, 1:1 Ratio

The limit price of a complex order/ quote for a vertical spread cannot be negative.

A sell market vertical spread order does not leg-in to the simple order book if the net trade results in a payout (negative net price) for the complex order.

4.11.6 Market Maker Protections (Curtailment/Speed Bump)

MMs are provided independent curtailment parameters for standard complex instruments and simple instruments. As with simple instruments, MMs must set curtailment parameters in order to quote standard complex instruments. There is no curtailment capability for stock complex instruments.

A trade of a standard complex instrument affects only the curtailment parameters for complex instruments. Similarly, curtailment parameters in the simple instruments have no effect on the complex instruments parameters. Trades in complex instruments are counted per option leg.

As in the regular market, the curtailment is checked after the trade. An MM is liable for the full value of the complex quote even if it exceeds the limit.

4.11.7 Matching in Complex Instruments

A complex order can trade either with other complex orders and quotes, or with orders and quotes on the individual leg instruments (legging in). A complex quote can trade only with other complex orders and quotes.

Matching in the complex instruments can be triggered either by a change in the complex instrument book or in the individual simple instruments book. Although trading a simple instrument has priority, matching between simple and complex is done on a best-effort basis.

Two types of market models are used for allocation in ISE's complex order book on a symbol-by-symbol basis. ISE periodically announces the list of symbols trading with each model.

1. **Price/time**, where resting orders/quotes trade with incoming orders/quotes in pure price/time priority.
2. **Pro-rata**, where an incoming order/quote trades against the book as follows:
 - Customer orders trade in time priority.
 - If the order is preferenced to an MM, and the preferred MM is present at the complex book BBO, the preferred MM receives the greater of:
 - The proportion of total size represented by its quotes, or
 - 60% of the remaining size if there is only one other quote or order resting on the book, or
 - 40% of the remaining size if there are two or more quotes or orders resting.
 - The balance is shared amongst the remaining MM quotes and non-priority customer orders on a pro-rata basis.

In both models if both regular books and complex instrument books are at the same price, the trade occurs with the complex order/quote in the complex instrument book as long as there is no priority customer at any of the individual legs. If there are priority customers on any individual legs, the trade occurs with priority customers orders on the individual legs.

4.11.8 Complex Instruments Trade Reports

Trade reports (**DIBs** and **TIBs**) for the individual legs are sent to market participants. Options leg trades are reported to OCC and clearinghouses via trade drop copy. Stock leg trades are reported to the Depository Trust & Clearing Corp. (DTCC), via a third party.

The **POB** or **PQB** reports the net trade price for the complex instrument, as well as the component leg prices.

4.11.9 Special Orders

The following special orders are allowed for complex instruments:

4.11.9.1 Preferenced and Reserved orders.

The allocation rules are the same as for simple instruments.

4.11.9.2 Facilitation, Solicitation, and PIM orders.

A complex PIM auction (standard or stock) must be entered at a price that is at least one cent better than both sides of the complex IBBO, and at least one cent better than the net price obtained from the BBOs of the individual legs on both sides. The allocation rules are the same as for simple instruments.

4.11.9.3 Customer-to-customer match orders

The behavior is the same as for simple instruments.

4.11.9.4 Exposure orders

The exposure order allows market participants to expose their orders in an effort to obtain price improvement. All market participants can respond with price improving responses. After the exposure period, the exposure order can trade with opposite side complex quotes, orders, response orders, and quotes and orders on individual legs. If an exposure order is not filled or only partially filled after the exposure period, any remaining quantity is added to complex instrument book.

For more information on the Exposure Order Auction, please see **Section 4.9.1.2, Exposure Auction**, on page 62.

NOTE: An unfilled or partially filled exposure order is canceled after the exposure period if *ExecInst* = z.

4.11.9.5 Underlying Price Contingency orders

Participants who wish to execute their order only if underlying is trading in a specific price range by using underlying price contingency (UPC) orders. The following contingencies are allowed:

- Underlying bid is less than, greater than, or within a price range
- Underlying offer is less than, greater than, or within a specific range

Note: If a UPC order is also marked as an exposure order, the exposure instruction is *ignored*.

The following fields on the **New Order Multileg** message are used:

- *RelatedLowPrice*
- *RelatedHighPrice*
- *RelatedPriceSource*

If the contingency low price (*RelatedLowPrice*) is specified, the UPC order is activated if the underlying national best bid (NBB) or offer (NBO), as specified in the *RelatedPriceSource* field, is greater than the contingency low price. If the contingency high price (*RelatedHighPrice*) is specified, the UPC order is activated if the underlying NBB or NBO, as specified in the *RelatedPriceSource* field, is less than the contingency high price. If both contingency price

fields are specified, the UPC order is activated if the underlying NBB or NBO (as specified) is greater than the contingency low price and less than the contingency high price.

Note: A UPC order is a “hidden” order: it is not exposed to the marketplace, and it does not update the IBBO.

4.11.9.6 Do Not Trade Through (DNTT)

Complex orders are allowed to trade without regard for the away market prices of the individual legs. “Do Not Trade Through” (DNTT), allows a member to indicate on the order that it should *not* trade through away market leg prices.

DNTT orders may be limit or market orders, and may be for any time validity and client category. DNTT orders may be modified, including changing DNTT to non-DNTT, and *vice versa*.

An order marked DNTT that does not trade rests on the book (unless it is an IOC or FOK order), updating the IBBO as appropriate.

The *PriceProtectionScope* field, on both the **New Order Multileg** and **Multileg Cancel Replace Request** messages, is used to indicate if the order is allowed to trade through the away markets. The default value is ‘1’ (Local) — the order is allowed to trade through. If set to ‘2’ (National), the order is *not* allowed to trade through.

4.11.10 Notable Differences with Simple Instruments

- Stop orders are not allowed for complex instruments.
- Directed orders are not allowed for complex instruments.
- Complex orders can be matched outside of the individual legs’ bids and offers established at other exchanges.
- There is no AMB lock for complex orders and complex orders are not eligible for linkage handling.
- There is no ISO flag for complex orders.
- There is no three-tick lock for complex orders.
- Market orders *can* rest on the complex instrument books.

5. DTI Message Specifications

This section describes each DTI message including:

- Purpose of the message
- Message format
- Response format
- Rules for completing input fields
- Special meaning of reply fields
- Tips for Developers

Refer to **Appendix G: DTI Field Descriptions (Alphabetical)**, on page 170, for detailed descriptions of general fields. This section does **not** describe error messages.

Group fields are shaded in the message tables to better show nesting.

5.1 Attributes Common to All DTI Messages

This section describes various attributes common across all, or most, messages.

5.1.1 Timestamp Precision

Time values in timestamp fields (*TransactTime*, *SendingTime*, *RequestTime*, etc.) are expressed in nanoseconds (ns) since the UNIX epoch (midnight, 1/1/1970 GMT). For example, the timestamp value 1,286,385,359,115,234,567 is equivalent to Wednesday, October 6, 2010, 1:15:59.115234567 p.m., ET.

5.1.2 Field numbers 8500 – 8999

Field numbers in the range 8500 to 8999 are temporary and subject to change pending permanent assignment by the FIX Protocol Organization.

5.1.3 Field Numbers ≥ 10000

DTI Application messages are numbered in the range of 10000 – 10999. Groups (which replace the FIX “Number of...” fields) are numbered ≥ 11000 .

5.1.4 MsgSeqNum

MsgSeqNum is required on all messages sent from the client application to the DTI, except the **Heartbeat** message. The *MsgSeqNum* entered in the request header is echoed in the response header of the message ACK or Reject.

5.1.5 ClOrdID

The *ClOrdID* field is used as a transaction ID and as a message ID. It is optional on all order-related messages. If supplied, it becomes the client’s reference ID for that order as well as the message.

ClOrdID must be unique for the session for the life of the order. Once an order has completed (filled, canceled, replaced), that *ClOrdID* may be reused.

5.1.6 Instrument Types

A simple instrument is one that represents a clearable instrument. A complex instrument is one that represents multiple simple instruments.

The instrument ID, represented by the *SecurityID* field, uniquely identifies an instrument across all instruments, products, and markets. The product ID, represented by the *MarketSegmentID* field, is also unique across all products and markets. All input and output messages contain both *MarketSegmentID* and *SecurityID* so that the message can be routed and filtered by the product. Complex instruments are also identified by *MarketSegmentID* and *SecurityID*. All messages that can refer to complex instruments also have the *ProductComplex* field, which indicates if this is a simple instrument, complex instrument (standard combination), or complex w/stock instrument (stock combination). This field defaults to simple instrument and is only required on messages that reference complex instruments.

There are two types of complex instruments:

- Combination instruments – which define two to eight simple instrument legs in the same product
- Stock combinations – which define one to eight simple instruments in the same product with the inclusion of the underlying stock

The underlying stock is assigned its own instrument ID to represent the stock leg. (Please see the *UnderlyingID* field in the **Product Snapshot** message in the *MDI Programming Manual*.)

Except for the **Mass Quote** message, which supports quoting multiple instrument types, different instrument types *cannot* be combined in a single request message — a separate message must be sent for each discrete instrument type. For example, to delete all orders for all instruments types, you must send three separate **Delete All Orders Request** messages, one for each instrument type (*ProductComplex*).

5.1.7 Market ID

The *MarketID* field is used on “mass action” requests (**Delete All Orders Request**, **Quote Action Request**, **Subscribe Request**, etc.) to explicitly identify the target exchange when there are no other identifying data on the message (*PartitionID*, *MarketSegmentID*, etc.). If the field is not provided, and there are no other identifying data, the system default value of “XISX” is used.

If the *MarketID* field is provided in conjunction with other identifying data, and those data conflict — for example, the specified *PartitionID* or *MarketSegmentID* does not exist in the specified *MarketID* — the transaction is rejected.

Mass action requests cannot span multiple markets, a transaction must be sent for each discrete *MarketID*.

The *MarketID* field is returned on the **LogonResp** in the *Partitions* group, to provide an explicit mapping of the given partition to the given exchange.

5.1.8 Parties Group

The *Parties* group is used to list the parties involved in an order or trade.

The parties that are listed in this group are:

- Business Unit
- Clearing Account
- CMTA
- Sub-Account
- Client Account
- Entering Broker's name

5.1.9 Order Capacity and Clearing Capacity

OrderCapacity specifies the client role of the member submitting the order: Customer, Professional Customer, Proprietary, Away Market Maker, ISE Market Maker, or Broker-Dealer.

ClearingCapacity specifies how the trade is to clear at OCC: Customer, Firm, or Market Maker.

Please see **Table 23: Order/Clearing Capacity Values**, on page 47, for the allowed order/clearing capacity combinations.

5.1.10 Order Reference IDs

When an order is being entered in response to an auction, or to execute another specific order, the *RefOrderID* field or *RefCLOrdID* field is used to identify the target order.

When an order is being modified or canceled, the *OrderID* field or *OrigCLOrdID* field is used to identify the order to be modified or canceled.

5.1.11 Misc. Reference/Free Text Fields

There are multiple free text fields on an order to assist in tracking the order and trade. While *CLOrdID* is a BU's primary identifier to track an order, there is also the *Text* field, the *PrivateReference* field, and the *BranchSeqNbr* field, all of which are copied through to the trade.

5.1.12 String (Data Type) Fields

Unless specified otherwise, the *maximum* length for any string field is 20 characters. In addition, the maximum length of a specific field (e.g. *PartyID*) may vary by context.

5.1.13 Default Field Values

Field values that are described as "(default)" indicate that the field need not be included in the message if the default value is used. The field must be included if a non-default value is used.

5.1.14 ExecID

The *ExecID* field contains the ID assigned by the ME to the request transaction. It is guaranteed to be unique per product per business date. Although typed as a string, internally this field is an unsigned, 8-byte integer.

The *ExecID* field is found on the following messages:

- **Order Acknowledge Response**, on page 88
- **Delete All Quote Response**, on page 108
- **Order Lock Response**, on page 113
- **Order Release Response**, on page 114
- **Private Order Broadcast**, on page 121
- **Private Quote Broadcast**, on page 130
- **Deal Item Broadcast**, on page 135
- **Trade Item Broadcast**, on page 138
- **Private MM Execution Report Broadcast**, on page 142

5.1.15 TradeReportID

The *TradeReportID* field contains the ID assigned by the ME for an individual Deal or Trade.

The *TradeReportID* field is present on both the **DIB** and the **TIB**. On the **DIB**, it is guaranteed to be unique per product per business date per side (the deal represents one side of the execution). On the **TIB**, it is guaranteed to be unique per product per business date (the trade represents *both* sides of the execution).

NOTE: The deal and trade *TradeItemIDs* for the same execution may be the same number, but they are not the same ID.

5.2 New Order Single Request

New Order Single is used only to send new orders and auction order responses for simple instruments. This message also is used by PMMs to optionally trade against a locked customer order. This request uses the FIX New Order Single message type.

Various fields are conditionally required depending upon the request type and order type:

- Reserve orders require the various *Display...* fields.
- Auction orders and responses require the auction-related fields.
- All BUs are configured with a default clearing account. The client may provide alternate clearing information using the *Parties* group.

When executing against a locked order, the PMM identifies that order using the *RefOrderID* field, and sets *OrdType* equal to "Q" (Counter Order). The *TradePublishIndicator* is used to indicate if the ISE should publish the trade (the PMM has "stepped-up").

If the **New Order Single** cannot be immediately processed, a **Pending Response** is returned. A final status update is sent when the request is processed.

5.2.1 Message Format

The format of the **New Order Single** is shown below:

Table 31: New Order Single Request (MsgID=10100)

Tag	Field	Req	Value(s)	Comment
10100		Y		New Order Single
<Request Header>		Y	MsgType=D	New Order Single
11	ClOrdID	N		Client's OrderID
14012	Parties	N		
448	> PartyID	Y		Give-up/CMTA = 5 char. numeric. Sub-account = 4 char. alphanumeric. Customer Acct. = 10 char. alphanumeric. See Appendix N: Away Market Codes , on page 217, for allowed PartyID values when PartyRole = 94.
452	> PartyRole	Y	4=Clearing Firm (CMTA) 12=Executing Trader (Broker ID) 24=Customer Account 45=Secondary Account Number (Sub-account) 83=Clearing Account (Give-up) 94=Away Market	Value 24 is used in lieu of FIX tag 1 (Account). Value 94 is used only for linkage handling.
14063	TargetParties	N	0-1	One Receiving Party
1462	> TargetPartyID	Y		3 char. BU acronym. For example, ABC
8552	> TargetPartyRoleQualifier	Y	5=Preferred Market Maker 6=Directed Market Maker	
18	ExecInst	N	1=Not Held 6=Participate don't initiate G=All-or-None (AON) H=Reinstate on System Failure (Persist) f=Intermarket Sweep (ISO) h=External Routing Not Allowed (DNR)	1 = ALO, re-price – MUST appear in conjunction with 6. 6 = ALO, do not re-price (cancel)
1300	MarketSegmentID	Y		Product ID
48	SecurityID	Y		Instrument ID

Tag	Field	Req	Value(s)	Comment
54	Side	Y	1=Buy 2=Sell	Buy/Sell
60	TransactTime	Y		Client's Time
38	OrderQty	Y		
40	OrdType	Y	1=Market 2=Limit 3=Stop 4=Stop Limit Q=Counter-order selection	.
44	Price	N		Req'd if OrdType = 2 or 4
59	TimeInForce	N	0=Day (default) 1=Good Till Canceled 2=At the Opening 3=Immediate or Cancel 4=Fill Or Kill 6=Good Till Date	Default is DAY
432	ExpireDate	N	YYYYMMDD	Req'd if TimeInForce = 6
528	OrderCapacity	Y	C=Customer D=Professional Customer B=Broker/Dealer G=Proprietary N=Far Market Maker M=ISE Market Maker	Type of Member/Client
58	Text	N		Customer reference field
8521	ClearingCapacity	Y	1=Customer 2=Firm 3=Market Maker	OCC Account type
77	PositionEffect	Y	O=Open C=Close	Open-Close
99	StopPx	N		Req'd if OrdType = 3 or 4
1390	TradePublishIndicator	N	0=Do Not Publish Trade 1=Publish Trade (Default)	Related to linkage handling
1080	RefOrderID	N		for Counter Order/Auction response
8516	RefClOrdID	N		EAM to submit initial auction response
5476	PrivateReference	N		Customer reference field
9861	BranchSeqNbr	N		Customer reference field
8601	AuctionInst	N	0=Start auction (default) 1=Do not start auction (DNF)	Do not flash. Only Non –cust order (i.e. tag 528 not equal "C") can opt out of Flash auction.
110	MinQty	N	$1 < n \leq \text{OrderQty}$	
8602	MinQtyType	N	O=Once R=Repeat	

Tag	Field	Req	Value(s)	Comment
1138	DisplayQty	N		Initial Display Qty
1083	DisplayWhen	N	1=Immediate 2=Exhaust	
1084	DisplayMethod	N	1=Initial 3=Random	
1085	DisplayLowQty	N		If DisplayMethod = Random
1086	DisplayHighQty	N		If DisplayMethod = Random
8522	AuctionType	N	See Table 30: Auction Types , on page 67	Required for auction responses.
8523	AuctionAllocationPct	N	$0 \leq n \leq 0.4$ (default = 40%)	Percentage required. E.g., 30% (0.3) is entered as 30,000,000
14018	DisclosureInstGrp	N	0-7	
8577	> DisclosureType	Y	1=Volume 2=Price 3=Side 4=AON 5=General 6=Clearing Account 7=CMTA Account	
8578	> DisclosureInst	Y	0=No (Hide) 1=Yes (Disclose) 2=Use Default	To disclose fields
1092	PriceProtectionScope	N	1=Local Market 2=National (default)	Auction orders, only. 1 = Stopped
913	AgreementDesc	N	String (40 chars.)	Free text
9303	RoutingInst	N	1=Sweep	Sweep orders, only.

5.2.2 Response Message

An **Order Acknowledge Response** message is sent in response to the **New Order Single** request, and uses the ISE-defined Order Acknowledge message type. The format is shown below:

Table 32: Order Acknowledge Response (MsgID=10103)

Tag	Field	Req	Value(s)	Field
10103		Y		Order Acknowledge
<Response Header>		Y	MsgType=U1	OrderAcknowledge
17	ExecID	Y		Request message ID
60	TransactTime	Y		ME timestamp

The **Order Acknowledge** message is also sent in response to:

- **Order Cancel Replace Request**, on page 89
- **Order Cancel Request**, on page 92

- **New Order Multileg**, on page 95
- **Multileg Order Cancel Replace**, on page 98
- **New Order Cross Request**, on page 101

5.3 Alter Order Single Request

The **Order Cancel Replace Request** is used to modify existing orders for simple instruments. This request uses the FIX Order Cancel Replace Request message type. Some fields are conditionally required depending upon the request type and order type.

- The order to be altered is identified using either the *OrigClOrdID* or the *OrderID* field.
- All attributes of an order that are to be carried forward to the modified order must be specified in the modification request, otherwise they are reset to default values as defined in the **New Order Single**. For example, if the original order specified a time in force of GTC, and that field was absent on the modify request, the altered order would have a new time in force of Day.

Note: An order modification request for a non-persistent order with the time-in-force Day to time-in-force GTC (either persistent or non-persistent) will be rejected. GTC orders must be persistent, and are rejected if marked non-persistent. Modification of an order from non-persistent to persistent is not allowed.

- If the alter transaction cannot be processed immediately, a **Pending Response** is returned. A final status update is sent when the request is processed.

5.3.1 Message Format

The format of the **Order Cancel Replace Request** is shown below:

Table 33: Order Cancel Replace Request (MsgID=10101)

Tag	Field	Req	Value(s)	Comment
10101		Y		Order Cancel Replace Request
<Request Header>		Y	MsgType=G	OrderCancelReplaceRequest
60	TransactTime	Y		Client's Time
37	OrderID	N		Order to be altered
41	OrigClOrdID	N		Order to be altered
11	ClOrdID	N		
1300	MarketSegmentID	Y		Product ID
48	SecurityID	Y		Instrument ID
54	Side	Y	1=Buy 2=Sell	Buy/Sell
38	OrderQty	Y		Should be the "Total Intended Order Quantity" (including the amount already executed for this chain of orders)

Tag	Field	Req	Value(s)	Comment
40	OrdType	Y	1=Market 2=Limit 3=Stop 4=Stop Limit Q=Counter-order selection	
44	Price	N		
59	TimeInForce	N	0=Day (default) 1=Good Till Canceled 2=At the Opening 3=Immediate or Cancel 4=Fill Or Kill 6=Good Till Date	
432	ExpireDate	N	YYYYMMDD	For GTD Orders
528	OrderCapacity	Y	C=Customer D=Customer Professional B=Broker/Dealer G=Proprietary N=Away Market Maker M=Market Maker	Client Category
58	Text	N		Primary Free Text.
8521	ClearingCapacity	Y	1=Customer 2=Firm 3=Market Maker	Clearing Type
77	PositionEffect	Y	O=Open C=Close	
14012	Parties	N		
448	> PartyID	Y	See Appendix N: Away Market Codes , on page 217, for allowed PartyID values	Give-up/CMTA = 5 char. numeric. Sub-account = 4 char. alphanumeric. Customer Acct. = 10 char. alphanumeric.
452	> PartyRole	Y	4=Clearing Firm (CMTA) 12=Executing Trader 24=Customer Account 45=Secondary Account Number (Sub-account) 83=Clearing Account (Give-up)	
14063	TargetParties	N	0-1	One Receiving Party
1462	> TargetPartyID	Y		3 char. Bus. Unit acronym
8552	> TargetPartyRoleQualifier	Y	5=Preferred Market Maker 6=Directed Market Maker	Directed or Preferred
99	StopPx	N		Stop Price

Tag	Field	Req	Value(s)	Comment
18	ExecInst	N	1=Not Held 6=Participate don't initiate G=AON H=Reinstate on System Failure (Persist) f=Intermarket Sweep (ISO) h=External Routing Not Allowed (DNR)	1 = ALO, re-price – MUST appear in conjunction with 6. 6 = ALO, do not re-price (cancel)
5476	PrivateReference	N		Secondary Free Text
9861	BranchSeqNbr	N		Customer reference
8601	AuctionInst	N	0=Start Auction (default) 1=Do not start auction	Reserved for future use.
110	MinQty	N	$1 < n \leq \text{OrderQty}$	
8602	MinQtyType	N	0=Once R=Repeat	
1138	DisplayQty	N		Initial Display Qty
1083	DisplayWhen	N	1=Immediate 2=Exhaust	
1084	DisplayMethod	N	1=Initial 3=Random	
1085	DisplayLowQty	N		
1086	DisplayHighQty	N		
8522	AuctionType	N	See Table 30: Auction Types , on page 67	Required for auction responses.
8523	AuctionAllocationPct	N	$0 \leq n \leq 0.4$ (default = 40%)	Percentage required. E.g., 30% (0.3) is entered as 30,000,000
14018	DisclosureInstGrp	N		
8577	> DisclosureType	Y	1=Volume 2=Price 3=Side 4=AON 5=General 6=Clearing Account 7=CMTA Account	
8578	> DisclosureInst	N	0=No (Hide) 1=Yes (Disclose) 2=Use Default	
1092	PriceProtectionScope	N	1=Local Market 2=National Market (default)	Auction orders, only. 1 = Stopped
913	AgreementDesc	N	String (40 chars.)	Free text
9303	RoutingInst	N	1=Sweep	A Sweep order cannot be cancel/replaced.

5.3.2 Response Message

An **Order Acknowledge** is sent in response to the **Order Cancel Replace Request**. The format of the response is shown in **Table 32: Order Acknowledge Response (MsgID=10103)**, on page 88.

NOTE: The **POB**, not the ACK, is confirmation that the order has been modified.

5.4 Delete Order Request

The **Order Cancel Request** message is used to delete a single order, and is used for both simple and complex instrument orders. This request uses the FIX Order Cancel Request message type.

- The order to be deleted is identified using either the *OrigClOrdID* or the *OrderID* field. If the order cannot be deleted, or cannot be found, the request is rejected.
- If the order cannot be deleted at the time, but the cancel request is queued, a **Pending Response** is returned. A final status is sent when the request is processed.
- This message cannot be used to delete a quote.
- A **POB** is sent to confirm that the order has been deleted.

5.4.1 Message Format

The format of the **Order Cancel Request** message is shown below:

Table 34: Order Cancel Request (MsgID=10102)

Tag	Field	Req	Value(s)	Comment
10102		Y		Order Cancel Request
<Request Header>		Y	MsgType=F	OrderCancelRequest
37	OrderID	N		Target Exchange's OrderID
41	OrigClOrdID	N		Target Client's OrderID
11	ClOrderID	N	String (20 bytes)	Client ID for this message
1300	MarketSegmentID	Y		Product ID
48	SecurityID	Y		Instrument ID
1227	ProductComplex	N	1=Simple (default) 2=Standard Combination 3=Stock Combination	

5.4.2 Response Message

An **Order Acknowledge** is sent in response to the **Order Cancel Request**. The format of the response is shown in **Table 32: Order Acknowledge Response (MsgID=10103)**, on page 88.

NOTE: The **POB**, not the ACK, is confirmation that the order has been deleted.

5.5 Delete All Orders Request

The **Delete All Orders Request** message is used to delete *all* orders for a BU. It can be used for simple or complex instruments. This request uses the FIX Order Mass Action Request message type.

- The scope of this message is limited to a single instrument type — simple, complex, or complex w/Stock.
- The scope of the orders deleted, as specified by the *MassActionScope* field, can be defined as:
 - all orders in the market
 - all orders in a specific partition or partitions
 - all orders for a specific product or products
 - all orders for a specific instrument

In addition, the scope can be further refined using the *TargetParties* group to delete orders only for a specific user ID.

- If *MassActionScope* is set to 1 (Security), then one instance of the *TargetMarketSegment* group and the *SecurityID* field must also be set.
- If *MassActionScope* is set to 9 (Market Segment), then one or more instances of the *TargetMarketSegment* group must also be set.
- If *MassActionScope* is set to 100 (Partition), then one or more instances of the *PartitionGrp* group must also be set.
- The *TargetMarketSegmentGrp* and *PartitionGrp* groups are mutually exclusive.
- The initial response indicates that the request has been received. After a short period, the system responds with an ACK per product affected.
- A **POB** is sent for each order that is deleted.
- This message cannot be used to delete a specific order. Use the **Order Cancel Request**.
- This message cannot be used to delete quotes. Use the **Mass Quote** or **Delete All Quotes Request**.

NOTE: The **Delete All Orders Request** for Complex and Complex with Stock instruments with **Product** or **Market** Scope is **not** subject to throttling – it is not counted against the transaction limit or outstanding request limit, nor does the response reduce the outstanding counter.

5.5.1 Message Format

The format of the **Delete All Orders Request** is shown below:

Table 35: Delete All Orders Request (MsgID=10105)

Tag	Field	Req	Value(s)	Comment
10105		Y		Delete All Orders Request
<Request Header>		Y	MsgType=CA	OrderMassActionRequest
1373	MassActionType	Y	3=Cancel Orders	

Tag	Field	Req	Value(s)	Comment
1374	MassActionScope	Y	1=Security (Instrument) 8=Market 9=Market Segment (Product — 1 or more) 100=Partition(s)	
1301	MarketID	N	XISX (default) GMNI	Recommended if MassActionScope = 8 (market)
14034	TargetMarketSegmentGrp	N	0 or more if SecurityID not filled 1 if SecurityID is filled	Required if MassActionScope = 1 or 9
8618	> TargetMarketSegmentID	Y		
1227	ProductComplex	N	1=Simple Instrument (default) 2=Standard Combination 3=Stock Combination	
48	SecurityID	N		Required if MassActionScope = 1
14063	TargetParties	N	0-1	Used to target orders for a specific user
1462	> TargetPartyID	Y		target user ID
14003	PartitionGrp	N	0 or more	Required if MassActionScope = 100
5948	> PartitionID	Y		Partition ID

5.5.2 Response Message

The **Delete All Orders Request** results in multiple responses.

The initial **Delete All Orders Response** (*MassActionResponse* = 1) simply indicates that the request has been handled by the DTI; it does not mean that any order has been deleted.

If the request cannot be forwarded by the DTI to a particular partition (ME), the response contains an exception list, the *NotAffectedMarketSegment* group, containing a list of the products for which the request could not be fulfilled.

If the request was a market- or partition-scope request, an additional response *may* be received for each partition covered by the original request. This response contains the *NotAffectedMarketSegment* group, containing the list of the products in that partition for which the request could not be fulfilled (for example, a product is in a “halt” state). The products enumerated in this message are different from the products that may have been enumerated in the initial DTI response.

Subsequent product-specific responses are sent as the request is processed by the ME(s) and indicate that the request is Accepted, Rejected, or Accepted with Warning (Pending). A pending response is always followed by another response indicating the final status for that product. These responses never contain the *NotAffectedMarketSegment* group.

NOTE: The **POBs**, not the response messages, are final confirmation that orders have been deleted.

The **Delete All Orders Response** uses the FIX Order Mass Action Report message type, and the format is shown below:

Table 36: Delete All Orders Response (MsgID=10106)

Tag	Field	Req	Value(s)	Comment
10106		Y		Delete All Orders Response
<Response Header>		Y	MsgType=BZ	OrderMassActionReport
1373	MassActionType	Y	3=Cancel Orders	
1374	MassActionScope	Y	1=Security 8=Market 9=Market Segment 100=Partition(s)	
1375	MassActionResponse	Y	0=Rejected 1=Accepted 2=Accepted w/ Warning (Pending)	Initial response is always 1 Subsequent responses may contain any code. 2 is always followed by a final response
1376	MassActionRejectReason	N		If MassActionResponse = 0
58	Text	N		If MassActionResponse = 0
1301	MarketID	Y		
1300	MarketSegmentID	N		
1227	ProductComplex	Y	1=Simple Instrument 2=Standard Combination 3=Stock Combination	
48	SecurityID	N		Initial response, if present on request
14033	NotAffectedMarketSegmentGrp	N	0 or more	
8612	> NotAffectedMarketSegmentID	Y		

NOTE: This response behavior is consistent for the following *MassActionType* request messages:

Delete All Orders Request, on page 93

Quote Action Request, on page 109

MM Parameter Definition Request, on page 117

5.6 New Order Multileg Request

New Order Multileg is used to send new orders and auction order responses for complex instruments, only. This request uses the FIX New Order Multileg message type.

Various fields are conditionally required depending upon the request type and order type.

ClearingCapacity and *Position* information must be provided on each leg. All other fields are the same as for a **New Order Single**.

NOTE: Single-sided complex orders are only allowed up to four options legs plus a stock leg. Two-sided complex auctions are allowed up to eight options legs plus a stock leg.

5.6.1 Message Format

The format of the **New Order Multileg** request is shown below:

Table 37: New Order Multileg Request (MsgID=10115)

Tag	Field	Req	Value(s)	Comment
10115		Y		New Order Multileg
<Request Header>		Y	MsgType=AB	NewOrderMultileg
60	TransactTime	Y		Client's Time
11	ClOrdID	N		
1300	MarketSegmentID	Y		Product ID
48	SecurityID	Y		Instrument ID
1227	ProductComplex	N	2=Std. Combination (default) 3=Stock Combination	
54	Side	Y	1=Buy 2=Sell	
38	OrderQty	Y		
44	Price	N		
40	OrdType	Y	1=Market 2=Limit Q=Counter-order selection	
59	TimeInForce	N	0=Day (default) 1=Good Till Canceled 2=At the Opening 3=Immediate or Cancel 4=Fill Or Kill 6=Good Till Date	
432	ExpireDate	N	YYYYMMDD	For GTD Orders
528	OrderCapacity	Y	C=Customer D=Customer Professional B=Broker/Dealer G=Proprietary N=Away Market Maker M=Market Maker	Type of Member/Client
58	Text	N		Primary Free Text
15025	LegOrdGrp	Y	> 1	
609	> LegSecurityType	Y	OPT=Option CS=Common Stock	

Tag	Field	Req	Value(s)	Comment
624	> LegSide	N	5=Sell Short 6=Sell Short Exempt	
8575	> LegClearingCapacity	Y	1=Customer 2=Firm 3=Market Maker	
564	> LegPositionEffect	N	O=Open C=Close	Required on option leg.
14008	> NestedParties	N		
524	>> NestedPartyID	Y		Stock leg Give-up = 4 char. alphanumeric. Option Give-up/CMTA = 5 char. numeric. Sub-account = 4 char. alphanumeric.
538	>> NestedPartyRole	Y	4=Clearing Firm (CMTA) 45=Secondary Account Number (Sub-account) 83=Clearing Account (Give-up)	
8572	RelatedLowPrice	N		
8573	RelatedHighPrice	N		
8574	RelatedPriceSource	N	1=Underlying NBB 2=Underlying NBO	
14012	Parties	N	0-2	
448	> PartyID	Y	See Appendix N: Away Market Codes , on page 217, for allowed PartyID values	Customer Acct. = 10 char. alphanumeric.
452	> PartyRole	Y	12=Executing Trader (Broker ID) 24=Customer Account	Value 24 is used in lieu of FIX tag 1 (Account)
18	ExecInst	N	G=All-or-None (AON) H=Reinstate on System Failure (Persist) z=Cancel Remaining	"z" may be used to indicate that qty remaining after a complex exposure auction should be canceled, and not rest on the book.
8522	AuctionType	N	See Table 30: Auction Types , on page 67	Required for auction responses.
110	MinQty	N	$1 < n \leq \text{OrderQty}$	
1092	PriceProtectionScope	O	1=Local Market (default) 2=National	1 = Allowed to trade through 2 = Do not trade through
8602	MinQtyType	N	O=Once R=Repeat	
1080	RefOrderID	N		for Counter Order
8516	RefClOrdID	N		EAM to submit response
5476	PrivateReference	N		Customer reference field
9861	BranchSeqNbr	N		Customer reference field

Tag	Field	Req	Value(s)	Comment
14063	TargetParties	N	0-2	
1462	> TargetPartyID	Y		3 char. Bus. Unit acronym
8552	> TargetPartyRoleQualifier	N	5=Preferred Market Maker 6=Directed Market Maker	Directed or Preferred
1138	DisplayQty	N		Initial Display Qty
1083	DisplayWhen	N	1=Immediate 2=Exhaust	Ever trade or exhaust
1084	DisplayMethod	N	1=Initial 3=Random	
1085	DisplayLowQty	N		For random
1086	DisplayHighQty	N		For Random
8523	AuctionAllocationPct	N	$0 \leq n \leq 0.4$ (default = 40%)	Percentage required. E.g., 30% (0.3) is entered as 30,000,000
14018	DisclosureInstGrp	N	0 – 7	To disclose fields
8577	> DisclosureInstType	Y	1=Volume 2=Price 3=Side 4=AON 5=General 6=Clearing Account 7=CMTA Account	Fields to disclose
8578	> DisclosureInst	Y	0=No (Hide) 1=Yes (Disclose) 2=Use Default	

5.6.2 Response Message

An **Order Acknowledge** is sent in response to the **New Order Multileg** request. The format of the response is shown in **Table 32: Order Acknowledge Response (MsgID=10103)**, on page 88.

5.7 Alter Order Multileg Request

Multileg Order Cancel Replace is used to modify (alter) existing orders for complex instruments. This request uses the FIX Multileg Order Cancel Replace message type. Some fields are conditionally required depending upon the request type and order type.

- The order to be altered is identified using either the *OrigClOrdID* or the *OrderID* field.
- All attributes of an order that are to be carried forward to the modified order must be specified in the modification request, otherwise they are reset to default values as defined in the **New Order Multileg**.
- If the alter transaction cannot be processed immediately, a **Pending Response** is returned. A final status update is sent when the request is processed.

5.7.1 Message Format

The format of the **Multileg Order Cancel Replace** is shown below:

Table 38: Multileg Order Cancel Replace Request (MsgID=10116)

Tag	Field	Req	Value(s)	Comment
10116		Y		Multileg Order Cancel Replace
<Request Header>		Y	MsgType=AC	MultilegOrderCancelReplace
60	TransactTime	Y		Client's Time
37	OrderID	N		Order to be altered
41	OrigClOrdID	N		Order to be altered
11	ClOrdID	N		Client's OrderID
1300	MarketSegmentID	Y		Product
48	SecurityID	Y		Security
1227	ProductComplex	N	2=Standard Combination (default) 3=Stock Combination	
54	Side	Y	1=Buy 2=Sell	
38	OrderQty	Y		Should be the "Total Intended Order Quantity" (including the amount already executed for this chain of orders)
44	Price	N		
40	OrdType	Y	1=Market 2=Limit Q=Counter-order selection	
59	TimeInForce	N	0=Day (default) 1=Good Till Canceled 2=At the Opening 3=Immediate or Cancel 4=Fill Or Kill 6=Good Till Date	Default is DAY
432	ExpireDate	N	YYYYMMDD	For GTD Orders
528	OrderCapacity	Y	C=Customer D=Customer Professional B=Broker/Dealer G=Proprietary N=Away Market Maker M=Market Maker	Type of Member/Client
58	Text	N		Primary Free Text
15025	LegOrdGrp	Y		
609	> LegSecurityType	Y	OPT=Option CS=Common Stock	

Tag	Field	Req	Value(s)	Comment
624	> LegSide	N	5=Sell Short 6=Sell Short Exempt	
8575	> LegClearingCapacity	N	1=Customer 2=Firm 3=Market Maker	
564	> LegPositionEffect	N	O=Open C=Close	Open/Close
14008	> NestedParties	N	0-3	
524	>> NestedPartyID	Y		Stock leg Give-up = 4 char. alphanumeric. Option Give-up/CMTA = 5 char. numeric. Sub-account = 4 char. alphanumeric.
538	>> NestedPartyRole	Y	4=Clearing Firm (CMTA) 45=Secondary Account Number (Sub-account) 83=Clearing Account (Give-Up)	
8572	RelatedLowPrice	N		
8573	RelatedHighPrice	N		
8574	RelatedPriceSource	N	1=Underlying NBB 2=Underlying NBO	
14012	Parties	N	0-2	
448	> PartyID	Y		Customer Acct. = 10 char. alphanumeric.
452	> PartyRole	Y	12=Executing Trader (Broker ID) 24=Customer Account	
18	ExecInst	N	G=AON H=Reinstate on System Failure (Persist) z=Cancel remaining	"z" may be used to indicate that qty remaining after a complex exposure auction should be canceled, and not rest on the book.
1092	PriceProtectionScope	O	1=Local Market (default) 2=National	1 = Allowed to trade through 2 = Do not trade through
5476	PrivateReference	N		Secondary Free Text
9861	BranchSeqNbr	N		customer reference
14063	TargetParties	N	0-1	One Receiving Party
448	> TargetPartyID	Y		3 char. Bus. Unit acronym
8552	> TargetPartyRoleQualifier	Y	5=Preferred Market Maker 6=Directed Market Maker	Directed or Preferred
110	MinQty	N	$1 < n \leq \text{OrderQty}$	
8602	MinQtyType	N	O=Once R=Repeat	
1138	DisplayQty	N		Initial Display Qty

Tag	Field	Req	Value(s)	Comment
1083	DisplayWhen	N	1=Immediate 2=Exhaust	Ever trade or exhaust
1084	DisplayMethod	N	1=Initial 3=Random	
1085	DisplayLowQty	N		For random
1086	DisplayHighQty	N		For Random
8522	AuctionType	N	See Table 30: Auction Types , on page 67	Required for auction responses.
8523	AuctionAllocationPct	N	$0 \leq n \leq 0.4$ (default = 40%)	Percentage required. E.g., 30% (0.3) is entered as 30,000,000
14018	DisclosureInstGrp	N	0-7	To disclose fields
8577	> DisclosureType	Y	1=Volume 2=Price 3=Side 4=AON 5=General 6=Clearing Account 7=CMTA Account	Fields to disclose
8578	> DisclosureInst	Y	0=No (Hide) 1=Yes (Disclose) 2=Use Default	

5.7.2 Response Message

An **Order Acknowledge** is sent in response to the **Multileg Order Cancel Replace** request. The format of the response is shown in **Table 32: Order Acknowledge Response (MsgID=10103)**, on page 88.

NOTE: The **POB**, not the ACK, is confirmation that the order has been modified.

5.8 New Order Cross Request

New Order Cross is used to cross two orders with each other where the orders are not exposed in an auction. This request uses the FIX New Order Cross message type.

This cross request is permitted for Qualified Contingent Cross (QCC) and Customer-to-Customer Cross (CCC) crossing types. The message contains details of both the buy-side and sell-side orders.

When submitting a QCC order, the minimum order quantity is 1,000 for regular options and 10,000 for mini options. There is no minimum quantity requirement for a CCC order.

This message is used for both simple and complex instruments. For complex instruments, the **ProductComplex** field is required to indicate the type of instrument and clearing information must be entered for each leg.

To enter a Stopped Cross order, set *PriceProtectionScope* = "1" (Local). The *AgreementDesc* field is used to record the time at which the stopped cross price was agreed between the parties. This is a free text field.

5.8.1 Message Format

The format of the **New Order Cross** is shown below.

Table 39: New Order Cross Request (MsgID=10114)

Tag	Field	Req	Value(s)	Comment
10114		Y		New Order Cross
<Request Header>		N	MsgType=s	NewOrderCross
549	CrossType	Y	See Table 27: Cross Order Types , on page 59	
1300	MarketSegmentID	Y		Product
48	SecurityID	Y		Instrument ID
1227	ProductComplex	N	1=Simple (default) 2=Standard Combination 3=Stock Combination	
40	OrdType	Y	2=Limit	
44	Price	Y		
1092	PriceProtectionScope	N	1=Local 2=National (default)	1 = Stopped Cross
18	ExecInst	N	f=Intermarket Sweep (ISO)	Valid only for Cust-To-Cust Cross (CCC) orders.
59	TimeInForce	Y	3=Immediate or Cancel	
60	TransactTime	Y		Client's Time
913	AgreementDesc	N	String (40 char.)	Agreed upon time for the stopped price.
15059	SideCrossOrdModGrp	Y	2	
54	> Side	Y	1=Buy 2=Sell	
11	> ClOrdID	N		Client's Reference
38	> OrderQty	Y		Must be same on both sides
528	> OrderCapacity	Y	C=Customer D=Customer Professional B=Broker/Dealer G=Proprietary N=Away Market Maker M=Market Maker	Client category
8521	> ClearingCapacity	N	1=Customer 2=Firm 3=Market Maker	

Tag	Field	Req	Value(s)	Comment
77	> PositionEffect	N	O=Open C=Close	
14012	> Parties	N		Clearing for Simple instruments
448	>> PartyID	Y		Give-up/CMTA = 5 char. numeric. Sub-account = 4 char. alphanumeric. Customer Acct. = 10 char. alphanumeric.
452	>> PartyRole	Y	4=Clearing Firm (CMTA) 12=Executing Trader 24=Customer Account 45=Secondary Account Number (Sub-account) 83=Clearing Account (Give-Up)	
14027	> LegCrossGrp	N	0 or > 1	
609	>> LegSecurityType	Y	OPT=Option CS=Common Stock	
624	>> LegSide	N	5=Sell Short 6=Sell Short Exempt	
8575	>> LegClearingCapacity	N	1=Customer 2=Firm 3=Market Maker	
564	>> LegPositionEffect	N	O=Open C=Close	
14010	>> NestedParties3	N	0-3	
949	>>> Nested3PartyID	N		Stock leg Give-up = 4 char. alphanumeric. Option Give-up/CMTA = 5 char. numeric. Sub-account = 4 char. alphanumeric.
951	>>> Nested3PartyRole	N	4=Clearing Firm (CMTA) 45=Secondary Account Number 83=Clearing Account (Give-Up)	
9861	> BranchSeqNbr	N		
58	> Text	N		
5476	> PrivateReference	N		

5.8.2 Response Message

An **Order Acknowledge** is sent in response to the **New Order Cross** message. The format of the response is shown in **Table 32: Order Acknowledge Response (MsgID=10103)**, on page 88.

5.9 Mass Quote Request

Mass Quote is used to submit or delete quotes. This request uses the FIX Mass Quote message type.

The **Mass Quote** can be used to enter quotes for up to 100 instruments in one, and only one, product. Quotes for simple and complex instrument types can be entered on the same **Mass Quote** message.

Each instrument item (*QuotEntryGrp*) can contain a bid quote, an offer quote, or both; however an instrument's quotes can only appear once in each message — you cannot specify the bid in one item, and the offer in a separate item.

NOTE: An MM should not submit quotes for both sides if only one side is being changed — it is not necessary to restate the existing, unchanging quote.

The *QuoteID* is the MM's unique reference ID and is a required field. This field is mapped to the *QuoteMsgID* field on the **PQB**, and the *ClOrdID* field on the deal and trade broadcasts.

Two fields are used to specify quantity, *[Bid|Offer]Size* and *[Bid|Offer]TotalSize*.

...Size is used to submit a “new” quote, or anytime the current quantity is zero. In other words, whatever the current quantity, replace it with quantity of this *Size*.

...TotalSize is used to *modify* an existing quote, and the ME takes into account any traded quantity against the original “new” quote and adjust the open quantity in the book accordingly. (This is analogous to the behavior of a FIX Order Cancel Replace Request, where the total quantity represents open + traded quantity.) The ME continues to calculate and adjust the quantity until a “new” quote is submitted — at which point, the calculation resets.

Quantity Examples:

Example 1 —

- A “new” quote (*...Size*) is submitted for 100 @ \$1.05
- A trade occurs for 30; the book shows 70 @ \$1.05
- A “modify” quote (*...TotalSize*) to improve the price is submitted for 100 @ \$1.00
- The book shows 70 @ \$1.00

Example 2 —

- A “new” quote (*...Size*) is submitted for 100 @ \$1.05
- A “new” quote (*...Size*) to improve the price is submitted for 100 @ \$1.00
- Simultaneously, a trade occurs on the original quote for 30 (@ \$1.05)
- The book shows 100 @ \$1.00

Example 3 —

- A “new” quote (*...Size*) is submitted for 100 @ \$1.05
- A “modify” quote (*...TotalSize*) to improve the price is submitted for 100 @ \$1.00
- Simultaneously, a trade occurs on the original quote for 30 (@ \$1.05)
- The book shows 70 @ \$1.00

To delete any quote, both ...*Size* and ...*TotalSize* fields are set to zero (0) with *no price* (that is, the price field, [*Bid/Offer*]Px is *not* populated in the message).

NOTE: Deleting quotes is the *only* time both quantity fields are populated. For regular quoting, only one quantity field can be populated.

The **Mass Quote** can be used to simultaneously add “new,” modify existing, and delete existing quotes.

5.9.1 Message Format

The format of the **Mass Quote** message is shown below:

Table 40: Mass Quote Request (MsgID=10203)

Tag	Field	Req	Value(s)	Comment
10203		Y		Mass Quote
<Request Header>		Y	MsgType=i	MassQuote
117	QuoteID	Y		MM’s unique reference
15049	QuotSetGrp	Y	1	Only 1
1300	> MarketSegmentID	Y		Product ID
15043	> QuotEntryGrp	Y	> 0	
48	>> SecurityID	Y		Instrument ID
1227	>> ProductComplex	N	1=Simple Instrument (default) 2=Std. Combination 3=Stock Combination	
132	>> BidPx	N		BidPx, OfferPx, or both must be specified.
134	>> BidSize	N		
8579	>> BidTotalSize	N		Open + Traded Qty
133	>> OfferPx	N		BidPx, OfferPx, or both must be specified.
135	>> OfferSize	N		
8580	>> OfferTotalSize	N		Open + Traded Qty

5.9.2 Response Message

The **Mass Quote Acknowledgement** is sent to indicate that the **Mass Quote** request has been processed. If any individual quote item was rejected, or could not otherwise be processed, the *QuotSetAckGrp* is present, listing the quotes that were not processed in the *QuotEntryAckGrp*. An invalid **Mass Quote** message, as opposed to an invalid quote, is rejected. For example, a **Mass Quote** where every quote is invalid is nevertheless acknowledged, with every quote rejected. A **Mass Quote** for instruments in a product that the MM is not authorized to quote is rejected.

The **Mass Quote Acknowledgement** uses the FIX Mass Quote Acknowledgement message type, and the format is shown below:

Table 41: Mass Quote Acknowledgement (MsgID=10204)

Tag	Field	Req	Value(s)	Comment
10204		Y		Mass Quote Acknowledgement
<Response Header>		Y	MsgType=b	MassQuoteAcknowledgement
60	TransactTime	Y		ME timestamp
117	QuoteID	Y		MM's unique reference
297	QuoteInactiveStatus	N	0=All active 1=Simple Inactive 2=Standard Combo Inactive 3=Simple and Standard Combo Inactive 4=Stock Combo Inactive, 5=Simple and Stock Combo Inactive 6=Standard Combo and Stock Combo Inactive 7= All Inactive	
15048	QuotSetAckGrp	N	0-1	
1300	> MarketSegmentID	Y		Product ID
15042	> QuotEntryAckGrp	Y	> 0	Number of reject items
48	>> SecurityID	Y		Instrument ID
1227	>> Product Complex	Y	1=Simple Instrument 2=Standard Combination 3=Stock Combination	
8547	>> BidCxlSize	N		Qty Canceled
8548	>> OfferCxlSize	N		Qty Canceled
1167	>> QuoteEntryStatus	Y	0=Accepted 5=Rejected 10=Pending 100=Accept With Warning	
368	>> QuoteEntryRejectReason	N		
1328	>> RejectText	N	Error Text	
8549	>> QuoteEntryRejectAction	N	0=None 1=Quote Canceled	

If any quote entry is rejected (*QuoteEntryStatus* = 5), both the resting bid and ask quotes, if present, are canceled. The actual quantity canceled is indicated in the *[Bid/Offer]CxlSize* fields.

QuoteEntryStatus = 10 (pending) occurs when an instrument or an order referencing that instrument is frozen. Upon processing after the instrument is unfrozen, an additional **Mass Quote ACK** with the actual (final) status of quote transaction is sent (0, 5, or 100).

QuoteEntryStatus is set to 100 (Accepted with Warning) in the following scenarios:

- A request for quote deletion resulted in no action because the quote side could not be found.
- A request for quote modification resulted in no action because the quote side could not be found.
- A request for quote modification resulted in the quote side being deleted, because it attempted to reduce the quantity of the quote side by more than what was still available.
- A new, single-sided quote crossed the resting quote on the other side, causing the resting quote to be deleted.

Except for the last case, the quote on the other side remains unaffected.

The *QuoteInactiveStatus* field provides current status of users quotes in the product by instrument type.

5.9.3 Quote Reject Reasons

Most quote rejects are straightforward and self-explanatory — “Invalid Limit Price.” The following table, ordered by reject reason code, lists some of the more unusual and esoteric quote reject reasons.

Table 42: Quote Reject Reasons

<i>QuoteEntryRejectReason</i>	<i>RejectText</i>	<i>Comment</i>
10053	INVALID_ORDER_QTY	The requested quantity is below the allowable minimum quote size. See Appendix B: ISE Parameters , on page 158.
26662	CROSSES_MINIMUM_NET_PRICE	A quote for a complex instrument is less than the absolute minimum price for that instrument. For example, a complex instrument defined as Buy 1 of leg 1 and Buy 1 of leg 2 <i>cannot</i> be priced less than \$0.02.
30110	INVALID_USAGE_QUOTING_MODEL	A quote as been entered with <i>both</i> quantity fields populated, and at least one qty field > 0.
30117	INVALID_USAGE_LIMIT_PRICE	1. A delete quote (both quantity fields == zero) also contains the price field. 2. The price field is populated, but <i>neither</i> quantity field is populated.
30140	INSTRUMENT_DUPLICATED	The specified SecurityID appears on multiple quote items. All instances of the duplicate are rejected.

5.10 Delete All Quotes Request

Delete All Quotes Request is used to delete *all* quotes for all instruments in one — and only one — product. This request uses the ISE-defined Quote Action Request message type.

Quotes to be deleted can also be filtered by user, using the *TargetParties* group, and/or side (Bid/Ask), using the *Side* field. When *Side* is specified, only quotes on that side (in all instruments) are deleted. When a user is specified, only quotes from that user are deleted. Please note that only one user can be specified.

To delete a single quote, or only a few specific quotes, use the **Mass Quote**.

5.10.1 Message Format

The format of the **Delete All Quote Request** message is shown below:

Table 43: Delete All Quotes Request (MsgID=10205)

Tag	Field	Req	Value(s)	Comment
10205		Y		Delete All Quotes Request
<Request Header>		Y	MsgType=U10	QuoteActionRequest
1373	MassActionType	Y	3=Cancel Quotes	
1374	MassActionScope	Y	9=All quotes for a market segment	
1300	MarketSegmentID	Y		Product ID
1227	ProductComplex	N	1=Simple Instrument (default) 2=Standard Combination 3=Stock Combination	
14063	TargetParties	N	0-1	
1462	> TargetPartyID	Y		User's quotes to be deleted
54	Side	N	1=Buy 2=Sell	

5.10.2 Response Message

The **Delete All Quotes Response** is positive acknowledgement that quotes for the indicated *MarketSegmentID* have been deleted. The message uses the ISE-defined Quote Action Report message type, and the format is shown below:

Table 44: Delete All Quotes Response (MsgID=10206)

Tag	Field	Req	Value(s)	Comment
10206		Y		Delete All Quotes Response
<Response Header>		Y	MsgType=U11	QuoteActionReport
1373	MassActionType	Y	3=Cancel quotes	Constant
1374	MassActionScope	Y	9=All quotes for a market segment	Constant
1375	MassActionResponse	Y	1=Accepted 2=Accepted with warning	
17	ExecID	Y		
60	TransactTime	Y		ME timestamp

Tag	Field	Req	Value(s)	Comment
1300	MarketSegmentID	Y		Product ID
1227	Product Complex	N	1=Simple Instrument (default) 2=Standard Combination 3=Stock Combination	

MassActionResponse is set to 2 (Accepted with Warning) to indicate that quote deletions are pending (because, for example, the instrument is in a frozen state). A final **Delete All Quotes Response** is sent when all pending deletions have been performed.

5.11 Quote Action Request

The **Quote Action Request** message is used to activate and inactivate quotes. This request uses the ISE-defined Quote Action Request message type.

(Please refer to **Section 4.10.3, Quote Inactivation**, on page 69, for a complete description of quote inactivation behavior.)

The scope of the request (activate or inactivate) is limited to a single instrument type, and is determined by the *MassActionScope* field and other conditionally required fields:

- 8 = All quotes in all products for the specified market (*MarketID* field) for the given instrument type
- 9 = All quotes for the specified product(s) (*TargetMarketSegment* group) for the given instrument type in the given market
- 100 = All quotes for all products in the specified partition(s) (*PartitionGrp*) for the given instrument type in the given market

5.11.1 Message Format

The format of the **Quote Action Request** is shown below:

Table 45: Quote Action Request (MsgID=10201)

Tag	Field	Req	Value(s)	Comment
10201		Y		Quote Action Request
<Request Header>		Y	MsgType=U10	QuoteActionRequest
1373	MassActionType	Y	1=Suspend quotes 2=Release quotes from suspension 100=Suspend and delete quotes	
1374	MassActionScope	Y	8=All quotes for a market 9=All quotes for a market segment (product) 100=All quotes for a partition	
1301	MarketID	N	XISX (default) GMNI	Recommended if MassActionScope = 8 (Market)

Tag	Field	Req	Value(s)	Comment
1227	ProductComplex	N	1=Simple (default) 2=Standard Combination 3=Stock Combination	
14034	TargetMarketSegmentGrp	N	0 or more	Required if MassActionScope = 9
8618	> TargetMarketSegmentID	Y		Product ID
14003	PartitionGrp	N	0 or more	Required if MassActionScope = 100
5948	> PartitionID	Y		Partition ID

5.11.2 Response Message

The **Quote Action Request** results in multiple responses.

The initial **Quote Action Report** (*MassActionResponse* = 1) simply indicates that the request has been handled by the DTI; it does not mean that any quotes have been affected.

If the request cannot be forwarded by the DTI to a particular partition (ME), the response contains an exception list, the *NotAffectedMarketSegment* group, containing a list of the products for which the request could not be fulfilled.

If the request was a market or partition-scope request, an additional response *may* be received for each partition covered by the original request. This response contains the *NotAffectedMarketSegment* group, containing the list of the products in that partition for which the request could not be fulfilled (for example, the MM is not assigned that product, or a product is in a “halt” state). The products enumerated in this message are different from the products that may have been enumerated in the initial DTI response.

Subsequent product-specific responses are sent as the request is processed by the ME(s) and indicate that the request is Accepted, Rejected, or Accepted with Warning (Pending). A Pending response is always followed by another response indicating the final status for that product. These responses never contain the *NotAffectedMarketSegment* group.

The final **Quote Action Report** for a product is positive acknowledgement that the requested action has taken place, or has been rejected.

The message uses the ISE-defined Quote Action Report message type, and the format is shown below:

Table 46: Quote Action Response (MsgID=10202)

Tag	Field	Req	Value(s)	Comment
10202		Y		Quote Action Report
<Response Header>		Y	MsgType=U11	QuoteActionReport

Tag	Field	Req	Value(s)	Comment
1373	MassActionType	Y	1=Suspend quotes 2=Release quotes from suspension 100=Suspend and delete quotes	
1374	MassActionScope	Y	8=All quotes for a market 9=All quotes for a market segment 100=All quotes for all products in the specified partition(s)	
1375	MassActionResponse	Y	0=Rejected 1=Accepted 2=Accepted With Warning	Initial response is always 1 Subsequent responses may contain any code. 2 is always followed by a final response
1376	MassActionRejectReason	N		If MassActionResponse = 0
58	Text	N		If MassActionResponse = 0
1301	MarketID	Y		
1300	MarketSegmentID	N		product-specific response
1227	ProductComplex	Y	1=Simple Instrument 2=Standard Combination 3=Stock Combination	
14033	NotAffectedMarketSegmentGrp	N	0 or more	
8612	> NotAffectedMarketSegmentID	Y		

NOTE: The **Quote Action Request** to inactivate or to suspend and delete quotes, is **not** subject to throttling – it is not counted against the transaction limit or outstanding request limit, nor does the response reduce the outstanding counter.

5.12 Change Instrument State Request

The **Change Instrument State Request** message is used by the PMM, only, to change the state of simple instruments to “ready to trade” at the opening or after a temporary trading halt. This request uses the ISE-defined Security Mass Status Change request message type.

This message can be used to open *all* instruments within a product, or a *single* instrument within a product.

NOTE: Complex instruments are opened automatically by the system after the simple instruments have been opened. There is no PMM procedure for opening complex instruments.

5.12.1 Message Format

The format of the **Change Instrument State Request** message is shown below:

Table 47: Change Instrument State Request (MsgID=10300)

Tag	Field	Req	Value(s)	Comment
10300		Y		Change Instrument State Request
<Request Header>			MsgType=U12	SecurityMassStatusChangeRequest
1300	MarketSegmentID	Y		Product ID
1679	SecurityMassTradingStatus	Y	17=Ready to trade	
15017	InstrmtGrp	N	0-1	If present, only 1 allowed
48	> SecurityID	Y		Instrument ID

5.12.2 Response Message

The **Change Instrument State Response** is positive acknowledgement that the requested instrument(s) has (have) changed state. If any instrument's state was not changed, the *InstrmtGrp* group contains details of the failed state change. The message uses the ISE-defined Security Mass Status Change Report message type, and the format is shown next.

Table 48: Change Instrument State Response (MsgID=10301)

Tag	Field	Req	Value(s)	Comment
10301		Y		Change Instrument State Response
<Response Header>		Y	MsgType=U13	SecurityMassStatusChangeReport
60	TransactTime	Y		Client's Time
1300	MarketSegmentID	Y		Product ID
1679	SecurityMassTradingStatus	Y	17=Ready to trade	
15017	InstrmtGrp	N		
48	> SecurityID	Y		Instrument ID
326	> SecurityTradingStatus	Y	1=Opening Delay 2=Trading Halt 17=Ready to Trade 18=Not available for Trading 22=Opening Rotation 23=Fast Market	Current trading state
1174	> SecurityTradingEvent	Y	100=State Change Requested 103=Failed Opening Rotation 107=Instrument Suspended 112=State Change Not Allowed 113=Acting PMM assigned	Reason code (see Appendix I: Security Trading Event Codes , on page 204, for the complete list)

NOTE: The **Change Instrument State Request** is **not** subject to throttling – it is not counted against the transaction limit or outstanding request limit, nor does the response reduce the outstanding counter.

5.13 Order Lock Request

The **Order Lock Request** message is used by the PMM to lock an order that is resting on the order book. This message can only be used for simple instruments — orders for complex instruments cannot be locked. This request uses the FIX Order Cancel Replace Request message type.

If the referenced order does not exist, the request is rejected.

5.13.1 Message Format

The format of the **Order Lock Request** is shown below:

Table 49: Order Lock Request (MsgID=10108)

Tag	Field	Req	Value(s)	Comment
10108		Y		Order Lock Request
<Request Header>		Y	MsgType=G	OrderCancelReplaceRequest
37	OrderID	Y		Target Exchange OrderID
1300	MarketSegmentID	Y		Product ID
48	SecurityID	Y		Instrument ID
18	ExecInst	Y	S=Suspend	Instruction to lock

5.13.2 Response Message

The **Order Lock Response** uses the FIX Execution Report message type, and the format is shown below:

Table 50: Order Lock Response (MsgID=10110)

Tag	Field	Req	Value(s)	Comment
10110		Y		Order Lock Response
<Response Header>		Y	MsgType=8	ExecutionReport
14012	Parties	N	0 or more	
448	> PartyID	Y		
452	> PartyRole	Y	4=Clearing Firm (CMTA) 59=Executing Unit (BU) 83=Clearing Account (Give-Up)	
17	ExecID	Y		
60	TransactTime	Y		ME timestamp
150	ExecType	Y	9=Suspended (Locked) L=Activated (Released)	
1300	MarketSegmentID	Y		Product ID
48	SecurityID	Y		Instrument ID
18	ExecInst	Y	S=Suspend	
37	OrderID	Y		Exchange OrderID

Tag	Field	Req	Value(s)	Comment
528	OrderCapacity	Y	C=Customer D=Customer Professional B=Broker/Dealer G=Proprietary N=Away Market Maker M=Market Maker	Client Category
54	Side	Y	1=Buy 2=Sell	
8533	LockedQty	Y		Total locked qty
8544	SecondaryLockedQty	Y		Qty this lock
40	OrdType	Y	1=Market 2=Limit	
44	Price	N		

5.14 Order Release Request

The **Order Release Request** is used to release a locked order. This request uses the FIX Order Cancel Replace Request message type.

5.14.1 Order Release Request Message Format

The format of the **Order Release Request** message is shown below:

Table 51: Order Release Request (MsgID=10109)

Tag	Field	Req	Value(s)	Comment
10109		Y		Order Release Request
<Request Header>		Y	MsgType=G	OrderCancelReplaceRequest
37	OrderID	Y		Target Exchange OrderID
1300	MarketSegmentID	Y		Product ID
48	SecurityID	Y		Instrument ID
8603	ReleaseInst	N	1=ISO 2=No ABBO Check	
18	ExecInst	Y	q=Release from suspension	Instruction to release
8543	ReleaseQty	N		Qty to be released
44	Price	N		Trade to price then lock again

5.14.2 Order Release Response Message

The format of the **Order Release Response** message is shown below:

Table 52: Order Release Response (MsgID=10111)

Tag	Field	Req	Value(s)	Comment
10111		Y		Order Release Response

Tag	Field	Req	Value(s)	Comment
<Response Header>		Y	MsgType=8	ExecutionReport
17	ExecID	Y		
60	TransactTime	Y		ME timestamp
150	ExecType	Y	L=Activated (Released)	
1300	MarketSegmentID	Y		Product ID
48	SecurityID	Y		Instrument ID
18	ExecInst	Y	q=Release from suspension	
8603	ReleaseInst	N	1=ISO 2=No ABBO Check	
37	OrderID	Y		
8533	LockedQty	Y		
8544	SecondaryLockedQty	N		
8543	ReleaseQty	Y		Qty released
1102	TriggerPrice	N		Released limit price

5.15 Add Complex Instrument Request

The **Add Complex Instrument Request** message is used to create a complex instrument. This request uses the FIX Security Definition Request message type.

Standard combinations can be created with two to eight legs. Stock combinations can be created with one stock leg and one to eight options legs.

Please see **Section 4.11.2, Creating Complex Instruments**, on page 74, for a complete discussion on complex instrument creation.

NOTE: A request for the creation of an already existing complex instrument does not result in the creation of an additional complex instrument. The response to this request describes the existing instrument.

5.15.1 Message Format

The format of the **Add Complex Instrument Request** message is shown below:

Table 53: Add Complex Instrument Request (MsgID=10306)

Tag	Field	Req	Value(s)	Comment
10306		Y		Add Complex Instrument Request
<Request Header>		Y	MsgType=c	SecurityDefinitionRequest
1300	MarketSegmentID	Y		Product ID
1227	ProductComplex	N	2=Standard Combination (default) 3=Stock Combination	
14017	InstrumentLegGrp	Y	> 1	

Tag	Field	Req	Value(s)	Comment
602	> LegSecurityID	Y		Simple instrument ID or reference data Underlying ID (for stock leg)
624	> LegSide	Y	1=Buy 2=Sell	Buy/Sell
623	> LegRatioQty	Y	$1 \leq n \leq 9,999$	Ratio
609	> LegSecurityType	N	OPT=Option (Default) CS=Common Stock	

5.15.2 Response Message

The **Add Complex Instrument Response** message is sent in response to the **Add Complex Instrument Request** and contains the instrument ID and definition of the created complex instrument.

NOTE: The leg sides (Buy/Sell) and/or the order of the legs may differ from the original request.

The message uses the FIX Security Definition message type. The format is shown below:

Table 54: Add Complex Instrument Response (MsgID=10307)

Tag	Field	Req	Value(s)	Comment
10307		Y		Add Complex Instrument Response
<Response Header>		Y	MsgType=d	SecurityDefinition
964	SecurityReportID	Y		
60	TransactTime	Y		ME timestamp
15132	MarketSegmentGrp	Y	1	
1300	> MarketSegmentID	Y		Product ID
48	SecurityID	Y		Complex instrument ID
1227	ProductComplex	Y	2=Standard Combination 3=Stock Combination	
779	LastUpdateTime	Y		Creation Time
14017	InstrumentLegGrp	Y	> 1	
602	> LegSecurityID	Y		
624	> LegSide	Y	1=Buy 2=Sell	
623	> LegRatioQty	Y		
609	> LegSecurityType	N	OPT=Option (default) CS=Common Stock	
8614	NumberOfSecurities	Y		Count of complex instruments created by the BU for this product.

5.16 MM Parameter Definition Request

The **MM Parameter Definition Request** message is used by MMs to set tick-worse, speed bump (curtailment), and/or stock leg short sale indicator parameters for one or more products. This request uses the ISE-defined MM Parameter Definition Request message type.

NOTE: All curtailment parameters *must be set* for a product/instrument-type in order to quote instruments in that product. There is no requirement to set other MM parameters in order to quote. The system does not provide default parameters.

Please see **Section 4.10, Quote Related Functions**, on page 68, for a complete discussion of the various MM parameters.

- The scope of the request is limited to a single instrument type.
- The *TargetMarketSegmentGrp* group and the *PartitionGrp* group are mutually exclusive – if *TargetMarketSegmentGrp* is present, *PartitionGrp* cannot be present, and *vice versa*.
- If the *TargetMarketSegmentGrp* group is present, the request is for the specified product(s).
- If the *PartitionGrp* group is present, the request is for all products in the specified partition(s).
- If neither the *TargetMarketSegmentGrp* nor the *PartitionGrp* group is present, the request is for the whole market. In this way, default parameters can be entered for the entire market, with subsequent requests redefining those parameters for subsets of products.
- Tick-worse parameters are set using the *QuoteOffsetValue* and *NewQuoteSize* fields.
- Speed bump (curtailment) parameters are set using the four counter fields: *CumQty*, *PctCount*, *Delta*, and *Vega*; and the timer field *ExposureDuration*.
- Short Sale indicator is set using the *LegSide* field.
- There is *no* speed bump for the stock combination instrument type.
- Tick-worse, speed bump, auto-rotate, and short sale indicator parameters are set independently of each other.
- To turn off (unset) an existing tick-worse parameter, set either *QuoteOffsetValue* or *NewQuoteSize* to zero (0).
- To turn off (unset) the Short Sale indicator, set the new value to 'B' ("As Defined").
- If *RotateRequestIndicator* is set ('Y'), and the PMM's offer is present, instruments are rotated *and* moved into regular trading at start of day or after a halt without any action on the part of the PMM.
- To turn off (unset) *RotateRequestIndicator*, set the new value to 'N' ("do not rotate").
- To effectively ignore an individual curtailment counter, set that counter field to the largest value (MAX_INT32).
- Setting *ExposureDuration* to the smallest value (500) and all counter fields to the largest value (MAX_INT32) effectively disables curtailment protections.

5.16.1 Message Format

The format of the **MM Parameter Definition Request** is shown below:

Table 55: MM Parameter Definition Request (MsgID=10308)

Tag	Field	Req	Value(s)	Comment
10308		Y		MM Parameter Definition Request
<Request Header>		Y	MsgType=U14	MMParameterDefinitionRequest
1301	MarketID	N	XISX (default) GMNI	Recommended for market-scope request
1227	ProductComplex	N	1=Simple Instrument (default) 2=Standard Combination 3=Stock Combination	
14034	TargetMarketSegmentGrp	N		
8618	> TargetMarketSegmentID	Y		Product ID
14003	PartitionGrp	N		
5948	> PartitionID	Y		Partition ID
14	CumQty	N	$1 \leq n \leq \text{MAX_INT32}$	
8615	PctCount	N	$99 \leq n \leq \text{MAX_INT32}$	Integer. E.g. 100% = 100
7966	Delta	N	$1 \leq n \leq \text{MAX_INT32}$	
7968	Vega	N	$1 \leq n \leq \text{MAX_INT32}$	
1629	ExposureDuration	N	$500 \leq n \leq 30,000$	milliseconds
8616	QuoteOffsetValue	N	$0 \leq n \leq 8$	ticks back (zero disables)
8617	NewQuoteSize	N	$n \geq 0$	quantity (zero disables)
624	LegSide	N	5=Sell Short B=As Defined (default — No Short Sale)	Short Sale Indicator
8647	RotateRequestIndicator	N	Y=Yes N=No	

5.16.2 Response Message

The **MM Parameter Definition Request** results in multiple responses.

The initial **MM Parameter Definition Request Ack** (*MassActionResponse* = 1) simply indicates that the request has been handled by the DTI; it does not mean that any parameters have been affected.

If the request cannot be forwarded by the DTI to a particular partition (ME), the response contains an exception list, the *NotAffectedMarketSegment* group, containing a list of the products for which the request could not be fulfilled.

If the request was a market or partition-scope request, an additional response *may* be received for each partition covered by the original request. This response contains the *NotAffectedMarketSegment* group, containing the list of the products in that partition for which the request could not be fulfilled (for example, the MM is not assigned that product, or a product is in a “halt” state). The products enumerated in this message are different from the products that may have been enumerated in the initial DTI response.

Subsequent product-specific responses are sent as the request is processed by the ME(s) and indicate that the request is Accepted, Rejected, or Accepted with Warning (Pending). A Pending response is always followed by another response indicating the final status for that product. These responses never contain the *NotAffectedMarketSegment* group.

The final product ACK (Accepted or Rejected) is confirmation that the intended action has taken place.

The message uses the ISE-defined **MM Parameters Definition Request Ack** message type. The format is shown below:

Table 56: MM Parameter Definition Request Acknowledge (MsgID=10309)

Tag	Field	Req	Value(s)	Comment
10309		Y		MM Parameter Definition Request Ack
<Request Header>		Y	MsgType=U14	MMPParametersDefinitionRequestAck
1301	MarketID	Y		
1300	MarketSegmentID	N		product-specific response
1227	ProductComplex	Y	1=Simple Instrument 2=Standard Combination 3=Stock Combination	
8636	MMReqStatus	Y	0=Accepted 1=Accepted With Warnings (Pending) 2=Rejected	
8637	MMReqResult	N		
58	Text	N		
14033	NotAffectedMarketSegmentGrp	N	0 or more	
8612	> NotAffectedMarketSegmentID	Y		Product ID

5.17 Modify Deal Request

The **Deal Item Maintenance Instruction Request** is used to change clearing information on an existing deal. This request uses the FIX Trade Capture Report message type.

A single deal can be broken up into multiple allocations, but the sum total quantity must equal the original deal quantity. The deal price cannot be changed. Only the current business day's deals can be adjusted.

The deal to be changed is referenced by the *TradeReportID* field. This field is captured from the **DIB**.

Fields that can be modified are:

- Free text fields (*Text*, *PrivateReference*, *BrSeqNum*)
- Clearing Type (*OrderCapacity*, *ClearingCapacity*, *PositionEffect*)
- Clearing Account information, as specified in the *Parties* group (Give-up, CMTA, Sub-account)

- Other tracking information, as specified in the *Parties* group (Executing Trader, Customer Account)

5.17.1 Message Format

The format of the Deal Item Maintenance Instruction Request is shown below:

Table 57: Deal Item Maintenance Instruction Request (MsgID=10402)

Tag	Field	Req	Value(s)	Comment
10402		Y		Deal Item Maintenance Instruction Request
<Request Header>		Y	MsgType=AE	TradeCaptureReport
571	TradeReportID	Y		The Deal to be changed
487	TradeReportTransType	Y	2=Replace	
1300	MarketSegmentID	Y		Product ID
48	SecurityID	Y		Instrument ID
32	LastQty	Y		Original deal qty
31	LastPx	Y		Original deal price
8582	MatchDate	Y		YYYYMMDD
14019	TrdCptRptSideGrp	Y	1	Only 1
54	> Side	Y	1=Buy 2=Sell	
15060	> TrdAllocGrp	Y	> 0	New Deal Items
80	>> AllocQty	Y		New deal quantity
14009	>> NestedParties2	Y	1 – 5	Clearing info for each new deal
757	>>> Nested2PartyID	Y		Stock leg Give-up = 4 char. alphanumeric. Option Give-up/CMTA = 5 char. numeric. Sub-account = 4 char. alphanumeric. Customer Acct. = 10 char. alphanumeric.
759	>>> Nested2PartyRole	Y	4=Clearing Firm (CMTA) 12=Executing Trader 24=Customer Account 45=Secondary Account Number (Sub-account) 83=Clearing Account (Give-Up)	Clearing Account must be specified
161	>> AllocText	N		
5476	>> PrivateReference	N		
9861	>> BranchSeqNbr	N		
1047	>> AllocPositionEffect	Y	O=Open C=Close	
8588	>> AllocClearingCapacity	Y	1=Customer 2=Firm 3=Market Maker	Must be an allowed value for the order capacity

Tag	Field	Req	Value(s)	Comment
8589	>> AllocOrderCapacity	Y	C=Customer D=Customer Professional B=Broker/Dealer G=Proprietary N=Away Market Maker M=Market Maker	Order capacity CANNOT be changed from Customer

5.17.2 Response Message

The **Deal Item Maintenance Instruction Response** is positive acknowledgement that the requested deal modification(s) took place.

In addition, the affected parties receive a **DIB** to reverse the original deal, and one or more **DIBs** for each new deal that has been created.

The message uses the FIX Trade Capture Report Ack message type. The format is shown below:

Table 58: Deal Item Maintenance Instruction Response (MsgID=10403)

Tag	Field	Req	Value(s)	Comment
10403		Y		Deal Item Maintenance Instruction Response
	<Response Header>	Y	MsgType=AR	TradeCaptureReportAck
487	TradeReportTransType	Y	2=Replace	
60	TransactTime	Y		TM timestamp
8582	MatchDate	Y		
1300	MarketSegmentID	Y		Product ID
48	SecurityID	Y		Instrument ID
571	TradeReportID	Y		The original deal ID
75	TradeDate	Y		Date of new deal
14019	TrdCptRptSideGrp	Y	1	
54	> Side	Y	1=Buy 2=Sell	
15060	> TrdAllocGrp	Y	> 0	
80	>> AllocQty	Y		Quantity of new deal
467	>> IndividualAllocID	Y		Identifier of new deal

5.18 Member Kill Switch Request

The **Member Kill Switch Request** is used to block another user's or BU's ability to enter new orders, alter existing orders, delete all orders, and create complex instruments. The transaction also causes all open orders for the specified BU/user to be deleted.

Only specifically entitled users can send a **Member Kill Switch Request**. If an unentitled user sends the request, or if the requested BU or user does not exist, the transaction is rejected.

Only one BU or user may be specified per request.

- To block (all users of) a BU, one instance of the *PartyDetailGrp* group is specified, with *PartyDetailRole* set to 59 (executing unit), and *PartyDetailID* set to the BU to be blocked ("ABC01E").
- To block a single user, two instances of the *PartyDetailGrp* group are specified. One instance identifies the BU, as described above, and the second instance identifies the specific user, with *PartyDetailRole* set to 55 (session ID), and *PartyDetailID* set to the user (session) to be blocked ("1").

If a user or BU is blocked, the following messages submitted by that user or BU would be rejected:

- **New Order Single**
- **New Order Multileg**
- **New Order Cross**
- **Order Cancel Replace Request**
- **Multileg Order Cancel Replace Request**
- **Delete All Orders Request**
- **Add Complex Instrument Request**
- **Member Kill Switch Request**

Except for those messages indicated above, a blocked user may still send all other messages.

5.18.1 Message Format

The format of the **Member Kill Switch Request** is shown below:

Table 59: Member Kill Switch Request (MsgID=10021)

Tag	Field	Req	Value(s)	Comment
10021		Y		Member Kill Switch Request
<Request Header>		Y	MsgType=DA	Party Entitlements Definition Request
1770	EntitlementRequestID	Y		
11041	PartyEntitlementUpdateGrp	Y	1	
1324	> ListUpdateAction	Y	D=Delete	
14040	> PartyDetailGrp	Y	1-2	
1691	>> PartyDetailID	Y		BU or user to be blocked, e.g. "ABC01E" or "1"
1693	>> PartyDetailRole	Y	59=Executing Unit 55=Session ID	

5.18.2 Response Message

The **Member Kill Switch Request Response** is a positive acknowledgement that the requested entitlement(s) took place. The message uses the FIX Party Entitlements Definition Request Ack message. The format is shown below:

Table 60: Member Kill Switch Response (MsgID=10022)

Tag	Field	Req	Value(s)	Comment
10022		Y		Member Kill Switch Response
<Response Header>		Y	MsgType=DB	PartyEntitlementsDefinitionRequestAck
1770	EntitlementRequestID	Y		
1882	EntitlementRequestStatus	Y	0=Accept	
11042	PartyEntitlementAckGrp	Y	1	
1324	> ListUpdateAction	Y	D=Delete	
1883	> EntitlementStatus		0=Accepted	
14040	> PartyDetailGrp	Y	1	
1691	>> PartyDetailID	Y		BU or user to be blocked, e.g. "ABC01E" or "1"
1693	>> PartyDetailRole	Y	59=Executing Unit 55=Session ID	

5.19 Private Order Broadcast

A **Private Order Broadcast** message is sent for all order-related activity. This message uses the FIX Execution Report message type.

The **POB** is a recoverable broadcast. The scope of the subscription can be for the entire BU or just for the subscribing user (session). Please refer to **Section 3.10, Broadcast Messages**, on page 31, for a complete discussion on the various aspects of broadcast subscriptions.

The *AppLastSeqNum* field is only present for a session-scope subscription. The field reports the sequence number of the last **POB** message that was delivered to the session – that is, the sequence number of the previous message the user received. The client application can use this value to detect whether a sequence number gap for this particular session has occurred.

For example, in this sample scenario, company ABC01E has three users (sessions) subscribed to the order status broadcasts:

- Two sessions subscribe to session-level broadcasts and receive only their respective broadcast traffic.
- One session subscribes at the BU-level and receives all broadcast traffic, including the traffic the other two sessions receive. BU-level **POBs** do **not** include the *AppLastSeqNum* field.

The following table shows how the *AppSeqNum* and *AppLastSeqNum* fields can be used to identify a sequence number gap on session-level subscriptions.

Table 61: Sequence Number Tracking Example

Message	Sequence Number Field	User 1 (Session)	User 2 (Session)	User 3 (BU Level)
Retransmitted	AppSeqNum	5699583599368470529	5699583599368470529	5699583599368470529

Message	Sequence Number Field	User 1 (Session)	User 2 (Session)	User 3 (BU Level)
Jump Msg (Start of Day)	RefApplLastSeqNum	0	0	0
POB #1	ApplSeqNum	...0530		...0530
	ApplLastSeqNum	0		Field Not Present
POB #2	ApplSeqNum		...0531	...0531
	ApplLastSeqNum		0	—
POB #3	ApplSeqNum		...0532	...0532
	ApplLastSeqNum		...0531	—
POB #4	ApplSeqNum	...0533		...0533
	ApplLastSeqNum	...0530		—

NOTE: In the table above, at system start-up, the start-of-day **Application Sequence Number Reset** message establishes the starting *ApplSeqNum* value at 5,699,583,599,368,470,529. This message can only be seen as a result of a retransmission request.

As **POBs** are broadcast, a session-level subscriber can determine whether there has been a sequence number gap by comparing *this POB's ApplLastSeqNum* field with the *last POB's ApplSeqNum* field. If a sequence number gap was detected, the client application could send a **Retransmission Request Message** to request the missing messages. However, note that recovery of order status messages is only at the BU level — the *ApplLastSeqNum* field is *not* present on retransmitted messages. Session-level subscribers must use other fields on the retransmitted **POBs** to filter relevant messages — for example, if *PartyRole* is set to 55 (session ID), the corresponding *PartyID* field identifies the session to which this **POB** belongs.

5.19.1 Private Order Broadcast Message Format

The format of the **Private Order Broadcast** message is shown below:

Table 62: Private Order Broadcast (MsgID=10104)

Tag	Field	Req	Value(s)	Comment
10104		Y		Private Order Broadcast
<Broadcast Header>			MsgType=8	Execution Report
1180	ApplID	Y	O=Order status	
1181	ApplSeqNum	Y		
1350	ApplLastSeqNum	N		Only present in user-level broadcast
5948	PartitionID	Y		
37	OrderID	Y		Exchange's ID
11	ClOrdID	N		Client's ID (if provided on order)
41	OrigClOrdID	N		

Tag	Field	Req	Value(s)	Comment
14012	Parties	Y	2-7	Number of Parties
448	> PartyID	Y		See Appendix N: Away Market Codes , on page 217, for allowed PartyID values when PartyRole = 94
452	> PartyRole	Y	4=Clearing Firm (CMTA) 12=Executing Trader 24=Customer Account 45=Secondary Account Number 55=Session ID 59=Executing Unit (BU) 83=Clearing Account (Give-Up) 94=Away Market	Value 24 is used in lieu of FIX tag 1 (Account).
14063	TargetParties	N	0-1	
1462	> TargetPartyID	Y		
8552	> TargetPartyRoleQualifier	Y	5=Preferred Market Maker 6=Directed Market Maker	
39	OrdStatus	Y	0=New 1=Partially Filled 2=Filled 4=Canceled 6=Pending Cancel 9=Suspended	Current "state" of the order
17	ExecID	Y		
1300	MarketSegmentID	Y		Product ID
48	SecurityID	Y		Instrument ID
1227	ProductComplex	Y	1=Simple Instrument 2=Standard Combination 3=Stock Combination	
54	Side	Y	1=Buy 2=Sell	
38	OrderQty	Y		
40	OrdType	Y	1=Market 2=Limit 3=Stop 4=Stop Limit Q=Counter-order selection	
44	Price	N		
1092	PriceProtectionScope	N	1=Local Market 2=National	
99	StopPx	N		
8572	RelatedLowPrice	N		
8573	RelatedHighPrice	N		

Tag	Field	Req	Value(s)	Comment
8574	RelatedPriceSource	N	1=Underlying NBB 2=Underlying NBO	
18	ExecInst	N	1=Not Held 6=Participate don't initiate G=All-or-None (AON) H=Reinstate on System Failure (Persist) f=Intermarket Sweep (ISO) h=External Routing Not Allowed (DNR) z=Cancel remaining	
8601	AuctionInst	N	0=Start auction 1=Do not start auction	
913	AgreementDesc	N		Free text up to 40 chars.
59	TimeInForce	Y	0=Day 1=Good Till Canceled 2=At the Opening 3=Immediate or Cancel 4=Fill Or Kill 6=Good Till Date	
432	ExpireDate	N	YYYYMMDD	
528	OrderCapacity	Y	C=Customer D=Customer Professional B=Broker/Dealer G=Proprietary N=Away Market Maker M=Market Maker	
8521	ClearingCapacity	N	1=Customer 2=Firm 3=Market Maker	
151	LeavesQty	Y		
14	CumQty	Y		
60	TransactTime	Y		ME timestamp
14026	OrdEventsGrp	Y	> 0	
8535	> OrdEventType	Y	1=Added 2=Modified 3=Deleted 4=Partially_filled 5=Filled 6=Locked 7=Released 8=Restated	
8536	> OrdEventReason	Y		
8554	> OrdEventPx	N		

Tag	Field	Req	Value(s)	Comment
9000	> OrdEventLastMkt	N	A = NYSE AMEX B = BOX C = CBOE H = ISE Gemini J = ISE Mercury I = ISE M = Miami N = NYSE ARCA Q = NASDAQ T = NASDAQ BX W = C2 X = NASDAQ PHLX Z = BATS	
8537	> OrdEventQty	N		
8620	> OrdEventMatchType	N	4=Auto-match Incoming Order 5=Cross 6=Counter-order Selection 9=Opening Auction 10=Auction 11=Auto-match Resting Order 12=Auto-match Pinging Attempt	
8621	> OrdEventMatchSubType	N		See Appendix G: DTI Field Descriptions (Alphabetical) , on page 170, for values.
14037	> OrdEventLegGrp	N		
14036	>> OrdEventLegExecGrp	Y		
8622	>>> OrdEventLegQty	Y		
8623	>>> OrdEventLegLastPx	Y		
8624	>>> OrdEventLegLastMkt	N	1=BNY ConvergeEx U.S. Transaction Services 3=BNY ConvergeEx Millennium ATS 4=Knight Match 5=Knight Link 6=Instinet CBX (US) 7=Deutsche Bank ATS 8=Cheevers 9=Libucki 10=FOG Equities 11=Smart Routing (KCG)	ID of the stock leg execution venue.
14046	>>>OrdEventDealItemGrp	N		
8701	>>>>DealItemID	Y		
8702	>>>>DealItemQty	Y		
14045	>>>>OrdEventTradeItemGrp	Y		
1003	>>>>>TradeID	Y		
1506	>>>>>SideTradeID	Y		

Tag	Field	Req	Value(s)	Comment
1843	>>>>>TradeQty	Y		
1444	>>>>>SideLiquidityInd	Y		
8704	>>>>>CounterpartyOrderCapacity	Y	C=Customer D=Customer Professional B=Broker/Dealer G=Proprietary N=Away Market Maker M=Market Maker	
8705	>>>>>CounterpartyClearingCapacity	N	1=Customer 2=Firm 3=Market Maker	
14009	>>>>>NestedParties2	Y		
757	>>>>>Nested2PartyID	Y		
759	>>>>>Nested2PartyRole	Y	59=Executing Unit	
14046	>OrdEventDealItemGrp	O		
8701	>>DealItemID	Y		
8702	>>DealItemQty	Y		
14045	>>OrdEventTradeItemGrp	Y		
1003	>>>TradeID	Y		
1506	>>>SideTradeID	Y		
1843	>>>TradeQty	Y		
1444	>>>SideLiquidityInd	Y		
8704	>>>CounterpartyOrderCapacity	Y	C=Customer D=Customer Professional B=Broker/Dealer G=Proprietary N=Away Market Maker M=Market Maker	
8705	>>>CounterpartyClearingCapacity	N	1=Customer 2=Firm 3=Market Maker	
14009	>>>NestedParties2	Y		
757	>>>>>Nested2PartyID	Y		
759	>>>>>Nested2PartyRole	Y	59=Executing Unit	
110	MinQty	N		
8602	MinQtyType	N	O=Once R=Repeat	
1608	InitialDisplayQty	N		
1138	DisplayQty	Y		
1083	DisplayWhen	N	1=Immediate 2=Exhaust	
1084	DisplayMethod	N	1=Initial 3=Random	

Tag	Field	Req	Value(s)	Comment
1085	DisplayLowQty	N		
1086	DisplayHighQty	N		
77	PositionEffect	N	O=Open C=Close	
15018	InstrmtLegExecGrp	N	0 or > 1	
624	> LegSide	N	5=Short Sell 6=Sell short exempt	
609	> LegSecurityType	Y	OPT=Option CS=Common Stock	
8575	> LegClearingCapacity	N	1=Customer 2=Firm 3=Market Maker	
564	> LegPositionEffect	N	O=Open C=Close	
14010	> NestedParties3	N	0 or more	
949	>> Nested3PartyID	Y		
951	>> Nested3PartyRole	Y	4=Clearing Firm (CMTA) 45=Secondary Account Number 83=Clearing Account (Give-Up)	
14020	TrdRegTimestamps	Y	3	Three Timestamps
769	> TrdRegTimestamp	Y		
770	> TrdRegTimestampType	Y	1=Execution Time 8=Time Priority 9=Order Creation	
1080	RefOrderID	N		
1390	TradePublishIndicator	N	0=Do Not Publish Trade 1=Publish Trade	
549	CrossType	N	See Table 27: Cross Order Types , on page 59	
8522	AuctionType	N	See Table 30: Auction Types , on page 67	Required for auction responses.
8523	AuctionAllocationPct	N	$0 \leq n \leq 0.4$ (default = 40%)	30% (0.3) is entered as 30,000,000
14018	NoDisclosureInstGrp	N	0-7	
8577	> DisclosureType	N	1=Volume 2=Price 3=Side 4=AON 5=General 6=Clearing Account 7=CMTA Account	Field to disclose
8578	> DisclosureInst	N	0=No 1=Yes	If disclosed or not
8529	TriggerIndicator	N	0=Not Triggered 1=Triggered	

Tag	Field	Req	Value(s)	Comment
8643	MinQtyExecIndicator	Y	0=No Exec 1=Exec	
8531	LockType	N	1=Away Market Better 2=Three Tick Locked 3=Locked by Market Maker 4=Directed Order Lock 5=Multileg Lock 6=Market Order Lock	
8533	LockedQty	N		
9861	BranchSeqNbr	N		
58	Text	N		
5476	PrivateReference	N		
9303	RoutingInst	N	1=Sweep	

5.20 Application Sequence Number Reset Broadcast

The **Application Sequence Number Reset Broadcast** (or, the **Jump** message) is provided to report a sequence number gap. This message uses the FIX Application Message Report message type.

This message is generated and sent on both the order status and MM information broadcasts at system start-up to establish the starting sequence number for those two broadcast types. This message may also be sent during the day, in the event of an ME restart.

Note: A sequence number reset indicates a sequence number gap. *It does not* necessarily indicate a message gap.

When this message is sent at system start-up, the *RefApplLastSeqNum* field is set to zero (0). If this message is sent after an ME restart, *RefApplLastSeqNum* indicates the last *persisted* message for the indicated broadcast type/partition.

- If the **Jump** message's *RefApplLastSeqNum* field value is less than the last status message's *ApplSeqNum* field value, then the last message **was not persisted**. The last message that the ME is aware of is the message whose *ApplSeqNum* matches that of the **Jump** message's *RefApplLastSeqNum* field. In this case, the BU must review the message(s) that the trading system has no record of to determine a course of action (such as re-submitting an order).
- If the **Jump** message's *RefApplLastSeqNum* field value is greater than the last status message's *ApplSeqNum* field value, the client application should recognize that a message gap *might* have occurred. In this case, the application can send a **Retransmit Req** message to determine whether any messages are missing.

5.20.1 Message Format

The message format of the **Application Sequence Number Reset (Jump)** message is shown below.

Table 63: Application Sequence Number Reset (MsgID=10020)

Tag	Field	Req	Value(s)	Comment
10020		Y		Application Sequence Number Reset
<Broadcast Header>			MsgType=BY	ApplicationMessageReport
1426	ApplReportType	Y	2=Heartbeat message	
15117	ApplIDReportGrp	Y	1	Only one entry sent per report.
1181	>ApplSeqNum	Y		Sequence number of this message
1355	>RefApplID	Y	0=Order status M=MM information	Broadcast type
5948	>PartitionID	Y		
1357	>RefApplLastSeqNum	Y		Last persisted sequence number for given partition and broadcast type

5.21 Private Quote Broadcast

A **Private Quote Broadcast** is sent for *unsolicited* quote-related activity. This broadcast uses the ISE-defined Quote Execution Report message type.

An MM who submits a new, or cancels/modifies an existing quote, or actively deletes or inactivates all quotes, does *not* receive a **PQB** for that activity. If a quote trades or is canceled due to external reasons — including canceled because a new quote is rejected — the MM receives a **PQB** message.

In the case of an unsolicited delete (by Market Operations) or inactivation of all quotes (curtailment), a single *QuoteEventsGrp* group conveys the deletion or inactivation, as specified by the *QuoteEventType* field. The *SecurityID* field within the group is not present, and indicates that the event does not represent a single instrument's quote, but rather all quotes for the affected product.

PQBs are not recoverable. The scope of the quote status subscription can be for the entire BU or just for the subscribing user (session). Please refer to **Section 3.10, Broadcast Messages**, on page 31, for a complete discussion on the various aspects of broadcast subscriptions.

NOTE: A BU-scope event, such as curtailment, is sent to all sessions, regardless of the subscription scope.

5.21.1 Private Quote Broadcast Message Format

The format of the **Private Quote Broadcast** message is shown below:

Table 64: Private Quote Broadcast (MsgID=10200)

Tag	Field	Req	Value(s)	Comment
10200		Y		Private Quote Broadcast
<Broadcast Header>			MsgType=U8	QuoteExecutionReport
1180	ApplID	Y	Q=Quote status	
1300	MarketSegmentID	Y		Product ID
60	TransactTime	Y		ME timestamp
17	ExecID	Y		
14011	QuoteEventsGrp	Y	> 0	
48	> SecurityID	N		If not present, the QuoteEventsGrp is reporting an event that affects all quotes in a product. E.g., quote inactivation due to curtailment.
1227	> ProductComplex	Y	1=Simple Instrument 2=Standard Combination 3=Stock Combination	
8539	> QuoteEventType	Y	1=Added_quote_side 2=Removed_quote_side 3=Partially_filled 4=Filled 5=Modify_quote_side 6=Suspended	
8540	> QuoteEventReason	Y		
14012	> Parties	N	0-1	
448	>> PartyID	Y		
452	>> PartyRole	Y	4=Clearing Firm (CMTA) 12=Executing Trader 24=Customer Account 45=Secondary Account Number 55=Session ID 59=Executing Unit (BU) 83=Clearing Account (Give-Up)	Value 24 is used in lieu of FIX tag 1 (Account).
1166	> QuoteMsgID	N		MM's QuoteID
8581	> QuoteEventSide	N	1=Buy 2=Sell	
8553	> QuoteEventPx	N		
8542	> QuoteEventQty	N		
574	> MatchType	N	4=Auto-match Incoming Quote 9=Opening Auction 10=Auction 11=Auto-match Resting Quote	

Tag	Field	Req	Value(s)	Comment
8610	> MatchSubType	N		See Appendix G: DTI Field Descriptions (Alphabetical) , on page 170, for values.
14039	> QuoteEventLegGrp	N		
14038	>> QuoteEventLegExecGrp	Y		
8630	>>> QuoteEventLegQty	Y		
8631	>>> QuoteEventLegLastPx	Y		
8632	>>> QuoteEventLegLastMkt	N	1=BNY ConvergeEx U.S. Transaction Services 3=BNY ConvergeEx Millennium ATS 4=Knight Match 5=Knight Link 6=Instinet CBX (US) 7=Deutsche Bank ATS 8=Cheevers 9=Libucki 10=FOG Equities 11=Smart Routing (KCG)	ID of the stock leg execution venue.
14044	>>>>QuoteEventDealItemGrp	N		
8701	>>>>>DealItemID	Y		
8702	>>>>>DealItemQty	Y		
14043	>>>>>QuoteEventTradeitemGrp	Y		
1003	>>>>>>TradeID	Y		
1506	>>>>>>SideTradeID	Y		
1843	>>>>>>TradeQty	Y		
1444	>>>>>>SideLiquidityInd	Y		
8704	>>>>>>CounterpartyOrderCapacity	Y	C=Customer D=Customer Professional B=Broker/Dealer G=Proprietary N=Away Market Maker M=Market Maker	
8705	>>>>>>CounterpartyClearingCapacity	N	1=Customer 2=Firm 3=Market Maker	
756	>>>>>>>NestedParties2	Y		
757	>>>>>>>>Nested2PartyID	Y		
759	>>>>>>>>Nested2PartyRole	Y	59=Executing Unit	
14044	>QuoteEventDealItemGrp	N		
8701	>>DealItemID	Y		
8702	>>DealItemQty	Y		
14043	>>QuoteEventTradeitemGrp	Y		

Tag	Field	Req	Value(s)	Comment
1003	>>>TradeID	Y		
1506	>>>SideTradeID	Y		
1843	>>>TradeQty	Y		
1444	>>>SideLiquidityInd	Y		
8704	>>>CounterpartyOrderCapacity	Y	C=Customer D=Customer Professional B=Broker/Dealer G=Propriety N=Away Market Maker M=Market Maker	
8705	>>>CounterpartyClearingCapacity	N	1=Customer 2=Firm 3=Market Maker	
756	>>>NestedParties2	Y		
757	>>>>Nested2PartyID	Y		
759	>>>>Nested2PartyRole	Y	59=Executing Unit	
14	> CumQty	N		
8615	> PctCount	N		
7966	> Delta	N		
7968	> Vega	N		

5.22 Product Reset Broadcast

The **Product Reset Broadcast** is sent *after* an ME fail-over or partition restart and is used to inform members that all non-persistent orders and quotes for the indicated product have been removed. This message uses the FIX Security Mass Status message type.

This message is sent as part of the order status, quote status, and MM information broadcast subscriptions.

5.22.1 Message Format

The format of the **Product Reset Broadcast** is shown below:

Table 65: Product Reset Broadcast (MsgID=10406)

Tag	Field	Req	Value(s)	Comment
10406		Y		Product Reset Broadcast
<Broadcast Header>		Y	MsgType=CO	SecurityMassStatus
1180	ApplID	Y	O=Order status Q=Quote status M=MM information	
1181	ApplSeqNum	N		Applicable for recoverable subscriptions, only

Tag	Field	Req	Value(s)	Comment
5948	PartitionID	N		
1300	MarketSegmentID	Y		
1679	SecurityMassTradingStatus	Y	3=Resume (after reset)	
60	TransactTime	Y		ME timestamp

NOTE: The **Product Reset Broadcast** is **not** sent if the broadcast subscription scope is for session messages only.

5.23 Deal Item Broadcast

A **Deal Item Broadcast** advises a BU of a deal. The deal is sent using the FIX Trade Capture Report message.

An order or quote that matches against one or more opposing orders or quotes generates a **deal** for each price level executed. For example, an order to buy 100 at market price executes in full, but is the result of two deals: 80 at price level 1, and 20 at price level 2. Each deal results in a **DIB**.

NOTE: Depending on how the match occurs, *more than one deal can occur* at a single price level. For example, the match above could be the result of three deals: 60 at price level 1 *and* 20 at price level 1, and 20 at price level 2.

- The **DIB** is available only as a recoverable broadcast. It is available to the entering BU, the clearing BU, the CMTA BU (which may differ from the clearing BU), and configured “drop-copy” BUs.
- Trade corrections — Busts and/or Adjustments — are made against the deals.
- Deals are broken down into matched trade pairs to send to OCC. Each matched pair results in a trade broadcast (see **Section 5.24, Trade Item Broadcast**, on page 138). A copy of the deal broadcast is also provided to the clearing BU and the CMTA BU. Drop copies can be configured for other BUs. A deal can be busted or adjusted. *TradeReportTransType* and *TradeReportType* indicate how this deal record was created.
- If the deal is the result of a manual trade entry (*MessageEventSource* = 2), the following fields are not present on the **DIB**:
 - *ExecID*
 - *SideLastPx*
 - *SideLastQty*
 - *OrderID*
 - *ClOrdID*
 - *OrderCategory*
 - *RelatedInstrumentGrp* (group)
 - *OrderSide*
 - *CumQty*
 - *LeavesQty*
 - *Price*

5.23.1 Message Format

The format of the **Deal Item Broadcast** is shown below:

Table 66: Deal Item Broadcast (MsgID=10400)

Tag	Field	Req	Value(s)	Comment
10400		Y		Deal Item Broadcast
<Broadcast Header>		Y	MsgType=AE	TradeCaptureReport
1180	ApplID	Y	D=Deal Item	
1181	ApplSeqNum	Y		
5948	PartitionID	Y		
571	TradeReportID	Y		Deal Item ID
487	TradeReportTransType	Y	0=New 2=Replace 4=Reverse	
1011	MessageEventSource	Y	1=Matching Engine 2=Manual Trade Entry 3=Deal Modification (user) 4=Deal Modification (exchange) 5=Trade Bust 6=Deal Bust 7=Deal Management (exchange)	Source of Deal
8583	TradeReportText	N		Explain why Deal is updated
572	TradeReportRefID	N		Previous Deal ID in Adjustment
1125	OrigTradeDate	N		For TradeReportRefID
830	TransferReason	Y	1=Owner 2=Clearer 3=CMTA 5=Deal Item Drop Copy	
17	ExecID	N		
14031	RootParties	N	1-2	Entity that initiated Deal
1117	> RootPartyID	Y		
1119	> RootPartyRole	Y	55=Session ID (user) 59=Executing Unit (BU)	
1300	MarketSegmentID	Y		Product ID
48	SecurityID	Y		Instrument ID
1227	ProductComplex	Y	1=Simple Instrument 4=Equity	Equity represents the stock leg of a stock combination
32	LastQty	Y		Deal Quantity
31	LastPx	Y		Deal Price

Tag	Field	Req	Value(s)	Comment
30	LastMkt	N	1=BNY ConvergeEx U.S. Transaction Services 3=BNY ConvergeEx Millennium ATS 4=Knight Match 5=Knight Link 6=Instinet CBX (US) 7=Deutsche Bank ATS 8=Cheevers 9=Libucki 10=FOG Equities 11=Smart Routing (KCG)	ID of the stock leg execution venue.
8582	MatchDate	Y		Of Deal, or of Original Deal
60	TransactTime	Y		TM timestamp
5507	TrdMatchTime	Y		ME timestamp
75	TradeDate	Y		Bus. date of this Deal
574	MatchType	Y	3=Confirmed trade report 4=Auto-match incoming order 5=Cross 6=Counter-order selection 9=Opening auction 10=Auction 11=Auto-match resting order 12=Auto-match pinging attempt	Reason Code
8610	MatchSubType	Y		See Appendix G: DTI Field Descriptions (Alphabetical) , on page 170, for values.
14019	TrdCptRptSideGrp	Y	1	Always 1
54	> Side	Y	1=Buy 2=Sell 5=Sell Short 6=Sell Short Exempt	Deal Side
8585	> SideLastPx	N		Complex Order
1009	> SideLastQty	N		Complex Order
8521	> ClearingCapacity	N	1=Customer 2=Firm 3=Market Maker	
14012	> Parties	N	0-5	
448	>> PartyID	Y		See Appendix N: Away Market Codes , on page 217, for allowed PartyID values when PartyRole = 94

Tag	Field	Req	Value(s)	Comment
452	>> PartyRole	Y	4=Clearing Firm (CMTA) 12=Executing Trader 24=Customer Account 45=Secondary Customer Account 83=Clearing Account (Give-Up) 94=Away Market	Value 24 is used in lieu of FIX tag 1 (Account).
77	> PositionEffect	N	O=Open C=Close	
37	> OrderID	N		
11	> ClOrdID	N		
1115	> OrderCategory	N	1=Order 2=Quote 5=Linked Order 8=Cross Order A=Auction D=Directed Order	
40	> OrdType	N	1=Market 2=Limit	Market or limit
14021	> RelatedInstrumentGrp	N	0 – 1	
1650	>> RelatedSecurityID	Y		ID of Multileg Instrument
8587	>> RelatedProductComplex	Y	1=Simple Instrument 2=Standard Combination 3=Stock Combination	
8586	> OrderSide	N	1=Buy 2=Sell	Order's side
14	> CumQty	N		Cum Qty traded of this order
151	> LeavesQty	N		Leaves Qty after this trade
44	> Price	N		Order's Limit Price
528	> OrderCapacity	N	C=Customer D=Customer Professional B=Broker/Dealer G=Proprietary N=Away Market Maker M=Market Maker	
9861	> BranchSeqNbr	N		Alternate Free text
58	> Text	N		Free Text
5476	> PrivateReference	N		Alternate Free Text

5.24 Trade Item Broadcast

A **Trade Item Broadcast** advises a BU of a trade. The trade is sent using the FIX Trade Capture Report message.

- A deal, as described above, is made up of one or more **trades**. For example, a deal for 80 at price level 1, is the result of two trades: 30 against an MM quote, and 50 against a resting order. Each trade results in a **TIB**.
- The **TIB** is available only as a recoverable broadcast. It is available to the entering BU, the clearing BU, the CMTA BU (which may differ from the clearing BU), and configured “drop-copy” BUs.
- The trade broadcast carries the same trade information that is sent to OCC.
- If the trade is the result of a manual trade entry (*MessageEventSource* = 2), the following fields are not present on the **TIB**:
 - *ExecID*
 - *OrderID*
 - *ClOrdID*
 - *SideLiquidityInd*
 - *RelatedInstrumentGrp* (group)

5.24.1 Message Format

The format of the **TIB** is shown below:

Table 67: Trade Item Broadcast (MsgID=10401)

Tag	Field	Req	Value(s)	Comment
10401		Y		Trade Item Broadcast
	<Broadcast Header>	Y	MsgType=AE	TradeCaptureReport
1180	ApplID	Y	T=Trade Item	
1181	ApplSeqNum	Y		
5948	PartitionID	Y		
571	TradeReportID	Y		
820	TradeLinkID	Y		Deal TradeReportID
487	TradeReportTransType	Y	0=New 2=Replace 4=Reverse	
1011	MessageEventSource	Y	1=Matching Engine 2=Manual Trade Entry 3=Deal Modification (user) 4=Deal Modification (exchange) 5=Trade Bust 6=Deal Bust 7=Deal Management (exchange)	
8583	TradeReportText	N		Free text to explain adjustment
572	TradeReportRefID	N		Trade-ID being Adjusted
17	ExecID	N		TX-ID that created trade
1125	OrigTradeDate	N		

Tag	Field	Req	Value(s)	Comment
830	TransferReason	Y	1=Owner 2=Clearer 3=CMTA 5=Trade Item Drop Copy	
14031	RootParties	N	1-2	Entity that initiated trade
1117	> RootPartyID	Y		
1119	> RootPartyRole	Y	59=Executing Unit 55=Session ID	
1300	MarketSegmentID	Y		Product ID
48	SecurityID	Y		Instrument ID
1227	ProductComplex	Y	1=Simple Instrument 4=Equity	Equity represents the stock leg of a stock combination instrument
32	LastQty	Y		Trade quantity
31	LastPx	Y		Trade price
30	LastMkt	N	1=BNY ConvergeEx U.S. Transaction Services 3=BNY ConvergeEx Millennium ATS 4=Knight Match 5=Knight Link 6=Instinet CBX (US) 7=Deutsche Bank ATS 8=Cheevers 9=Libucki 10=FOG Equities 11=Smart Routing (KCG)	ID of the stock leg execution venue.
8582	MatchDate	Y		
5507	TrdMatchTime	Y		ME timestamp
60	TransactTime	Y		TM timestamp
75	TradeDate	Y		
574	MatchType	Y	3=Confirmed trade report 4=Auto-match incoming order 5=Cross 6=Counter-order selection 9=Opening auction 10=Auction 11=Auto-match resting order 12=Auto-match pinging attempt	
8610	MatchSubType	Y		See Appendix G: DTI Field Descriptions (Alphabetical) , on page 170, for values.

Tag	Field	Req	Value(s)	Comment
14019	TrdCptRptSideGrp	Y	1 or 2 (if counter-party disclosed)	1 st instance is always the member's side of the trade (Buy or Sell). 2 nd instance is the counter-party side (if disclosed).
54	> Side	Y	1=Buy 2=Sell 5=Sell Short 6=Sell Short Exempt	
1506	> SideTradeID	Y		Carried to OCC TrdID field
14012	> Parties	N	0 – 5	
448	>> PartyID	Y		See Appendix N: Away Market Codes , on page 217, for allowed PartyID values when PartyRole = 94
452	>> PartyRole	Y	4=Clearing Firm (CMTA) 12=Executing Trader 24=Customer Account 45=Secondary Customer Account 59=Executing Unit (counter-party side, only) 83=Clearing Account (Give-Up) 94=Away Market	Value 24 is used in lieu of FIX tag 1 (Account).
77	> PositionEffect	N	O=Open C=Close	Member's side, only.
1444	> SideLiquidityInd	N	See Table 24: Liquidity Indicator Values , on page 49	Member's side, only.
37	> OrderID	N		Member's side, only.
11	> ClOrdID	N		Member's side, only.
14021	> RelatedInstrumentGrp	N	0 – 1	Member's side, only.
1650	>> RelatedSecurityID	Y		
8587	>> RelatedProductComplex	Y	1=Simple Instrument 2=Standard Combination 3=Stock Combination	
528	> OrderCapacity	N	C=Customer D=Customer Professional B=Broker/Dealer G=Proprietary N=Away Market Maker M=Market Maker	
5476	> PrivateReference	N		Member's side, only.
9861	> BranchSeqNbr	N		Member's side, only.
58	> Text	N		Member's side, only.

5.25 Locked Order Broadcast

A **Private MM Execution Report Broadcast** is sent to inform an MM of a locked order, or of an order released from a lock. The broadcast is sent using the FIX Execution Report message type.

The **Private MM Execution Report Broadcast** is a recoverable broadcast, and is available through the MM information broadcast subscription. There is one message for each order that is locked or released.

5.25.1 Message Format

The format of the **Private MM Execution Report Broadcast** message is shown below:

Table 68: Private MM Execution Report Broadcast (MsgID=10305)

Tag	Field	Req	Value(s)	Comment
10305		Y		Private MM Execution Report Broadcast
<Request Header>		Y	MsgType=8	ExecutionReport
1180	ApplID	Y	M=MM information	
17	ExecID	Y		Reference for the transaction
60	TransactTime	Y		ME timestamp
1181	ApplSeqNum	Y		
5948	PartitionID	Y		
37	OrderID	Y		Order being locked/released
14012	Parties	N	0-3	Number of Parties
448	> PartyID	Y		Name of the Party
452	> PartyRole	Y	4=Clearing Firm 59=Executing Unit 83=Clearing Acct.	
1300	MarketSegmentID	Y		Product ID
48	SecurityID	N		
528	OrderCapacity	Y	C=Customer D=Customer Professional	
54	Side	Y	1=Buy 2=Sell	
8533	LockedQty	Y		Total locked qty
8544	SecondaryLockedQty	N		Qty THIS lock
8543	ReleaseQty	N		
40	OrdType	Y	1=Market 2=Limit	
44	Price	N		Order limit price
845	DiscretionPrice	N		Release limit price
18	ExecInst	N	f=Intermarket sweep (ISO)	Mutually exclusive of PriceProtectionScope

Tag	Field	Req	Value(s)	Comment
1092	PriceProtectionScope	N	1=Local Market	Mutually exclusive of ExecInst
150	ExecType	Y	6=Pending Cancel 9=Suspended (Locked) L=Activated (Released)	
378	ExecRestatementReason	Y		Lock/Release reason (see Appendix J: Event Reason Codes , on page 205)
9303	RoutingInst	N	1=Sweep	
8521	ClearingCapacity	N	1=Customer 2=Firm 3=Market Maker	

5.26 Business End Of Stream Broadcast

The **Business End Of Stream Broadcast** is sent to inform that a post-trade facility (for example, OCC) is closing for the day. This message uses the FIX Trade Session Status message type.

The **Business End Of Stream Broadcast** is a recoverable broadcast and is sent as part of the Deal and Trade broadcast subscriptions.

5.26.1 Message Format

The format of the **Business End Of Stream Broadcast** is shown below:

Table 69: Business End of Stream Broadcast (MsgID=10404)

Tag	Field	Req	Value(s)	Comment
10404		Y		Business End of Stream
<Broadcast Header>		Y	MsgType=h	TradingSessionStatus
1180	ApplID	Y	D=Deal Item T=Trade Item	
1181	ApplSeqNum	Y		
5948	PartitionID	Y		
5816	ClearingVenue	Y		
336	TradingSessionID	Y	1=Day	
75	TradeDate	Y		
340	TradSesStatus	Y	3=Closed	

5.27 Business End Of Day Broadcast

The **Business End Of Day Broadcast** is sent to inform that the referenced product is closing for the day. This message uses the FIX Trade Session Status message type.

The **Business End Of Day Broadcast** is sent as part of the order status, quote status, and MM information broadcast subscriptions.

5.27.1 Message Format

The format of the **Business End Of Day Broadcast** is shown below:

Table 70: Business End of Day Broadcast (MsgID=10405)

Tag	Field	Req	Value(s)	Comment
10405		Y		Business End of Day
<Broadcast Header>		Y	MsgType=h	TradingSessionStatus
1180	ApplID	Y	O=Order status Q=Quote status M=MM Information.	
1181	ApplSeqNum	N		
5948	PartitionID	N		
1300	MarketSegmentID	Y		
336	TradingSessionID	Y	1=Day	
75	TradeDate	Y		
340	TradeSesStatus	Y	3=Closed	

NOTE: The **Business End of Day Broadcast** is **not** sent if the broadcast subscription scope is for session messages only.

5.28 End Of Stream Broadcast

The **End Of Stream Broadcast** is sent to inform that broadcasts for the referenced partition are ending for the day. This message uses the FIX Application Message Report message type.

The **End Of Stream Broadcast** is a recoverable broadcast and is sent as part of all recoverable broadcast subscriptions.

5.28.1 Message Format

The format of the **End Of Stream Broadcast** is shown below:

Table 71: End of Stream Broadcast (MsgID=10015)

Tag	Field	Req	Value(s)	Comment
10015				End of Stream
<Broadcast Header>		Y	MsgType=BY	ApplicationMessageReport
1426	ApplReportType	Y	1=Last message has been sent	
15117	ApplIDReportGrp	Y	1	Only one entry per report.
1355	> RefApplID	Y	O=Order status M=MM information T=Trade item D=Deal item N=Market Messages (News)	Broadcast type (only recoverable types are relevant)

Tag	Field	Req	Value(s)	Comment
5948	> PartitionID	Y		
1357	> RefApplLastSeqNum	Y		Last sequence number for given partition and broadcast type

5.29 Trading Session Marker Message Broadcast

When the ME reloads orders, either during “Start of Day” processing or after a market reset event due to an ME failure, the reloaded orders are “restated” on the order status broadcast.

In order to indicate exactly when all re-statement and other start-up broadcasts have ended, the **Trading Session Marker Message** is sent after all start-up broadcasts have been sent. A **Trading Session Marker Message** is sent for each product.

The **Trading Session Marker Message** field, *TradSesStatus*, can be set to the following values: OPEN, CLOSED, and RESET.

- *TradSesStatus* = 2 (Open) is sent after all start-up broadcasts have been sent for each product after a normal start up of the Matcher.
- *TradSesStatus* = 7 (Reset) is sent after all start-up broadcasts have been sent after a failover or re-start of the Matcher.
- *TradSesStatus* = 3 (Closed) is sent at the end of day *before* any re-statements or “end-of-day broadcasts” are sent.

At present, OPEN is the only valid *TradSesStatus* value.

5.29.1 Message Format

The format of the Trading Session Marker Message Broadcast is shown below:

Table 72: Trading Session Marker Message Broadcast (MsgID=10407)

Tag	Field	Req	Value(s)	Comment
10407				Trading Session Marker Broadcast
<Broadcast Header>		Y	MsgType=h	TradingSessionStatus
1180	ApplID	Y	O=Order status Q=Quote status M=MM information	
1181	ApplSeqNum	N		
5948	PartitionID	N		
1300	MarketSegmentID	Y		
336	TradingSessionID	Y	1=Day	
75	TradeDate	Y		
340	TradSesStatus	Y	2=Open	

NOTE: The **Trading Session Marker Message Broadcast** is **not** sent if the broadcast subscription scope is for session messages only.

5.30 User Notification Broadcast

The **User Notification** broadcast is used for the Service Availability subscription to notify the client of changes in partition status. Clients subscribe to this broadcast for either ME notifications or TM notifications (clients can also subscribe to both notifications).

If an ME service becomes “Unavailable,” that ME is no longer able to process new transactions. Any order- or quote-related request messages sent to the affected partition are rejected.

“Unavailable” is not necessarily an indication of an actual ME fail-over or partition restart. The **Product Reset Broadcast** (see **Section 5.22**, on page 134) is the indication of an actual ME fail-over or partition restart.

NOTE: The **Product Reset Broadcast**, not the **User Notification** broadcast, is confirmation that quotes and non-persistent orders have been deleted.

If a TM service becomes “Unavailable,” Deal and Trade broadcasts from the affected partition are no longer available. Deal/Trade retransmission requests to the affected partition are rejected.

The **User Notification** broadcast is also sent prior to a forced disconnect by the exchange to inform the session that it is being disconnected. For example, if two sessions logon with the same credentials, the second session forces out the first session. The first session receives the **User Notification** broadcast with *UserStatus* = 7 (forced logout) and is disconnected. The second session remains active. The receipt of the broadcast message in this case is automatic and does not require a broadcast subscription.

5.30.1 Message Format

The format of the **User Notification** message is shown below:

Table 73: User Notification Broadcast (MsgID=10012)

Tag	Field	Req	Value(s)	Comments
10012				User Notification
<Broadcast header>		Y	MsgType=CB	UserNotification
926	UserStatus	Y	7=Forced user logout by Exchange 9=Service availability	
7930	ServiceID	N	ME=Matching Engine TM=Trade Manager	Present when UserStatus=9
5948	PartitionID	N		Present when UserStatus=9
5969	PartitionStatus	N	0=Unavailable 1=Available	Present when UserStatus=9

5.31 Invalid Data Broadcast

The **Invalid Data Broadcast** indicates that there is a message for the given broadcast type and partition that contained invalid data, or that the DTI could not correctly format for transmission. There is no application level content except a reference to the sequence number of the message with invalid data. This informs the user that he cannot ask for a retransmission of this message and has to resolve the issue out of band.

5.31.1 Message Format

The format of the **Invalid Data** message is shown below:

Table 74: Invalid Data Broadcast (MsgID=10016)

Tag	Field	Req	Value(s)	Comments
10016				Invalid Data Broadcast
	<Broadcast header>		MsgType=BY	ApplicationMessageReport
1426	ApplReportType	Y	4=Last message has invalid data	
15117	ApplIDReportGrp	Y	1	Only one entry sent per report
1355	> RefApplID	Y	O=Order status M=MM information T=Trade item D=Deal item N=Market Messages (News)	Broadcast type (only recoverable types are relevant)
5948	> PartitionID	Y		
1357	> RefApplLastSeqNum	Y		Sequence number of message with invalid data

5.32 News Broadcast

The **News Broadcast** provides general messages from the Market Operations Group. **News** is a recoverable broadcast.

5.32.1 Message Format

The format of the **News** message is shown below:

Table 75: News Broadcast (MsgID=10013)

Tag	Field	Req	Value(s)	Comments
10013				News Broadcast
<Broadcast header>			MsgType=B	News
1180	ApplID	Y	N=Market Message	
1181	ApplSeqNum	Y		
1472	NewsID	Y		Unique reference for News Item
15144	NewsRefGrp	N	0-1	
1476	> NewsRefID	Y		Reference to existing news items
1477	> NewsRefType	Y	0=Replacement 3=Removal	
1301	MarketID	Y		
42	OrigTime	Y		
61	Urgency	Y	0=Normal (Medium) 1=Flash (High) 2=Background (Low)	
148	Headline	Y		News title
15029	LinesOfTextGrp	N		Needed only if message cannot be conveyed in Headline field
58	> Text	Y		News Content

5.33 Heartbeat

The **Heartbeat** message is used between the client application and the DTI as a keep-alive mechanism across an otherwise idle connection.

5.33.1 Message Format

The format of the **Heartbeat** message is shown below:

Table 76: Heartbeat (MsgID=10011)

Tag	Field	Req	Value(s)	Comments
10011				Heartbeat
11006	BroadcastHeader	Y		
35	> MsgType	Y	0=Heartbeat	
49	> SenderCompID	N	"HB"	Populated DTI -> client
52	> SendingTime	N		Populated DTI -> client

List of Appendices

This section provides additional reference information to support the content of this document:

- Appendix A: LTV Encoding (page 150).
- Appendix B: ISE Parameters (page 158)
- Appendix C: Glossary (page 160)
- Appendix D: DTI Conformance Test (page 164)
- Appendix E: DTI Application Messages (page 166)
- Appendix F: DTI Groups (page 168)
- Appendix G: DTI Field Descriptions (Alphabetical) (page 170)
- Appendix H: DTI Field Descriptions (Numerical) (page 187)
- Appendix I: Security Trading Event Codes (page 204)
- Appendix J: Event Reason Codes (page 205)
- Appendix K: Throttles (page 210)
- Appendix L: Locked Order Handling (page 213)
- Appendix M: Standard Strategies (page 215)
- Appendix N: Away Market Codes (page 217)
- Appendix O: Unicast IP Addresses (page 218)

Appendix A: LTV Encoding

Messages sent between the client and the DTI are encoded using LTV Encoding.

LTV encoding is a simple, self-describing and self-delimiting, variable length, binary encoding algorithm designed to make efficient use of the TCP byte stream.

Logical Organization

A DTI message is simply a logical collection— a group, or structure— of data fields. Data fields are the basic unit that carry actual data, and may be organized within sub-groups or repeating sub-groups (arrays) within the larger structure of the message itself.

A data field is encoded within an LTV “container.”

A collection of encoded data field containers (and/or other group containers) may be encoded within a group (structure or array) LTV container, as required.

The collection of all data field and group containers is encoded within the DTI message LTV container.

In other words, the content of an LTV encoded DTI message is simply a sequence of nested LTV containers. Those nested LTV containers may be discrete data fields or groups (structures or arrays), which themselves contain nested LTV containers.

LTV Container Layout

An LTV container consists of three fields:

- **Length:** A variable length unsigned integer field that describes the total length, in bytes, of the container, including the Length field itself.
- **Tag:** A variable length unsigned integer field that identifies the data in the Value field.
- **Value:** A variable length untyped raw data field. The actual content is defined by the Tag field.



Byte Ordering

All integer fields are represented using the little-endian (LE) convention, where low order bytes precede high order bytes.

Length and Tag Field Encoding

An important property of LTV encoding is the use of *continuation-bit* encoding for the Length and Tag fields, which allows for variable-length Tag and Length fields.

The Length (and Tag) field is a sequence of bytes where the most significant (left-most) bit in each byte— the continuation-bit — is used to indicate if the *next* byte is part of the field. If the continuation bit is set (1), another byte follows for the field. If the continuation bit is not set (0), this is the last byte. The seven bits following the continuation-bit are the significant data bits, resulting in 7-bit binary numbers: numbers up to 127 are represented in one 7-bit byte; numbers up to 16,383 are represented in two 7-bit bytes; etc.

For example:

To **decode** the hexadecimal value 0xDE02 (LE) —

Examine byte-by-byte —

0xDE (1101 1110): The continuation-bit (highlighted) is set (1), indicating that the following byte belongs to the field. Strip the continuation-bit and save the 7-bit byte 1011110.

0x02 (0000 0010): The continuation-bit is not set (0), indicating that this is the last byte for the field. Strip the continuation-bit and save the 7-bit byte 0000010.

Swap the two bytes to big-endian (BE — to be more human-readable):

0000010 1011110

Concatenate and convert to 8-bit binary:

0000010 1011110 (two 7-bit bytes) → 00000001 01011110 (two 8-bit bytes)
= 0x015E = 350₁₀

This may be described programmatically as (pseudo-code):

```

value = 0
for ( each byte 0 to n ) do
    if ( byte AND 0x80 )      # if continuation-bit is set
        # turn off continuation-bit
        byte = byte XOR 0x80
    else
        # this is the last encoded byte
        last_byte = TRUE
    fi

    # m'ply byte by 128 raised to the power
    # of the byte number (0 - n)
    temp = byte * (128byte_no)

    # add to value
    value = value + temp

    if (last_byte)
        break out of FOR loop

```



```
fi
done
```

To **encode** the integer value 201 (0xC9 [BE]) —

Convert to 7-bit binary:

11001001 (one 8-bit byte) —> 0000001 1001001 (two 7-bit bytes).

Swap the bytes (LE):

1001001 0000001.

Add continuation-bits:

11001001 00000001 = 0xC901 (LE).

This may be described programmatically as (pseudo-code):

```
byte = 1st_byte of buffer to store encoded data
while ( value > 127 ) do
    # mask value by 127 and store in byte
    byte = (value AND 0x7f)

    # turn on continuation-bit
    byte = byte OR 0x80

    # divide value (integer division) by 128
    value = value DIV 128

    byte = next_byte of buffer
done

# fill last byte
byte = value
```

Special Length/Tag Field Encoding for the DTI Message

The Length field for the DTI Message LTV container is *not* continuation-bit encoded and is always encoded as a fixed-length, 4-byte unsigned integer.

The Tag field for the DTI Message LTV container is *not* continuation-bit encoded and is always encoded as a fixed-length, 2-byte unsigned integer.

Value Field Encoding

The Value field comprises the actual raw data for the LTV container and is *not* continuation-bit encoded. The field may contain the data for a single field, or one or more nested LTV containers—the actual content is defined by the Tag field. For example: the raw value 0x41 may represent the integer 64, or the ASCII character “A”, depending on the Tag field.

When decoding an LTV container, the length of the Value field— the length of the raw data— is determined by subtracting the number of bytes used for the Length and Tag fields from the value of the Length field.

Theoretically, the Value field cannot be empty. That is, the length of the Value field cannot be zero— every valid Tag must have a Value. It is an error to encode “tag =”. However, the DTI *ignores* LTV containers if the tag is unknown or out of context. It is possible to send an LTV container with a 1-byte Length set to two and a 1-byte Tag set to zero (and Value not set). As a Tag value of zero is unknown, the DTI simply ignores those two bytes of data.

For any Length = n , if Tag is “unknown,” the DTI ignores those n bytes of data.

• Data Field Content

If the Tag field identifies a data field, the Value field contains the actual data for the field. The representation of the data is according to the data type of the identified field.

It is necessary, but not required, to send only the minimum number of bytes required— leading zeros (integer values) and trailing spaces or nulls (string values) can be stripped.

For example: the integer value 65,536 requires at least four bytes of computer memory; however, because the extra byte of leading zeros can be stripped for LTV encoding, 65,536 can be represented in only three bytes in the Value field: 0x0000 01 (LE)

The following table shows the four-byte LTV container for an integer field, Tag ID 108, set to 5,000. The last row shows the byte-by-byte hexadecimal encoding:

Table 77: LTV Data Element

HeartBtInt			
L _{ength}	T _{ag}	V _{alue}	
04	6C	88	13

➤ Special Encoding for Negative Numbers

As the Value field can be variable in length, it is necessary to identify, unambiguously, positive and negative integer values in two’s complement, binary number representation.

In order to differentiate between positive and negative numbers, all negative numbers *must* be encoded using **eight bytes of data**. For example, the value 0x0080 (LE) is *always* interpreted as +32,768, never as -32,768. The encoding for -32,768 is 0x0080 FFFF FFFF FFFF (LE).

• Group Content— Structures

For a structure, each discrete element (data field or sub-group) of the structure is LTV encoded and nested within the structure’s Value field. The order of the nested LTV containers is not significant. Unused elements (in context) are not required to be included.

NOTE: Structures are identified and defined in the isefix.xml template file as **Groups** with *cardinality* = “1.”

The following table shows a generic LTV container for a structure containing n data fields:

Table 78: Generic LTV Structure

Structure								
Length	Tag	Value						
		Field ₁			...	Field _n		
		Len	Tag	Val		Len	Tag	Val

The following table shows an example nine-byte LTV container for the *RequestHeader* structure with two, of three defined, fields. The last row shows the byte-by-byte encoding:

Table 79: RequestHeader LTV structure encoding

RequestHeader								
Length	Tag		Value					
			MsgType			MsgSeqNum		
			Len	Tag	Val	Len	Tag	Val
09	F8	55	03	23	41	03	22	01

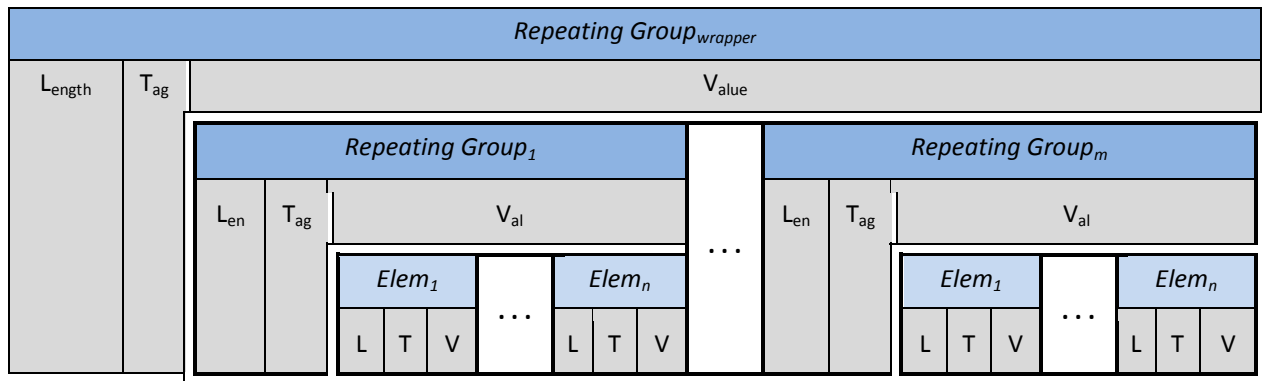
• Repeating Group Content— Arrays

For a repeating group (an array), as with structures, the discrete array elements are encoded and nested within a container for the array instance; and then all instances of the array are nested again within a “wrapper” array container.

NOTE: Arrays are identified and defined in the isefix.xml template file as **Groups** with *cardinality* = “unbounded.”

The following table shows a generic LTV container for an array of n elements, containing m iterations of the array:

Table 80: Generic LTV Array



The following table shows an example 14-byte LTV container for the Session Instruction repeating group, containing just a single iteration of the array. The last row shows the byte-by-byte encoding:

Table 81: SessionInstGrp LTV array encoding

SessionInstGrp _{wrapper}													
L _{length}	T _{ag}		V _{value}										
			SessionInstGrp ₁										
			L _{en}	T _{ag}	V _{al}								
					SessionInstType					SessionInstScope			
			L	T		V		L	T		V		
0E	B1	6D	0B	B1	6D	04	B6	42	01	04	B7	42	01

Example LTV Encoded Message

The following pseudo-code is an example of the **Logon** message, showing individual data fields and values, as well as the *RequestHeader* group (structure), and one iteration of the *SessionInstGrp* repeating group (array).

```

structure LogonReq                                     # ApplID = 10000
{
    structure RequestHeader                             # ID = 11000
    {
        char *MsgType = "A"                           # ID = 35
        int MsgSeqNum = 1                             # ID = 34
    }
}

```

```

int HeartBtInt = 5000           # ID = 108
char *MemberName = "ABC01E"    # ID = 5364
char *Username = "1"           # ID = 553
char *Password = "password"     # ID = 554
structure SessionInstGrp [1]    # ID = 14001
{
    int SessionInstType = 1      # ID = 8502
    int SessionInstScope = 1     # ID = 8503
}
}

```

The table below provides a detailed view of the LTV encoded message data, as well as the actual encoded bytes. The last row depicts the actual byte-by-byte encoding, in hexadecimal, for the corresponding LTV element (shaded to show nesting).

Table 82: LogonReq LTV message encoding

LogonReq																																								
Length	Tag	V _{alue}																																						
57	10000	RequestHeader							HeartBtInt			MemberName				Username			Password				SessionInstGrp _{wrapper}																	
		Len	Tag	V _{al}				Len	Tag	V _{al}	Len	Tag	V _{al}		Len	Tag	V _{al}	Len	Tag	V _{al}		Len	Tag	V _{al}																
		9	11000	MsgType			MsgSeqNum		4	108	5000	9	5364	“ABC01E”		4	553	“1”	11	554	“password”		14	14001	SessionInstGrp ₁															
																							SessionInstGrp ₁																	
		L	T	V	L	T	V																																	
					3	35	“A”	3	34	1																														
																									SessionInstType									SessionInstScope						

NOTE: The Length and Tag fields for the message itself are fixed-length, and are not continuation-bit encoded.

Appendix B: ISE Parameters

Various ISE parameters and their corresponding values are listed in the following table.

ISE Parameters		Description	Value	
Order/Quote Maximum Price			\$99,999.9999	
Order/Quote Maximum Quantity		Simple and Standard Combos	999,999	
		Stock Combos	9,999,999	
Combo Leg Maximum Ratio		Standard Combo	9,999	
		Stock Combo	9,999	
Combo OrderMax. Legs for “Legging-In”		Maximum number of legs on the order to leg-in to the simple order book.	Three	
EAM Crossing Order Percentages		Percentage of an order the EAM is guaranteed to execute if the EAM enters a Block size order into the Crossing Mechanism.	40%	
		Minimum allowable percentage	0%	
Order Minimum Sizes		Minimum allowed volume for an order.	Regular Option	Mini Option
	Regular Order		One contract	One contract
	Customer-to-Customer Cross (CCC)		One contract	One contract
	Qual. Contingent Cross (QCC)		1,000 contracts	10,000 contracts
	Block Order		50 contracts	500 contracts
	Facilitation Order		50 contracts	500 contracts
	PIM Order		One contract	One contract
	Solicitation Order		500 contracts	5,000 Contracts
Auction Response Times		Time allocated to respond to an Auction order.		
	Block Order		500 milliseconds	
	Combo Exposure Order		One second	
	Facilitation		500 milliseconds	
	Flash Order		150 milliseconds	
	PIM Order		500 milliseconds	
	Solicitation		500 milliseconds	
MM Parameters				
	Minimum Percentage	Minimum value for percentage counter.	99%	
	Exposure Timer	Min. & max. value for speed bump timer.	150 ≤ t ≤ 30,000 milliseconds	
	Ticks back	Min. & max. value for how many levels back to re-quote.	1 ≤ n ≤ 8	
Minimum Quote Size		Minimum size of regular quote at entry.	10 for all products unless explicitly communicated by ISE Market Operations that there is a different minimum quote size for a specific product.	
		Minimum size of complex quote at entry.	One.	

ISE Parameters	Description	Value
Min. Order Size to PMM	Minimum order size that is traded exclusively by the PMM.	$1 \leq n \leq 5$
Opening Time Intervals	Number of milliseconds between repeat calculations of the opening algorithm.	1,000 milliseconds (1 second)
PMM/CMM Size Distribution	Maximum percentages PMM receives, depending on number of MMs at the same quote.	60% (PMM and 1 MM) 40% (PMM and 2 MMs) 30% (PMM and 3+ MMs)

Appendix C: Glossary

The terms that are used in this document are described below:

Terms	Description
American Options	American options can be exercised at any time during the option life.
AMM	Away Market Maker. An options market maker on an away exchange.
At-the-Money series (ATM)	The strike price that is closest to the market price of the underlying security.
Away Market	All US option exchanges, other than ISE.
Best Bid and Offer (BBO)	Consolidated quantity, and best price for bids and offers on the ISE market. An away BBO refers to the best bid and offer price on the away exchanges.
Bin	To manage the assignment of market makers, each symbol is assigned to a bin. There are 10 bins, each with one PMM and multiple CMMs. A market maker is required to quote for symbols in their own bin and cannot quote outside the assigned bin. The bin concept is not applicable to EAMs. EAMs can trade on all symbols, regardless of the assigned bin. The symbols in a bin might not all be assigned to the same order book server.
Broadcast	A message sent to participants.
Broker-Dealer	A US-registered broker-dealer.
Cabinet Order	A Cabinet order is used to give away a position for tax reasons. A premium is not specified. However, the price is set implicitly by the ISE.
Call Option	An option contract that entitles the buyer to purchase a fixed number of shares of the underlying security at a stated price on or before a fixed expiry date.
Central Database	The ISE database that contains member names, user names, stock symbols, and series descriptions.
Class	An option class refers to all options of the same type – calls or puts, which also have the same underlying security.
Client	The client of a broker-dealer, or EAM. The client categories are: Away Market Maker (AMM) Proprietary (Clearing Firm or Customer) Broker Dealer (Clearing Firm or Customer) Customer
CMTA	Clearing Member Transfer Agreement. CMTA is recorded with OCC and permits one clearing member to transfer contracts to another clearing member.
Combo	A trade or order that consists of two or more option series, or stocks and options in one combination order. Also known as spreads.
Commodity	See Underlying or Product.

Terms	Description
Competitive Market Maker (CMM)	A market maker that is required to provide continuous quotations in a portion of the options classes within their assigned group. There are multiple CMMs appointed to each of the 10 groups of options that are traded on the exchange. Each CMM quotes independently. In addition to providing quotes in their assigned options classes, CMMs can also conduct a limited amount of trading in other options classes that are traded on the exchange. CMMs must purchase or lease CMM memberships. Each membership entitles a CMM to enter quotes in one group of options. CMMs are not permitted to represent agency orders.
Contract	Options are traded in contracts. At ISE, a contract is usually for 100 shares of the underlying security. A “mini” contract is usually for 10 shares. The contract size might change due to corporate actions.
Crossing Order Mechanism	Crossing orders permit an EAM to execute a portion of its own Block size orders (50 contracts or more) as principal.
Customer order	An order that is entered on behalf of a member’s customer, who is not a registered broker-dealer.
Deal	A deal occurs when buy orders are matched with sell orders. One deal involves at least two orders and generates at least two trades. However, there can be several orders and several trades in one deal.
Direct Trading Interface (DTI)	This is also referred to as the DTI. The DTI contains the sub-routines that must be used to communicate with the T7 trading system.
Directed Market Maker (DMM)	A market maker that receives a directed order.
Directed Order	A Directed Order is a simple order that is sent to a specific market participant. The order is no longer directed if it is released or changed to a PIM order.
Electronic Access Member (EAM)	A broker-dealer that represents customer and non-customer orders on the exchange, as well as their own proprietary orders.
European Options	European options can only be exercised at maturity.
Exercise	Declaring the desire to use the right to buy or to sell an option.
Exercise Price	The price at which the underlying security can be bought or sold. This is also known as the strike price.
Expiration Date	The last day on which the option can be exercised. All the rights and obligations conferred by the option are null and void after the expiration date.
Far Market Maker (FARMM)	See AMM, above.
Fast Market	Indicates exceptional market conditions. A fast market occurs when the PMM is present but has not rotated the series, or there is a market broadcast that has made the market volatile. In fast markets, the away market prices are not checked and there are no minimum volume requirements.
Fill Or Kill (FOK)	An order that immediately trades the whole order within the limit price, or is deleted without trading at all. The order is not placed on the order book.
Firm Order	An order that is entered for the member’s proprietary account.
Guaranteed Directed Order (GDO)	A snapshot of a quote at the NBBO that is only available for a specific directed order. The GDO is saved for later use by the market maker that is on the opposite side of the order.

Terms	Description
Immediate Or Cancel (IOC)	An order that trades up to the limit price. Any remaining quantity is deleted. The order is not placed on the order book.
Instrument	Represents a trade-able entity at the exchange. Also called a Series.
Intermarket Sweep Order (ISO)	An order that should execute, if possible, without regard to protection of other markets' prices.
Limit Order	An order that must specify the premium.
Market	A forum in which market participants can trade. An exchange can operate multiple markets. Each market can open and close at different times, and trade different securities.
Market Order	An order without a limit price that trades at the BBO until the quantity is satisfied.
Member	US-registered broker-dealer that has received approval to trade on ISE.
Mini Option	An option delivering 10 shares of the underlying security per contract, rather than the usual 100 shares per contract.
Non-Customer	Primary Market Maker (PMM), Competitive Market Maker (CMM), FIRM, Broker-Dealer, or Far Market Maker (FARMM).
OPRA	Options Price Reporting Authority. The authority that disseminates the prices from all US options exchanges through a single feed.
Option	An agreement between a buyer and a seller that gives the buyer the right, through exercise, to require the seller to perform certain specific obligations. An option, which is left unexercised, expires worthless after a stated period of time.
Order	An entry into the trading system that indicates the intention to buy or sell.
Order book	The central order book contains the orders and quotes for all participants. A participant might have a local order book on their trading application to manage the orders. The trading system operates multiple order book servers to process trading transactions.
PrecISE Trade	A trader workstation that is used by EAMs.
Premium	The price of an option.
Price Improvement Mechanism (PIM)	An order that allows an EAM to trade against its own orders as principal and at least one cent better than the same-side IBBO and at least at the National Best Bid or Offer (NBBO) on either side.
Primary Market	Each underlying is assigned to one primary market by ISE, even if it is traded in several markets or systems. Only the price from the designated primary market is posted.
Primary Market Maker (PMM)	A market maker that has significant market-making responsibilities, such as overseeing the opening, providing continuous quotations in all of their assigned options classes, and handling customer's orders (with respect to available prices from other options exchanges). One PMM is assigned to each of the 10 options bins that are traded on the exchange.
Product	The set of options over an Underlying Security. Also known as Commodity, Symbol, or Name.
Put Option	An option contract that entitles the buyer to sell a fixed number of shares of the underlying security at the stated prices on or before a fixed expiry date.
Special Orders	Block, Crossing, and Cabinet order transactions.
Spot	The underlying security.

Terms	Description
Stock option	A contract that gives its holder the right to buy or sell shares of the underlying security at a specified price by a specified date. There are two types of options: Calls – the right to buy an underlying security Puts – the right to sell an underlying security
Strike Price	The price at which the option can be exercised.
T7	The computer system that processes the transactions and generates broadcasts. This system is operated by ISE.
The trading system	The computer system that processes the transactions and generates broadcasts. This system is operated by ISE.
Tick-Worse	A mechanism that moves a quote away from the market.
Time-In-Force (TIF)	See Validity Time
Time Value	The amount by which the option premium (actual cost) is above the intrinsic value of the option.
Trade	One side of a deal. When a buy order matches a sell order, the buy side is one trade and the sell side is another trade. Each user is informed of their member's trades.
Transaction	A trade. This term also refers to the processing cycle in which a message is received by the trading system, processed, and a reply is returned.
Underlying (underlying security)	An underlying security is the stock on which the options contract is based. Options can trade on many kinds of assets, such as commodity options and stock options. See Product.
User	For Direct Trading Interface applications, the user is the member's computer system that is logged onto ISE. A Valid Username and password are needed to establish a trading session to the DTI.
Validity Time	Also known as Time-In-Force. The life of an order that does not trade out completely. Validity can be applied to Day, Good until Canceled (GTC), Fill Or Kill (FOK), and Immediate Or Cancel (IOC) orders.

Appendix D: DTI Conformance Test

In order to connect to the DTI, a client application must demonstrate that the application:

- Is stable in its operation
- Conforms with specific design requirements

ISE has developed several conformance tests (one for each membership type). The type of conformance test that is performed on the client application depends on the purpose of that application. Member firms are required to pass a detailed conformance test with the ISE, addressing each area outlined in the test requirements for the functionality implemented. The ISE reserves the right to ask a member to revisit conformance testing to address issues. The tests are conducted jointly by TMS and the participant.

Testing all DTI Applications

The participant must prove that the DTI application can:

- Establish and maintain communications with the DTI
- Use correct product and instrument IDs
- Successfully send messages and process responses and broadcasts
- React accordingly to changing market and/or instrument statuses
- Recover after application logout/disconnect

NOTE: The application must not perform repeat queries to obtain information contained in broadcasts.

Testing Order Routing Applications

In addition to the testing that all applications must perform, applications that send orders to ISE must demonstrate they can:

- Enter orders of different types and time validities (for example Limit, Market, FOK, IOC, Day, etc.), with different prices (outside market, within market, at market, cross-market, and market orders)
- Have a different order source (Customer, Firm, MM, FARMM, etc.), that must also show the application maintains the correct order number
- Amend an order
- Delete an order
- Record the trades that are generated by the orders
- Record the current state of orders

Testing MM Applications

In addition to the testing that all applications must perform, applications that send quotes to ISE must demonstrate that they can:

- Correctly set MM parameters

- Enter mass quotes for assigned products
- Update, inactivate, and delete quotes
- Correctly process locked orders (Directed MMs)

Testing PMM Applications

In addition to the testing that all MM applications must perform, PMM applications must demonstrate that they can:

- Change instrument status
- Clear opening imbalance
- Correctly process Three Tick and Market Order locks

Appendix E: DTI Application Messages

The following tables enumerate the DTI Application messages, first alphabetically by name, then by message number.

DTI Application Messages, by name

Application Msg Name	Numeric ID	Application Msg Name	Numeric ID
AddComplexInstrumentRequest	10306	MultilegOrderCancelReplace	10116
AddComplexInstrumentResponse	10307	NewOrderCross	10114
ApplicationSequenceNumberReset (Jump)	10020	NewOrderMultileg	10115
BusinessEndOfDayBroadcast	10405	NewOrderSingle	10100
BusinessEndOfStreamBroadcast	10404	News	10013
ChangeInstrumentStateRequest	10300	OrderAcknowledge	10103
ChangeInstrumentStateResponse	10301	OrderCancelReplaceRequest	10101
DealItemBroadcast	10400	OrderCancelRequest	10102
DealItemMaintenanceInstructionRequest	10402	OrderLockRequest	10108
DealItemMaintenanceInstructionResponse	10403	OrderLockResponse	10110
DeleteAllOrdersRequest	10105	OrderReleaseRequest	10109
DeleteAllOrdersResponse	10106	OrderReleaseResponse	10111
DeleteAllQuotesRequest	10205	PendingResponse	10014
DeleteAllQuotesResponse	10206	PrivateMMExecutionReportBroadcast	10305
EndOfStreamBroadcast	10015	PrivateOrderBroadcast	10104
GatewayRequest	10018	PrivateQuoteBroadcast	10200
GatewayResponse	10019	ProductResetBroadcast	10406
Heartbeat	10011	QuoteActionReport	10202
InvalidDataBroadcast	10016	QuoteActionRequest	10201
InvalidResponse	10017	Reject	10010
LogonReq	10000	RetransmitReq	10008
LogonResp	10001	RetransmitResp	10009
LogoutReq	10002	SubscribeReq	10004
LogoutResp	10003	SubscribeResp	10005
MassQuote	10203	TradeItemBroadcast	10401
MassQuoteAcknowledgement	10204	TradingSessionMarkerBroadcast	10407
Member Kill Switch Request	10021	UnsubscribeReq	10006
Member Kill Switch Response	10022	UnsubscribeResp	10007
MMPParameterDefinitionRequest	10308	UserNotification	10012
MMPParameterDefinitionRequestAck	10309		

DTI Application Messages, by numeric ID

Application Msg Name	Numeric ID	Application Msg Name	Numeric ID
LogonReq	10000	OrderLockRequest	10108
LogonResp	10001	OrderReleaseRequest	10109
LogoutReq	10002	OrderLockResponse	10110
LogoutResp	10003	OrderReleaseResponse	10111
SubscribeReq	10004	NewOrderCross	10114
SubscribeResp	10005	NewOrderMultileg	10115
UnsubscribeReq	10006	MultilegOrderCancelReplace	10116
UnsubscribeResp	10007	PrivateQuoteBroadcast	10200
RetransmitReq	10008	QuoteActionRequest	10201
RetransmitResp	10009	QuoteActionReport	10202
Reject	10010	MassQuote	10203
Heartbeat	10011	MassQuoteAcknowledgement	10204
UserNotification	10012	DeleteAllQuotesRequest	10205
News	10013	DeleteAllQuotesResponse	10206
PendingResponse	10014	ChangeInstrumentStateRequest	10300
EndOfStreamBroadcast	10015	ChangeInstrumentStateResponse	10301
InvalidDataBroadcast	10016	PrivateMMExecutionReportBroadcast	10305
InvalidResponse	10017	AddComplexInstrumentRequest	10306
GatewayRequest	10018	AddComplexInstrumentResponse	10307
GatewayResponse	10019	MMPParameterDefinitionRequest	10308
ApplicationSequenceNumberReset (Jump)	10020	MMPParameterDefinitionRequestAck	10309
MemberKillSwitchRequest	10021	DealItemBroadcast	10400
MemberKillSwitchResponse	10022	TradeItemBroadcast	10401
NewOrderSingle	10100	DealItemMaintenanceInstructionRequest	10402
OrderCancelReplaceRequest	10101	DealItemMaintenanceInstructionResponse	10403
OrderCancelRequest	10102	BusinessEndOfStreamBroadcast	10404
OrderAcknowledge	10103	BusinessEndOfDayBroadcast	10405
PrivateOrderBroadcast	10104	ProductResetBroadcast	10406
DeleteAllOrdersRequest	10105	TradingSessionMarkerBroadcast	10407
DeleteAllOrdersResponse	10106		

Appendix F: DTI Groups

The following tables enumerate the DTI Groups.

DTI Groups, by name

Group Name	Numeric ID
ApplIDReportGrp	15117
ApplIDRequestAckGrp	14032
ApplIDRequestGrp	14029
BroadcastHeader	11006
DisclosureInstGrp	14018
GatewayGrp	14035
InstrmtGrp	15017
InstrmtLegExecGrp	15018
InstrmtMDReqGrp	15022
InstrumentLegGrp	14017
LegCrossGrp	14027
LegOrdGrp	15025
LinesOfTextGrp	15029
MarketSegmentGrp	15132
MDFullGrp	15031
NestedParties	14008
NestedParties2	14009
NestedParties3	14010
NewsRefGrp	15144
NotAffectedMarketSegmentGrp	14033
OrdEventDealItemGrp	14046
OrdEventTradeItemGrp	14045
OrdEventLegExecGrp	14036
OrdEventLegGrp	14037
OrdEventsGrp	14026
Parties	14012

Group Name	Numeric ID
PartitionGrp	14003
PartyDetailGrp	14040
PartyEntitlementAckGrp	11042
PartyEntitlementUpdateGrp	11041
QuoteEventDealItemGrp	14044
QuoteEventTradeItemGrp	14043
QuoteEventLegExecGrp	14038
QuoteEventLegGrp	14039
QuoteEventsGrp	14011
QuotEntryAckGrp	15042
QuotEntryGrp	15043
QuotSetAckGrp	15048
QuotSetGrp	15049
RelatedInstrumentGrp	14021
RequestHeader	11000
ResponseHeader	11002
RootParties	14031
SecSizesGrp	15102
SessionInstGrp	14001
SideCrossOrdModGrp	15059
TargetMarketSegmentGrp	14034
TargetParties	14063
ThrottleParamsGrp	14030
TrdAllocGrp	15060
TrdCptRptSideGrp	14019
TrdRegTimestamps	14020

DTI Groups, by numeric ID

Group Name	Numeric ID
RequestHeader	11000
ResponseHeader	11002
BroadcastHeader	11006
PartyEntitlementUpdateGrp	11041
PartyEntitlementAckGrp	11042
SessionInstGrp	14001
PartitionGrp	14003
NestedParties	14008
NestedParties2	14009
NestedParties3	14010
QuoteEventsGrp	14011
Parties	14012
InstrumentLegGrp	14017
DisclosureInstGrp	14018
TrdCptRptSideGrp	14019
TrdRegTimestamps	14020
RelatedInstrumentGrp	14021
OrdEventsGrp	14026
LegCrossGrp	14027
ApplIDRequestGrp	14029
ThrottleParamsGrp	14030
RootParties	14031
ApplIDRequestAckGrp	14032
NotAffectedMarketSegmentGrp	14033
TargetMarketSegmentGrp	14034
GatewayGrp	14035

Group Name	Numeric ID
OrdEventLegExecGrp	14036
OrdEventLegGrp	14037
QuoteEventLegExecGrp	14038
QuoteEventLegGrp	14039
PartyDetailGrp	14040
QuoteEventTradeItemGrp	14043
QuoteEventDealItemGrp	14044
OrdEventTradeItemGrp	14045
OrdEventDealItemGrp	14046
TargetParties	14063
InstrmtGrp	15017
InstrmtLegExecGrp	15018
InstrmtMDReqGrp	15022
LegOrdGrp	15025
LinesOfTextGrp	15029
MDFullGrp	15031
QuotEntryAckGrp	15042
QuotEntryGrp	15043
QuotSetAckGrp	15048
QuotSetGrp	15049
SideCrossOrdModGrp	15059
TrdAllocGrp	15060
SecSizesGrp	15102
ApplIDReportGrp	15117
MarketSegmentGrp	15132
NewsRefGrp	15144

Appendix G: DTI Field Descriptions (Alphabetical)

DTI fields, with elemental data type, are listed in alphabetical order.

Allowed values listed here are defined in the isefix.xml template file, unless marked with an asterisk (*).

Unless specified otherwise, the *maximum* length for any string field is 20 characters.

Field Name	Tag	Type	Allowed Value(s)
AgreementDesc	913	String	≤ 40 chars.
AllocClearingCapacity	8588	int	1=Customer 2=Firm 3=Market_Maker
AllocOrderCapacity	8589	char	G=Proprietary C=Customer D=Customer_Professional B=Broker_Dealer N=Away_Market_Maker M=Market_Maker
AllocPositionEffect	1047	char	C=Close O=Open
AllocQty	80	int	
AllocText	161	String	
ApplBegSeqNum	1182	int	
ApplEndSeqNum	1183	int	
ApplID	1180	String	O=Order_status Q=Quote_status M=Market_maker_information T=Trade_item D=Deal_item N=News S=Service_availability
ApplLastSeqNum	1350	int	
ApplReportType	1426	int	1=Last_Message 2=Heartbeat 4=Invalid_Data
ApplReqScope	8569	int	0=All_messages 1=Only_for_user
ApplReqType	1347	int	0=Retransmission 1=Subscription 4=Unsubscribe
ApplResponseID	1353	String	

Field Name	Tag	Type	Allowed Value(s)
ApplSeqNum	1181	int	
AuctionAllocationPct	8523	float	$0 \leq n \leq 0.4$ (default = 40%) 30% (0.3) is entered as 30,000,000
AuctionInst	8601	int	0=Start_auction 1=Do_not_start_auction
AuctionType	8522	int	(*) See Table 30: Auction Types , on page 67
BidCxlSize	8547	int	
BidPx	132	float	
BidSize	134	int	
BidTotalSize	8579	int	
BranchSeqNbr	9861	String	
ClearingCapacity	8521	int	1=Customer 2=Firm 3=Market_Maker
ClearingVenue	8613	String	
ClOrdID	11	String	
CompletionIndicator	6228	char	N=Not_last_message Y=Last_message
ContraDisplayPrice	8648	float	INTERNAL ISE USE ONLY (defined in isefix.xml)
CounterpartyClearingCapacity	8705	int	1=Customer 2=Firm 3=Market Maker
CounterpartyOrderCapacity	8704	char	C=Customer (*) D=Customer Professional (*) B=Broker/Dealer (*) G=Proprietary N=Away Market Maker (*) M=Market Maker (*)
CrossType	549	int	(*) See Table 27: Cross Order Types , on page 59
CumQty	14	int	
DealItemID	8701	String	
DealItemQty	8702	int	
DefaultCstmApplVerID	1408	String	Version=007.000.000
Delta	7966	int	

Field Name	Tag	Type	Allowed Value(s)
DisclosureInst	8578	int	0=No 1=Yes 2=Use_Default
DisclosureType	8577	int	1=Volume 2=Price 3=Side 4=AON 5=General 6=Clearing_Account 7=CMTA_Account
DiscretionPrice	845	float	
DisplayHighQty	1086	int	
DisplayLowQty	1085	int	
DisplayMethod	1084	char	1=Initial 3=Random
DisplayQty	1138	int	
DisplayWhen	1083	char	1=Immediate 2=Exhaust
EntitlementRequestID	1770		
EntitlementRequestStatus	1882	int	0=Accept 2=Reject
EntitlementStatus	1883	int	0=Accepted
ExecID	17	String	
ExecInst	18	String	1=Not Held 6=Participate don't initiate G=AON H=Reinstate_on_failure (persist) S=Suspend f=Intermarket_Sweep h=External_Routing_Not_Alwd q=Release_from_suspension z=Cancel_remaining
ExecRestatementReason	378	int	

Field Name	Tag	Type	Allowed Value(s)
ExecType	150	char	0=New 4=Canceled 5=Replaced 6=Pending_Cancel 9=Suspended C=Expired D=Restated F=Trade L=Activated
ExpireDate	432	String	
ExposureDuration	1629	int	
GatewayID	8644	char	
GatewaySubID	8645	char	
Headline	148	String	
HeartBtInt	108	int	
IndividualAllocID	467	String	
InitialDisplayQty	1608	int	
LastMkt	30	String	1=BNY ConvergeEx U.S. Transaction Services (*) 3=BNY ConvergeEx Millennium ATS (*) 4=Knight Match (*) 5=Knight Link (*) 6=Instinet CBX (US) (*) 7=Deutsche Bank ATS (*) 8=Cheevers (*) 9=Libucki (*) 10=FOG Equities (*) 11=Smart Routing (KCG) (*)
LastPx	31	float	
LastQty	32	int	
LastUpdateTime	779	int	
LeavesQty	151	int	
LegClearingCapacity	8575	int	1=Customer 2=Firm 3=Market_Maker
LegPositionEffect	564	char	C=Close O=Open
LegRatioQty	623	int	

Field Name	Tag	Type	Allowed Value(s)
LegSecurityID	602	String	
LegSecurityType	609	String	CS=Common_Stock OPT=Option
LegSide	624	char	1=Buy 2=Sell 5=Sell_short 6=Sell_short_exempt B=As_Defined
ListUpdateAction	1324	char	D=Delete
LockedQty	8533	int	
LockType	8531	int	1=Away_Market_Better 2=Three_Tick_Locked 3=Locked_by_MM 4=Directed_Order_Lock 5=Multileg_Lock 6=Market_Order_Lock
MarketID	1301	String	XISX=ISE Exchange GMNI=ISE Gemini Exchange MCRY=ISE Mercury Exchange
MarketMakerEventReason	8566	int	
MarketMakerEventType	8565	int	1=Assignment_of_Security 2=Deassignment_of_Security
MarketSegmentID	1300	String	
MassActionRejectReason	1376	int	
MassActionResponse	1375	int	0=Rejected 1=Accepted 2=Accepted_Warning
MassActionScope	1374	int	1=Security 8=Market 9=Market_Segment 13=Product_Complex 100=Partition
MassActionType	1373	int	1=Suspend 2=Release 3=Cancel 100=Suspend and delete quotes
MatchDate	8582	String	

Field Name	Tag	Type	Allowed Value(s)
MatchSubType	8610	int	0=NONE (*) 1=BLOCK_ORDER_SIMPLE (*) 2=DIRECTED_ORDER_SIMPLE (*) 3=FLASH_ORDER_SIMPLE (*) 4=EXPOSURE_ORDER_STANDARD (*) 5= EXPOSURE_ORDER_STOCK (*) 6= FACILITATION_ORDER_SIMPLE (*) 7=FACILITATION_ORDER_STANDARD (*) 8=FACILITATION_ORDER_STOCK (*) 9=SOLICITATION_ORDER_SIMPLE (*) 10=SOLICITATION_ORDER_STANDARD (*) 11=SOLICITATION_ORDER_STOCK (*) 12=PRICE_IMPROVEMENT_MECHANISM_SIMPLE (*) 13=PRICE_IMPROVEMENT_MECHANISM_STD (*) 14=PRICE_IMPROVEMENT_MECHANISM_STOCK (*) 15= DIRECTED_ORDER_PIM_SIMPLE (*)
MatchType	574	int	3=Confirmed_Trade_Report 4=Auto_match_incoming 5=Cross_Auction 6=Counter_Order_Selection 9=Opening_Auction 10=Auction 11=Auto_match_resting 12=Auto_match_pinging
MDEntryPx	270	float	
MDEntrySize	271	int	
MDEntryType	269	char	0=Bid 1=Offer b=Auction_Price
MDReportID	963	int	
MDSecSize	1179	int	
MDSecSizeType	1178	int	2=Total_Open_Size
MemberName	5364	String	
MessageEventSource	1011	int	1=Matching_Engine 2=Manual_Trade_Entry 3=Deal_Modification_User 4=Deal_Modification_Exchange 5=Trade_Bust 6=Deal_Bust 7=Deal_Management_Exchange
MinQty	110	int	0=No Exec 1=Exec

Field Name	Tag	Type	Allowed Value(s)
MinQtyExecIndicator	8643	int	0=No Exec 1=Exec
MinQtyType	8602	char	O=Once R=Repeat
MMReqResult	8637	int	
MMReqStatus	8636	int	0=Accepted 1=Accepted_warning 2=Rejected
MsgSeqNum	34	int	
MsgType	35	String	0=Heartbeat 5=Logout 8=ExecutionReport A=Logon AB=NewOrderMultileg AC=MultilegOrderCancelReplace AE=TradeCaptureReport AR=TradeCaptureReportAck B=News BW=ApplicationMessageRequest BX=ApplicationMessageRequestAck BY=ApplicationMessageReport BZ=OrderMassActionReport CA=OrderMassActionRequest CB=UserNotification CO=SecurityMassStatus D=NewOrderSingle DA=PartyEntitlementsDefinitionRequest DB=PartyEntitlementsDefinitionRequestAck F=OrderCancelRequest G=OrderCancelReplaceRequest V=MarketDataRequest W=MarketDataSnapshotFullRefresh b=MassQuoteAcknowledgement c=SecurityDefinitionRequest d=SecurityDefinition h=TradingSessionStatus i=MassQuote s=NewOrderCross U1=OrderAcknowledge U2=Pending U3=Reject U4=InvalidResponse U8=QuoteExecutionReport U10=QuoteActionRequest U11=QuoteActionReport

Field Name	Tag	Type	Allowed Value(s)
			U12=ChangeInstrumentStateRequest U13=ChangeInstrumentStateResponse U14=MMPParameterDefinitionRequest U20=MarketMakerUpdateReport
Nested2PartyID	757	String	
Nested2PartyRole	759	int	2=Broker_of_Credit 4=Clearing_Firm 12=Executing_Trader 24=Customer_Account 45=Secondary_Account_Number 55=Session_ID 59=Executing_Unit 66=Market_Maker 83=Clearing_Account 94=Away_Market
Nested3PartyID	949	String	
Nested3PartyRole	951	int	2=Broker_of_Credit 4=Clearing_Firm 12=Executing_Trader 24=Customer_Account 45=Secondary_Account_Number 55=Session_ID 59=Executing_Unit 66=Market_Maker 83=Clearing_Account 94=Away_Market
NestedPartyID	524	String	
NestedPartyRole	538	int	2=Broker_of_Credit 4=Clearing_Firm 12=Executing_Trader 24=Customer_Account 45=Secondary_Account_Number 55=Session_ID 59=Executing_Unit 66=Market_Maker 83=Clearing_Account 94=Away_Market
NewQuoteSize	8617	int	
NewsID	1472	String	
NewsRefID	1476	String	
NewsRefType	1477	int	0=Replacement 3=Removal

Field Name	Tag	Type	Allowed Value(s)
NotAffectedMarketSegmentID	8612	String	
NumberOfSecurities	8614	int	
OfferCxlSize	8548	int	
OfferPx	133	float	
OfferSize	135	int	
OfferTotalSize	8580	int	
OrderCapacity	528	char	G=Proprietary C=Customer D=Customer_Professional B=Broker_Dealer N=Away_Market_Maker M=Market_Maker
OrderCategory	1115	char	1=Order 2=Quote 5=Linked_order 8=Cross_Order A=Auction D=Directed_Order
OrderID	37	String	
OrderQty	38	int	
OrderSide	8586	char	1=Buy 2=Sell 5=Sell_Short 6=Sell_Short_Exempt
OrdEventLastMkt	9000	String	A = NYSE AMEX B = BOX C = CBOE H = ISE Gemini I = ISE J = ISE Mercury M = Miami N = NYSE ARCA Q = NASDAQ T = NASDAQ BX W = C2 X = NASDAQ PHLX Z = BATS

Field Name	Tag	Type	Allowed Value(s)
OrdEventLegLastMkt	8624	String	1=BNY ConvergeX U.S. Transaction Services (*) 3=BNY ConvergeX Millennium ATS (*) 4=Knight Match (*) 5=Knight Link (*) 6=Instinet CBX (US) (*) 7=Deutsche Bank ATS (*) 8=Cheevers (*) 9=Libucki (*) 10=FOG Equities (*) 11=Smart Routing (KCG) (*)
OrdEventLegLastPx	8623	float	
OrdEventLegQty	8622	int	
OrdEventMatchSubType	8621	int	(*) See MatchSubType (tag 8610)
OrdEventMatchType	8620	int	3=Confirmed_Trade_Report 4=Auto_match_incoming 5=Cross_Auction 6=Counter_Order_Selection 9=Opening_Auction 10=Auction 11=Auto_match_resting 12=Auto_match_pinging
OrdEventPx	8554	float	
OrdEventQty	8537	int	
OrdEventReason	8536	int	
OrdEventType	8535	int	1=Added 2=Modified 3=Deleted 4=Partially_filled 5=Filled 6=Locked 7=Released 8=Restated
OrdStatus	39	char	0=New 1=Partially_filled 2=Filled 4=Canceled 6=Pending_Cancel 9=Suspended

Field Name	Tag	Type	Allowed Value(s)
OrdType	40	char	1=Market 2=Limit 3=Stop 4=Stop_Limit Q=Counter_order_selection
OrigClOrdID	41	String	
OrigTime	42	int	
OrigTradeDate	1125	String	
PartitionID	5948	int	
PartitionStatus	5969	int	0=Unavailable 1=Available
PartyDetailID	1691		
PartyDetailRole	1693	int	59=Executing Unit 55=Session
PartyID	448	String	
PartyRole	452	int	2=Broker_of_Credit 4=Clearing_Firm 12=Executing_Trader 24=Customer_Account 45=Secondary_Account_Number 55=Session_ID 59=Executing_Unit 66=Market_Maker 83=Clearing_Account 94=Away_Market
Password	554	String	
PctCount	8615	int	
PendingReason	5946	int	
PositionEffect	77	char	C=Close O=Open
Price	44	float	
PriceProtectionScope	1092	char	1=Local 2=National
PrivateReference	5476	String	
ProductComplex	1227	int	1=Simple_instrument 2=Standard_combination 3=Stock_combination 4=Equity

Field Name	Tag	Type	Allowed Value(s)
QuoteEntryRejectAction	8549	int	0=None 1=Canceled
QuoteEntryRejectReason	368	int	
QuoteEntryStatus	1167	int	0=Accepted 5=Rejected 10=Pending 100=Accepted_warning
QuoteEventLegLastMkt	8632	String	1=BNY ConvergeEx U.S. Transaction Services (*) 3=BNY ConvergeEx Millennium ATS (*) 4=Knight Match (*) 5=Knight Link (*) 6=Instinet CBX (US) (*) 7=Deutsche Bank ATS (*) 8=Cheevers (*) 9=Libucki (*) 10=FOG Equities (*) 11=Smart Routing (KCG) (*)
QuoteEventLegLastPx	8631	float	
QuoteEventLegQty	8630	int	
QuoteEventLiquidityInd	8641	int	(Not currently used.)
QuoteEventPx	8553	float	
QuoteEventQty	8542	int	
QuoteEventReason	8540	int	
QuoteEventSide	8581	char	1=Buy 2=Sell 5=Sell_Short 6=Sell_Short_Exempt
QuoteEventType	8539	int	1=Added_quote_side 2=Removed_quote_side 3=Partially_filled 4=Filled 5=Modify_quote_side 6=Suspended
QuoteID	117	String	

Field Name	Tag	Type	Allowed Value(s)
QuoteInactiveStatus	297	int	0=All active 1=Simple Inactive 2=Standard Combo Inactive 3=Simple and Standard Combo Inactive 4=Stock Combo Inactive, 5=Simple and Stock Combo Inactive 6=Standard Combo and Stock Combo Inactive 7= All Inactive
QuoteMsgID	1166	String	
QuoteOffsetValue	8616	int	
RefApplID	1355	String	O=Order_status Q=Quote_status M=Market_maker_information T=Trade_item D=Deal_item N=News S=Service_availability
RefApplLastSeqNum	1357	int	
RefApplResponseID	8515	String	
RefApplSeqMode (Deprecated)	8514	int	0=Unrecoverable 1=Recoverable
RefClOrdID	8516	String	
RefOrderID	1080	String	
RejectText	1328	String	
RelatedHighPrice	8573	float	
RelatedLowPrice	8572	float	
RelatedPriceSource	8574	int	1=Underlying_NBB 2=Underlying_NBO
RelatedProductComplex	8587	int	1=Simple_instrument 2=Standard_combination 3=Stock_combination 4=Equity
RelatedSecurityID	1650	String	
ReleaseInst	8603	int	1=ISO 2=No_ABBO_Check
ReleaseQty	8543	int	
RequestTime	5979	int	

Field Name	Tag	Type	Allowed Value(s)
RootPartyID	1117	String	
RootPartyRole	1119	int	2=Broker_of_Credit 4=Clearing_Firm 12=Executing_Trader 24=Customer_Account 45=Secondary_Account_Number 55=Session_ID 59=Executing_Unit 66=Market_Maker 83=Clearing_Account 94=Away_Market
RotateRequestIndicator	8647	char	Y=Rotate N=Do_Not_Rotate
RoutingInst	9303	int	1=Sweep
SecondaryLockedQty	8544	int	
SecurityID	48	String	
SecurityMassTradingStatus	1679	int	1=Opening_delay 2=Trading_halt 3=Resume 17=Ready_to_trade 18=Not_available 22=Opening_Rotation 23=Fast_Market
SecurityReportID	964	int	
SecurityTradingEvent	1174	int	
SecurityTradingStatus	326	int	1=Opening_delay 2=Trading_halt 3=Resume 17=Ready_to_trade 18=Not_available 22=Opening_Rotation 23=Fast_Market
SenderCompID	49	String	
SenderSubID	50	String	
SendingTime	52	int	
ServiceID	7930	String	ME=Matching_Engine TM=Trade_Manager
SessionInstScope	8503	int	1=Orders 2=Orders_NP 3=Quotes

Field Name	Tag	Type	Allowed Value(s)
SessionInstType	8502	int	0=None 1=Cancel
SessionRejectReason	373	int	0=Invalid_Tag_Number 1=Required_Tag_Missing 2=Tag_not_defined 3=Undefined_tag 4=Tag_specified_without 5=Value_is_incorrect 6=Incorrect_data_format 7=Decryption_problem 8=Signature_problem 9=CompID_problem 10=SendingTime_Accuracy_Problem 11=Invalid_MsgType 12=XML_Validation_Error 13=Tag_appears_more 14=Tag_specified_out 15=Repeating_group_fields 16=Incorrect_NumInGroup_count 17=Non_Data_value 18=Invalid_Unsupported_Application 99=Other 101=Internal_technical_error 102=Rejected_by_throttle 104=Transaction_timeout 151=Invalid MsgSeqNum Received (*) 152=Too Many Heartbeats (*) 158=Throttle Window Size Violation (*)
SessionStatus	1409	int	0=Active 4=Logout 9=Takeover
Side	54	char	1=Buy 2=Sell 5=Sell_Short 6=Sell_Short_Exempt
SideLastPx	8585	float	
SideLastQty	1009	int	
SideLiquidityInd	1444	int	(*) See Table 24: Liquidity Indicator Values , on page 49
SideTradeID	1506	string	Carried to OCC TrdID field
StopPx	99	float	
TargetMarketSegmentID	8618	String	

Field Name	Tag	Type	Allowed Value(s)
TargetPartyID	1462	String	
TargetPartyRoleQualifier	8552	int	0=Agency 1=Principal 2=Riskless_Principal 3=General_Clearing_Member 4=Individual_Clearing_Member 5=Preferred_MM 6=Direct_MM
Text	58	String	
ThrottleAction	1611	int	1=Queue_outbound 2=Reject 3=Disconnect
ThrottleCountIndicator	1686	int	0=Unchanged 1=Decreased
ThrottleNoMsgs	1613	int	
ThrottleStatus	1609	int	0=Not_Queued 1=Queued
ThrottleTimeInterval	1614	int	
ThrottleTimeUnit	1615	int	0=Seconds 1=Seconds_10 2=Seconds_100 3=Seconds_1000
ThrottleType	1612	int	0=Inbound_Rate 1=Outstanding_Requests
TimeInForce	59	char	0=Day 1=GTC 2=OPG 3=IOC 4=FOK 6=GTD
TradeDate	75	String	
TradeID	1003	String	
TradeLinkID	820	String	
TradePublishIndicator	1390	int	0=Do_Not_Publish 1=Publish_Trade
TradeQty	1843	int	
TradeReportID	571	String	
TradeReportRefID	572	String	

Field Name	Tag	Type	Allowed Value(s)
TradeReportText	8583	String	
TradeReportTransType	487	int	0=New 2=Replace 4=Reverse
TradingSessionID	336	String	1=Day
TradSesStatus	340	int	2=Open 3=Closed 7=Reset
TransactTime	60	int	
TransferReason	830	int	1=Owner 2=Clearer 3=CMTA 4=Deal_Item_Drop_Copy 5=Trade_Item_Drop_Copy
TrdMatchTime	5507	int	
TrdRegTimestamp	769	int	
TrdRegTimestampType	770	int	1=Execution_Time 8=Time_Priority 9=Order_Creation
TriggerIndicator	8529	int	0=Not_triggered 1=Triggered
TriggerPrice	1102	float	
Urgency	61	char	0=Normal 1=Flash 2=Background
Username	553	String	
UserStatus	926	int	7=Forced_user_logout 9=Application_available
Vega	7968	int	

Appendix H: DTI Field Descriptions (Numerical)

DTI fields, with elemental data type, are listed in numerical order.

Allowed values listed here are defined in the isefix.xml template file, unless marked with an asterisk (*).

Unless specified otherwise, the *maximum* length for any string field is 20 characters.

Field Name	Tag	Type	Allowed Value(s)
ClOrdID	11	String	
CumQty	14	int	
ExecID	17	String	
ExecInst	18	String	1=Not Held 6=Participate don't initiate G=AON H=Reinstate_on_failure (persist) S=Suspend f=Intermarket_Sweep h=External_Routing_Not_Alwd q=Release_from_suspension z=Cancel_remaining
LastMkt	30	String	1=BNY ConvergeEx U.S. Transaction Services (*) 3=BNY ConvergeEx Millennium ATS (*) 4=Knight Match (*) 5=Knight Link (*) 6=Instinet CBX (US) (*) 7=Deutsche Bank ATS (*) 8=Cheevers (*) 9=Libucki (*) 10=FOG Equities (*) 11=Smart Routing (KCG) (*)
LastPx	31	float	
LastQty	32	int	
MsgSeqNum	34	int	
MsgType	35	String	0=Heartbeat 5=Logout 8=ExecutionReport A=Logon AB=NewOrderMultileg AC=MultilegOrderCancelReplace AE=TradeCaptureReport AR=TradeCaptureReportAck B=News BW=ApplicationMessageRequest BX=ApplicationMessageRequestAck

Field Name	Tag	Type	Allowed Value(s)
			BY=ApplicationMessageReport BZ=OrderMassActionReport CA=OrderMassActionRequest CB=UserNotification CO=SecurityMassStatus D=NewOrderSingle DA=PartyEntitlementsDefinitionRequest DB=PartyEntitlementsDefinitionRequestAck F=OrderCancelRequest G=OrderCancelReplaceRequest V=MarketDataRequest W=MarketDataSnapshotFullRefresh b=MassQuoteAcknowledgement c=SecurityDefinitionRequest d=SecurityDefinition h=TradingSessionStatus i=MassQuote s=NewOrderCross U1=OrderAcknowledge U2=Pending U3=Reject U4=InvalidResponse U8=QuoteExecutionReport U10=QuoteActionRequest U11=QuoteActionReport U12=ChangeInstrumentStateRequest U13=ChangeInstrumentStateResponse U14=MMPParameterDefinitionRequest U20=MarketMakerUpdateReport
OrderID	37	String	
OrderQty	38	int	
OrdStatus	39	char	0=New 1=Partially_filled 2=Filled 4=Canceled 6=Pending_Cancel 9=Suspended
OrdType	40	char	1=Market 2=Limit 3=Stop 4=Stop_Limit Q=Counter_order_selection
OrigClOrdID	41	String	
OrigTime	42	int	

Field Name	Tag	Type	Allowed Value(s)
Price	44	float	
SecurityID	48	String	
SenderCompID	49	String	
SenderSubID	50	String	
SendingTime	52	int	
Side	54	char	1=Buy 2=Sell 5=Sell_Short 6=Sell_Short_Exempt
Text	58	String	
TimeInForce	59	char	0=Day 1=GTC 2=OPG 3=IOC 4=FOK 6=GTD
TransactTime	60	int	
Urgency	61	char	0=Normal 1=Flash 2=Background
TradeDate	75	String	
PositionEffect	77	char	C=Close O=Open
AllocQty	80	int	
StopPx	99	float	
HeartBtInt	108	int	
MinQty	110	int	
QuotID	117	String	
BidPx	132	float	
OfferPx	133	float	
BidSize	134	int	
OfferSize	135	int	
Headline	148	String	

Field Name	Tag	Type	Allowed Value(s)
ExecType	150	char	0=New 4=Canceled 5=Replaced 6=Pending_Cancel 9=Suspended C=Expired D=Restated F=Trade L=Activated
LeavesQty	151	int	
AllocText	161	String	
MDEntryType	269	char	0=Bid 1=Offer b=Auction_Price
MDEntryPx	270	float	
MDEntrySize	271	int	
QuoteInactiveStatus	297	int	0=All active 1=Simple Inactive 2=Standard Combo Inactive 3=Simple and Standard Combo Inactive 4=Stock Combo Inactive, 5=Simple and Stock Combo Inactive 6=Standard Combo and Stock Combo Inactive 7= All Inactive
SecurityTradingStatus	326	int	1=Opening_delay 2=Trading_halt 3=Resume 17=Ready_to_trade 18=Not_available 22=Opening_Rotation 23=Fast_Market
TradingSessionID	336	String	1=Day
TradSesStatus	340	int	2=Open 3=Closed 7=Reset
QuoteEntryRejectReason	368	int	
SessionRejectReason	373	int	0=Invalid_Tag_Number 1=Required_Tag_Missing 2=Tag_not_defined 3=Undefined_tag 4=Tag_specified_without 5=Value_is_incorrect

Field Name	Tag	Type	Allowed Value(s)
			6=Incorrect_data_format 7=Decryption_problem 8=Signature_problem 9=ComplID_problem 10=SendingTime_Accuracy_Problem 11=Invalid_MsgType 12=XML_Validation_Error 13=Tag_appears_more 14=Tag_specified_out 15=Repeating_group_fields 16=Incorrect_NumInGroup_count 17=Non_Data_value 18=Invalid_Unsupported_Application 99=Other 101=Internal_technical_error 102=Rejected_by_throttle 104=Transaction_timeout 151=Invalid MsgSeqNum Received (*) 152=Too Many Heartbeats (*) 158=Throttle Window Size Violation (*)
ExecRestatementReason	378	int	
ExpireDate	432	String	
PartyID	448	String	
PartyRole	452	int	2=Broker_of_Credit 4=Clearing_Firm 12=Executing_Trader 24=Customer_Account 45=Secondary_Account_Number 55=Session_ID 59=Executing_Unit 66=Market_Maker 83=Clearing_Account 94=Away_Market
IndividualAllocID	467	String	
TradeReportTransType	487	int	0=New 2=Replace 4=Reverse
NestedPartyID	524	String	

Field Name	Tag	Type	Allowed Value(s)
OrderCapacity	528	char	G=Proprietary C=Customer D=Customer_Professional B=Broker_Dealer N=Away_Market_Maker M=Market_Maker
NestedPartyRole	538	int	2=Broker_of_Credit 4=Clearing_Firm 12=Executing_Trader 24=Customer_Account 45=Secondary_Account_Number 55=Session_ID 59=Executing_Unit 66=Market_Maker 83=Clearing_Account 94=Away_Market
CrossType	549	int	(*) See Table 27: Cross Order Types , on page 59
Username	553	String	
Password	554	String	
LegPositionEffect	564	char	C=Close O=Open
TradeReportID	571	String	
TradeReportRefID	572	String	
MatchType	574	int	3=Confirmed_Trade_Report 4=Auto_match_incoming 5=Cross_Auction 6=Counter_Order_Selection 9=Opening_Auction 10=Auction 11=Auto_match_resting 12=Auto_match_pinging
LegSecurityID	602	String	
LegSecurityType	609	String	CS=Common_Stock OPT=Option
LegRatioQty	623	int	
LegSide	624	char	1=Buy 2=Sell 5=Sell_short 6=Sell_short_exempt B=As_Defined

Field Name	Tag	Type	Allowed Value(s)
Nested2PartyID	757	String	
Nested2PartyRole	759	int	2=Broker_of_Credit 4=Clearing_Firm 12=Executing_Trader 24=Customer_Account 45=Secondary_Account_Number 55=Session_ID 59=Executing_Unit 66=Market_Maker 83=Clearing_Account 94=Away_Market
TrdRegTimestamp	769	int	
TrdRegTimestampType	770	int	1=Execution_Time 8=Time_Priority 9=Order_Creation
LastUpdateTime	779	int	
TradeLinkID	820	String	
TransferReason	830	int	1=Owner 2=Clearer 3=CMTA 4=Deal_Item_Drop_Copy 5=Trade_Item_Drop_Copy
DiscretionPrice	845	float	
AgreementDesc	913	String	≤ 40 chars.
UserStatus	926	int	7=Forced_user_logout 9=Application_available
Nested3PartyID	949	String	
Nested3PartyRole	951	int	2=Broker_of_Credit 4=Clearing_Firm 12=Executing_Trader 24=Customer_Account 45=Secondary_Account_Number 55=Session_ID 59=Executing_Unit 66=Market_Maker 83=Clearing_Account 94=Away_Market
MDReportID	963	int	
SecurityReportID	964	int	

Field Name	Tag	Type	Allowed Value(s)
TradeID	1003	String	
SideLastQty	1009	int	
MessageEventSource	1011	int	1=Matching_Engine 2=Manual_Trade_Entry 3=Deal_Modification_User 4=Deal_Modification_Exchange 5=Trade_Bust 6=Deal_Bust 7=Deal_Management_Exchange
AllocPositionEffect	1047	char	C=Close O=Open
RefOrderID	1080	String	
DisplayWhen	1083	char	1=Immediate 2=Exhaust
DisplayMethod	1084	char	1=Initial 3=Random
DisplayLowQty	1085	int	
DisplayHighQty	1086	int	
PriceProtectionScope	1092	char	1=Local 2=National
TriggerPrice	1102	float	
OrderCategory	1115	char	1=Order 2=Quote 5=Linked_order 8=Cross_Order A=Auction D=Directed_Order
RootPartyID	1117	String	
RootPartyRole	1119	int	2=Broker_of_Credit 4=Clearing_Firm 12=Executing_Trader 24=Customer_Account 45=Secondary_Account_Number 55=Session_ID 59=Executing_Unit 66=Market_Maker 83=Clearing_Account 94=Away_Market
OrigTradeDate	1125	String	

Field Name	Tag	Type	Allowed Value(s)
DisplayQty	1138	int	
QuoteMsgID	1166	String	
QuoteEntryStatus	1167	int	0=Accepted 5=Rejected 10=Pending 100=Accepted_warning
SecurityTradingEvent	1174	int	
MDSecSizeType	1178	int	2=Total_Open_Size
MDSecSize	1179	int	
ApplID	1180	String	O=Order_status Q=Quote_status M=Market_maker_information T=Trade_item D=Deal_item N=News S=Service_availability
ApplSeqNum	1181	int	
ApplBegSeqNum	1182	int	
ApplEndSeqNum	1183	int	
ProductComplex	1227	int	1=Simple_instrument 2=Standard_combination 3=Stock_combination 4=Equity
MarketSegmentID	1300	String	
MarketID	1301	String	XISX=ISE Exchange GMNI=ISE Gemini Exchange MCRY=ISE Mercury Exchange
RejectText	1328	String	
ListUpdateAction	1324	char	D=Delete
ApplReqType	1347	int	0=Retransmission 1=Subscription 4=Unsubscribe
ApplLastSeqNum	1350	int	
ApplResponseID	1353	String	

Field Name	Tag	Type	Allowed Value(s)
RefApplID	1355	String	O=Order_status Q=Quote_status M=Market_maker_information T=Trade_item D=Deal_item N=News S=Service_availability
RefApplLastSeqNum	1357	int	
MassActionType	1373	int	1=Suspend 2=Release 3=Cancel 100=Suspend and delete quotes
MassActionScope	1374	int	1=Security 8=Market 9=Market_Segment 13=Product_Complex 100=Partition
MassActionResponse	1375	int	0=Rejected 1=Accepted 2=Accepted_Warning
MassActionRejectReason	1376	int	
TradePublishIndicator	1390	int	0=Do_Not_Publish 1=Publish_Trade
DefaultCstmApplVerID	1408	String	Version=007.000.000
SessionStatus	1409	int	0=Active 4=Logout 9=Takeover
ApplReportType	1426	int	1=Last_Message 2=Heartbeat4=Invalid_Data
SideLiquidityInd	1444	int	(*) See Table 24: Liquidity Indicator Values , on page 49
TargetPartyID	1462	String	
NewsID	1472	String	
NewsRefID	1476	String	
NewsRefType	1477	int	0=Replacement 3=Removal
SideTradeID	1506	String	Carried to OCC TrdID field
InitialDisplayQty	1608	int	
ThrottleStatus	1609	int	0=Not_Queued 1=Queued

Field Name	Tag	Type	Allowed Value(s)
ThrottleAction	1611	int	1=Queue_outbound 2=Reject 3=Disconnect
ThrottleType	1612	int	0=Inbound_Rate 1=Outstanding_Requests
ThrottleNoMsgs	1613	int	
ThrottleTimeInterval	1614	int	
ThrottleTimeUnit	1615	int	0=Seconds 1=Seconds_10 2=Seconds_100 3=Seconds_1000
ExposureDuration	1629	int	
RelatedSecurityID	1650	String	
SecurityMassTradingStatus	1679	int	1=Opening_delay 2=Trading_halt 3=Resume 17=Ready_to_trade 18=Not_available 22=Opening_Rotation 23=Fast_Market
ThrottleCountIndicator	1686	int	0=Unchanged 1=Decreased
PartyDetailID	1691		
PartyDetailRole	1693		59=Executing_Unit 55=Session_ID
EntitlementRequestID	1770		
TradeQty	1843	Int	
EntitlementRequestStatus	1882	int	0=Accept 2=Reject
EntitlementStatus	1883	int	0=Accepted
MemberName	5364	String	
PrivateReference	5476	String	
TrdMatchTime	5507	int	
PendingReason	5946	int	
PartitionID	5948	int	

Field Name	Tag	Type	Allowed Value(s)
PartitionStatus	5969	int	0=Unavailable 1=Available
RequestTime	5979	int	
CompletionIndicator	6228	char	N=Not_last_message Y=Last_message
ServiceID	7930	String	ME=Matching_Engine TM=Trade_Manager
Delta	7966	int	
Vega	7968	int	
SessionInstType	8502	int	0=None 1=Cancel
SessionInstScope	8503	int	1=Orders 2=Orders_NP 3=Quotes
RefApplSeqMode (Deprecated)	8514	int	0=Unrecoverable 1=Recoverable
RefApplResponseID	8515	String	
RefClOrdID	8516	String	
ClearingCapacity	8521	int	1=Customer 2=Firm 3=Market_Maker
AuctionType	8522	int	(*) See Table 30: Auction Types , on page 67
AuctionAllocationPct	8523	float	$0 \leq n \leq 0.4$ (default = 40%) 30% (0.3) is entered as 30,000,000
TriggerIndicator	8529	int	0=Not_triggered 1=Triggered
LockType	8531	int	1=Away_Market_Better 2=Three_Tick_Locked 3=Locked_by_MM 4=Directed_Order_Lock 5=Multileg_Lock 6=Market_Order_Lock
LockedQty	8533	int	

Field Name	Tag	Type	Allowed Value(s)
OrdEventType	8535	int	1=Added 2=Modified 3=Deleted 4=Partially_filled 5=Filled 6=Locked 7=Released 8=Restated
OrdEventReason	8536	int	
OrdEventQty	8537	int	
QuoteEventType	8539	int	1=Added_quote_side 2=Removed_quote_side 3=Partially_filled 4=Filled 5=Modify_quote_side 6=Suspended
QuoteEventReason	8540	int	
QuoteEventQty	8542	int	
ReleaseQty	8543	int	
SecondaryLockedQty	8544	int	
BidCxlSize	8547	int	
OfferCxlSize	8548	int	
QuoteEntryRejectAction	8549	int	0=None 1=Canceled
TargetPartyRoleQualifier	8552	int	0=Agency 1=Principal 2=Riskless_Principal 3=General_Clearing_Member 4=Individual_Clearing_Member 5=Preferred_MM 6=Direct_MM
QuoteEventPx	8553	float	
OrdEventPx	8554	float	
MarketMakerEventType	8565	int	1=Assignment_of_Security 2=Deassignment_of_Security
MarketMakerEventReason	8566	int	
ApplReqScope	8569	int	0=All_messages 1=Only_for_user

Field Name	Tag	Type	Allowed Value(s)
RelatedLowPrice	8572	float	
RelatedHighPrice	8573	float	
RelatedPriceSource	8574	int	1=Underlying_NBB 2=Underlying_NBO
LegClearingCapacity	8575	int	1=Customer 2=Firm 3=Market_Maker
DisclosureType	8577	int	1=Volume 2=Price 3=Side 4=AON 5=General 6=Clearing_Account 7=CMTA_Account
DisclosureInst	8578	int	0=No 1=Yes 2=Use_Default
BidTotalSize	8579	int	
OfferTotalSize	8580	int	
QuoteEventSide	8581	char	1=Buy 2=Sell 5=Sell_Short 6=Sell_Short_Exempt
MatchDate	8582	String	
TradeReportText	8583	String	
SideLastPx	8585	float	
OrderSide	8586	char	1=Buy 2=Sell 5=Sell_Short 6=Sell_Short_Exempt
RelatedProductComplex	8587	int	1=Simple_instrument 2=Standard_combination 3=Stock_combination 4=Equity
AllocClearingCapacity	8588	int	1=Customer 2=Firm 3=Market_Maker

Field Name	Tag	Type	Allowed Value(s)
AllocOrderCapacity	8589	char	G=Proprietary C=Customer D=Customer_Professional B=Broker_Dealer N=Away_Market_Maker M=Market_Maker
AuctionInst	8601	int	0=Start_auction 1=Do_not_start_auction
MinQtyType	8602	char	O=Once R=Repeat
ReleaseInst	8603	int	1=ISO 2=No_ABBO_Check
MatchSubType	8610	int	0=NONE (*) 1=BLOCK_ORDER_SIMPLE (*) 2=DIRECTED_ORDER_SIMPLE (*) 3=FLASH_ORDER_SIMPLE (*) 4=EXPOSURE_ORDER_STANDARD (*) 5= EXPOSURE_ORDER_STOCK (*) 6= FACILITATION_ORDER_SIMPLE (*) 7=FACILITATION_ORDER_STANDARD (*) 8=FACILITATION_ORDER_STOCK (*) 9=SOLICITATION_ORDER_SIMPLE (*) 10=SOLICITATION_ORDER_STANDARD (*) 11=SOLICITATION_ORDER_STOCK (*) 12=PRICE_IMPROVEMENT_MECHANISM_SIMPLE (*) 13=PRICE_IMPROVEMENT_MECHANISM_STD (*) 14=PRICE_IMPROVEMENT_MECHANISM_STOCK (*) 15= DIRECTED_ORDER_PIM_SIMPLE (*)
NotAffectedMarketSegmentID	8612	String	
ClearingVenue	8613	String	
NumberOfSecurities	8614	int	
PctCount	8615	int	
QuoteOffsetValue	8616	int	
NewQuoteSize	8617	int	
TargetMarketSegmentID	8618	String	

Field Name	Tag	Type	Allowed Value(s)
OrdEventMatchType	8620	int	3=Confirmed_Trade_Report 4=Auto_match_incoming 5=Cross_Auction 6=Counter_Order_Selection 9=Opening_Auction 10=Auction 11=Auto_match_resting 12=Auto_match_pinging
OrdEventMatchSubType	8621	int	See MatchSubType (tag 8610)
OrdEventLegQty	8622	int	
OrdEventLegLastPx	8623	float	
OrdEventLegLastMkt	8624	String	1=BNY ConvergeX U.S. Transaction Services (*) 3=BNY ConvergeX Millennium ATS (*) 4=Knight Match (*) 5=Knight Link (*) 6=Instinet CBX (US) (*) 7=Deutsche Bank ATS (*) 8=Cheevers (*) 9=Libucki (*) 10=FOG Equities (*) 11=Smart Routing (KCG) (*)
QuoteEventLegQty	8630	int	
QuoteEventLegLastPx	8631	float	
QuoteEventLegLastMkt	8632	String	1=BNY ConvergeX U.S. Transaction Services (*) 3=BNY ConvergeX Millennium ATS (*) 4=Knight Match (*) 5=Knight Link (*) 6=Instinet CBX (US) (*) 7=Deutsche Bank ATS (*) 8=Cheevers (*) 9=Libucki (*) 10=FOG Equities (*) 11=Smart Routing (KCG) (*)
MMReqStatus	8636	int	0=Accepted 1=Accepted_warning 2=Rejected
MMReqResult	8637	int	
QuoteEventLiquidityInd	8641	int	(Not currently used.)
MinQtyExecIndicator	8643	int	0=No Exec 1=Exec
GatewayID	8644	char	

Field Name	Tag	Type	Allowed Value(s)
GatewaySubID	8645	char	
RotateRequestIndicator	8647	char	Y=Rotate N=Do_Not_Rotate
ContraDisplayPrice	8648	float	INTERNAL ISE USE ONLY (defined in isefix.xml)
DealItemID	8701	String	
DealItemQty	8702	Int	
CounterpartyOrderCapacity	8704	char	C=Customer (*) D=Customer Professional (*) B=Broker/Dealer (*) G=Proprietary N=Away Market Maker (*) M=Market Maker (*)
CounterpartyClearingCapacity	8705	int	1=Customer 2=Firm 3=Market Maker
OrdEventLastMkt	9000	String	A = NYSE AMEX B = BOX C = CBOE H = ISE Gemini J= ISE Mercury I = ISE M = Miami N = NYSE ARCA Q = NASDAQ T = NASDAQ BX W = C2 X = NASDAQ PHLX Z = BATS
RoutingInst	9303	int	1=Sweep
BranchSeqNbr	9861	String	

Appendix I: Security Trading Event Codes

The table below enumerates the values seen in the *SecurityTradingEvent* field:

Code	Value
REQUESTED_STATE_CHANGE	100
PRODUCT_STATE_CHANGE	101
AUTOMATED_OPENING	102
FAILED_OPENING_ROTATION	103
NO_ACTING_PMM_ASSIGNED	104
UNDERLYING_FAST	105
UNDERLYING_NOT_OPEN_FOR_TRADING	106
INSTRUMENT_SUSPENDED	107
RESTING_MRKT_ORDERS	108
MM_IS_NOT_ACTING_PMM	109
UNDERLYING_OPEN_FOR_TRADING	110
AUTO_INSTR_STATE_CHG_DISABLED	111
STATE_CHANGE_NOT_ALLOWED	112
ACTING_PMM_ASSIGNED	113
LEG_INSTRUMENT_STATE_CHANGE	114
SIMPLE_INSTRUMENT_FROZEN	115
PENDING	116
INITIAL_STATE_CHANGE	117
INSTRUMENT_DELETE	118
PMM_ROTATE_REQUEST	119
LINKAGE_HANDLER_AVAILABLE	120
LINKAGE_HANDLER_UNAVAILABLE	121
EQUITY_LIMIT_TRADING	122
NO_MM_QUOTES	125

Appendix J: Event Reason Codes

The table below enumerates the values seen in the *ExecRestatementReason*, *OrdEventReason*, *QuoteEventReason*, and *MarketMakerEventReason* fields:

Reason Code	Value
NO_SPECIFIC_REASON	100
ADD_ORDER_REQUEST	101
MODIFY_ORDER_REQUEST	102
DELETE_ORDER_REQUEST	103
DELETE_ALL_ORDERS_REQUEST	104
IOC	105
THREE_TICK_RULE	106
BOOK_MARKET_ORDER_CONVERSION	107
BOOK_ORDER_MATCH	108
ADD_QUOTE_SIDE_REQUEST	109
MODIFY_QUOTE_SIDE_REQUEST	110
DELETE_QUOTE_SIDE_REQUEST	111
DELETE_ALL_QUOTES_REQUEST	112
QUOTE_SIDE_REJECTION	113
TICK_WORSE	114
CHANGE_INSTRUMENT_STATE	122
CHANGE_PRODUCT_STATE	123
LINKAGE_HANDLING_OFF	124
LINKAGE_HANDLING_ON	125
REFERENCE_DATA_LOAD	126
IMAGE_START_UP	127
IMAGE_SHUT_DOWN	128
MARKET_ORDER_VS_MARKET_ORDER	129
INCOMING_REQUEST	130
MOPS_QUOTE_DELETION_REQUEST	131
MOPS_QUOTE_INACTIVATION_REQUEST	132
MOPS_ORDER_DELETION_REQUEST	133
QUOTE_SIDES_CROSSING	134
AUTOMATIC_PMM_CONTROL	135
MARKET_OPERATIONS_PMM_CONTROL	136
AMB_LOCK_LOCKED	137
AMB_LOCK_RELEASED	138
THREE_TICK_LOCK_LOCKED	139
THREE_TICK_LOCK_RELEASED	140
PMM_INITIATED_LOCK_LOCKED	141

Reason Code	Value
PMM_INITIATED_LOCK_RELEASED	142
MARKET_ORDER_LOCK_LOCKED	143
MARKET_ORDER_LOCK_RELEASED	144
START_OF_DAY_PROCESSING	145
END_OF_DAY_PROCESSING	146
SERIES_EXPIRATION	147
ORDER_EXPIRATION	148
AWAY_BBO_UPDATE	149
UNDERLYING_BBO_UPDATE	150
AUCTION_ORDER_ENTRY	151
AUTOMATIC_AUCTION	152
AUCTION_PRICE_IMPROVED	153
AUCTION_QUANTITY_IMPROVED	154
AUCTION_TIMER_EXPIRED	155
EARLY_TERMINATION	156
AUTO_ORDER_RELEASE_TIMER_EXPIRED	157
MISSING_MM_QUOTE_TIMER_EXPIRED	158
AUTOMATIC_OPENING_TIMER_EXPIRED	161
AUTOMATIC_OPENING_RECHECK_TIMER_EXPIRED	162
OB_IMBALANCE_TIMER_EXPIRED	163
THREE_TICK_LOCK_TIMER_EXPIRED	164
CREATE_NEW_ID_NUMBER	165
ADD_COMPLEX_INSTRUMENT_REQUEST	166
DELETE_COMPLEX_INSTRUMENT_REQUEST	167
AUCTION_RESPONSE_IMPROVED	168
ADD_CROSS_REQUEST	169
STOP_ORDER_TRIGGERING	170
RESERVE_ORDER_REFRESH	171
MARKET_MAKER_PROTECTION	172
QUOTE_INACTIVATION	173
QUOTE_REACTIVATION	174
REFERENCE_DATA_UPDATE	175
IBBO_UPDATE	176
DELETED_DUE_TO_AMB	177
REFERENCE_DATA_UPDATE_ADD	178
REFERENCE_DATA_UPDATE_CHANGE	179
REFERENCE_DATA_UPDATE_DELETE	180
INITIAL_RESPONSE_REJECTED	183
MARKET_MAKER_PARAMETER_UPDATE_REQUEST	184

Reason Code	Value
MOPS_MARKET_MAKER_PARAMETER_UPDATE_REQUEST	185
TIMER_EXPIRED	186
SPREAD_PRICE_PROTECTION	187
IMAGE_FAILOVER	188
TIER_RUN_DOWN	189
INTERNAL_PROCESSING	190
POSTPONED_INCOMING_REQUEST	191
POSTPONED_TIMER_EXPIRY	192
POSTPONED_INTERNAL_PROCESSING	193
POSTPONED_AUTO_INSTRUMENT_STATE_CHANGE	194
POSTPONED_MM_PROTECTION	195
POSTPONED_REF_DATA_UPDATE	196
INCOMING_MOPS_REQUEST	197
INCOMING_SEM_REQUEST	198
INSTRUMENT_DELETION	199
MEQ_ORDER_TRIGGERING	200
DIRECTED_ORDER_LOCK_LOCKED	201
DIRECTED_ORDER_LOCK_RELEASED	202
DIRECTED_ORDER_LOCK_TIMER_EXPIRED	203
DIRECTED_ORDER_PIM_STARTED	204
GUARANTEED_DIRECTED_ORDER_CREATED	205
GUARANTEED_DIRECTED_ORDER_DELETED	206
INITIAL_RESPONSE_WAIT_TIME_ELAPSED	207
ORDER_FREEZE_FOR_STOCK_MATCH	210
QUOTE_FREEZE_FOR_STOCK_MATCH	211
SIMPLE_INSTRUMENT_FREEZE_FOR_STOCK_MATCH	212
SYNTHETICSORRESPONSETIMER_TIME_OUT	213
COMBOTOCOMBOSORRESPONSETIMER_TIME_OUT	214
INVALID_STOCK_ORDER_RESPONSE	215
INVALID_CROSSING_STOCK_ORDER_RESPONSE	216
BEST_RESPONSE_CHANGE	217
ORDER_UNFREEZE	218
QUOTE_UNFREEZE	219
SIMPLE_INSTRUMENT_UNFREEZE	220
IOC_UPC_COND_NOT_SATISFIED	221
RECOVERY_INITIATED	222
RECOVERY_COMPLETED	223
RECOVERY_RESPONSE_TIMER_EXPIRED	224
END_OF_DAY_PROCESSING_INITIATED	225

Reason Code	Value
END_OF_DAY_PROCESSING_COMPLETED	226
LEGGING_RETRY_TIMER_EXPIRED	227
LEGGING_RETRY_TIMER_CANCELLED	228
BEST_EFFORT_PROCESSING	229
SYNTH_SOR_RESPONSE_TIMER_EXPIRED	231
SYNTH_SOR_RESPONSE_TIMER_CANCELLED	232
CMB2CMB_SOR_RESPONSE_TIMER_EXPIRED	233
CMB2CMB_SOR_RESPONSE_TIMER_CANCELLED	234
MAX_INIT_RESPONSE_WAIT_TIMER_EXPIRED	235
MAX_INIT_RESPONSE_WAIT_TIMER_CANCELLED	236
RECOV_EXCEEDS_MAXIMUM_QTY	237
RECOV_INVALID_LIMIT_PRICE	238
RECOV_NONEXISTENT_OWNING_BUID	239
RECOV_NOT_PERSISTENT	240
SOR_ORDER_BROADCAST	241
SOR_ORDER_BROADCAST_ACK	242
SOR_ORDER_BROADCAST_NAK	243
SOR_TRADE_REQUEST	244
SOR_NO_TRADE_REQUEST	245
SOR_TRADE_ACK	246
SOR_TRADE_NAK	247
SOR_TRADE_EVENT	248
SOR_TRADE_FAILURE	249
SOR_CROSS_ORDER_BROADCAST	251
SOR_CROSS_ORDER_BROADCAST_ACK	252
SOR_CROSS_ORDER_BROADCAST_NAK	253
SOR_CROSS_TRADE_REQUEST	254
SOR_NO_CROSS_TRADE_REQUEST	255
SOR_CROSS_TRADE_ACK	256
SOR_CROSS_TRADE_NAK	257
SOR_CROSS_TRADE_EVENT	258
SOR_CROSS_TRADE_FAILURE	259
DELETE_ALL_NON_PERSISTENT_ORDERS_SEM_REQUEST	261
DELETE_ALL_ORDERS_SEM_REQUEST	262
QUOTE_ACTIVATION_SEM_REQUEST	263
FAILED_MATCH_ATTEMPT	264
SERVICE_AVAILABILITY_RESPONSE	265
SERVICE_STATE_CHANGE	266
DELETE_ALL_QUOTES_SEM_REQUEST	267

Reason Code	Value
RECOV_NONEXISTENT_MPID	268
IMMEDIATE_MATCHING_OF_AUCTION_ORDER	269
AWAY_MARKET_BETTER	270
RECOV_STOCK_COMBO_NOT_ALLOWED_FOR_BUID	271
LOCKED_ORDER_ALERT_TIMER_EXPIRED	272
MEMBER_KILL_SWITCH_REQUEST	273
MEMBER_KILL_SWITCH_MOPS_REQUEST	274
USER_BLOCKED	275
ALO_REPRICE	276
ALO_DELETE	277
EQUITY_LIMIT_STATE_IN_EFFECT	282
MOPS_LOCKED_QUOTE_INACTIVATION_REQUEST	286
TM_REPLACEMENT_RULE_EXECUTION_SUCCESS	1101
TM_MATCH_EVENT_ENRICHMENT	1102
TM_USER_DEALITEM_MAINT_INSTRUCTION	1103
TM_MOPS_DEALITEM_MAINT_INSTRUCTION	1104
TM_MOPS_DEAL_MAINT_INSTRUCTION	1105
TM_MOPS_TRADE_BUST	1106
TM_MOPS_DEAL_BUST	1107
TM_MOPS_ENTER_TRADE	1108
TM_BUSINESS_EVENT_MESSAGE	1109

Appendix K: Throttles

The DTI is asynchronous. A client application can send multiple messages without waiting for a response. However, the number of messages allowed within a given time frame, as well as the number of outstanding messages (messages awaiting a response), is controlled by the use of throttles. The throttle values are configured by the ISE for each Username, and are provided to the client application in the **LogonResp** message.

The throttles are upper limits and do not guarantee throughput rates. As load fluctuates in the exchange system, actual throughput rates can vary.

The throttle mechanism uses three components:

- Transaction Limit
- Outstanding Message Limit
- Reject/Disconnect Limit

Each component is described below.

Transaction Limit

The Transaction Limit is the maximum number of messages that a client application may send within a configured time interval.

The system allows for very granular configuration of this parameter, and the maximum rate can be specified in seconds (for example, n messages per 1 sec.), tenths of seconds (n messages per 0.1 sec.), hundredths of seconds (n messages per 0.01 sec.), or milliseconds (n messages per 0.001.). The interval may also be specified in multiples of the time unit; for example, n messages per 5 seconds, etc.

If the number of messages exceeds the configured rate, the DTI delays delivery of ACKs to the client application until the time interval has expired. Broadcast messages are never delayed.

Required heartbeats do not count against the transaction limit.

For example, a Transaction Limit of 100 messages/sec. could be represented in the **LogonResp**, *ThrottleParamsGrp* group as:

ThrottleAction (tag 1611) = 1 (Queue Outbound)

ThrottleType (tag 1612) = 0 (Inbound Rate)

ThrottleNoMsgs (tag 1613) = 100

ThrottleTimeInterval (tag 1614) = 1000

ThrottleTimeUnit (tag 1615) = 3 (Seconds/1000)

Note that the same time interval could also be represented as:

ThrottleTimeInterval (tag 1614) = 1

ThrottleTimeUnit (tag 1615) = 0 (Seconds)

It is possible, and easy, to use up the transaction limit early in the time interval; therefore, Members should smooth out their message load to spread messages evenly across the configured interval and not exceed their transaction limit.

Outstanding Message Limit

The maximum number of messages sent for which the client application has not yet received a response.

A client application must count the number of outstanding requests and, when it reaches the Outstanding Message Limit, must wait for an ACK before sending another message; otherwise, the next message sent is rejected by the DTI. If too many requests are rejected due to exceeding the request limit, the session is terminated.

- Heartbeat messages do not count as request messages. If the client is waiting for a response, it must still send a heartbeat message every heartbeat interval to maintain the session.
- Some request messages can receive multiple responses. In this case, the first response is counted. The Response Header contains the *ThrottleCountIndicator* field. If the *ThrottleCountIndicator* is set to "N," the response does not count in the reduction of the open message count.
- Broadcast messages do not reduce the count of open requests.

Example of throttle values from a Logon response

An Outstanding Request Limit of five messages is represented in the **LogonResp.ThrottleParamsGrp** as:

ThrottleAction (tag 1611) = 2 (Reject)
ThrottleType (tag 1612) = 2 (Outstanding Requests)
ThrottleNoMsgs (tag 1613) = 5

Reject/Disconnect Limit

The maximum number of sequential message rejects allowed by the DTI.

Once an ACK has been sent, the Reject/Disconnect Limit counter is reset.

If the client application continues to send messages that are rejected for exceeding the Outstanding Message Limit (as described above) and the Reject/Disconnect Limit is exceeded, the DTI disconnects the session.

For example, a Reject Limit of 50 is represented in the **LogonResp.ThrottleParamsGrp** as:

ThrottleAction (tag 1611) = 3 (Disconnect)
ThrottleType (tag 1612) = 2 (Outstanding Requests)
ThrottleNoMsgs (tag 1613) = 50

Summary

An application can send messages without waiting for ACKs until it reaches its outstanding message limit. At that point, it must wait for an ACK that reduces the number of outstanding messages before sending another message. If messages are sent faster than the throttle rate, acknowledgments are delayed.

For example, when an order is submitted, the system responds with an acknowledgment message. If subscribed to, the system also responds with a **POB** message. The **POB** contains information about how the order was processed, including any trades that were executed, and the final state of the order. The ACK is used for flow control. The **POB** is the business response to the order. Normally, the latency measured between the order and the ACK or the **POB** is always as small as possible. However, if a client application exceeds the configured transaction limit, the **POB** is sent back immediately, but the ACK is delayed.

Appendix L: Locked Order Handling

Under existing ISE rules, a PMM must handle an order that locks with a Three-Tick or Market Order lock (see **Section 4.5.8, Order Locks**, on page 45).

Processing a Locked Order

A PMM is notified of a locked order via the **Private MM Execution Broadcast** (see **Section 5.25, Locked Order Broadcast**, on page 142). A Three-Tick or Market Order lock generally indicates a lack of liquidity in the market for the locked order. Upon receipt of the locked order broadcast, the PMM has a number of options:

- Add quantity to the book on the other side of the order (to create a market for the order) and release the order.
- "Step-up" to (trade directly with) the order for some or all of the quantity.
- Simply release the order.

Executing the Locked Order

Upon receipt of the locked order, the PMM may elect to trade directly with (step-up to) the locked order using the **New Order Single** message (see **Section 5.2, New Order Single Request**, on page 85). Specific fields must be completed as follows (fields not specified here should be populated as required for the message):

Tag	Field	Value(s)
1080	RefOrderID	ISE order number of locked order
1300	MarketSegmentID	Product ID
48	SecurityID	Instrument ID
40	OrdType	Q=Counter-order selection
38	OrderQty	Quantity to execute
44	Price	Price to execute (cannot be worse than the BBO)
54	Side	Opposite side of locked order
59	TimeInForce	3=Immediate or Cancel
77	PositionEffect	O=Open

Releasing the Locked Order

Unless the PMM steps up to the entire order quantity, the PMM must release the locked order back to the order book. The order is released using the **Order Release Request** (see **Section 5.14, Order Release Request**, on page 114). Specific fields must be completed as follows:

Tag	Field	Value(s)
37	OrderID	ISE order number of locked order
1300	MarketSegmentID	Product ID
48	SecurityID	Instrument ID

Tag	Field	Value(s)
18	ExecInst	q=Release from suspension
8603	ReleaseInst	As indicated on the locked order broadcast

NOTE: When a locked order is successfully released, the PMM will receive both the **Order Release Response** and the **Private MM Execution Report Broadcast** confirming the lock release.

Auto-Release of Locked Quantity

If the PMM loses acting status in an instrument while working a locked order for that instrument, the system automatically releases the order after a configurable amount of time (currently five seconds), unless the PMM regains acting status within that time.

If the system auto-releases a locked order, a **Private MM Execution Report** is sent to the holder of the lock informing them that the order has been released.

Appendix M: Standard Strategies

Most of the combo order flow in the ISE Market is concentrated in a few types of strategies. These strategies are defined as “standard strategies” at ISE and are listed in the following tables. Standard options strategies have exactly two option legs in the ratio 1:1 and no stock leg (i.e. only standard combinations).

Standard strategies can be classified as debit or credit strategies. On creation, credit strategies are inverted to their debit strategy equivalent.

Table 83: Standard Options Strategies

Strategy Name	Description	Debit/ Credit	Inverted (Y/N)
Bull Call Vertical Spread	Buy Low Strike Call, Sell High Strike Call, Same Expiry	Debit	N
Bear Call Vertical Spread	Buy High Strike Call, Sell Low Strike Call, Same Expiry	Credit	Y
Bear Put Vertical Spread	Buy High Strike Put, Sell Low Strike Put, Same Expiry	Debit	N
Bull Put Vertical Spread	Buy Low Strike Put, Sell High Strike Put, Same Expiry	Credit	Y
Long Call Calendar Spread	Buy Far Expiry Call, Sell Near Expiry Call, Same Strike	Debit	N
Short Call Calendar Spread	Buy Near Expiry Call, Sell Far Expiry Call, Same Strike	Credit	Y
Long Put Calendar Spread	Buy Far Expiry Put, Sell Near Expiry Put, Same Strike	Debit	N
Short Put Calendar Spread	Buy Near Expiry Put, Sell Far Expiry Put, Same Strike	Credit	Y
Long Straddle	Buy Call, Buy Put, Same Strike, Same Expiry	Debit	N
Short Straddle	Sell Call, Sell Put, Same Strike, Same Expiry	Credit	Y
Long Strangle	Buy Call, Buy Put, Different Strike, Same Expiry	Debit	N
Short Strangle	Sell Call, Sell Put, Different Strike, Same Expiry	Credit	Y

Table 84: Standard Stock-Combo Strategies

Strategy Name	Description	Debit/ Credit	Inverted (Y/N)
Buy/Write	Buy Stock, Sell Call	Debit	N
Synthetic Long Put	Sell Stock, Buy Call	Credit	Y
Protective Put	Buy Stock, Buy Put	Debit	N
Covered Put	Sell Stock, Sell Put	Credit	Y
Combo 1	Buy Stock with Bull Call Vertical Spread	Debit	N
Combo 2	Sell Stock with Bear Call Vertical Spread	Credit	Y
Combo 3	Buy Stock with Bear Put Vertical Spread	Debit	N
Combo 4	Sell Stock with Bull Put Vertical Spread	Credit	Y
Combo 5	Buy Stock with Long Call Calendar Spread	Debit	N
Combo 6	Sell Stock with Short Call Calendar Spread	Credit	Y
Combo 7	Buy Stock with Long Put Calendar Spread	Debit	N
Combo 8	Sell Stock with Short Put Calendar Spread	Credit	Y

Strategy Name	Description	Debit/ Credit	Inverted (Y/N)
Combo 9	Buy Stock with Long Straddle	Debit	N
Combo 10	Sell Stock with Short Straddle	Credit	Y
Combo 11	Buy Stock with Long Strangle	Debit	N
Combo 12	Sell Stock with Short Strangle	Credit	Y

Strategy Examples

Example 1:

Request to create the instrument: Buy 1 May XYZ 45 C, Sell 1 May XYZ 50 C

Action: The instrument is a debit strategy and is created as is.

Example 2:

Request to create the instrument: Sell 1 May XYZ 45 C, Buy 1 May XYZ 50 C

Action: The instrument is a credit strategy and is inverted as:

Buy 1 May XYZ 45 C, Sell 1 May XYZ 50 C

Unconventional Strategies

Unconventional strategies, such as orders in which both sides are buy or both are sell, for example, can also be employed. Orders in unconventional strategies cannot leg into the simple order book. The following list presents the common unconventional strategies.

- Buy/Buy, in which both sides are to buy
- Sell/Sell, in which both sides are to sell
- Call/Call, in which both sides are calls
- Put/Put, in which both sides are puts
- Complex instruments with 3 or more option legs where all legs are buying or all legs are selling

Appendix N: Away Market Codes

The table below enumerates the values used for the *PartyID* field when the *PartyRole* field is set to 94 (Away Market) on the **New Order Single** for linkage handling. These values are also carried to the resulting **POBs**, **DIBs**, and **TIBs**.

Table 85: Away Market Codes

OPRA Code	Away Exchange
A	NYSE AMEX
B	BOX
C	CBOE
H	ISE Gemini
I	ISE
J	ISE Mercury
M	Miami
N	NYSE ARCA
Q	NASDAQ
T	NASDAQ BX
W	C2
X	NASDAQ PHLX
Z	BATS

Away market codes are used for the *PartyID* field in these messages:

- **New Order Single**, on page 86
- **Private Order Broadcast**, on page 124
- **Deal Item Broadcast**, on page 136
- **Trade Item Broadcast**, on page 139

Appendix O: Unicast IP Addresses

The following IP addresses are used to access the trading system production environment.

Production Unicast IP Addresses and Ports

Table 86: Production Unicast IP Addresses and Ports

Env	IP Address	Interface	Port(s)
Primary Equinix NY4	207.231.197.5	VIP for DTI Interface (All Members)	20011 / 20012 NOTE: DTI sessions should be split between ports.
	207.231.197.6	VIP for Prem. Access Connection Gateway	20021
	207.231.196.64/27 207.231.196.96/27	VIP subnet for Prem. Access DTI interface	20031 / 20032
	207.231.197.37	VIP for PrecISE Trade Interface	SSL 443
	74.120.84.118 or precise.iseoptions.com	PrecISE Trade via the Internet	SSL 443
	207.231.197.38	VIP for FIX Interface	Member-specific
	207.231.197.39	Member FTP VIP	N/A
	74.120.84.126	VPN Peer1-IPSEC Primary	N/A
Backup / DR TelX	207.231.197.133	VIP for DTI Interface (All Members)	20011 / 20012
	Premium Access is not available at TelX		
	207.231.197.164	VIP for PrecISE Trade	SSL 443
	74.120.84.250 or precise.iseoptions.com	PrecISE Trade via the Internet	SSL 443
	207.231.197.165	VIP for FIX	Member-specific
	207.231.197.39	Member FTP VIP	N/A
	74.120.84.254	VPN Peer2-IPSEC Secondary	N/A

NOTE: Please make sure your DTI sessions are split and balanced between ports. Connectivity to the DTI is through a load balancer with a single VIP (Versatile Interface Processor) with two TCP/IP ports. Load balancing DTI sessions across both ports ensures that the ISE infrastructure load is optimal.

IP Addresses for Multicast Services

Multicast IP Addresses are documented in the *MDI Programming Guide*, available on the Members Area site (<https://members.ise.com>).

Member Test IP Addresses and Ports

Connectivity information, both uni- and multicast, for the member test environments (MT1 and MT2) is documented in the **Member Simulation Guide**, available on the Members Area site (<https://members.ise.com>).

Additional Information

Member Website

In order to ensure market participants are informed in a timely manner about the detailed changes that come with the ISE T7 system, more information, including (but not limited to) FAQs and other documentation, is available on the member website at <https://members.ise.com>.

Contact Information

Questions	ISE Contacts		
General Release Questions	Technology Member Services	(212) 897-0244	tms@ise.com
Connectivity Questions	Joseph Maresca	(212) 897-8158	jmaresca@ise.com
Functional and Technical Questions	Technology Member Services	(212) 897-0244	tms@ise.com
Business Questions	Jeanine Hightower	(212) 897-0357	jhightower@ise.com
Market Operations Questions	Daniel Amar	(212) 897-8146	damar@ise.com

Document Revision Table

Version	Date	Changes
1.0	May 10, 2010	Initial Version
1.0.1	July 20, 2010	Please see version 1.0.1 for revision details.
1.0.2	August 16, 2010	Please see version 1.0.2 for revision details.
1.0.3	September 9, 2010	Please see version 1.0.3 for revision details.
1.0.4	September 17, 2010	Please see revision 1.0.4 for revision details.
1.0.5	November 2010	Major review. Please see revision 1.0.5 for revision details.
1.0.6	December 2010	Major review. Please see revision 1.0.6 for revision details.
1.1.0	January 2011	Member Simulation Release version 1.1 page design updates and new content: <ul style="list-style-type: none"> ▪ 4.1.12 ExecID on page 85 ▪ 4.1.13 TradeReportID on page 85 ▪ 4.29 Trading Session Marker Message Broadcast on page 145
1.1.1	January 2011	Updated message tables to conform to latest ISE message definitions. Updated field descriptions in Appendix F and Appendix G to conform with latest ISE message field standards: SecurityTradingEvent (Tag 1174) – added values TradePublishIndicator (Tag 1390) – added value ApplNewSeqNum (Tag 1399) – added this field, used in Message 10015 (Table 63: End of Stream Broadcast) SecurityResponseID (Tag 322) – added this field, used in message 10301 (Table 41: Change Instrument State Response)
1.2.0	February 2011	Member Simulation Release version 1.2 page new content to update messages and miscellaneous text.
1.2.1 1.2.2 1.2.3 1.2.4	February 21, 2011	Miscellaneous minor document corrections. New appendices F and G. Original F and G (and subsequent appendices) pushed down alphabetically.
1.2.5	March 7, 2011	<i>MassActionRejectReason</i> numeric ID changed from 8635 to 1376. (Delete All Orders Response and Quote Action Report messages.) Miscellaneous document corrections.
1.2.8	April 2011	Updated Appendix K: Event Reason Codes: values 186-199 added, values 267-271 added. Explained three-tick rule in Section 3.10.3.
1.2.9	April 2011	Removed STOCK_TRADE_REJECTION (value 121) from Appendix K: Event Reason Codes Added MinQtyExecIndicator to Private Order Broadcast Message description Clarified AuctionType (tag 8522) values through the document
1.3.0	May 2010	Miscellaneous clarifications and corrections to include document with Optimise 1.3.0 release
1.3.1	May 2010	Updated RootParties 14031 to Optional (N) from Mandatory
1.4.0	June 2011	Updated Side tags for release of version 1.4 in MT2 environment

Version	Date	Changes
1.5.0	August 2011	Release Version update New Section 3.4.8 – Order Locks Appendix L – additional example provided
1.5.1	September 2011	Removed references to OM Click, the legacy trading application
1.6.0	September 2011	Release Version update. Combo instruments allow up-to eight option legs.
1.6.1	September 2011	Sect. 3.5.7: Identifying and Cross-referencing trade-related broadcasts – corrected Deal/Trade Item B'cast descriptions. Sect. 4.23: Trade Item Broadcast – updated to reflect disclosure of counter-party. Appendix B: ISE Parameters table updated.
1.6.2	October 2011	Minor updates to connectivity tables
2.0	October 2011	Updated message tables to conform with latest ISE message definitions: Table 30: Delete All Orders Request Table 31: Delete All Orders Response Table 39: Quote Activation Request Table 40: Quote Activation Response Table 51: MM Parameter Definition Request Table 56: Private Quote Broadcast Section 3.3.2. Clarified MM percentages when trading against the book. Section 3.10.7. Updated description of complex trade allocation. Appendix O: Added recommendation to balance DTI sessions across two ports.
2.0.1	November 2011	Private Market Maker Broadcast is no longer supported. Section 4.12: The Change Instrument State Request is not subject to throttling.
2.0.2	November 2011	Section 2.5.3. A new method for establishing a DTI connection, Premium Access Connectivity, is available. Two new application messages defined. Section 3.9.4, third bullet, the time interval start time was changed from 2 seconds to 1 second. T-cross timer is turned off by default.
3.0	January 2012	Fast Order Status broadcasts are now recoverable via sequence number. See the following sections: Section 3.10, Changes in POB behavior; Note that <i>RefApp/SeqMode</i> field is deprecated. Section 5.18, Description of fields used to detect sequence number gaps. Section 5.20, Application Sequence Number Reset broadcast available for Order Status and Market Maker Information message streams. Sections 5.2, 5.3, Added ALO support to New Order Single and Cancel Replace requests. Section 5.24, Trade Item Broadcast now provides counterparty clearing information. Appendix A rewritten
3.0.1	January 2012	Minor edits.
3.0.2	February, 2012	Flash timer reduced to 500 ms Heartbeat message added to message descriptions, Section 4.33 New section 4.7.6 Add Liquidity Only Orders (ALO) Add'l minor edits.
3.0.3	February 2012	Added 2 ports (20031 and 20032) to Table 80 for DTI Premium Access gateway connection.

Version	Date	Changes
3.0.4	February 2012	App. L: AMB-Lock Order Handling. Clarification of auto-release behavior. Misc. edits and corrections.
3.0.5	March 2012	App. B: Flash timer reduced to 150 ms. Misc. edits and corrections.
3.1	April 2012	Added Member Kill Switch Request and Response: Table 59 and Table 60. Added additional stock execution venues to the OrdEventLegLastMkt field in the Private Order Broadcast.
3.1.1	May 2012	Sect. 3.9.2.1 — Corrected tag ID for Text field, table 12. Sect. 4.6.7 — Deal/Trade adjustment clarification. Sect. 4.6.8 — new Table 26: Mapping status broadcasts to deal/trade broadcasts (complex instruments). Sect. 4.11 — corrected definition of Stock Combo instrument. Sect. 5.12.2 — removed unused field SecurityResponseID (tag 322), table 45. Sect. 5.18.2 — corrected format/definition of message, table 59. Sect. 5.30 — section renamed and behavior clarified. Appendix B: — BBO Holdback timer removed. Appendix J: — new reason codes 272, 276, and 277. Add'l minor edits and corrections.
3.1.2	May 30, 2012	<i>LastMkt</i> codes updated Normalized & standardized acronym usage throughout doc. About This Doc. — New acronym table Section 4.7.1 — Rewritten to cover all minimum qty order types: AON, FOK, and MEQ. Section 4.10 — Updated for new Trading Rights model. Section 5.9.3 — New section. Includes new Table 42: Quote Reject Reasons. (All following tables renumbered.) Section 5.23 — DIB Clarification of Manual Trade behavior. Section 5.24 — TIB Clarification of Manual Trade behavior. Add'l minor edits and corrections.
3.1.3	June 11, 2012	Appendix N: Away Market Codes — OPRA codes added for new Miami and NASDAQ BX exchanges.
4.0	June 28, 2012	Section 4.9.1.3 — All orders now eligible for flash auction Section 4.11.9.6 — New section: No Trade Through Section 5.6.1 — New field, Table 37 Section 5.7.1 — New field, Table 38 Section 5.16 — new parameter; new field, Table 55
4.0.1	July 19, 2012	ALO acronym corrected. Add'l minor edits & corrections.

Version	Date	Changes
5.0.0	Oct. 12, 2012	Release version update. Explicit support for multiple exchanges. New MarketID value defined — “XTPZ”. Section 2.5.2 — MarketID field added, PartitionStatus field (unused) deleted, Table 6. Section 2.10.1 — Deprecated field, RefApplSeqMode, removed, Table 15. Section 2.10.5.1 — MarketID field added, Table 19. New section 4.1.6 — MarketID field description/usage. Appendix N — New OPRA code for 2 nd exchange. Add'l minor edits & corrections.
5.1.0	Dec. 18, 2012	Release version update. Removed references to HOLDRS products. New Section 3.9.2.1 Notable Reject Messages New section 3.10.5.3 Retransmission Request Examples Add'l minor edits & corrections.
5.1.1	Jan. 18, 2013	Table 22 — Missing Partition group added to Retransmit Response Appendix A: — Corrected <i>MemberName</i> field in Logon example. Add'l minor edits & corrections.
5.1.2	Feb. 7, 2013	New LastMkt value 7=Deutsche Bank ATS XML Template version updated to 5.1.24. Mini options added to Glossary. Add'l minor edits & corrections.
5.1.3	Mar. 13, 2013	New field, <i>ContraDisplayPrice</i> —INTERNAL USE ONLY. Updated order minimum quantity requirements for mini options. Add'l minor edits & corrections.
6.0.0	Apr. 9, 2013	New release version. Appendix B: ISE Parameters — Updated auction response times. New section 4.6.4: Liquidity (Maker/Taker) Codes. New section 4.9.3: Auction Step-Up Price to clarify functionality. Section 4.10.4 — Speed bump (curtailment) parameters required , and behavior clarified. New Section 4.10.4.1 Curtailment Examples. Replaces “Percentage Count Speed Bump Example” table (removed, with following table #s adjusted) Table 67 — New TIB field: <i>SideTradeID</i> . Appendix M: Standard Strategies — added new table for standard stock-combos Add'l minor edits & corrections.
6.0.1	May 2, 2013	New <i>MarketID</i> value defined — “GMNI” (“XTPZ” removed)
6.0.2	May 8, 2013	Ignore Away Market (IAM) functionality removed. Section 4.9.1.2 — Clarification of Exposure auction behavior. Inserted new Section 5.1.1 clarifying timestamp precision.
6.0.3	May 14, 2013	Section 5.30 — Clarification of User Notification broadcast behavior. Add'l minor edits & corrections.

Version	Date	Changes
6.0.4	July 5, 2013	Rebrand to T7 New Section 1.2— Intended Audience New Section 1.3— Related Documents Section 5.12 — Corrected <i>InstrmtGrp</i> tag number. Appendix L: — Rewritten as gen'l PMM locked-order handling. No AMB lock handling. Add'l minor edits & corrections.
6.0.5	Aug. 1, 2013	Addition of new stock execution venue, Cheevers (<i>LastMkt</i> = 8).
6.0.6	Aug. 16, 2013	Section 3.3.2.3 & Appendix A: — Clarified distinction between Structure Groups and Array Groups. Section 4.10 — Updated CMM quoting requirements (rule change). Section 5.18.1 — Corrected <i>PartyDetailGrp</i> ID on Member Kill Switch Request .
7.0.0	Sep. 25, 2013	New release version. “About This Document” and “Introduction” sections reorganized. New section number (1) added to “About This Document” (all following main section numbers adjusted accordingly — old Sect. 1 → Sect. 2, etc.). Sect. 3.5.1 — Corrected minimum heartbeat interval. Appendix B: — Additional parameters added to table. Appendix J: — Deleted obsolete T-cross timer codes. New stock execution venues, Libucki (<i>LastMkt</i> = 9), FOG Equities (<i>LastMkt</i> = 10). Add'l minor edits & corrections.
7.0.1	Oct. 11, 2013	Section 3.5.1 — Clarified heartbeat interval
8.0.0	Nov. 16, 2013	New release version. Section 5.22, 5.20 — Updated Private Quote and Order Broadcast message structure. Section 5.17 — Updated minimum exposure duration in MM Parameter Definition Request. Section 5.9.2 — Updated Mass Quote Response Message Structure. Section 4.6.4 — Added new liquidity codes for auctions.
8.0.1	Jan 3, 2014	Added QuoteCondition field to Order Book Inquiry Response Message.
8.0.2	Jan 10, 2014	Updated Appendix J with Reason Code 286. Table 41: Mass Quote Acknowledgement (MsgID=10204), changed QuoteInactiveStatus to not required. Added description for Market Wide Speed Bump (MWSB)
9.0.0	April 28, 2014	New Release Version.
9.0.1	April 30, 2014	Section 5.5 : <i>Delete All Order Request</i> for complex and complex with stock instruments with Market scope not subject to throttles.
9.0.2	July 7, 2014	New stock execution venue, Smart Routing (KCG) (<i>LastMkt</i> = 11). Additional new liquidity codes for auctions.
9.0.3	July 28, 2014	Section 4.9.2.2: Update PIM auction functionality to align with new PIM rules.

10.0.0	July 28, 2014	New release version. Section 4.7.7: Added new Sweep Order. Section 4.10.3: Added new QuoteMassAction type in Quote Action Request that allows PMMs to inactivate and delete quotes. Section 4.5.9: GTC/GTD orders are now persistent, and are rejected if not marked persistent. Section 4.5.8: Non-customer orders are now handled by linkage.
10.0.1	August 12, 2014	Section 4.6.4: Maker Taker Liquidity Codes for sweep orders. Appendix B: Updated definition of MinimumQuoteSize.
10.0.2	September 2, 2014	Order Book Inquiry Request no longer supported.
10.0.3	September 24, 2014	Section 4.10.5, Added Cross Market Wide Speed Bump functionality. Section 4.9.2.2, Updated behavior for ISO PIM auctions. Section 5.25. Added ClearingCapacity field to Locked Order Broadcast.
10.0.4	February 13, 2015	Added ISE Mercury support