International Securities Exchange.

T7™

Market Data Interface (MDI)
Programming Manual

Version: 10.0.4

Date: March 10, 2015

Abstract
This document provides information on business descriptions, programming interfaces and protocols for connecting client applications to ISE's 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.

Table of Contents

Tables

1. About This Document

This document is a programmer's reference guide intended to aid in the development and integration of the market data feeds produced by the ISE T7 trading system. It covers the general business behavior of the market data feeds and the technology standards and techniques employed to provide this service. 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. 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 with the trading 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 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.

This document describes the Market Data Interface (MDI).

1.2 Intended Audience

This document is for programmers, analysts, and IT managers who are developing applications to process market data feeds from the ISE's trading system.

1.3 FIX/FAST-Related Documents

The ISE MDI makes use of the FAST protocol (**F**IX **A**dapted for **St**reaming data). Users without prior knowledge of FAST should review these documents.

NOTE: The documents listed below are not under the ISE's control. As documents may be moved, deleted, or updated, we recommend that you navigate to the main FIX Protocol web site at http://fixprotocol.org to find the latest documents in the document repository directory.

Table 1: FAST Protocol Documentation

Document	Description	Location
FAST Technical Overview	Explains in detail how FAST successfully presents a solution to the problem of spiraling market data volumes.	http://fixprotocol.org/documents/2801/FIX%20Adapted%20for%20STreaming%20-%20FAST%20Protocol.pdf
FAST Protocol Specification v1.1	Defines the structure and semantics of FAST	http://www.fixprotocol.org/documents/3066/FAST%20Specification%201%20x%201.pdf

Document	Description	Location
Transfer Encoding Specification v1.01	Describes the serialization process used to reduce the size of a data stream	http://www.fixprotocol.org/documents/3062/FAST%2520Transfer%2520Encoding%2520Specification%25201.0.2.pdf
Field Encoding Specification v1.0	Describes field-level operations used to reduce redundant information	http://www.fixprotocol.org/documents/3063/FAST%2520Field%2520Encoding%2520Specification%25201.0.pdf
Basic FAST Users Guide	Describes the proper use of the FAST Protocol in a one-way exchange of data	http://fixprotocol.org/documents/2301/A%20Basic%20Guide%20to%20FAST%20v1.0.pdf
FIX Protocol Version 5.0 SP2 Recommended Book Management Practices	FIX Protocol Standard Specification	www.fixprotocol.org/specifications

1.4 ISE Related Documents

The following are documents related to T7.

Table 2: ISE-Related Documentation

Document	Description	Location
Direct Trading Interface (DTI) Programming Manual	Guide to developing trading applications to connect to T7.	https://members.ise.com
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

2. Introduction

2.1 Available Data Feeds

The feeds available over the MDI consist of market data and reference data.

Two formats are available:

- FAST encoded
- Raw, unencoded binary

All feeds are duplicated using an A/B configuration. T7 may be configured for multiple markets, in which case each market will have its own set of feeds.

2.1.1 FAST Encoded Data Feeds

The following data feeds are available in FAST encoded format.

- **Depth Of Market** (Depth) – The Depth feed shows the top five price levels with the aggregated quantity on each level. Customer quantity and Professional Customer quantity are also shown. This feed also contains trading status messages, but not trades.
- **Order** – The Order feed provides information about orders that did not trade on entry and currently rest on the order book. It also provides information about auctions, including Flash orders, solicitations, facilitations, and PIMs.
- **Spread Feed** – The market data feeds described above —Depth of Market, Top Quote, Trade, and Order feeds — are segregated by instrument type — simple or complex — and sent out on different multicast channels. The collection of market data feeds for complex instruments is called the "Spread feed," and the descriptions of the discrete feeds within this document serve for both simple and complex instruments. Any specific differences between the simple and complex feed instances are noted in that feed's description.
- **Reference Data** – The Reference Data feed describes all products (underlyings) and all simple and complex instruments (series). Each instrument is fully described with internal identifier and OSI symbol. Each product also defines its internal identifier and has a full description of its trading rules. In addition, the snapshot for each product lists the multicast channels used by each market data feed for that product.

2.1.2 Binary Data Feeds

The following data feeds are available in raw, unencoded binary format.

- **Depth Of Market** (Depth) – The Depth feed shows the top five price levels with the aggregated quantity on each level. Customer quantity and Professional Customer quantity are also shown. This feed also contains trading status messages, but not trades.
- **Top Quote** – The Top Quote feed contains the same Best Bid/Offer quotes and trades information that is sent to OPRA. Customer quantity and Professional Customer quantity,

which are not part of OPRA, are also shown. This feed also provides trading state information.

- **Trade** – The Trade feed provides trade information for all ISE executed orders.
- **Order** – – The Order feed provides information about orders that did not trade on entry and currently rest on the order book. It also provides information about auctions, including Flash orders, solicitations, facilitations, and PIMs.
- **Spread Feed** – The market data feeds described above —Depth of Market, Top Quote, Pre-Open, Trade, and Order feeds — are segregated by instrument type — simple or complex — and sent out on different multicast channels. The collection of market data feeds for complex instruments is called the "Spread feed," and the descriptions of the discrete feeds within this document serve for both simple and complex instruments. Any specific differences between the simple and complex feed instances are noted in that feed's description.
- **Reference Data** – The Reference Data feed describes all products (underlyings) and all simple and complex instruments (series). Each instrument is fully described with internal identifier and OSI symbol. Each product also defines its internal identifier and has a full description of its trading rules. In addition, the snapshot for each product lists the multicast channels used by each market data feed for that product.

2.2 Overview of the Data Feeds

The ISE market data feeds only provide information about ISE's markets. They do not contain data from other options exchanges. The information provided in the Top Quote feed and the top price-level of the Depth feed contain identical quotes to those provided to OPRA except that "customer quantity" and "professional customer quantity" are not sent to OPRA. The Spread feed provides quote and order data on complex instruments that are not provided to OPRA.

- The ISE market data service is based on industry and technology standards including Financial Information Exchange (FIX) protocol for business level messaging, FIX Adapted for Streaming (FAST) protocol for the FAST encoded feeds, and UDP and IPv4 standards for transmission of all broadcast data.
- The feeds are distributed over multiple multicast channels and the ISE may proactively balance the load across the channels from day to day. In other words, each Symbol for each product is assigned to a set of channels that can change from day to day, but not within a trading day.
- The data feeds are multicast over various networks in an A-feed / B-feed format, similar to OPRA.
- There are no recovery channels, nor is it possible to request a retransmission of missed blocks. If you are late to join the data feed or a packet is lost, you must process a complete cycle of the snapshot messages (as defined for each feed) to ensure that the order book data are accurate. The full refresh or complete rotation of the order book for all instruments takes approximately three minutes.
- FAST messages are defined using the FIX.5.0 SP2 standard for market data, and follow the best practices outlined by the FIX Market Data Working Group. The data are transmitted in the FAST v1.1 encoding method. There are minor deviations from the FIX 5.0 SP2 standards to improve the efficiency of the feeds.

- The binary messages are defined wholly by the ISE. Various message and field names may be similar to the FIX messages — this is purely coincidental, and simply reflects the similarities of data carried in those fields or messages.
- The feeds that are available in both binary and FAST formats carry the same information. Except for formatting, no distinctions are made between the feeds.

For more information about the FIX and FAST Protocols and specifications, please see the FIX Protocol web site: http://fixprotocol.org/specification and http://fixprotocol.org/fast.

2.3 Definitions

Channel: One multicast IP address and port number. For example, 172.2.3.4:200 is one channel.

Stream: A Market data feed can be partitioned over multiple streams depending upon its capacity requirements. For example, the Depth feed may be partitioned over 24 Streams.

Each data stream is sent over two channels in an A/B configuration (like OPRA). For example, the Trade feed is sent over A channel 172.2.3.4:200 and B channel 172.3.3.4:300.

Each market data feed is separated by instrument type: simple instruments and complex instruments. For example, the Top Quote feed may have 16 streams for the simple instruments and another 16 streams for complex instruments. (The actual number of streams is configurable based on capacity requirements.)

Reference data are sent on two separate feeds: a snapshot feed of all products and instruments (simple and complex); and an incremental updates feed which describes products and/or instruments (simple and complex) as they are added or changed intraday.

The IP/Port of each channel for each market data feed is described in the reference data for each product.

2.4 Hours of Operation

Normal trading hours for the ISE markets are from 9:30 a.m. to 4:00 p.m. Eastern Time (ET) for equity options and from 9:30 a.m. to 4:15 p.m. ET for ETF and Index Options.

The ISE MDI is available at approximately 6:00 a.m. and continues to broadcast information throughout the trading day. The service stops broadcasting at approximately 5:45 p.m.

Table 3: Hours of Operation Schedule

Time	Activity
4:00 a.m.	RefData: the snapshot feed provides a complete snapshot of all products and instruments at regular intervals. This feed walks through all products and instruments on a constant basis every one minute. Some new instruments and products can be added or changed prior to the opening of the market and intraday.

Time	Activity
6:00 a.m.	All other data feeds begin. Top Quote and Depth: Market Data Full Refresh messages show the BBO and size of market orders as the book is updated. The Security Trading Status = 21 (pre-open) and is sent for all instruments every three minutes. Order and Trade feeds contain just Heartbeat messages.
9:30 a.m.	The market opens and regular trading begins. Top Quote and Depth: An **Instrument List Status** message is sent as each product is opened. Quote messages are sent with market updates. Snapshots continue every three minutes. Trade messages are sent on the Trade and Top Quote feeds. Orders that rest on the book and auctions are sent on the Order feed. No further information is sent on the Pre-Open feed (Heartbeats only). RefData: incremental feed sends Instrument Incremental Updates as complex and simple Instruments are created. Snapshot feed continues to send snapshots of all products and instruments at regular intervals.
4:00 p.m./4:15 p.m.	Regular trading ends. Top Quote and Depth: An **Instrument List Status** message is sent as each product is closed. Snapshot messages continue to be disseminated however all BBO prices are zero. Order: No further information is sent (Heartbeats only).
5:31 p.m.	Reference Data feeds: Continue to provide snapshot information every minute until 5:31 p.m.
5:45 p.m.	All feeds are closed

2.5 Support and Connectivity

ISE support for the MDI is available from 8 am to 6 pm (Eastern Time) on market days and the contacts are as follows:

Table 4: ISE Contact List

ISE Contact List		
Business Issues	212-897-8160	marketdata@ise.com
Technical Support	212-897-0284	computeroperations@ise.com
Market Data Support	212-897-0244, #1	tms@ise.com
Member Connectivity	212-897-0244, #3	connect@ise.com

ISE market feeds are currently distributed by:

Table 5: ISE MDI Distributors

MDI Distributor		
Activ Financial	Atrium Networks	BT Radianz
Essex Radez	GuavaTech	Interactive Data 7ticks
Lime Brokerage	NYSE SFTI	NYSE SuperFeed
OptionsIT	Pico Quantitative Trading	SAVVIS
Verizon Financial Network		

Members may use their existing connections to access these data but their routers may need to be upgraded to support multicast data.

Contact information for the MDI distributors may be found in **Section 6.2, Service Providers**, beginning on page 89.

3. FAST Feed Descriptions

This section provides a description of each FAST encoded data feed. All messages are defined according to the FIX 5.0 SP2 standard.

- All multicast data are sent in blocks (UDP packets) in which the *application data* do not exceed 1000 bytes:
- The first message in each block is the FAST Reset message.
- The second message in each block is a **Block Header** message, which contains a block number and timestamp.
- The rest of the block contains the FIX market data messages.
- As per FAST, all integer data are represented in **Big-Endian** byte order.

The FIX messages do not have standard headers. The only fields at the start of each message are the message type and sequence number. The Timestamp, Source ID, and version number have been moved into the Header message of each Block to improve efficiency.

Each data feed has its own version of a snapshot and incremental message, defined using specific message formats. The feeds available on the MDI are:

- Depth of Market (Depth)
- Order
- Reference Data (RefData)

Except for the Reference Data, the feeds for simple and complex instruments are separated, and sent out on different multicast channels. As indicated previously, the market data feeds for complex instruments are collectively known as the "Spread feed."

A single FIX message may be used for multiple feeds. For example, the Market Data Incremental Refresh is used on both the Trade and the Top Quote feeds. In this document, and in the FAST template file, messages are assigned discrete, feed-specific names to better describe their purpose.

3.1 Depth of Market (Depth) Feed

All feeds are disseminated in duplicate over two multicast streams, as is done for OPRA, and referred to as the A feed and the B feed. The reference data provide the IP/Port addresses of each feed, both A and B, for each product.

The Depth of Market feed is described below:

3.1.1 Purpose

The Depth feed provides subscribers with the bids and offers at the top five price levels of the order book. All quotes and orders at each price level are aggregated into the total quantity. The quantity of Customer Orders and Customer Professional orders are also supplied in separate fields. Trade data are not present on this feed.

Price depth data are sent starting at 6:00 a.m. through the close of each trading day.

Depth for simple instruments and complex instruments are sent on separate multicast streams.

3.1.2 Messages

The Depth feed utilizes four (five) messages:

- **Depth Snapshot** is used to send Snapshots and as the heartbeat for the feed
- **Depth Incremental** for changes to the quotes within the top 5 price levels
- **Instrument Status** for trading state change of an instrument
- **Instrument List Status** for trading state changes of many instruments in a Product
- **(Heartbeat** is sent only if no other data exist to indicate activity on an otherwise "dark" feed)

The FIX message types used for each message are:

Table 6: Depth Feed FIX Message Types

Message	FIX Message	Msg Type	Notes
Depth Incremental	Market Data Incremental Refresh	X	Used to send new quotes (may update multiple levels)
Depth Snapshot	Market Data Snapshot Full Refresh	W	A snapshot of an Instrument, giving Bid and Offer quotes for five price levels, Trading Status, and some reference data.
Instrument Status	Security Status	f	Change of Trading Status of one instrument
Instrument List Status	Security Mass Status	CO	Status change of all instruments in a product
(Heartbeat)	(Heartbeat)	(0 (zero))	(**Section 3.4.3, Heartbeat Message**, pg 45)

3.1.3 Concepts

The Depth feed provides a view of five levels of depth, showing the aggregate quantity of quotes and orders on each price level. Customer quantity and Professional Customer quantity is also shown on each level.

Example: the top five price levels provided in the Depth feed:

Instrument: IBM 17Jun2011 80 C, Instrument ID 2026, Product ID 427								
Status: – Regular (17)								
Bid					Offer			
Qty Market Orders=-30					Qty Market Orders=100			
Level	Price	Qty	Cust	CustProf	Price	Qty	Cust	CustProf
1	0.98	20	10	5	1.00	50	0	10
2	0.97	30	0	10	1.01	30	0	0
3	0.96	10	5	5	1.03	10	5	0
4	0.94	80	40	0	1.05	10	0	0
5	0.93	10	0	10	1.08	10	0	0

Note the following Depth feed features:

1. The price level field, *MDPriceLevel*, indicates where each price is to be inserted or changed in the depth display.
2. **Depth Incremental** uses update actions "new," "change," and "delete" at each price level.

Snapshots are sent using the **Depth Snapshot** message, and are sent for each instrument at regular three-minute intervals throughout the day, starting at 6:00 a.m. The **Depth Snapshot** message provides a description of each instrument along with a unique product identifier and instrument identifier, the bids and offers up to five levels, and trading state. The **Depth Snapshot** message does not contain trade information. Snapshot messages are sent in-band.

Changes in trading state are sent using the **Instrument Status** message or the **Instrument List Status** message. The **Instrument Status** message is used when one instrument changes state independently, while the **Instrument List Status** message is used when all instruments in a product change state at the same time—for example, at the open, or at the close.

Price updates within any of the top five levels are sent with the **Depth Incremental** message, starting at 6:00 a.m. with pre-open BBOs, and continuing during regular trading with open market price updates.

Quantity fields on the Depth messages provide separate Customer and Customer Professional quantities; as well as the total quantity, which includes the Customer and Customer Professional quantities.

The **Depth Snapshot** message contains the Instrument and Product Identifiers, as well as the OSI name. The **Depth Incremental** and **Instrument Status** messages have only the product identifier and instrument identifier. Recipients can use the information in the **Depth Snapshot** messages to create a mapping table between the product identifiers and instrument identifiers to the OSI names. This information can also be obtained from the Reference Data feed.

3.1.4 Daily Schedule

The Depth feed is on the same schedule as the Top Quote feed. Please see **Section Error! Reference source not found., Error! Reference source not found.** on page **Error! Bookmark not defined.** for a complete description.

3.1.5 Depth Snapshot Message

The **Depth Snapshot** message provides a snapshot of an instrument including the bids and offers for the top five price levels, and the trading state. This message is sent once every three minutes for every instrument, and serves as the heartbeat for the Depth feed.

The recipient should process one complete pass of **Depth Snapshot** messages when first connecting for the day. After processing one complete pass, the **Depth Snapshot** can be ignored unless the *RefreshIndicator* field is set to "Y". This is used in the event that a new instrument is added intraday, or a system failure that requires the ISE to send snapshots for all instruments to refresh the order book. It is also set to "Y" when new Complex Instruments are created.

3.1.5.1 Format – Depth Snapshot

The following table shows the format of the **Depth Snapshot** message.

Table 7: Depth Snapshot (Template ID 8)

Tag	Tag Name	Req	Comments
34	MsgSeqNum	Y	
35	MsgType	Y	W=Market Data Snapshot Full Refresh
1022	MDFeedType	Y	PD=Price Depth
1683	MDFeedSubType	Y	F=Full (Snapshot)
1187	RefreshIndicator	Y	N=Process if required (Default) Y=Mandatory refresh
1300	MarketSegmentID	Y	Unique Product ID
6426	Underlying Symbol	Y	Content depends on product category – e.g. for options, it is the identifier for the stock leg of a complex instrument
48	SecurityID	Y	Unique Instrument ID
55	Symbol	N	OPRA Root Symbol, only present for simple instruments
201	PutOrCall	N	0=Put, 1=Call Only present for simple instruments
541	MaturityDate	N	YYYYMMDD Only present for simple instruments
202	Strike Price	N	Only present for simple instruments
555	NoLegs	N	Only present for complex instruments
602	> LegSecurityID	Y	
609	> LegSecurityType	Y	OPT=Option (default) CS=Stock
623	> LegRatioQty	Y	
624	> LegSide	Y	1=Buy, 2=Sell
1682	MDSecurityTradingStatus	Y	See SecurityTradingStatus (tag 326)

Tag	Tag Name	Req	Comments
8642	LinkageHandlingIndicator	N	Used only for simple instruments. O=Linkage Handling Inactive 1=Linkage Handling active
5292	BidMarketSize	Y	Quantity of Market order contracts on the bid side. Only provided during regular trading or fast market. Default=0.
5293	AskMarketSize	Y	Quantity of Market order contracts on the offer side. Only provided during regular trading or fast market. Default=0.
26001	BidMarketNTTSize	Y	Quantity of NTT market order contracts on the bid side. Only provided during regular trading or fast market.
26002	AskMarketNTTSize	Y	Quantity of NTT market order contracts on the offer side. Only provided during regular trading or fast market.
268	NoMDEntries	Y	
269	> MDEntryType	Y	0=Bid, 1=Offer (Default), J=empty book. J=Empty Book required when switching to regular trading as receivers of the pre-open feed might have left the feed with a non-empty book that became empty during opening.
270	> MDEntryPx	N	Not present if empty book
271	> MDEntrySize	N	Not present if empty book
6709	> MDCustomerSize	N	Customer quantity. Default=0.
6208	> MDSecondaryCustomerSize	N	Customer professional quantity. Default=0.
26000	> MDNTTSize	N	Conveys quantity of limit orders that cannot be traded through (complex orders only).
1023	> MDPriceLevel	N	Price Level

3.1.5.2 Examples

Example of a **Depth Snapshot** sent after the open:

Table 8: Depth Snapshot after Open

Tag	Tag Name	Value	Description
34	MsgSeqNum	123056	
35	MsgType	W	Market Data Snapshot Full Refresh
1022	MDFeedType	PD	Price Depth
1683	MDFeedSubType	F	Full (Snapshot)
1187	RefreshIndicator	N	No need to process
1300	MarketSegmentID	427	Product ID
48	SecurityID	2026	Instrument ID
55	Symbol	IBM1	OCC Root Symbol
201	PutOrCall	1	Call

Tag	Tag Name	Value	Description
541	MaturityDate	20110614	Expiration date
202	Strike Price	80.00	
1682	MDSecurityTradingStatus	17	Regular
5292	BidMarketSize	0	
5293	AskMarketSize	0	
26001	BidMarketNTTSize	0	
26002	AskMarketNTTSize	0	
268	NoMDEntries	3	
269	> MDEntryType	0	Bid
270	> MDEntryPx	0.98	
271	> MDEntrySize	20	
6709	> MDCustomerSize	10	Customer
6208	> MDSecondaryCustomerSize	5	Customer Professional
1023	> MDPriceLevel	1	top price level
269	> MDEntryType	0	Bid
270	> MDEntryPx	0.96	
271	> MDEntrySize	100	
1023	> MDPriceLevel	2	second price level
269	> MDEntryType	1	Offer
270	> MDEntryPx	1.01	
271	> MDEntrySize	100	
1023	> MDPriceLevel	1	top price level on Offer

3.1.6 Depth Incremental

Depth Incremental messages are used to send changes to any of the top five price levels. The **Depth Incremental** contains multiple repeating items, each of which updates one side of one price level. All repeating items in one message relate to the same instrument.

3.1.6.1 Format — Depth Incremental Message

The following table shows the format of the **Depth Incremental** message.

Table 9: Depth Incremental (Template ID 4)

Tag	Tag Name	Req	Comments
34	MsgSeqNum	Y	
35	MsgType	Y	X=Market Data Incremental Refresh
1022	MDFeedType	Y	PD=Price Depth
1683	MDFeedSubType	Y	U=Update (Incremental)
1300	MarketSegmentID	Y	Product ID
48	SecurityID	Y	Instrument ID

Tag	Tag Name	Req	Comments
5292	BidMarketSize	Y	Quantity of Market order contracts on the bid side. Only provided during regular trading or fast market.
5293	AskMarketSize	Y	Quantity of Market order contracts on the offer side. Only provided during regular trading or fast market.
26001	BidMarketNTTSize	N	Conveys NTT market order quantity on the bid side that cannot be traded through (only for complex orders).
26002	AskMarketNTTSize	N	Conveys NTT market order quantity on the offer side that cannot be traded through (only for complex orders).
268	NoMDEntries	Y	Default=1
279	> MDUpdateAction	Y	0=New, 1=Change, 2=Delete, 4=Delete From
269	> MDEntryType	Y	0=Bid, 1=Offer
270	> MDEntryPx	N	Price
271	> MDEntrySize	N	Quantity, not including market orders. Conditionally required for MDUpdateAction = 0 or 1.
6709	> MDCustomerSize	N	Quantity of Customer orders included in MDEntrySize. Conditionally required for MDUpdateAction = 0 or 1. Default=0.
6208	> MDSecondaryCustomerSize	N	Quantity of Customer Professional orders included in MDEntrySize
26000	> MDNTTSize	N	Conveys quantity of limit orders that cannot be traded through (complex instruments only)
1023	> MDPriceLevel	Y	Price Level. Default=1.

NOTE: This format deviates from the standard—several fields have been moved out of the MDEntries repeating group to improve the efficiency of the message.

The following sections explain how the update action field (*MDUpdateAction*) is used to maintain the view of the book:

3.1.6.2 Update Action — New Price Level

When a new price level is created in the order book, a **Depth Incremental** message is sent with *MDUpdateAction* set to zero ("New"). This indicates:

- That the new price level is to be inserted at the specified price level.
- All existing rows in the order book at this level and lower are to be pushed down. If there were already five price levels then the last level should be deleted.
- There is no explicit instruction to delete the bottom price level when inserting a new price level.

The field *MDPriceLevel* is used to identify which level is being inserted. If set to 1:

- It is to be inserted at the top, regardless of the prices.
- The subscriber's application should check that there are no prices higher than this price level and if they do exist then they should be deleted. This should not happen in normal operation.

Example 1: An order creates a new Best Bid:

Initial State of the book:

Instrument: IBM 17Jun2011 80 C, Instrument ID 2026, Product ID 427								
Status: – Regular (17)								
Bid					Offer			
Qty Market Orders=0					Qty Market Orders=0			
Level	Price	Qty	Cust	CustProf	Price	Qty	Cust	CustProf
1	0.97	30	15	0	1.00	50	0	0
2	0.94	80	0	10				
3	0.92	60	0	0				
4	0.90	50	0	0				
5	0.88	10	0	0				

A new **Customer** order to Buy 20 @ 0.98 is added to the book.

Tag	Tag Name	Value	Description
34	MsgSeqNum	123056	
35	MsgType	X	Market Data Incremental Refresh
1022	MDFeedType	PD	Price Depth
1683	MDFeedSubType	U	Update (Incremental)
1300	MarketSegmentID	427	Product ID
48	SecurityID	2026	Instrument ID
268	NoMDEntries	1	
279	> MDUpdateAction	0	New
269	> MDEntryType	0	Bid
270	> MDEntryPx	0.98	Price
271	> MDEntrySize	20	Total Quantity is 20
6709	> MDCustomerSize	20	
1023	> MDPriceLevel	1	

The new row is inserted as price level 1 and all subsequent rows are pushed down. The old row number 5 is deleted.

State of the book after the order is entered:

Instrument: IBM 17Jun2011 80 C, Instrument ID 2026, Product ID 427								
Status: – Regular (17)								
Bid					Offer			
Qty Market Orders=0					Qty Market Orders=0			
Level	Price	Qty	Cust	CustProf	Price	Qty	Cust	CustProf
1	0.98	20	20	0	1.00	50	0	0
2	0.97	30	15	0				
3	0.94	80	0	10				
4	0.92	60	0	0				
5	0.90	50	0	0				

3.1.6.3 Update Action — Change Price Level

A **Depth Incremental** Message with *MDUpdateAction* equal to one ("Change") indicates:

- A change at a given price level
- All fields on the specified side at the price level should be updated.

Example: An order at the top price level is partially executed:

The quantity of an existing Customer buy order is reduced from 20 contracts to 10.

Tag	Tag Name	Value	Description
34	MsgSeqNum	123555	
35	MsgType	X	Market Data Incremental Refresh
1022	MDFeedType	PD	Price Depth
1683	MDFeedSubType	U	Update (Incremental)
1300	MarketSegmentID	427	Product ID
48	SecurityID	2026	Instrument ID
268	NoMDEntries	1	
279	> MDUpdateAction	1	Change
269	> MDEntryType	0	Bid
270	> MDEntryPx	0.98	
271	> MDEntrySize	10	Total Quantity is 10
6709	> MDCustomerSize	10	
1023	> MDPriceLevel	1	

State of the book after the order is executed:

Instrument: IBM 17Jun2011 80 C, Instrument ID 2026, Product ID 427	
Status: – Regular (17)	
Bid	Offer
Qty Market Orders=0	Qty Market Orders=0

Instrument: IBM 17Jun2011 80 C, Instrument ID 2026, Product ID 427								
Level	Price	Qty	Cust	CustProf	Price	Qty	Cust	CustProf
1	0.98	10	10	0	1.00	50	0	0
2	0.97	30	15	0				
3	0.94	80	0	10				
4	0.92	60	0	0				
5	0.90	50	0	0				

3.1.6.4 Update Action — Delete Price Level

A **Depth Incremental** with *MDUpdateAction* equal to two ("Delete") is used to delete a price level.

Example: The remaining quantity at the top price level on the Bid is deleted:

A **Depth Incremental** is sent to delete Price Level 1 on the Bid side. As a result, all lower bid levels move up.

Tag	Tag Name	Value	Description
34	MsgSeqNum	123555	
35	MsgType	X	Market Data Incremental Refresh
1022	MDFeedType	PD	Price Depth
1683	MDFeedSubType	U	Update (Incremental)
1300	MarketSegmentID	427	Product ID
48	SecurityID	2026	Instrument ID
268	NoMDEntries	1	
279	> MDUpdateAction	2	Delete
269	> MDEntryType	0	Bid
1023	> MDPriceLevel	1	
271	> MDEntrySize	10	Total Quantity is 10
1023	> MDPriceLevel	5	

The state of the book after deleting Price Level 1.

Instrument: IBM 17Jun2011 80 C, Instrument ID 2026, Product ID 427								
Status: – Regular (17)								
Bid					Offer			
Qty Market Orders=0					Qty Market Orders=0			
Level	Price	Qty	Cust	CustProf	Price	Qty	Cust	CustProf
1	0.97	30	15	0	1.00	50	0	0
2	0.94	80	0	10				
3	0.92	60	0	0				
4	0.90	50	0	0				

Instrument: IBM 17Jun2011 80 C, Instrument ID 2026, Product ID 427								
5								

3.1.6.5 Update Action — Delete From Price Level

The *MDUpdateAction,* 4 ("Delete From"), is used to clear the book on one side of the book starting at the indicate price level. It is also used to delete the whole book when an instrument goes into Halt.

Example: Clear the book.

A **Depth Incremental** message is sent to clear the book:

Tag	Tag Name	Value	Description
34	MsgSeqNum	123555	
35	MsgType	X	Market Data Incremental Refresh
1022	MDFeedType	PD	Price Depth
1683	MDFeedSubType	U	Update (Incremental)
1300	MarketSegmentID	427	Product ID
48	SecurityID	2026	Instrument ID
268	NoMDEntries	2	
279	> MDUpdateAction	4	Delete From – Clear the book from price level
269	> MDEntryType	0	Bid
1023	> MDPriceLevel	1	
279	> MDUpdateAction	4	Delete From –Clear the book from price level
269	> MDEntryType	1	Offer
1023	> MDPriceLevel	1	

3.1.6.6 Example — Multiple Updates

There can be multiple updates in one message. The bid is updated first, and in a rising market, the bid can overlap the offer before the offer is moved out of the way. The recipient must apply all items in a message before evaluating the resulting book.

NOTE: The Price Level changes as each update is applied within a message. For example, delete Price Level 3, Change Price Level 3:
The first update to delete Price Level 3 results in row 4 being moved up to row 3.
The next update to change Price Level 3 results in a change to the new row 3.

Example: Multiple updates

The top-level quotes are updated, from:
15 @ 0.97 x 50 @ 1.00 to:
50 @ 1.00 x 50 @ 1.02

In addition, a new Bid price level 2 is added, and Bid quotes on other levels are updated.

State of the book before the update:

Instrument: IBM 17Jun2011 80 C, Instrument ID 2026, Product ID 427								
Status: – Regular (17)								
Bid					Offer			
Qty Market Orders=0					Qty Market Orders=0			
Level	Price	Qty	Cust	CustProf	Price	Qty	Cust	CustProf
1	0.97	15	0	0	1.00	50	0	0
2	0.94	80	0	10				
3	0.92	60	0	0				
4	0.90	50	0	0				
5								

A **Depth Incremental** message is sent to Delete the current quotes, insert new quotes, and update existing quotes. (Shaded rows indicate MDEntries iterations.)

Tag	Tag Name	Value	Description
34	MsgSeqNum	123555	
35	MsgType	X	Market Data Incremental Refresh
1022	MDFeedType	PD	Price Depth
1683	MDFeedSubType	U	Update (Incremental)
1300	MarketSegmentID	427	Product ID
48	SecurityID	2026	Instrument ID
268	NoMDEntries	7	
279	> MDUpdateAction	2	Delete
269	> MDEntryType	0	Bid
1023	> MDPriceLevel	1	
279	> MDUpdateAction	0	New
269	> MDEntryType	0	Bid
270	> MDEntryPx	1.00	Price
271	> MDEntrySize	50	Quantity
1023	> MDPriceLevel	1	
279	> MDUpdateAction	2	Delete
269	> MDEntryType	1	Offer
1023	> MDPriceLevel	1	
279	> MDUpdateAction	0	New
269	> MDEntryType	1	Offer
270	> MDEntryPx	1.02	Price
271	> MDEntrySize	50	Quantity
1023	> MDPriceLevel	1	

Tag	Tag Name	Value	Description
279	> MDUpdateAction	0	New
269	> MDEntryType	0	Bid
270	> MDEntryPx	0.98	Price
271	> MDEntrySize	30	Quantity
1023	> MDPriceLevel	2	
279	> MDUpdateAction	1	Change
269	> MDEntryType	0	Bid
270	> MDEntryPx	0.94	
271	> MDEntrySize	60	
6709	> MDCustomerSize	10	
1023	> MDPriceLevel	3	
279	> MDUpdateAction	1	Change
269	> MDEntryType	0	Bid
270	> MDEntryPx	0.90	
271	> MDEntrySize	60	
6709	> MDCustomerSize	10	
1023	> MDPriceLevel	5	

The state of the book after applying all changes:

Instrument: IBM 17Jun2011 80 C, Instrument ID 2026, Product ID 427								
Status: – Regular (17)								
Bid					Offer			
Qty Market Orders=0					Qty Market Orders=0			
Level	Price	Qty	Cust	CustProf	Price	Qty	Cust	CustProf
1	1.00	50	0	0	1.02	50	0	0
2	0.98	30	0	0				
3	0.94	80	0	10				
4	0.92	60	0	0				
5	0.90	60	10	0				

3.1.7 Instrument Status Messages

The **Instrument Status** message is sent when a *single* instrument changes state during the day. For example, when quotes are removed from one instrument, or if one instrument is manually halted by Market Operations.

The **Instrument List Status** message is sent when *all* instruments for a product change state at the same time; for example, at the opening or the close. The field *SecurityMassTradingStatus* defines the current state for the instruments. If one or more instruments cannot change state, the **Instrument List Status** message contains an exception list identifying the instruments that could

not change state. The field *SecurityTradingStatus* defines the current state for the excepted instrument(s). As those excepted instruments are handled, the **Instrument List Status** message is sent with an ever-decreasing exceptions list.

3.1.7.1 Format – Instrument Status

The following table shows the format of the **Instrument Status** Message:

Table 10: Instrument Status (Template ID 6)

Tag	Tag Name	Req	Comments
34	MsgSeqNum	Y	
35	MsgType	Y	f=Security Status.
1300	MarketSegmentID	Y	Unique product identifier
1227	ProductComplex	Y	See **Appendix C: FIX Field Descriptions** on page 120
48	SecurityID	Y	Unique instrument identifier
326	SecurityTradingStatus	Y	See **Appendix C: FIX Field Descriptions** on page 120
1174	SecurityTradingEvent	N	6=Change of Security Trading Status (Default) 100=Change of Linkage Handling
8642	LinkageHandlingIndicator	N	0=Linkage Handling Inactive 1=Linkage Handling Active (Default)

3.1.7.2 Format—Instrument List Status

The following table shows the format of the **Instrument List Status** Message:

Table 11: Instrument List Status (Template ID 5)

Tag	Tag Name	Req	Comments
34	MsgSeqNum	Y	
35	MsgType	Y	CO=Security Mass Status
1300	MarketSegmentID	Y	Product ID
1544	InstrumentScopeProductComplex	Y	See **Appendix C: FIX Field Descriptions** on page 120
1679	SecurityMassTradingStatus	Y	See **Appendix C: FIX Field Descriptions** on page 120
146	NoRelatedSym	Y	Number of exceptions. Use to convey exception list of instruments. Default=0.
48	> SecurityID	N	Req'd if NoRelatedSym > 0.
326	> SecurityTradingStatus	N	See **Appendix C: FIX Field Descriptions** on page 120

NOTE: Instrument Status and **Instrument List Status** messages are also sent on the Top Quote feed.

3.1.7.3 Examples

Example: Market Operations halt one instrument

Tag	Tag Name	Value	Description
34	MsgSeqNum	123760	
35	MsgType	f	Security Status
1300	MarketSegmentID	427	
1227	ProductComplex	1	
48	SecurityID	2026	
326	SecurityTradingStatus	21	Pre-open

Example: All instruments in a product open at 9:30 a.m., except one that has an imbalance:

Tag	Tag Name	Value	Description
34	MsgSeqNum	123770	
35	MsgType	CO	Security Mass Status
1300	MarketSegmentID	427	
1544	InstrumentScopeProductComplex	1	
1679	SecurityMassTradingStatus	17	Regular
146	NoRelatedSym	1	
48	> SecurityID	2026	
326	> SecurityTradingStatus	1	Opening Delay

Example: The imbalance is removed and now all instruments are open

Tag	Tag Name	Value	Description
34	MsgSeqNum	123789	
35	MsgType	CO	Security Mass Status
1300	MarketSegmentID	427	
1544	InstrumentScopeProductComplex	1	
1679	SecurityMassTradingStatus	17	Regular
146	NoRelatedSym	0	

3.2 Order Feed

The Order feed is described below:

3.2.1 Purpose

The Order feed advises participants that a new order is now resting on the book. The quantity and price of the new order are disclosed. The Order feed also announces that a new Auction

order is in the market. Auction orders include Flash, Facilitation, Solicitation, etc. For public (exposed) auctions, auction responses are also disclosed.

NOTE: Auction announcements are only available via the Order feed; there are no auction order broadcasts through the DTI.

The Order feed uses the **Order on Book** message to supply the information about each order.

The Order Feeds for simple instruments and for complex instruments are sent as separate streams.

3.2.2 Messages

The Order feed utilizes two messages:

- **Order on Book** message
- **Heartbeat** message

The FIX message types used for each message are:

Table 12: Order Feed FIX Message Types

Message	FIX Message	Msg Type	Notes
Order on Book	Market Data Snapshot Full Refresh	W	A description of one order.
Heartbeat	Heartbeat	0 (zero)	(**Section 3.4.3, Heartbeat Message**, pg 45)

3.2.3 Concepts

The purpose of this feed is simply to notify participants that a new order has arrived and is resting on the book. This feed is also used to announce the start and end of auctions (e.g. Flash, Facilitation, Solicitation, PIM, etc.), and public auction responses (complex exposure auctions).

- **Order on Book** messages are sent for any new orders that arrive and *rest* on the book. The message is sent even if the order is outside the current market.
- **Order on Book** messages are *not* sent for orders that fill or are canceled on entry; for resting orders that are modified, canceled or traded; or for quotes.
- **This feed cannot be used to build the ISE order book.**
- For Reserve orders, only the **displayed quantity** is disclosed.

3.2.4 Daily Schedule

The Order feed commences each day at 6:00 a.m. ET with **Heartbeats**.

Starting at 6:00 a.m. ET, **Order on Book** messages are sent for new resting orders as well as new auctions. The **Order on Book** messages continue until each instrument closes.

After market close (4:00 p.m./4:15 p.m. ET), only **Heartbeats** are sent until the feed closes at 5:45 p.m. ET.

3.2.5 Order on Book Message

The **Order on Book** message is sent for each new order that rests on the order book. Each message describes one order, including price, size, order capacity, and, if the order is an Attributable order, may also disclose the identities of the sending and clearing firms.

For Block auctions, some fields, including price and size, may not be disclosed, in which case the fields are not present in the message. For exposed auctions (complex exposure), the first iteration of the *NoMDEntries* repeating group specifies the order being auctioned and the second iteration, if present, specifies only the aggregate quantity at the best response price.

All-or-None orders are identified by the *ExecInst* field.

3.2.5.1 Format – Order on Book

The following table shows the format of the **Order on Book** message.

Table 13: Order on Book (Template ID 9)

Tag	Tag Name	Req	Comments
34	MsgSeqNum	Y	
35	MsgType	Y	W=Market Data Snapshot Full Refresh
1022	MDFeedType	Y	OB=Order On Book
1683	MDFeedSubType	Y	O=Order A=Auction
1300	MarketSegmentID	Y	Product ID
48	SecurityID	Y	Instrument ID
555	NoLegs	N	Only present for complex instruments
602	> LegSecurityID	Y	
609	> LegSecurityType	Y	OPT=Option (default) CS=Common Stock
623	> LegRatioQty	Y	
624	> LegSide	Y	1=Buy, 2=Sell
6682	> LegAllocAccount	N	Clearing account (PartyRole 83), if disclosed.
6684	> LegAllocClearingAccount	N	Clearing CMTA firm (PartyRole 4), if disclosed.
268	NoMDEntries	Y	1-2 (Default=1) 2 entries are possible only for exposed auctions.
269	> MDEntryType	Y	0=Bid 1=Offer (Default) Q=Auction Price (Side not disclosed)
270	> MDEntryPx	N	Price/Premium. Not present for market orders or auctions w/o price disclosure
271	> MDEntrySize	N	Quantity. Not present for auctions w/o volume disclosure
40	> OrdType	N	1=Market, 2=Limit Not present for auction responses.

Tag	Tag Name	Req	Comments
8522	> AuctionType	N	Present for auction orders. Not present for auction responses. See **Appendix C: FIX Field Descriptions** on page 120
276	> QuoteCondition	N	A=Start/Update auction, B=End of auction
37	> OrderID	N	Identifies the (auction) order. Not present for auction responses.
110	> MinQty	N	for minimum execution quantity orders
18	> ExecInst	N	G=AON
528	> OrderCapacity	N	See **Appendix C: FIX Field Descriptions** on page 120
546	> Scope	N	1=Local (ignore away market) 2=National
453	> NoPartyIDs	N	Present only if Parties are disclosed.
448	>> PartyID	Y	
452	>> PartyRole	Y	4=Clearing Firm (CMTA) 59=Member ID 83=Clearing Acct (Give-Up)

3.2.5.2 Examples

An example of an **Order on Book** message for a Limit order during trading:

Tag	Tag Name	Value	Description
34	MsgSeqNum	123056	
35	MsgType	W	Market Data Snapshot Full Refresh
1022	MDFeedType	OB	Order on Book
1683	MDFeedSubType	O	Order
1300	MarketSegmentID	427	Product ID
48	SecurityID	2026	Instrument ID
268	NoMDEntries	1	
269	> MDEntryType	1	Offer
271	> MDEntryPx	0.99	
271	> MDEntrySize	20	
40	> OrdType	2	Limit
528	> OrderCapacity	C	Customer

An example of a FLASH Auction announcement:

Tag	Tag Name	Value	Description
34	MsgSeqNum	123056	
35	MsgType	W	Market Data Snapshot Full Refresh
1022	MDFeedType	OB	Order feed
1683	MDFeedSubType	A	Auction

Tag	Tag Name	Value	Description
1300	MarketSegmentID	427	product identifier
48	SecurityID	2026	instrument identifier
268	NoMDEntries	1	
269	> MDEntryType	0	Bid
270	> MDEntryPx	3.60	
271	> MDEntrySize	20	
40	> OrdType	2	Limit
8522	> AuctionType	3	FLASH auction
276	> QuoteCondition	A	Start auction
37	> OrderID	1336060744892086015	identifies the auction order
528	> OrderCapacity	C	Customer

An example of an ongoing PIM Auction:

Tag	Tag Name	Value	Description
34	MsgSeqNum	123056	
35	MsgType	W	Market Data Snapshot Full Refresh
1022	MDFeedType	OB	Order on Book
1683	MDFeedSubType	A	Auction
1300	MarketSegmentID	427	product identifier
48	SecurityID	2026	instrument identifier
268	NoMDEntries	1	
269	> MDEntryType	0	Bid
270	> MDEntryPx	23.75	
271	> MDEntrySize	20	
40	> OrdType	2	Limit
8522	> AuctionType	12	PIM auction (simple instr.)
276	> QuoteCondition	A	Start (update) auction
37	> OrderID	1336060744892086015	identifies the auction order
528	> OrderCapacity	C	Customer

3.2.6 Heartbeat Message

The **Heartbeat** message is sent once per minute while the feed is open if nothing else is sent in that minute.

Please see **Section 3.4.3, Heartbeat Message** on page 45 for a complete description of this message.

3.3 Reference Data Feed

The Reference Data (RefData) feed is described below:

3.3.1 Purpose

The RefData feed continuously streams a complete list of all products and instruments (simple and complex) traded at the ISE.

The RefData is actually sent as two separate feeds:

- The **RefData Snapshot** feed—provides a continuous cycle of all product and instrument definitions on one-minute intervals.
- The **RefData Incremental** feed—provides real-time information about products and instruments that are added, changed, or deleted intraday. Note that the change (add/delete) shown on the RefData Incremental feed appear in the *next* snapshot cycle.

Reference data for simple and complex instruments appear on the *same* feed.

3.3.2 Messages

The RefData feed utilizes seven messages:

3.3.2.1 RefData Snapshot Feed:

- **Product Snapshot**
- **Instrument Snapshot**
- **Start of Snapshot Cycle** which flags the start of the snapshot
- **End of Snapshot Cycle** which flags the end of the snapshot

3.3.2.2 RefData Incremental Feed:

- **Product Incremental**
- **Instrument Incremental**
- **Heartbeat**

The FIX message types used for each message are as follows:

Table 14: RefData FIX Message Types

Message	FIX Message	Msg Type	Notes
Product Snapshot	Market Definition	BU	
Instrument Snapshot	Security Definition	d	

Message	FIX Message	Msg Type	Notes
Start of Snapshot Cycle	<none>	<none>	
End of Snapshot Cycle	<none>	<none>	
Product Incremental	Market Definition Update Report	BV	
Instrument Incremental	Security Definition Update Report	BP	
Heartbeat	Heartbeat	0 (zero)	(**Section 3.4.3, Heartbeat Message**, pg 45)

NOTE: Shaded messages indicate non-standard, ISE-defined messages.

3.3.3 Concepts

The RefData Snapshot feed is a complete snapshot of all reference data (products and instruments) and is sent in a continuous cycle throughout the full day.

- A complete snapshot cycle starts with the **Start of Snapshot Cycle** message, and proceeds with a **Product Snapshot**, followed by an **Instrument Snapshot** for each instrument (simple and complex) for that product. Each product and its associated instruments are defined in turn until all products and all instruments have been sent. The cycle ends with the **End of Snapshot Cycle** message, which includes counters indicating the total number of products and instruments sent in that cycle. The next full cycle begins immediately.
- When describing the instruments for a product, simple instruments are defined first, then the complex instruments. All instruments for a product are defined before moving on to the next product.
- If products or instruments are added, changed, or deleted intraday, that change is immediately reported on the RefData Incremental feed, using the **Product Incremental** or the **Instrument Incremental** message, as appropriate.
- The system generates the snapshot messages for *all* products and instruments at the start of the snapshot cycle, which are then disseminated smoothly over the snapshot interval. If a product or instrument is added (changed, deleted) after a snapshot cycle has started, the change is not reflected in the snapshot until the *next* full cycle.

NOTE: Some reference data are included in the **Top Quote** and **Depth** feed snapshot messages. Those messages supply minimal, but sufficient data to map the ISE Product and Instrument IDs to regular OSI Symbology.

3.3.4 Daily Schedule

The RefData Feeds (Snapshot and Incremental) start at 4:00 a.m. ET.

The snapshot feed continuously streams **Product/Instrument Snapshot** messages, bounded by the **Start/End of Snapshot Cycle** messages.

The incremental feed sends **Heartbeats**, unless a change is made to the reference data, at which point a **Product Incremental** or **Instrument Incremental**, as appropriate, is sent. **Product** and **Instrument Incremental** messages may be sent at any time.

The RefData feeds close at 5:30 p.m. ET

3.3.5 Product Snapshot Message

A **Product Snapshot** message provides a complete description of a product, including its trading parameters and the market data feed channels over which its instrument market data are streamed.

Note that Tick Rules, Match Rules, and Auction Rules are defined for each of the Instrument Types (Simple, Combination, and Stock Combination). The Price Step table, for example, has one set of entries for simple instruments, another set for combination instruments, and a third set for stock combination instruments.

3.3.5.1 Format — Product Snapshot

The following table shows the format of the **Product Snapshot** message.

Table 15: Product Snapshot (Template ID 12)

Tag	Tag Name	Req	Description
34	MsgSeqNum	Y	
35	MsgType	Y	BU=Market Definition
1301	MarketID	Y	XSIX=ISO 10383 MIC for ISE GMNI=ISO 10383 MIC for ISE Gemini MCRY=ISO 10383 MIC for ISE Mercury
1300	MarketSegmentID	Y	Product ID
1396	MarketSegmentDesc	N	Product ID from legacy trading system (deprecated)
8599	MarketSegmentStatus	Y	1=Active, 2=Inactive
1325	ParentMktSegmID	N	Concatenation of Market Segment and Bin ID: IXS=Primary Options Market Segment IXT=Secondary Options Market Segment IXC=FX Market Segment
5948	PartitionID	Y	Partition of the product
5949	BinID	Y	Bin of the product
6653	UnderlyingSecurityType	Y	product category, e.g. Stock, ETF, Index, etc.
5336	UnderlyingID	N	LegSecurityID for the stock leg of a stock-complex instrument
6426	UnderlyingSymbol	N	Stock symbol
75	TradeDate	Y	Current business date
6254	StartTime	Y	product opens for trading
6255	EndTime	Y	product closes
453	NoPartyIDs	Y	Always 1
448	> PartyID	Y	Primary Market Maker (PMM)
1205	NoTickRules	Y	Price Step (Tick) Table
8596	> TickRuleID	Y	Instrument Type: 1=Simple, 2=Combination, 3=Stock Combination

Tag	Tag Name	Req	Description
1206	> StartTickPriceRange	Y	
1207	> EndTickPriceRange	Y	
1208	> TickIncrement	Y	
1235	NoMatchRules	Y	Allocation rules for matching
8597	> MatchRuleID	Y	Instrument Type: 1=Simple, 2=Combination, 3=Stock Combination
1142	> MatchAlgorithm	Y	PT=Price time, PR=Pro-rata
8595	> CustomerPriorityIndicator	Y	0=no priority for customer orders 1=priority for customer orders
8594	NoAuctionRules	Y	
8598	> AuctionRuleID	Y	Instrument Type: 1=Simple, 2=Combination, 3=Stock Combination
8522	> AuctionType	Y	See **Appendix C: FIX Field Descriptions** on page 120
1141	NoMDFeedTypes	Y	
1022	> MDFeedType	Y	POS=Pre-Open Simple POC=Pre-Open Complex TBS=Top-of-Book Simple TBC=Top-of-Book Complex TIS=Ticker Simple TIC=Ticker Complex PDS=Price Depth Simple PDC=Price Depth Complex OBS=Order on Book Simple OBC=Order on Book Complex POSB=Pre-Open Simple Binary POCB=Pre-Open Complex Binary TBSB=Top-of-Book Simple Binary TBCB=Top-of-Book Complex Binary TISB=Ticker Simple Binary TICB=Ticker Complex Binary
264	> MarketDepth	N	Number of Price levels if Depth feed
8590	> MDPrimaryFeedLineID	Y	IP Address A
8591	> MDPrimaryFeedLineSubID	Y	Port number for IP address A
8592	> MDSecondaryFeedLineID	N	IP Address B
8593	> MDSecondaryFeedLineSubID	N	Port number for IP address B

3.3.6 Instrument Snapshot

The Instrument Snapshot message provides a complete description of an instrument.

3.3.6.1 Format — Instrument Snapshot

The following table shows the format of the **Instrument Snapshot** message.

Table 16: Instrument Snapshot (Template ID 14)

Tag	Tag Name	Req	Description
34	MsgSeqNum	Y	
35	MsgType	Y	d=Security Definition
48	SecurityID	Y	
1227	ProductComplex	Y	Default=1. See **Appendix C: FIX Field Descriptions** on page 120
965	SecurityStatus	Y	1=Active (default) 2=Suspended 3=Active, closing orders only 4=Expired 5=Delisted
55	Symbol	N	OPRA root Symbol, only present for simple instruments
201	PutOrCall	N	0=Put, 1=Call. Only present for simple instruments.
541	MaturityDate	N	Only present for simple instruments
202	StrikePrice	N	Only present for simple instruments
231	ContractMultiplier	N	contract size, only present for simple instruments
206	OptAttribute	N	Only present for simple instruments
1194	ExerciseStyle	N	0=European, 1=American (default)
947	StrikeCurrency	N	<ISO 4217 values>
555	NoLegs	N	Used to describe complex instruments
602	> LegSecurityID	Y	instrument identifier of leg
623	> LegRatioQty	Y	
624	> LegSide	Y	1=Buy, 2=Sell
167	SecurityType	N	MLEG, only present for complex instruments
762	SecuritySubType	N	Vertical Calendar Straddle Strangle Other Non-Std Only present for complex instruments
864	NoEvents	N	1 (deactivation time)
865	> EventType	N	6=Inactivation
866	> EventDate	N	last trading day

Tag	Tag Name	Req	Description
1145	> EventTime	N	last point in time for trading (for FX options)
711	NoUnderlyings	N	Information about deliverable
311	> UnderlyingSymbol	Y	
246	> UnderlyingFactor	Y	
973	> UnderlyingCashAmount	N	
318	> UnderlyingCurrency	N	<ISO 4217 values>
1310	NoMarketSegments	Y	Always 1
1300	> MarketSegmentID	Y	Product ID

3.3.7 Product Incremental Message

The Product Incremental message is sent if a new product is added, or an existing product is changed or deleted.

3.3.7.1 Format — Product Incremental

The following table shows the format of the **Product Incremental** message.

Table 17: Product Incremental (Template ID 13)

Tag	Tag Name	Req	Description
34	MsgSeqNum	Y	
35	MsgType	Y	BV=Market Definition Update Report
1395	MarketUpdateAction	Y	A=Add, M=Modify, D=Delete
1301	MarketID	Y	XSIX=ISO 10383 MIC for ISE GMNI=ISO 10383 MIC for ISE Gemini MCRY=ISO 10383 MIC for ISE Mercury
1300	MarketSegmentID	Y	Product ID
1396	MarketSegmentDesc	N	
8599	MarketSegmentStatus	Y	1=Active, 2=Inactive
1325	ParentMktSegmID	N	Concatenation of Market Segment and Bin ID.
5948	PartitionID	N	Partition Number of Product
5949	BinID	N	Bin Assigned to this Product
6653	UnderlyingSecurityType	N	product category, e.g., Stock, ETF, Index , etc.
5336	UnderlyingID	N	LegSecurityID for the stock leg of a stock-complex instrument
6426	UnderlyingSymbol	N	Stock symbol
75	TradeDate	N	Current business date in the system
6254	StartTime	N	Product opens for trading
6255	EndTime	N	Product closes
453	NoPartyIDs	Y	1 – to specify Primary Market Maker
448	> PartyID	Y	Primary Market Maker (PMM)
1205	NoTickRules	N	Price Step table

Tag	Tag Name	Req	Description
8596	> TickRuleID	Y	Instrument Type: 1=Simple, 2=Combination, 3=Stock Combination
1206	> StartTickPriceRange	Y	
1207	> EndTickPriceRange	Y	
1208	> TickIncrement	Y	
1235	NoMatchRules	N	Allocation rules for matching
8597	> MatchRuleID	Y	Instrument Type: 1=Simple, 2=Combination, 3=Stock Combination
1142	> MatchAlgorithm	Y	PR=Pro-Rata, PT=Price Time
8595	> CustomerPriorityIndicator	Y	0=no priority, 1=Priority
8594	NoAuctionRules	N	
8598	> AuctionRuleID	Y	Instrument Type: 1=Simple, 2=Combination, 3=Stock Combination
8522	> AuctionType	Y	See **Appendix C: FIX Field Descriptions** on page 120
1141	NoMDFeedTypes	N	
1022	> MDFeedType	Y	POS=Pre-Open Simple POC=Pre-Open Complex TBS=Top-of-Book Simple TBC=Top-of-Book Complex TIS=Ticker Simple TIC=Ticker Complex PDS=Price Depth Simple PDC=Price Depth Complex OBS=Order on Book Simple OBC=Order on Book Complex POSB=Pre-Open Simple Binary POCB=Pre-Open Complex Binary TBSB=Top-of-Book Simple Binary TBCB=Top-of-Book Complex Binary TISB=Ticker Simple Binary TICB=Ticker Complex Binary
264	> MarketDepth	N	Use for price depth feeds of book data
8590	> MDPrimaryFeedLineID	Y	IP Address A
8591	> MDPrimaryFeedLineSubID	Y	Port number for IP address A
8592	> MDSecondaryFeedLineID	N	IP Address B
8593	> MDSecondaryFeedLineSubID	N	Port number for IP address B

3.3.8 Instrument Incremental Message

The Instrument Incremental message is sent if a new instrument is added, or an existing instrument is changed or deleted.

3.3.8.1 Format — Instrument Incremental

The following table shows the format of the **Instrument Incremental** message.

Table 18: Instrument Incremental (Template ID 15)

Tag	Tag Name	Req	Description
34	MsgSeqNum	Y	
35	MsgType	Y	BP=Security Definition Update Report
980	SecurityUpdateAction	Y	A=Add, M=Modify, D=Delete
48	SecurityID	Y	Binary ID for the lifetime of the instrument
1227	ProductComplex	N	Only present for complex instruments. See **Appendix C: FIX Field Descriptions** on page 120
965	SecurityStatus	N	1=Active (default) 2=Suspended 3=Active, closing orders only 4=Expired 5=Delisted
55	Symbol	N	OPRA root Symbol, only for simple instruments
201	PutOrCall	N	0=Put, 1=Call
541	MaturityDate	N	Only present for simple instruments
202	StrikePrice	N	Only present for simple instruments
231	ContractMultiplier	N	contract size, only present for simple instruments
206	OptAttribute	N	Only present for simple instruments
1194	ExerciseStyle	N	0=European, 1=American
947	StrikeCurrency	N	<ISO 4217 values>
555	NoLegs	N	Used to describe complex instruments
602	> LegSecurityID	Y	
623	> LegRatioQty	Y	
624	> LegSide	Y	1=Buy, 2=Sell
167	SecurityType	N	MLEG, only present for complex instruments
762	SecuritySubType	N	Vertical Calendar Straddle Strangle Other Non-Std — Only present for complex instruments
864	NoEvents	N	deactivation time
865	> EventType	N	6=Inactivation
866	> EventDate	N	last trading day
1145	> EventTime	N	last point in time for trading (for FX options)
711	NoUnderlyings	Y	
311	> UnderlyingSymbol	Y	
246	> UnderlyingFactor	Y	

Tag	Tag Name	Req	Description
973	> UnderlyingCashAmount	N	
318	> UnderlyingCurrency	N	<ISO 4217 values>
1310	NoMarketSegments	Y	Instrument is associated with only a single product
1300	> MarketSegmentID	Y	Product ID

3.3.9 Start of Snapshot Cycle Message

The **Start of Snapshot Cycle** message indicates the start of a snapshot cycle and gives the time at which all messages in the snapshot were prepared. It also provides the sequence number of the last Incremental message sent on the Reference Data Incremental feed.

3.3.9.1 Format — Start of Snapshot Cycle

The following table shows the format of the **Start of Snapshot Cycle** message.

Table 19: Start of Snapshot Cycle (Template ID 16)

Tag	Tag Name	Req	Comments
34	MsgSeqNum	Y	
	LastMsgSeqNumProcessed	Y	
	SnapshotCreationTime	Y	

NOTE: Shaded fields indicate non-standard, ISE-defined fields.

3.3.10 End of Snapshot Cycle Message

The **End of Snapshot Cycle** message indicates the end of a snapshot cycle and gives the time at which all messages in the snapshot were prepared. It also provides both the number of products and number of instruments contained in the just completed snapshot cycle.

3.3.10.1 Format — End of Snapshot Cycle

The following table shows the format of the **End of Snapshot Cycle** message.

Table 20: End of Snapshot Cycle (Template ID 17)

Tag	Tag Name	Req	Comments
34	MsgSeqNum	Y	
	SnapshotCreationTime	Y	
	NumOfProducts	Y	
	NumofInstruments	Y	

NOTE: Shaded fields indicate non-standard, ISE-defined fields.

3.3.11 Heartbeat Message

The **Heartbeat** message is sent once per minute while the feed is open if nothing else is sent in that minute.

Please see Section **3.4.3, Heartbeat Message** on page 45 for a complete description of this message.

3.4 Other FAST Encoded Messages

This section describes various administrative, session, and FAST protocol messages.

If there are discrepancies between this document and the ISE's FAST template file, please conform to the template file.

3.4.1 FAST Reset Message

3.4.1.1 Purpose

This message is used to clear the application data cache. This message is the first message in each UDP packet.

3.4.1.2 Format — FAST Reset

The **FAST Reset** message does not contain any data fields. It is represented on the data stream by only the presence map, followed by the template ID, 120.

3.4.1.3 Example

The following is a hex dump of a reset message.

|RESET |

1) C0 F8

3.4.2 Block Header Message

3.4.2.1 Purpose

The **Block Header** message contains a Block Sequence Number, Sending Time, and other fields normally found in the FIX message header. For performance reasons, some fields have been moved from each FIX message header to the **Block Header** message, such as the timestamp.

Each network block begins with a **FAST Reset** message, followed by a **Block Header** message.

The fields in the header record are sent as Byte Vectors so that they occupy the same byte positions in every block. The fields always have the same length, as described below.

The header record contains some information about the environment producing the data feed. *Exchange* is "I" for ISE, "H" for ISE Gemini or "J" for ISE Mercury. *Area* is "P" or "S" for Production or Simulation. *Environment* is used when there are multiple test systems and has a value 1 to 99. Member Test-1 is 33, Member Test 2 is 34. Production is environment number 1.

A timestamp is only sent once in each header record, and represents when the packet was sent to the wire. It is expressed as microseconds (µs) since the UNIX epoch (midnight, 1/1/1970 GMT).

For example, the *SendingTime* Byte Vector 0x88 0004 91F5 EE5F D3E2 is translated to: 1,286,385,359,115,234, or Wed., Oct 6, 2010, 13:15:59.115234 EDT.

The comments below describe the byte positions within the UDP packet at which the field can be found without decoding the packet.

3.4.2.2 Format — Block Header

The following table shows the format of the **Block Header** message.

Table 21: Block Header (Template ID 1)

Tag	Field Name	Req	Data Type	Field Encoding	Bytes	Comment
	MsgPartition	Y	uInt32	Byte Vector	4	Partition number (1 to 99), found at byte 9 (offset 8) of the packet
	SequenceNumber	Y	uInt32	Byte Vector	4	Block sequence number, found at bytes 11 – 14 (offset 10) of the packet
	Exchange	Y	String	Byte Vector	1	I=ISE / H=ISE Gemini / J=ISE Mercury, found at byte 16 (offset 15) of the packet
	Area	Y	String	Byte Vector	1	P=Production / S=Simulation, found at byte 18 (offset 17) of the packet
	Environment	Y	uInt32	Byte Vector	4	Production is 1 (one), Member test is 33 or 34; found at byte 20 (offset 19) of the packet
	SendingTime	Y	uInt64	Byte Vector	8	µs from 1/1/1970, found at bytes 25 – 32 (offset 24) of the packet.

NOTE: All **Block Header** fields are non-standard, ISE-defined fields.

3.4.2.3 Example

The following example shows the start of a UDP packet. It begins with the **FAST Reset** message (shaded), followed by the **Block Header** message. This is then followed by other FIX application messages (not shown). The fields in the header record always occupy the same positions within the UDP packet. The data fields from the header record, as described above, are highlighted in the following example.

0000h: C0 F8 C0 81 84 00 00 00 01 84 00 01 F0 D2 81 49

0010h: 81 53 84 00 00 00 21 88 00 04 91 F5 EE 5F D3 E2

3.4.3 Heartbeat Message

3.4.3.1 Purpose

The **Heartbeat** message is sent to indicate activity on a feed if there are no other messages to send.

The *MsgSeqNum* field contains the sequence number of the *previous* FIX message, or zero, if no other FIX messages have been broadcast.

3.4.3.2 Format — Heartbeat

The following table shows the format of the **Heartbeat** message.

Table 22: Heartbeat (Template ID 10)

Tag	Tag Name	Req	Comments
35	MsgType	Y	0 (zero)=Heartbeat
34	MsgSeqNum	Y	*Previous* FIX MsgSeqNum or zero (0)

3.4.4 Sequence Number Reset Message

3.4.4.1 Purpose

The **Sequence Number Reset** message defines the next sequence numbers to expect on a block level and on a message level.

3.4.4.2 Format

The template for the **Sequence Number Reset** message has the following format:

Table 23: Sequence Number Reset (Template ID 11)

Tag	Tag Name	Req	Comments
34	MsgSeqNum	Y	Always set to 1
36	NewSeqNum	Y	Always set to 1
6591	BlockSeqNum	Y	Always set to 1

3.5 FAST Message Decoding

All FAST messages are encoded as per the FAST v1.1 specification. Consumers of the FAST feeds must use a FAST v1.1 decoder to decode the data.

On receipt of a UDP packet by the subscriber's application, the byte stream must be decoded.

A UDP packet contains one or more FAST messages. The first message in each packet is a **FAST Reset** (Template ID = 120) which resets the FAST dictionaries. Values are not cached across UDP packets.

The ISE implementation utilizes the following data types:

- decimal

- length
- string
- Int32/Int64
- uInt32/uInt64
- Byte Vector

The ISE implementation utilizes the following operators:

- constant
- copy
- default
- delta
- increment
- tail

The ISE implementation utilizes the following attributes:

- dictionary — “template” and “global”
- presence — “optional” and “mandatory” (If not otherwise explicitly stated, presence = mandatory)
- value

The ISE implementation utilizes both the “global” and the “template” dictionary caches. The template file specifies which dictionary cache to use for each field.

The maximum UDP packet size contains up to 1000 bytes of *application data*.

Messages may contain both optional groups and repeating groups.

A detailed description of the FAST protocol is beyond the scope of this document. Please see the FIX Protocol Ltd. website at http://fixprotocol.org for more information. Please see **Section 1.3, FIX/FAST-Related Documents** on page 8 for a complete list of the documents relevant to the ISE’s FAST implementation.

4. Binary Feed Descriptions

This section describes the binary data feeds and defines the messages used on those feeds. All data messages are defined within this document.

- All multicast data are sent in blocks (UDP packets) in which the *application data* do not exceed 1000 bytes.
- The first message in each block is a **Block Header** message.
- The **Block Header** message identifies the message type (*MsgType* field) and related product (*MarketSegmentID* field) for *all* messages contained in the block.
    - For example, all messages in this block are snapshot m*essages* for AAPL.
- Messages of differ*ent message type*s are *not* m*ixe*d in a single block.
- The **Block Header** contains a count (*MsgCount* field) of data messages contained within the block.
    - *MsgCount* does *not* include the **Block *Header*** message itself.
- The rest of the block contains *MsgCount (zero* or m*ore*) data messages of the type specified.
- Price data are represented in integer format, only, with an inferred exponent as defined by the field's data type.
- All integer data are represented in **Little-Endian** (LE) byte order.
- All feeds are disseminated in duplicate over two multicast streams, as is done for OPRA, and referred to as the A feed and the B feed. The reference data provide the IP/Port addresses of each feed, both A and B, for each product.

4.1 Data Messages

The binary feeds are comprised of fifteen discrete data messages. Two messages are for feed and data management:

- **Block Header**
- **Heartbeat**

The following messages are for market data:

- **Quote**
- **Long Quote**
- **Combo Quote**
- **Ticker**
- **Snapshot**
- **Combo Snapshot**
- **Mass Status**
- **Status**
- **Combo Status**
- **Depth Incremental**
- **Combo Depth Incremental**
- **Depth Snapshot**
- **Combo Depth Snapshot**

- **Simple Instrument Order On Book**
- **Complex Instrument Order On Book**
- **Simple Instrument Auction**
- **Complex Instrument Auction**

Four messages are for reference data:

- **Product**
- **Instrument**
- **Complex Instrument**
- **RefData Cycle**

The message structures are described in following sections. All messages structures are static and all defined fields are required. Most messages are fixed-length. Variable length messages contain a variable number of fixed-length data vectors. The number of included data vectors is specified in the message itself; however, no single variable-length message can exceed 984 bytes. All message definitions, below, include the message's minimum and, if applicable, maximum size.

Messages are associated with **message types**. A single message may be associated with more than one message type; however, a single message type can only be associated with one, and only one, message structure.

4.2 Data Types

For simplicity, all fields are defined with either a basic or extended data type. All extended data types are extensions of basic data types. A field's data type explicitly defines the usage and scope of the data carried in that field. Fields will be set to largest value representing NULL or No Value. For example, if the field is defined as UInt8, then NULL will be represented as 255 in a message.

NOTE: All integer data are represented in little-endian byte order.

The binary feed data types are defined in the following table.

Table 24: Binary Data Types

Data Type	Description
BASIC DATA TYPES	
Int8	signed 8-bit integer -128 to 127
Int64	signed 64-bit integer -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
UInt8	unsigned 8-bit integer 0 to 255
UInt16	unsigned 16-bit integer 0 to 65,535

Data Type	Description
UInt32	unsigned 32-bit integer 0 to 4,294,967,295
UInt64	unsigned 64-bit integer 0 to 18,446,744,073,709,551,615
EXTENDED DATA TYPES	
ASCII char	Int8 Any printable ASCII character in the range 0x20 - 0x7E E.g. 0x58 = 'X'
Decimal	UInt64 Base 10, inferred exponent -8. E.g. 315,000,000 = 3.15
Price – short	UInt16 Any price, $0 ≤ *p* ≤ $655.35. Base 10, inferred exponent -2. E.g. 315 = 3.15
Price – long	Int64 Any price. Base 10, inferred exponent -8. E.g. 315,000,000 = 3.15
Symbol	5 ASCII char Any five-character, left-justified, space-padded string. E.g. 0x58 0x59 0x5A 0x20 0x20 = 'X' 'Y' 'Z' ' ' ' ' ' = "XYZ"
Timestamp	UInt64 Microseconds since the UNIX epoch. E.g. 1,286,385,359,115,234 = Wed., Oct 6, 2010, 13:15:59.115234 EDT

4.3 Message Types

The **Block Header**, *MsgType* field identifies how the included data messages are to be handled. Some messages perform double- and triple-duty depending on the *MsgType* value. For example, in addition to identifying the messages contained in the block, the **Block Header** message is used as a sequence number reset message, and to signal both the start and end of a snapshot cycle.

Message types are defined in the structure for the **Block Header** message, below. Their usage is given in the descriptions of the feeds.

4.4 Feed/Data Management Messages

This section describes the messages used for feed and data management

- **Block Header**
- **Heartbeat**

4.4.1 Block Header Message

The **Block Header** is the first message in each UDP packet. It contains a strictly monotonically increasing sequence number for gap checking, and a timestamp indicating when the packet was sent out on the wire.

The **Block Header** identifies the message type of all messages within the packet. All messages in a packet are of one, and only one, message type (and therefore, are one, and only one, message structure).

The value of the *MsgCount* field is the number of data messages contained in the packet. That number does not include the **Block Header** message, itself.

If the packet contains market or reference data, the **Block Header** identifies the underlying product (*MarketSegmentID* field) for all data messages in the packet. All data messages in a packet are for one, and only one, product.

If the packet does not contain data (heartbeat, start/end snapshot cycle, or sequence number reset), *MarketSegmentID* is set equal to 0xFFFF ($65{,}535_{10}$).

If *MsgType* = 8 (SeqNo Reset), 15 (Start Snapshot Cycle), or 16 (End Snapshot Cycle), the **Block Header** message is the *only* message in the packet.

- Sequence Number Reset (*MsgType* = 8)
  - The value of the Block Header, SeqNo field is the current sequence number, and increases strictly monotonically, from this point, regardless of the previous sequence number.
  - The Block Header message is the only message in the packet.
- **Start Snapshot Cycle** (*MsgType* = 15)
  - This **Block Header** message marks the start of a periodic snapshot cycle.
  - The **Block Header** message is the only message in the packet.
- **End Snapshot Cycle** (*MsgType* = 16)
  - This **Block Header** message marks the end of a periodic snapshot cycle.
  - The **Block Header** message is the only message in the packet.

4.4.1.1 Structure — Block Header

The following table shows the structure of the **Block Header** message.

Table 25: Block Header (Binary)

Pos	Name	Data Type	Values	Comment
1	SeqNo	UInt32		
2	SendingTime	Timestamp		
3	MsgType	UInt8	0=Heartbeat	

Pos	Name	Data Type	Values	Comment
			1=Quote 2=Long Quote 3=Combo Quote 4=Snapshot — Optional 5=Snapshot — Mandatory 6=Combo Snapshot — Optional 7=Combo Snapshot — Mandatory 8=SeqNo Reset 9=Ticker 10=Ticker Snapshot — Optional 11=Ticker Snapshot — Mandatory 12=Mass Status 13=Status 14=Combo Status 15=Start Snapshot Cycle 16=End Snapshot Cycle 17=Simple Depth Incremental 18=Complex Depth Incremental 19=Simple Depth Snapshot Optional 20=Simple Depth Snapshot Mandatory 21=Complex Depth Snapshot Optional 22=Complex Depth Snapshot Mandatory 23=Simple Order On Book 24=Complex Order On Book 25=Simple Auction 26=Complex Auction 100=Add Product 101=Change Product 102=Delete Product 103=Product Snapshot 104=Add Simple Instrument 105=Change Simple Instrument 106=Delete Simple Instrument 107=Simple Instrument Snapshot 108=Add Complex Instrument 109=Change Complex Instrument 110=Delete Complex Instrument 111=Complex Instrument Snapshot 112=Start RefData Snapshot Cycle 113=End RefData Snapshot Cycle	
4	MarketSegmentID	UInt16		Product ID.
5	MsgCount	UInt8		

4.4.2 Heartbeat Message

The Heartbeat message is sent on a continuous, periodic basis, on every feed, regardless of other traffic on the feed.

The Heartbeat identifies the exchange, the partition, and MDI version number.

4.4.2.1 Structure — Heartbeat

The following table shows the structure of the Heartbeat message.

Table 26: Heartbeat (Binary)

Pos	Name	Data Type	Values	Comment
1	Partition	UInt8		
2	Exchange	ASCII Char	'I'=ISE 'H'=ISE Gemini 'J'=ISE Mercury	OPRA exchange code
3	Version	UInt8		MDI version number
Associated message type(s): MsgType = 0 (Heartbeat)				

4.5 Top Quote Feed

The Top Quote feed is described below:

- Purpose
- Messages
- Concepts
- Daily Schedule
- Quote Message
- Long Quote Message
- Combo Quote Message
- Snapshot Message
- Combo Snapshot Message
- Ticker Message
- Mass Status Message
- Status Message
- Combo Status Message

4.5.1 Purpose

Top Quote provides subscribers with the ISE Best Bid and Offer (IBBO) at the top price level of the order book, as well as trades and trading status information.

- All quotes and orders at the top price level are aggregated into the total quantity.
- The quantity of Customer Orders and Customer Professional orders are also supplied in separate fields.
- For complex instruments, the aggregated quantity that cannot trade through the away markets (NTT) is also indicated.

- Top Quote updates match those sent to OPRA (simple instruments only).
- Top Quote data are sent during pre-open and when the market is open for trading. See **Section Error! Reference source not found., Error! Reference source not found.** on page **Error! Bookmark not defined.** for more information.
- Top Quote for simple instruments and Top Quote for complex instruments are sent as separate streams.
- The Top Quote feed is available to all parties.

4.5.2 Messages

The Top Quote feed utilizes nine message structures:

Simple instruments, only:

- **Quote** — quote updates with very specific criteria
- **Long Quote** — quote updates when the **Quote** message cannot be used
- **Snapshot**
- **Mass Status** — trading state change for all (or many) instruments in a product
- **Status** — trading state change for a single instrument

Complex instruments only:

- **Combo Quote**
- **Combo Snapshot**
- **Combo Status**

Both instrument types:

- **Ticker** — new trades

The message types associated with each message are:

Table 27: Top Quote Binary Message Types

Message Structure	MsgType	Notes
Simple instruments:		
Quote	1=Quote	Updates with specific data criteria
Long Quote	2=Long Quote	
Snapshot	4=Snapshot – Optional 5=Snapshot – Mandatory	Process as necessary. Processing required (suggested).
Mass Status	12=Mass Status	Status change of all (most) instruments in a product.
Status	13=Status	Status change of a single instrument.
Complex instruments:		
Combo Quote	3=Combo Quote	
Combo Snapshot	6=Combo Snapshot — Optional 7=Combo Snapshot — Mandatory	Process as necessary. Processing required (suggested).

Message Structure	MsgType	Notes
Combo Status	14=Combo Status	Status change of a single instrument.
Both instrument types:		
Ticker	9=Ticker	Real-time trade.

4.5.3 Concepts

The Top Quote feed provides quote, trade, and status information for each Instrument.

An example snapshot of the top of the simple order book provided in the Top Quote feed:

Product: IBM, Product ID 427 Instrument: IBM 17Jun2011 80 C, Instrument ID 2026							
Status: – Regular (17)							
Trade Information: Last = 30@0.97, Open = 0.95, High = 0.99, Low = 0.92, Volume = 2050							
Bid				Offer			
Qty Market Orders=0				Qty Market Orders =0			
Price	Quantity	Cust	CustProf	Price	Quantity	Cust	CustProf
0.98	20	10	5	1.00	50	10	0

- Snapshots are sent using the **Snapshot** and **Combo Snapshot** messages, and are sent for each instrument at regular intervals throughout the day. The snapshot messages provide a description of each instrument along with a unique instrument identifier, the bid and offer quotes, trading state, and trade information. (The product ID is contained in the **Block Header** message.) Snapshot messages are sent in-band.
- Changes in trading state for *simple* instruments are sent using the **Status** message or the **Mass Status** message. The **Status** message is used when one instrument changes state independently, while the **Mass Status** message is used when all instruments in a product change state at the same time—for example, at the open, or at the close.
- Changes in trading state for *complex* instruments are sent using the **Combo Status** message.
- Updates to the IBBO are sent with the **Quote** or **Long Quote** message (simple instruments), or the **Combo Quote** message (complex instruments) during pre-open and regular trading. Each quote message updates only the bid or the offer. Trades are sent with the **Ticker** message.
- Quantity fields on the **Quote** message provide separate Customer and Customer Professional quantities, as well as the total quantity, which includes the Customer and Customer Professional quantities.
- The **Long Quote** and **Combo Quote** messages, in addition to the data above, include the quantities of Bid and Offer Market Orders. These quantities are *not* included in the total quantity, and are only filled when Market Orders are present, such as during Pre-Open trading state. (Complex instruments can have Market Orders on the book, unable to trade, during regular trading.)

- The snapshot message for simple instruments, only, contains the OSI name. The quote, ticker, and status messages contain only the instrument identifier. (The product ID is contained in the **Block Header** message.) Recipients can use these data to create a mapping table between the product and instrument identifiers, and the OSI names. (This information can also be obtained from the Reference Data feed.)
- The IBBO and trade information is disseminated at all times and can be received before the market has opened. Trade information can be disseminated after the market has closed.

4.5.4 Daily Schedule

The Top Quote feed commences each day at 6:00 a.m. ET with the dissemination of the snapshot messages. The instrument trading status is "Pre-open" (21).

- All Instruments receive a snapshot message every three minutes, with the messages spread evenly over the interval. Snapshot messages are populated with prices as they become available during pre-open.
- When the market opens at 9:30 a.m., a **Mass Status** (or **Status)** message is disseminated for each product (or instrument) with *Status* set to "Rotation" (22). This is the indication that the PMMs can open the market.
- When a product is rotated, a **Mass Status** (or **Status**) message is disseminated with *Status* set to "Regular" (17). Occasionally, individual instruments within a product cannot open. The **Status** message indicates the status of those instruments and provides further updates as the instruments are opened.
- Once an instrument is open, quote messages (**Quote, Long Quote,** and **Combo Quote**) are disseminated for every BBO change. Each message updates either the bid or the offer.
- If a single instrument changes state—for example, it halts because there are no quotes—then a **Status** or **Combo Status** message is sent for that one instrument.
- **Ticker** messages are sent for each trade and include the opening, high, and low prices; and total traded volume.
- When the market closes at either 4:00 p.m. or 4:15 p.m. ET, a **Mass Status** messages is sent for each product. (The **Product Snapshot** message on the Reference Data feed indicates the actual closing time for each product.)
- It is possible to receive **Ticker** messages before the markets open and after the close.
- Snapshot messages continue on the Top Quote feed, but with bid and offer set to zero, until the feed stops at 5:45 p.m.

4.5.5 Quote Message

The **Quote** message provides updates to the IBBO for simple instruments, only. Only the bid or the offer is reported in each message. Updates utilizing this message must meet the following criteria:

- Instrument ID < 4.29 billion (precludes complex instruments)
- Price ≤ $655.35
- Total size ≤ 65,535

- No market order quantity
- Instrument status is “Ready To Trade”

The **Block Header**, *MarketSegmentID* field identifies the product to which this message applies.

4.5.5.1 Structure — Quote

The following table shows the structure of the **Quote** message.

Table 28: Quote (Binary)

Pos	Name	Data Type	Values	Comment
1	SecurityID	UInt32		Instrument ID
2	Price	Price-short		
3	Size	UInt16		
4	ProCustSize	UInt16		Customer professional quantity
5	CustSize	UInt16		Customer quantity
6	Side	UInt8	0=Bid 1=Offer	
Associated message type(s): MsgType = 1 (Quote)				

4.5.6 Long Quote Message

The **Long Quote** message provides updates to the IBBO for simple instruments that do not otherwise meet the criteria for the **Quote** message. Only the bid or the offer is reported in each message.

The **Block Header**, *MarketSegmentID* field identifies the product to which this message applies.

4.5.6.1 Structure — Long Quote

The following table shows the structure of the Long Quote message.

Table 29: Long Quote (Binary)

Pos	Name	Data Type	Values	Comment
1	SecurityID	UInt64		Instrument ID
2	InstType	UInt8	1=Simple Instrument	
3	Price	Price-long		
4	MarketSize	UInt32		Market order quantity
5	Size	UInt32		Quantity
6	CustSize	UInt32		Customer quantity
7	ProCustSize	UInt32		Customer professional quantity
8	Side	UInt8	0=Bid 1=Offer	
Associated message type(s): MsgType = 2 (Long Quote)				

4.5.7 Combo Quote Message

The **Combo Quote** message provides updates to the IBBO for complex instruments. Only the bid or the offer is reported in each message.

The **Block Header**, *MarketSegmentID* field identifies the product to which this message applies.

NOTE: Through position eight, the **Combo Quote** and the **Long Quote** are the same structure.

4.5.7.1 Structure — Combo Quote

The following table shows the structure of the **Combo Quote** message.

Table 30: Combo Quote (Binary)

Pos	Name	Data Type	Values	Comment
1	SecurityID	UInt64		Product ID
2	InstType	UInt8	2=Standard Combination 3=Stock Combination	
3	Price	Price-long		
4	MarketSize	UInt32		Market order quantity
5	Size	UInt32		Quantity
6	CustSize	UInt32		Customer quantity
7	ProCustSize	UInt32		Customer professional quantity
8	Side	UInt8	0=Bid 1=Offer	
9	NTTMarketSize	UInt32		
10	NTTSize	UInt32		
Associated message type(s): MsgType = 3 (Combo Quote)				

4.5.8 Snapshot Message

The **Snapshot** message provides a continuous, periodic snapshot of a simple instrument. Data conveyed include the bid, offer, last trade, current trading status, and more. The value of the **Block Header**, *MsgType* field determines if the message is **optional** or **mandatory.**

Optional snapshots need only be processed as required. Mandatory snapshots *should* be processed, and are typically sent after a system restart to refresh the order book.

The *StateFlag* field indicates the current linkage handling and underlying equity limit states for the specified instrument. This field is a bit-string, using Least Significant Bit (LSB) 0 numbering (right-most bit is the least significant bit). Linkage handling is indicated in bit[0] and equity limit state is indicated in bit[1]. If the bit is off (‘0’), the state is disabled. If the bit is on (‘1’), the state is enabled.

The **Block Header**, *MarketSegmentID* field identifies the product to which this message applies.

4.5.8.1 Structure — Snapshot

The following table shows the structure of the **Snapshot** message.

Table 31: Snapshot (Binary)

Pos	Name	Data Type	Values	Comment
1	SecurityID	UInt64		Instrument ID
2	ProductComplex	UInt8	1=Simple Instrument	
3	Status	UInt8	1=Opening Delay 2=Trading Halt 17=Ready to trade 18=Not available for Trading 21=Pre-Open 22=Opening Rotation 23=Fast Market	
4	BidMarketSize	UInt32		
5	AskMarketSize	UInt32		
6	BidSize	UInt32		
7	BidCustSize	UInt32		
8	BidProCustSize	UInt32		
9	AskSize	UInt32		
10	AskCustSize	UInt32		
11	AskProCustSize	UInt32		
12	TradePrice	Price-long		
13	BidPrice	Price-long		
14	AskPrice	Price-long		
15	TradeSize	UInt32		
16	Volume	UInt32		
17	High	Price-long		
18	Low	Price-long		
19	First	Price-long		
20	Underlying Symbol	Symbol		Stock symbol
21	Symbol	Symbol		OSI symbol
22	PutOrCall	UInt8	0=Put 1=Call	
23	StateFlag	UInt8	$0 \le n \le 3$	Bit-string: bit[0]=Linkage handling indicator bit[1]=Equity limit indicator
24	StrikePrice	Decimal		
25	MaturityYear	UInt16		YYYY
26	MaturityMonth	UInt8		MM
27	MaturityDay	UInt8		DD

Pos	Name	Data Type	Values	Comment
Associated message type(s):			MsgType = 4 (Snapshot — optional) MsgType = 5 (Snapshot — mandatory)	

4.5.9 Combo Snapshot Message

The **Combo Snapshot** message provides a continuous, periodic snapshot of a complex instrument. Data conveyed include the bid, offer, last trade, current trading status, and more. The value of the **Block Header**, *MsgType* field determines if the message is **optional** or **mandatory.**

Optional snapshots need only be processed as required. Mandatory snapshots *should* be processed, and are typically sent when a new complex instrument is defined, or after a system restart to refresh the order book.

The **Block Header**, *MarketSegmentID* field identifies the product to which this message applies.

NOTE: Through position 20, the **Combo Snapshot** and the **Snapshot** are the same structure.

4.5.9.1 Structure — Combo Snapshot

The following table shows the structure of the **Combo Snapshot** message.

Table 32: Combo Snapshot (Binary)

Pos	Name	Data Type	Values	Comment
1	SecurityID	UInt64		
2	ProductComplex	UInt8	2=Standard Combination 3=Stock Combination	
3	Status	UInt8	2=Trading Halt 17=Ready to trade 18=Not available for Trading 21=Pre-Open	
4	BidMarketSize	UInt32		
5	AskMarketSize	UInt32		
6	BidSize	UInt32		
7	BidCustSize	UInt32		
8	BidProCustSize	UInt32		
9	AskSize	UInt32		
10	AskCustSize	UInt32		
11	AskProCustSize	UInt32		
12	TradePrice	Price-long		
13	BidPrice	Price-long		
14	AskPrice	Price-long		
15	TradeSize	UInt32		
16	Volume	UInt32		

Pos	Name	Data Type	Values	Comment
17	High	Price-long		
18	Low	Price-long		
19	First	Price-long		
20	Underlying Symbol	Symbol		Stock symbol
21	BidMarketNTTSize	UInt32		
22	AskMarketNTTSize	UInt32		
23	BidNTTSize	UInt32		
24	AskNTTSize	UInt32		
25	Items	UInt8		
<Leg Data Vector>				
25.1	LegSecurityID	UInt64		
25.2	LegType	UInt8	1=Option 2=Stock	
25.3	LegRatioQty	UInt16	$1 \le r \le 999$ $1 \le r \le 9999$	Option leg Stock leg
25.4	LegSide	UInt8	0=Buy 1=Sell	
Associated message type(s):		MsgType = 6 (Combo Snapshot — optional) MsgType = 7 (Combo Snapshot — mandatory)		

4.5.10 Ticker Message

The **Ticker** message is used to send real-time trade information.

The **Block Header**, *MarketSegmentID* field identifies the product to which this message applies.

4.5.10.1 Structure — Ticker

The following table shows the structure of the **Ticker** message.

Table 33: Ticker (Binary)

Pos	Name	Data Type	Values	Comment
1	SecurityID	UInt64		Instrument ID
2	LastPrice	Price-long		Most recent price
3	Size	UInt32		Last traded quantity.
4	Volume	UInt32		Total traded quantity
5	High	Price-long		High price for the day
6	Low	Price-long		Low price for the day
7	First	Price-long		Opening price for the day

Pos	Name	Data Type	Values	Comment
8	TradeCondition	UInt8	0=Exchange Last 1=Out of Sequence 2=Spread 3=Straddle 4=Combo 5=Stopped 6=Intermarket Sweep 7=Trade Through Exempt 8=Multi Asset Class Multileg Trade 9=Cancel Last 10=Cancel Open 11=Cancel Only 12=Cancel	
9	TradeTime	Timestamp		
Associated message type(s):			MsgType = 9 (Ticker — new trade) The following are associated with the Trade Feed, only: MsgType = 10 (Ticker Snapshot — optional) MsgType = 11 (Ticker Snapshot — mandatory)	

4.5.11 Mass Status Message

The **Mass Status** message is sent for simple instruments, only. It is sent when all, or most, instruments for a product change state at the same time; for example, at the opening or the close.

The **Block Header**, *MarketSegmentID* field identifies the product to which this message applies.

If one or more instruments cannot change state, the message contains an exception list identifying the instruments that could not change.

If the **Mass Status** message contains exceptions, then a **Status** message is sent for each instrument in the exception list, identifying that instrument's actual status, immediately following the **Mass Status** message.

If the number of exceptions exceeds 122, then this message is *not* sent. **Status** messages for *every* instrument are sent instead.

4.5.11.1 Structure — Mass Status

The following table shows the structure of the **Mass Status** message.

Table 34: Mass Status (Binary)

Pos	Name	Data Type	Values	Comment
1	InstType	UInt8	1=Simple Instrument	
2	Status	UInt8	1=Opening Delay 2=Trading Halt 17=Ready to trade 18=Not available for Trading 21=Pre-Open 22=Opening Rotation 23=Fast Market	
3	Items	UInt8	$0 \le n \le 122$	Number of exceptions.
<Exception Data Vector>				
3.1	SecurityID	UInt64		Instrument ID
Associated message type(s): MsgType = 12 (Mass Status)				

4.5.12 Status Message

The **Status** message is sent when a *single* simple instrument changes state during the day. For example, when all quotes are removed from an instrument or if an instrument is manually halted by Market Operations. This message is also used to indicate changes in linkage handling or equity limit state (Limit Up/Down).

The **Block Header**, *MarketSegmentID* field identifies the product to which this message applies.

The **Status** message is also sent as a follow-up message to the **Mass Status** message, to identify an excepted instrument's actual status.

The *Event* field identifies the state change that is taking place: trading status, linkage handling status, or equity limit status. Only one state change is allowed.

If trading status is changing, the *Status* field indicates the new status and the *StateFlag* field is ignored.

If linkage handling or equity limit status is changing, the *StateFlag* field indicates the new status and the *Status* field is ignored.

The *StateFlag* field is a bit-string, using LSB 0 numbering (right-most bit is the least significant bit). Linkage handling is indicated in bit[0] and equity limit state is indicated in bit[1]. If the bit is off ('0'), the state is disabled. If the bit is on ('1'), the state is enabled.

4.5.12.1 Structure —Status

The following table shows the structure of the **Status** message.

Table 35: Status (Binary)

Pos	Name	Data Type	Values	Comment
1	SecurityID	UInt64		Instrument ID
2	InstType	UInt8	1=Simple Instrument	
3	Status	UInt8	1=Opening Delay 2=Trading Halt 17=Ready to trade 18=Not available for Trading 21=Pre-Open 22=Opening Rotation 23=Fast Market	
4	Event	UInt8	0=Equity limit state change 6=Trading status change 100=Linkage status change	
5	StateFlag	UInt8	$0 \le n \le 3$	Bit-string — bit[0]=Linkage handling indicator bit[1]=Equity limit indicator
Associated message type(s): MsgType = 13 (Status)				

4.5.13 Combo Status Message

The **Combo Status** message is sent when a *single* complex instrument changes state during the day. For example, when a new complex instrument is created or if an instrument is manually halted by Market Operations.

The **Block Header**, *MarketSegmentID* field identifies the product to which this message applies.

There is no mass status message for complex instruments.

NOTE: Through position 4, the **Combo Status** and **Status** message are the same structure.

4.5.13.1 Structure —Combo Status

The following table shows the structure of the **Combo Status** message.

Table 36: Combo Status (Binary)

Pos	Name	Data Type	Values	Comment
1	SecurityID	UInt64	Instrument ID	
2	InstType	UInt8	2=Standard Combination 3=Stock Combination	
3	Status	UInt8	2=Trading Halt 17=Ready to trade 18=Not available for Trading 21=Pre-Open	
4	Event	UInt8	6= Trading status change	
Associated message type(s): MsgType = 14 (Combo Status)				

4.6 Trade Feed

The Trade feed is described below:

- Purpose
- Messages
- Concepts
- Daily Schedule
- Ticker Message

4.6.1 Purpose

The Trade feed contains only trade information.

New trade messages are sent as trades occur.

Continuous, periodic trade snapshots are also sent, allowing for recovery of the last trade.

Trades for simple and complex instruments are sent as separate streams.

4.6.2 Messages

The Trade feed utilizes one message:

- **Ticker** — new trades and trade snapshots

The message types associated with this message are:

Table 37: Trade Feed Binary Message Types

Message Structure	MsgType	Notes
Ticker	9=Ticker 10=Ticker Snapshot — Optional 11=Ticker Snapshot — Mandatory	Real-time trade. Process as necessary. Processing required (suggested).

4.6.3 Concepts

The Trade feed contains only trade information.

New trade messages are sent as trades occur in the market and may occur at any time. Pre-open trades indicate adjustments to previous days' trades. Post-close trades indicate adjustments to this day's trades.

Trade snapshots are sent on a continuous, periodic basis, allowing for recovery of the last trade. Snapshots are sent only after an instrument has traded. If an instrument has not traded, snapshot messages are not sent for that instrument.

4.6.4 Daily Schedule

The Trade feed commences at 6:00 a.m. ET.

Trade messages are sent as trades occur in the market.

Snapshot messages are sent periodically once an instrument has traded, and for as long as the feed is active.

Heartbeats/snapshots continue until the feed is closed at 5:45 p.m. ET.

4.6.5 Ticker Message

The **Ticker** message is used to send real-time trade information and continuous, periodic snapshots of the most recent trade. The value of the **Block Header**, *MsgType* field determines the usage. Snapshots may be **optional** or **mandatory.**

Optional snapshots need only be processed as required. Mandatory snapshots *should* be processed, and are typically sent after a system restart to refresh the order book.

Please see **Section 4.5.10, Ticker Message** on page 60 for a full description of this message.

4.7 Depth of Market (Depth) Feed

The binary Depth feed is described as follows:

- Purpose
- Messages
- Concepts
- Daily Schedule
- Depth Incremental Message
- Combo Depth Incremental Message
- Depth Snapshot Message
- Combo Depth Snapshot Message

4.7.1 Purpose

The Depth feed provides subscribers with the bids and offers at the top five price levels of the order book. All quotes and orders at each price level are aggregated into the total quantity. The

quantity of Customer Orders and Customer Professional orders are also supplied in separate fields. Trade data are not present on this feed.

Depth for simple instruments and complex instruments are sent on separate multicast streams.

4.7.2 Messages

The Depth feed utilizes four messages:

Simple instruments Only:

- **Depth Incremental**
- **Depth Snapshot**
- **Mass Status** — trading state change for all (or many) instruments in a product
- **Status** — trading state change for a single instrument

Complex instruments Only:

- **Combo Depth Incremental**
- **Combo Depth Snapshot**
- **Combo Status**

Table 38 : Depth feed Message Types

Message Structure	MsgType	Notes
Simple instruments:		
Depth Incremental	17= Depth Incremental	
Depth Snapshot	19= Depth Snapshot - Optional 20= Depth Snapshot - Mandatory	Process as necessary. Processing required (suggested).
Mass Status	12=Mass Status	Status change of all (most) instruments in a product.
Status	13=Status	Status change of a single instrument.
Complex instruments:		
Combo Depth Incremental	18=Combo Depth Incremental	
Combo Depth Snapshot	6=Combo Depth Snapshot — Optional 7=Combo Depth Snapshot — Mandatory	Process as necessary. Processing required (suggested).
Combo Status	14=Combo Status	Status change of a single instrument.

4.7.3 Concepts

The Depth feed provides a view of five levels of depth, showing the aggregate quantity of quotes and orders on each price level. Customer quantity and Professional Customer quantity is also shown on each level.

Example: The top five price levels provided in the Depth feed:

Instrument: IBM 17Jun2011 80 C, Instrument ID 2026, Product ID 427								
Status: – Regular (17)								
Bid					Offer			
Qty Market Orders=-30					Qty Market Orders=100			
Level	Price	Qty	Cust	CustProf	Price	Qty	Cust	CustProf
1	0.98	20	10	5	1.00	50	0	10
2	0.97	30	0	10	1.01	30	0	0
3	0.96	10	5	5	1.03	10	5	0
4	0.94	80	40	0	1.05	10	0	0
5	0.93	10	0	10	1.08	10	0	0

- There are two differences between the messages in Depth vs. Top Quote:
  - The price level field, Price *Level*, indicates where each price is to be inserted or changed in the depth display.
  - **Depth Incremental** uses update actions "new," "change," and "delete" at each price level. **TOB Quote** uses only "new."

- Snapshots are sent using the **Depth Snapshot** message, and are sent for each instrument at regular three-minute intervals throughout the day. The **Depth Snapshot** message provides a description of each instrument along with a unique product identifier and instrument identifier, the bids and offers up to five levels, and trading state. The **Depth Snapshot** message does not contain trade information. Snapshot messages are sent in-band.
- Changes in trading state are sent using the **Status** message or the **Mass Status** message. The **Status** message is used when one instrument changes state independently, while the **Mass Status** message is used when all instruments in a product change state at the same time—for example, at the open, or at the close. Please see **section 4.5.10** and **4.5.11** for detail description of **Status** and **Mass Status** message.
- Once an instrument has opened, updates within any of the top five levels are sent with the **Depth Incremental** message.
- Quantity fields on the Depth messages provide separate Customer and Customer Professional quantities; as well as the total quantity, which includes the Customer and Customer Professional quantities.
- The **Depth Snapshot** message for simple instruments only contains the Instrument Identifiers, as well as the OSI name. The **Depth Incremental** and **Status** messages have only instrument identifier.(The product ID is contained in the **Block Header** message) Recipients can use the information in the **Depth Snapshot** messages to create a mapping table between the product identifiers and instrument identifiers to the OSI names. This information can also be obtained from the Reference Data feed.
- Market data are only disseminated on the Depth feed while the market is open for trading. At other times, the order book is shown to be empty.

4.7.4 Daily Schedule

The Depth feed is on the same schedule as the Top Quote feed. Please see **Section Error! Reference source not found., Error! Reference source not found.** on page **Error! Bookmark not defined.** for a complete description.

4.7.5 Depth Incremental Message

Changes in the price depth for simple instruments are reported on the depth incremental message.

The **Block Header**, *MarketSegmentID* field identifies the product to which this message applies.

Table 39: Depth Incremental Message (Binary)

Pos	Name	Data Type	Values	Comment
1	SecurityID	UInt64	Instrument ID	
2	InstType	UInt8	1=Simple Instrument	
3	bidMktSize	UInt32		
4	askMktSize	UInt32		
5	noOfDepthIncrementals	UInt8		
6	>updateAction	UInt8	0 = New, 1 = Change, 2 = Delete, 4 = Delete From	
7	> side	UInt8	0=Bid 1=Offer	
8	> level	UInt8		
9	> price	price-long	price-long	
10	>size	UInt32		
11	>custSize	UInt32		
12	>custProfSize	UInt32		
Associated message type(s): MsgType = 17				

4.7.6 Combo Depth Incremental Message

Changes in the price depth for complex instruments are reported on the combo depth incremental message.

Table 40: Combo Depth Incremental Message (Binary)

Pos	Name	Data Type	Values	Comment
1	SecurityID	UInt64	Instrument ID	
2	InstType	UInt8	2=Standard Combination 3=Stock Combination	

Pos	Name	Data Type	Values	Comment
3	bidMktSize	UInt32		
4	askMktSize	UInt32		
5	bidMktNTTSize	UInt32		
6	askMktNTTSize	UInt32		
7	noOfDepthIncrementals	UInt8		
8	>updateAction	UInt8	0 = New, 1 = Change, 2 = Delete, 4 = Delete From	
9	>side	UInt8	0=Bid 1=Offer	
10	>level	UInt8		
11	>price	price-long	price-long	
12	>size	UInt32		
13	>custSize	UInt32		
14	>custProfSize	UInt32		
15	>NTTSize	UInt32		
Associated message type(s): MsgType = 18				

4.7.7 Depth Snapshot Message

The **Snapshot** message provides a continuous, periodic snapshot of a simple instrument. Data conveyed include the bid, offer, current trading status, and more. The value of the **Block Header**, *MsgType* field determines if the message is **optional** or **mandatory.**

Optional snapshots need only be processed as required. Mandatory snapshots *should* be processed, and are typically sent after a system restart to refresh the order book.

The *StateFlag* field indicates the current linkage handling and underlying equity limit states for the specified instrument. This field is a bit-string, using Least Significant Bit (LSB) 0 numbering (right-most bit is the least significant bit). Linkage handling is indicated in bit[0] and equity limit state is indicated in bit[1]. If the bit is off ('0'), the state is disabled. If the bit is on ('1'), the state is enabled.

The **Block Header**, *MarketSegmentID* field identifies the product to which this message applies.

The following table shows the structure of the **Depth Snapshot** message.

Table 41: Depth Snapshot Message (Binary)

Pos	Name	Data Type	Values	Comment
1	SecurityID	UInt64	Instrument ID	
2	InstType	UInt8	1=Simple Instrument	

Pos	Name	Data Type	Values	Comment
3	Status	UInt8	1 =Opening Delay 2 =Trading Halt 17 =Ready to trade 18 =Not available for Trading 21 = Pre-Open 22 =Opening Rotation 23 = Fast Market	
4	bidMktSize	UInt32		
5	askMktSize	UInt32		
6	StateFlag	UInt8	bit[0]=Linkage handling indicator bit[1]=Equity limit indicator	
7	Underlying	Symbol		
8	Symbol	Symbol		
9	PutOrCall	UInt8	0=Put 1=Call	
10	StrikePrice	price-long	price-long	
11	MaturityYear	MaturityYear	YYYY	
12	MaturityMonth	MaturityMonth	MM	
13	MaturityDay	MaturityDay	DD	
14	noOfDepthEntries	UInt8		
15	> side	UInt8	0=Bid 1=Offer 2=Empty Book	
16	> level	UInt8		
17	> price	price-long		
18	>size	UInt32		
19	>custSize	UInt32		
20	>custProfSize	UInt32		
Associated message type(s): MsgType = 19 (Snapshot – Optional) MsgType = 20 (Snapshot –Mandatory)				

4.7.8 Combo Depth Snapshot Message

The **Combo Snapshot** message provides a continuous, periodic snapshot of a complex instrument. Data conveyed include the bid, offer, current trading status, and more. The value of the **Block Header**, *MsgType* field determines if the message is **optional** or **mandatory.**

Optional snapshots need only be processed as required. Mandatory snapshots *should* be processed, and are typically sent when a new complex instrument is defined, or after a system restart to refresh the order book.

The **Block Header**, *MarketSegmentID* field identifies the product to which this message applies.

The following table shows the structure of the **Combo Depth Snapshot** message.

Table 42: Combo Depth Snapshot Message (Binary)

Pos	Name	Data Type	Values	Comment
1	SecurityID	UInt64	Instrument ID	
2	InstType	UInt8	2=Standard Combination 3=Stock Combination	
3	Status	UInt8	2 =Trading Halt 17 =Ready to trade 18 =Not available for Trading 21 = Pre-Open	
4	bidMktSize	UInt32		
5	askMktSize	UInt32		
6	StateFlag	UInt8	Bit-string — bit[0]=Linkage handling indicator bit[1]=Equity limit indicator	
7	Underlying	Symbol		
8	bidMktNTTSize	UInt32		
9	askMktNTTSize	UInt32		
10	noOfLegs	UInt8		
11	>legSecurityID	UInt64		
12	>legSecurityType	UInt8	0=Option, 1=Stock	
13	>legRatio	UInt16		
14	>legSide	UInt8	0=Buy, 1=Sell	
15	noOfDepthEntries	UInt8		
16	> side	UInt8	0=Bid 1=Offer	
17	> level	UInt8		
18	> price	price-long	price-long	
19	>size	UInt32		
20	>custSize	UInt32		
21	>custProfSize	UInt32		
22	>NTTSize	UInt32		
Associated message type(s): MsgType = 21 (Snapshot – Optional) MsgType = 22 (Snapshot – Mandatory)				

4.8 Order Feed

The order on book binary feed is described as follows:

- Purpose
- Messages
- Concepts
- Daily Schedule
- Simple order on book message
- Complex order on book message

- Simple auction message
- Complex auction message

Table 43 Order feed Message Types

Message Structure	MsgType	Notes
Simple instruments:		
Simple Order On Book	23=Simple Order On Book	Provides information about order posted on the book
Simple Auction	25=Simple Auction	Provides simple instrument auction information.
Complex instruments:		
Complex Order On book	24=Complex Order On Book	Provides information about complex order posted on the book
Comple Auction	26=Complex Auction	Provides complex instrument auction information.

4.8.1 Purpose

The Order feed advises participants that a new order is now resting on the book. The quantity and price of the new order are disclosed. The Order feed also announces that a new Auction order is in the market. Auction orders include Flash, Facilitation, Solicitation, etc. For public (exposed) auctions, auction responses are also disclosed.

NOTE: Auction announcements are only available via the Order feed; there are no auction order broadcasts through the DTI.

4.8.2 Messages

The Order On Book feed includes the following messages:

- Simple Order On Book
- Simple auction message
- Complex Order On Book
- Complex auction message

The Order Feeds for **simple instruments and** for **complex instruments** are sent as separate streams.

4.8.3 Concepts

The purpose of this feed is simply to notify participants that a new order has arrived and is resting on the book. This feed is also used to announce the start and end of auctions (e.g. Flash, Facilitation, Solicitation, PIM, etc.), and public auction responses (complex exposure auctions).

- **Order on Book** messages are sent for any new orders that arrive and *rest* on the book. The message is sent even if the order is outside the current market.
- **Order on Book** messages are not sent for orders that fill or are canceled on entry; for resting orders that are modified, canceled or traded; or for quotes.
- **This feed cannot be used to build the ISE order book.**
- For Reserve orders, only the **displayed quantity** is disclosed.

4.8.4 Daily Schedule

The Order feed commences each day at 6:00 a.m. ET with **Heartbeats**.

Starting at 6:00 a.m. ET, Simple or Complex Order on Book messages are sent for new resting orders. The Order on Book messages continue until each instrument closes.

After market close (4:00 p.m./4:15 p.m. ET), only Heartbeats are sent until the feed closes at 5:45 p.m. ET.

4.8.5 Order On Book Message

The simple or complex **Order on Book** message is sent for each new order that rests on the order book. Each message describes one order, including price, size, order capacity, and, if the order is an Attributable order, may also disclose the identities of the sending and clearing firms.

For Underlying Price Continegency (UPC) Order , related Low price , High price and Price source is disclosed in UPCLow, UPCHigh and UPCContingentSide fields.

All-or-None orders are identified by the ExecFlag field.

4.8.5.1 Simple Order On Book Message

The following table shows the format of the **Simple Order on Book** message

Table 44: Simple Order On Book Message (Binary)

Pos	Name	Data Type	Values	Comment
1	SecurityID	UInt64	InstrumentID	
2	InstType	UInt8	1=Simple Instrument	
3	OrdType	UInt8	1=Market, 2=Limit	
4	Side	Uint8	0=Bid 1=Offer	
5	Price	Price-long		
6	Size	UInt32		
7	MinExecQty	UInt32		
8	ExecFlag	UInt8	0=None, 1=AON	
9	OrderCapacity	ASCII char 1	"C"=Customer "D"=Customer Professional "B"=Broker/Dealer "G"=Proprietary "N"=Away Market Maker "M"=Market Maker	
10	OwnerID	ASCII char 6	Spaces when not set	
11	Giveup	ASCII char 6	Spaces when not set	
12	CMTA	ASCII char 6	Spaces when not set	
Associated message type(s): MsgType = 23				

4.8.5.2 Complex Order On Book Message

The following table shows the format of the **Complex Order on Book** message.

Table 45: Complex Order On Book Message (Binary)

Pos	Name	Data Type	Values	Comment
1	SecurityID	UInt64	InstrumentID	
2	InstType	UInt8	2=Standard Combination 3=Stock Combination	
3	OrdType	Uint8	1=Market, 2=Limit	
4	Side	Uint8	0=Bid 1=Offer	
5	Price	Price-long		Set to MAX(INT64) for Market Order.
6	Size	UInt32		
7	MinExecQty	UInt32		
8	ExecFlag	UInt8	0=None, 1=AON	
9	OrderCapacity	ASCII char 1	"C"=Customer "D"=Customer Professional "B"=Broker/Dealer "G"=Proprietary "N"=Away Market Maker "M"=Market Maker	
10	OwnerID	ASCII char 6	Spaces when not set	
11	Scope	UInt8	1=Local (ignore away market) 2=National	
12	UPClow	price-long	UPC low price	
13	UPChigh	price-long	UPC high price	
14	UPCContingentSide	UInt8	0=Buy, 1=Sell	
15	noOfLegs	UInt8		
16	>legSecurityID	UInt64		
17	>legSecurityType	UInt8	1=Option, 2=Stock	
18	>legRatio	UInt16		
19	>legSide	UInt8	0=Buy, 1=Sell	
20	>giveup	ASCII char 6		
21	>CMTA	ASCII char 6		
Associated message type(s): MsgType = 24				

4.8.6 Auction Message

Simple or Complex auction message is sent for every new auction in the market. For Block auctions, some fields, including price, size and side, may not be disclosed, in which case the fields are set to max value for the field datatype. For exposed auctions (complex exposure), noOfAuctionResponses repeating group specifies only the aggregate quantity at the best response

price. For closed auctions (Block, Facilitation,Solicitation, PIM and Flash) noOfAuctionResponses field will be 0 since responses are not exposed.

For AuctionEvent field in auction message indicate start of auction, end of auction or update to an existing ongoing auction.

4.8.6.1 Simple Auction Message

The following table shows the format of the **Simple Auction** message.

Table 46: Simple Auction Message (Binary)

Pos	Name	Data Type	Values	Comment
1	SecurityID	UInt64	InstrumentID	
2	InstType	UInt8	1=Simple Instrument	
3	OrdType	UInt8	1=Market, 2=Limit	
4	Side	UInt8	0=Bid; 1=Offer 2=Hidden Side	
5	Price	Price-long		
6	Size	UInt32		
7	ExchOrderID	UInt64		
8	OrderCapacity	ASCII char 1	"C"=Customer "D"=Customer Professional "B"=Broker/Dealer "G"=Proprietary "N"=Away Market Maker "M"=Market Maker	
9	ExecFlag	Uint8	0 = None 1 = AON	
10	OwnerID	ASCII char 6	Spaces when not set	
11	Giveup	ASCII char 6	Spaces when not set	
12	CMTA	ASCII char 6	Spaces when not set	
13	AuctionEvent	UInt8	0=Start 1=Auction Update 2=End of Auction	
14	AuctionType	UInt8	See Appendix C for valid values.	
15	noOfAuctionResponses	UInt8		
16	>Price	Price-long		
17	>Size	UInt32		
Associated message type(s): MsgType = 25				

4.8.6.2 Complex Auction Message

The following table shows the format of the Complex Auction message.

Table 47: Complex Auction Message (Binary)

Pos	Name	Data Type	Values	Comment
1	SecurityID	UInt64	InstrumentID	
2	InstType	UInt8	2=Standard Combination 3=Stock Combination	
3	OrdType	UInt8	1=Market, 2=Limit	
4	Side	UInt8	0=Bid; 1=Offer	
5	Price	Price-long		Set to MAX(INT64) for Market Order.
6	Size	UInt32		
7	ExchOrderID	UInt64		
8	OrderCapacity	ASCII char 1	"C"=Customer "D"=Customer Professional "B"=Broker/Dealer "G"=Proprietary "N"=Away Market Maker "M"=Market Maker	
9	ExecFlag	Uint8	0 = None;1 = AON	
10	OwnerID	ASCII char 6	Spaces when not set	
11	AuctionEvent	UInt8	0=Start 1=Auction Update 2=End of Auction	
12	AuctionType	UInt8	See Appendix C for valid values.	
13	Scope	UInt8		
14	noOfAuctionResponses	UInt8		
15	>Price	Price-long		
16	>Size	UInt32		
17	noOfLegs	UInt8		
18	>legSecurityID	UInt64		
19	>legSecurityType	UInt8	1=Option 2=Stock	
20	>legRatio	UInt16		
21	>legSide	UInt8	0=Buy, 1=Sell	
22	>Giveup	ASCII char 6		
23	>CMTA	ASCII char 6		
Associated message type(s): MsgType = 26				

4.9 Reference Data Feed

The Reference Data Feed is described below:

- Purpose
- Messages
- Concepts
- Daily Schedule
- Product Message
- Instrument Message
- Complex Instrument Message
- RefData Cycle Message

4.9.1 Purpose

The RefData feed continuously streams a complete list of all products and instruments (simple and complex) traded at the ISE.

The RefData is actually two separate feeds:

- The **RefData Snapshot** feed — provides a continuous cycle of all product and instrument definitions on one-minute intervals.
- The **RefData Incremental** feed — provides real-time information about products and instruments that are added, changed, or deleted intraday.

NOTE: Updates (add/change/delete) shown on the RefData Incremental feed appear in the *next* snapshot cycle.

Reference data for simple and complex instruments appear on the *same* feed.

4.9.2 Messages

The binary RefData feed utilizes four message structures:

- **Product** — product definition
- **Instrument** — simple instrument definition
- **Complex Instrument** — complex instrument definition
- **RefData Cycle** — correlation data about the snapshot cycle

The message types associated with each message are:

Table 48: Reference Data Binary Message Types

Message Structure	MsgType	Notes
RefData Snapshot Feed:		
Product	103=Product Snapshot	
Instrument	107= Instrument Snapshot	

Message Structure	MsgType	Notes
Complex Instrument	111=Complex Instrument Snapshot	
RefData Cycle	112=Start RefData Snapshot Cycle 113=End RefData Snapshot Cycle	
RefData Incremental Feed:		
Product	100=Add Product 101=Change Product 102=Delete Product	
Instrument	104=Add Instrument 105=Change Instrument 106=Delete Instrument	
Complex Instrument	108=Add Complex Instrument 109=Change Complex Instrument 110=Delete Complex Instrument	

4.9.3 Concepts

The RefData Snapshot feed is a complete snapshot of all reference data (products and instruments) and is sent in a continuous cycle throughout the day.

- A complete snapshot cycle starts with the **Block Header, Start RefData Snapshot Cycle** message, and proceeds with a **Product Snapshot**, followed by **Instrument Snapshot** and **Complex Instrument Snapshot** messages for that product. A product and all its instruments are defined before the definition of the next product and its instruments. The cycle ends with the **Block Header, End RefData Snapshot Cycle** message. The next full cycle begins immediately.
- When describing the instruments, simple instruments are defined first, then the complex instruments. All instruments for a product are defined before moving on to the next product.
- If products or instruments are added, changed, or deleted intraday, those changes are immediately reported on the RefData Incremental feed, using the appropriate **Product** or **Instrument/Complex Instrument** messages.
- The system generates the snapshot messages for *all* products and instruments at the start of the snapshot cycle, which are then disseminated smoothly over the snapshot interval. If a product or instrument is added (changed, deleted) after a snapshot cycle has started, the change is not reflected in the snapshot until the *next* full cycle.

NOTE: Some reference data are included in the **Top Quote** feed snapshot messages. Those messages supply minimal, but sufficient data to map the ISE Product and Instrument IDs to regular OSI Symbology.

4.9.4 Daily Schedule

The RefData Feeds (Snapshot and Incremental) start at 4:00 a.m. ET.

The snapshot feed continuously streams **Product** and **Instrument/Complex Instrument** messages, bounded by the **Start/End RefData Snapshot Cycle** messages.

The incremental feed sends **Heartbeats**, unless a change is made to the reference data, at which point a **Product** or **Instrument/Complex Instrument**, as appropriate, is sent. **Product** and **Instrument/Complex Instrument** messages may be sent at any time.

The RefData feeds close at 5:30 p.m. ET

4.9.5 Product Message

A **Product** message provides the current, complete description of a product, including its trading parameters and the market data feed channels over which its instrument market data are streamed.

The **Product** message is a 26-byte static structure followed by one instance each of four variable length data vectors, each of which contains a variable number of *item* data vectors:

- *Match Algorithm* vector (1 byte + 2-byte *item* vector)
- *Auction Type* vector (1 byte + 2-byte *item* vector)
- *Tick Rule* vector (1 byte + 17-byte *item* vector)
- *Feed Type* vector (1 byte + 13-byte *item* vector)

Tick increments, matching algorithms, and auctions are defined for each of the instrument types (Simple, Combination, and Stock Combination). The complete definition of the tick increments, for example, has one set of Tick Rule data vectors for simple instruments, another set for combination instruments, and a third set for stock combination instruments.

The tick increment definitions contain only the starting price for a tick price range — the end price of the range must be inferred. For example, a product whose simple instruments are priced $0.05 below $3.00 and $0.10 above $3.00 contains two Tick Rule item vectors for *InstType* = 1, the first indicating a start range of $0 and a tick increment of $0.05, and the second indicating a start range of $3.00 and a tick increment of $0.10. It must be inferred from the data given that the end of the price range for the $0.05 increment is $3.00, and the end of the price range for the $0.10 increment is the maximum price at the exchange.

The **Block Header**, *MarketSegmentID* field identifies the product to which this message applies.

4.9.5.1 Structure — Product

The following table shows the structure of the **Product** message.

Table 49: Product (Binary)

Pos	Name	Data Type	Values	Comment
1	Partition	Uint8		
2	Bin	Uint8		
3	SecType	Uint8	1=Agency 2=Commodity 3=Corporate 4=Currency 5=Equity	

Pos	Name	Data Type	Values	Comment
			6=Government 7=Index 8=Loan 9=Money Market 10=Mortgage 11=Municipal 12=Other 13=Financing 14=ETF	
4	UnderlyingID	Uint32		LegSecurityID for the stock leg of a stock-complex instrument. 0xFFFFFFFF if non-stock.
5	Underlying	Symbol		Stock Symbol
6	CurrentYear	Uint16	YYYY	Current Business Year
7	CurrentMonth	Uint8	MM	Current Business Month
8	CurrentDay	Uint8	DD	Current Business Day
9	StartHour	Uint8	HH	Start of trading
10	StartMinute	Uint8	MM	
11	EndHour	Uint8	HH	End of trading
12	EndMinute	Uint8	MM	
13	PMM	6 ASCII char		E.g., “CDL01M”
<Match Algorithm Vector> — One instance				
14	Items	Uint8	$n \geq 1$	
<Match Algorithm Item Vector> — *n* instances				
14.1	InstType	Uint8	1=Simple Instrument 2=Standard Combination 3=Stock Combination	
14.2	MatchAlgorithm	Uint8	0=Price time 1=Pro rata	
<Auction Type Vector> One instance				
15	Items	Uint8	$n \geq 0$	
<Auction Item Vector> — *n* instances				
15.1	InstType	Uint8	1=Simple Instrument 2=Standard Combination 3=Stock Combination	
15.2	AuctionType	Uint8	**Simple Instruments:** 1=Block 2=Directed Order 3=Flash 6=Facilitation 9=Solicitation 12=PIM 15=Directed Order PIM	

Pos	Name	Data Type	Values	Comment
			Standard combination: 4=Exposure 7=Facilitation 10=Solicitation 13=PIM **Stock combination:** 5=Exposure 8=Facilitation 11=Solicitation 14=PIM	
<Tick Rule Vector> One instance				
16	Items	Uint8	$n \geq 1$	
<Tick Rule Item Vector> — *n* instances				
16.1	InstType	Uint8	1=Simple Instrument 2=Standard Combination 3=Stock Combination	
16.2	StartTickPrice	Price-long		The end of tick range is not provided and must be inferred.
16.3	TickIncr	Price-long		
<Feed Type Vector> One instance				
17	Items	Uint8	$n \geq 1$	
<Feed Type Item Vector> — *n* instances				
17.1	FeedType	Uint8	**Binary Feeds:** 0=POSB (Pre-Open Simple) 1=POCB (Pre-Open Complex) 2=TBSB (Top-of-Book Simple) 3=TBCB (Top-of-Book Complex) 4=TISB (Ticker Simple) 5=TICB (Ticker Complex) 6=PDSB (Price Depth Simple) 7=PDCB(Price Depth Complex) 8=OBSB(Order on Book Simple) 9=OBCB(Order on Book Complex) **FAST Feeds:** 10=POS (Pre-Open Simple) 11= POC(Pre-Open Complex) 12=TBS(Top-of-Book Simple) 13=TBC(Top-of-Book Complex) 14=TIS (Ticker Simple) 15=TIC (Ticker Complex) 16=PDS (Price Depth Simple) 17=PDC (Price Depth Complex)	

Pos	Name	Data Type	Values	Comment
			18=OBS (Order on Book Simple) 19=OBC (Order on Book Complex)	
17.2	PrimeAddr	Uint32		To be read logically as: 4th octet.3rd octet.2nd octet.1st octet. E.g., the IP address 224.0.75.70 would appear as $3,758,115,654_{10}$
	4th octet	Uint8		
	3rd octet	Uint8		
	2nd octet	Uint8		
	1st octet	Uint8		
17.3	PrimePort	Uint16		
17.4	SecondAddr	Uint32		To be read logically as: 4th octet.3rd octet.2nd octet.1st octet
	4th octet	Uint8		
	3rd octet	Uint8		
	2nd octet	Uint8		
	1st octet	Uint8		
17.5	SecondPort	Uint16		
Associated message type(s):		**Incremental Feed:** MsgType = 100 (Add Product) MsgType = 101 (Change Product) MsgType = 102 (Delete Product) **Snapshot Feed:** MsgType = 103 (Product Snapshot)		

4.9.6 Instrument Message

An **Instrument** message provides the current, complete description of a simple instrument, only.

The **Instrument** message is a 53-byte static structure followed by one instance each of two variable length data vectors, each of which contains a variable number of *item* data vectors:

- *Security Deliverable* vector (1 byte + 13-byte *item* vector)
- *Cash Deliverable* vector (1 byte + 11-byte *item* vector)

The **Block Header**, *MarketSegmentID* field identifies the product to which this message applies.

4.9.6.1 Structure —Instrument

The following table shows the structure of the **Instrument** message.

Table 50: Instrument (Binary)

Pos	Name	Data Type	Values	Comment
1	SecurityID	Uint64		
2	Symbol	Symbol		OSI Symbol
3	MaturityYear	Uint16	YYYY	
4	MaturityMonth	Uint8	MM	
5	MaturityDay	Uint8	DD	
6	StrikePrice	Decimal		

Pos	Name	Data Type	Values	Comment
7	ContractSize	Decimal		
8	PutOrCall	Uint8	0=Put 1=Call	
9	Modifier	Uint64		
10	SecStatus	Uint8	1=Active 3=Active for closing orders 4=Expired 5=Delisted	
11	Currency	3 ASCII char		E.g., “USD”
12	ExerciseStyle	Uint8	0=American 1=European	
13	LastTradeYear	Uint16	YYYY	
14	LastTradeMonth	Uint8	MM	
15	LastTradeDay	Uint8	DD	
16	LastTradeHour	Uint8	HH	Exchange local time
17	LastTradeMinute	Uint8	MM	
<Security Deliverable Vector> One instance				
18	Items	Uint8	$n \geq 0$	
<Security Deliverable Item Vector> *n* instances				
18.1	UnderlyingSymbol	symbol		
18.2	Factor	Decimal		
<Cash Deliverable Vector> One instance				
19	Items	Uint8	$n \geq 0$	
<Cash Deliverable Item Vector> *n* instances				
19.1	Currency	3 ASCII char		E.g., “USD”
19.2	CashAmt	Price-long		
Associated message type(s):	**Incremental Feed:** MsgType = 104 (Add Instrument) MsgType = 105 (Change Instrument) MsgType = 106 (Delete Instrument) **Snapshot Feed:** MsgType = 107 (Instrument Snapshot)			

4.9.7 Complex Instrument Message

A **Complex Instrument** message provides the current, complete description of a complex or stock-combination instrument.

The **Block Header**, *MarketSegmentID* field identifies the product to which this message applies.

4.9.7.1 Structure — Complex Instrument

The following table shows the structure of the **Complex Instrument** message.

Table 51: Complex Instrument (Binary)

Pos	Name	Data Type	Values	Comment
1	SecurityID	Uint64		
2	InstType	UInt8	2=Standard Combination 3=Stock Combination	
3	SpreadType	Uint8	1=Vertical 2=Calendar 3=Straddle 4=Strangle 5=Other 6=Non-standard	
4	Items	UInt8	$2 \le n \le 9$	
<Leg Data Vector>				
4.1	LegSecurityID	UInt64		
4.2	LegType	UInt8	1=Option 2=Stock	
4.3	LegRatio	UInt16	$1 \le r \le 999$ $1 \le r \le 9999$	Option leg Stock leg
4.4	LegSide	UInt8	0=Buy 1=Sell	
Associated message type(s):			**Incremental Feed:** MsgType = 108 (Add Complex Instrument) MsgType = 109 (Change Complex Instrument) MsgType = 110 (Delete Complex Instrument) **Snapshot Feed:** MsgType = 111 (Complex Instrument Snapshot)	

4.9.8 RefData Cycle Message

The **RefData Cycle** message is broadcast at the start and end of the reference data snapshot cycle to provide data about the snapshot cycle.

At the start of the snapshot cycle, the message provides the time at which the snapshot was created and the sequence number of the last incremental update on the Reference Data Incremental feed.

At the end of the snapshot cycle, the message reiterates the snapshot time and provides a raw count of all products and instruments in the just completed snapshot.

4.9.8.1 Structure — RefData Cycle

The following table shows the structure of the **RefData Cycle** message.

Table 52: RefData Cycle (Binary)

Pos	Name	Data Type	Values	Comment
1	LastSeqNo	Uint32		Valid data at “Start” 0xFFFFFFFF at “End”
2	SnapshotTime	Timestamp		
3	ProdCount	Uint32		Valid data at “End” 0xFFFFFFFF at “Start”
4	InstrCount	Uint32		Valid data at “End” 0xFFFFFFFF at “Start”
Associated message type(s):		**Snapshot Feed:** MsgType = 112 (Start RefData Snapshot Cycle) MsgType = 113 (End RefData Snapshot Cycle)		

5. Message Recovery

5.1 Introduction

This section discusses how messages can be recovered. The MDI transmits market data using UDP. The advantage of this type of transmission is the very low latency; however, there is no guarantee that all messages will be delivered. It is possible that network routers can join or split packets and even cause them to arrive out of sequence.

The data are sent in blocks such that the *application data* do not exceed 1000 bytes.

NOTE: This reduces the chance that a block is split into multiple network packets, although it does not remove this risk entirely.

Every **Block Header** message contains a sequence number so that recipients can detect missing blocks.

Example

Matching Engine (ME) partitions one through four may be on IP address 1. The block sequence numbers for blocks from ME-1 start at one and increment throughout the day. Blocks from ME-2 also start at one and increment throughout the day. It is possible that blocks from ME-2 appear on multiple IP addresses in which case they update separately. The blocks from ME-2 on IP-1 start from one and increment throughout the day, and the blocks from ME-2 on IP-2 start from one.

In the event that a block is missed on the A feed, it might be possible to recover the block from the B feed. If it is not possible to recover the block, then you must re-initialize the feed using the in-band snapshot data.

5.2 In-Band Recovery

The Top Quote feed, for example, sends incremental changes using the quote messages. In the event that a message is lost, the recipient must initialize the current state of every instrument on that feed using the snapshot messages.

The snapshot messages are sent on a continual basis, and do a complete cycle of all instruments every three minutes. Changes to the book are never sent using the snapshot messages; they always reflect the state of the book as of the last incremental message.

Once the recipient has processed a complete cycle of snapshot messages upon connection to the feed, further snapshot messages can be ignored. The snapshot messages account for less than 5% of the total bandwidth on each feed.

When joining a data feed, the recipient must first process the snapshot messages. Once processed, a snapshot for an instrument, the recipient may then begin to process incremental messages for

that instrument. Once the recipient gets a snapshot for a known instrument, you have processed a complete cycle of snapshots and you can then ignore further snapshot messages.

A snapshot may be received with *RefreshIndicator* set to "Y" (FAST feeds) or *MsgType* set to "mandatory" (binary feeds). This indicates that this snapshot message must be processed. Mandatory snapshots are sent at the start of the day, when an Instrument is added intra-day, or in the case of a failover to a backup server at ISE.

5.3 Out-Of-Band Recovery

The Reference Data (RefData) feed is sent as two feeds, the RefData Snapshot feed, and the RefData Incremental feed.

The Snapshot feed provides a complete snapshot of all reference data (products and instruments) at regular intervals throughout the day. Any changes to the reference data (additions, changes, deletes) are broadcast in real-time over the RefData Incremental feed.

When joining the RefData feed, the recipient must listen to the RefData Incremental feed *before* listening to the RefData Snapshot feed, and coordinate received incremental messages with the snapshot cycle.

In other words, snapshot messages represent a static point in time while incremental messages represent real-time. A message received on the Incremental feed before the **End of Snapshot Cycle** message on the Snapshot feed indicates a reference data change that is *not* part of that snapshot cycle—the recipient must hold, and apply that change once the complete snapshot has been processed.

Once the recipient has processed a complete snapshot cycle (with any received incremental messages), it is no longer necessary to listen to the RefData Snapshot feed — the RefData Incremental feed provides all necessary updates.

If an incremental message is lost, the recipient *must* again listen to the Snapshot feed and process a full snapshot cycle (with any received incremental messages). The full snapshot cycle takes approximately one minute.

5.4 Backup Feed Recovery

ISE sends each data feed on two Multicast streams, the A-feed and the B-feed. Members can receive the A-feed, or the B-feed, or both. Recipients should process both feeds and discriminate between the two by always taking the next data block from whichever feed provides it first.

6. Communications

6.1 Bandwidth Requirements

The following are **estimates** of the required bandwidth for each feed — FAST or binary — in the MDI. There may be growth due to changes in the market including the addition of new exchanges. The continued migration to pennies could also cause a significant increase in bandwidth requirements.

These specifications provide for 100% headroom based on the peak data rates as of April 2013.

All PMMs are **required** to have two data lines:

- A line to the primary data center in Secaucus, New Jersey
- A line to the backup data center in Clifton, New Jersey

All Market Data Feeds can be combined onto the same data lines, and used for FIX interface, Direct Trading Interface and PrecISE Trade.

Table 53: Bandwidth Requirements

Feed	ISE Options		ISE Gemini		ISE Mercury	
	A Stream	B Stream	A Stream	B Stream	A Stream	B Stream
Top Quote Feed	200 Mb	200 Mb	200 Mb	200 Mb	200 Mb	200 Mb
Depth of Market Feed	500 Mb	500 Mb	500 Mb	500 Mb	500 Mb	500 Mb
Order Feed	5Mb	5Mb	5Mb	5Mb	5Mb	5Mb
Pre-Open Feed	10Mb	10Mb	10Mb	10Mb	10Mb	10Mb
Reference Data	30Mb	30Mb	30Mb	30Mb	30Mb	30Mb
Spread Feed	25Mb	25Mb	25Mb	25Mb	25Mb	25Mb
Trade Feed	5Mb	5Mb	5Mb	5Mb	5Mb	5Mb

NOTE: There is no primary stream as the same server sends both A and B feeds. Members must also calculate total bandwidth if participating in ISE, ISE Gemini and ISE Mercury exchanges.

6.2 Service Providers

The ISE market data feeds are currently distributed by a number of managed service providers. These providers use advanced telecommunications protocols, designed to support a number of industry-standard protocols including IP and UDP as defined by the Internet Engineering Task Force (IETF).

The ISE market data feeds are disseminated using multicast via two redundant lines (A and B) intended to provide a level of fault tolerance.

The contacts for support and connectivity are as follows:

Table 54: Activ Financial Contact Information

Activ Financial		
Department	**Phone**	**Email**
Sales	212-599-1600	sales@activfinancial.com
Support	212-964-2600	mailto:clientservices@activfinancial.com

Table 55: Atrium Networks Contact Information

Atrium Networks		
Department	**Phone**	**Email**
Sales	212-387-2178	Contact-us@atriumnetwork.com
Support	212-387-2179	support@atriumnetwork.com

Table 56: BT Radianz Contact Information

BT Radianz		
Department	**Phone**	**Email**
Sales – Walt Terbrusch	212-205-1995	walter.terbrusch@bt.com
Support	877-228-1497	

Table 57: Essex Radez Contact Information

Essex Radez		
Department	**Phone**	**Email**
Sales	312-212-1815	sales@radez.com
Support		support@radez.com

Table 58: GuavaTech Contact Information

GuavaTech		
Department	**Phone**	**Email**
Sales	312-604-4300	sales@guavatech.com
Support	312-604-4444	support@guavatech.com

Table 59: Interactive Data 7ticks Contact Information

Interactive Data 7ticks		
Department	**Phone**	**Email**
Sales	212-771-6565 / 312-896-0300	info@interactivedata.com
Support	312-896-0302	support@7ticks.com

Table 60: NYSE Technologies – SFTI & Superfeed Contact Information

NYSE Technologies - SFTI (Secure Financial Transaction Infrastructure) and Superfeed		
Department	**Phone**	**Email**
Sales-Mike Misiaszek	212-510-3614	mmisiaszek@nyx.com
Support	800-873-7422	support@SIAC.com

Table 61: Options IT Contact Information

Options IT		
Department	**Phone**	**Email**
Sales	646-205-2586	sales@options-it.com
Support	646-205-2555	support@options-it.com

Table 62: Pico Quantitative Trading Contact Information

Pico Quantitative Trading		
Department	**Phone**	**Email**
Sales – Bruce Boytim	312-446-5766	Bruce.boytim@picotrading.com
Support		tradesupport@picotrading.com

Table 63: SAVVIS Contact Information

SAVVIS Financial Services		
Department	**Phone**	**Email**
Sales	800-463-8294	teamise@savvis.net
Support	800-639-6771	

Table 64: Verizon Contact Information

Verizon Financial Networking		
Department	**Phone**	**Email**
Sales	800-825-9163	VFNsales@lists.verizonbusiness.com
Support	800-838-7928	fsnoc@lists.verizonbusiness.com

Table 72: R2G Services LLC

R2G Services LLC		
Department	**Phone**	**Email**
Sales	312-834-9000	sales@r2g.com
Support	312-834-9000	support@r2g.com

For additional information, please send email to marketdata@ise.com.

6.3 Transmission Standards

The MDI utilizes the User Datagram Protocol (UDP) over IP version 4 (Ipv4), implemented as per the IETF protocol standards RFC 768 (UDP) and RFC 791 (Ipv4).

6.4 Failover

The architecture of the market data service is that our primary datacenter (Equinix NY4) is the source of all feeds. Each market data server outputs all feeds. Our disaster recovery data center (60 Broad St, NY) has the WAN ability to disseminate market data sourced from Equinix (via an ISE cross-site). Only in the event of a disaster is market data sourced from servers physically located in DR site.

6.4.1 Sequencing

Sequencing is the responsibility of a single server (primary and backup configuration). In the event of a server failure, a delay of several seconds may occur while the backup server resumes operation. In this instance, snapshot messages of all instruments are sent before the updates resume. These snapshots could include state changes of the book that have not been included in update messages, and must be processed by client systems to assure data integrity. The full refresh of the order book takes approximately three minutes.

When there has been a failure in the service at ISE, the *RefreshIndicator* field in the message is set to ”Y.” Setting *RefreshIndicator* to “Y” indicates that the subscriber should discard the contents in the order book completely and replace it with the contents of this snapshot message. The *RefreshIndicator* field also allows the subscriber to only process snapshots that are set to “Y” once the market is open.

6.5 Testing IP Groups

6.5.1 How to access the MDI in Member Simulation

The MDI disseminates market data and reference data over a multicast network. As market and reference data are not provided through the DTI, all DTI users must use the MDI as well.

All market participants are entitled to receive all test market and reference data streams.

Similar to the DTI, member simulation market and reference data can be received using modified lines to the current data center or Internet VPN.

The following tables provide the multicast channels used for the different feeds, depending on the type of connection used.

6.5.2 Member Test 1 (MT1) Environment

6.5.2.1 ISE Exchange

Table 65: MT1 Multicast Channels (ISE)

Feed Name		Ports	A Feed Source:207.231.198.199 or 200 or 201	B feed
FAST FEEDS				
Reference Data Feed	Snapshot	53150	224.0.75.1	224.0.75.7
	Incremental	53151		
Depth of Market Feed		53150 53250	224.0.75.2	224.0.75.8
Trade Feed		53151 53251	224.0.75.3	224.0.75.9
Order Feed		53150 53250	224.0.75.5	224.0.75.11
Spread Feed	Depth	53150 53250	224.0.75.6	224.0.75.12
	Top Quote	53151 53251		
	Order	53152 53252		
	Trade	53153 53253		
BINARY FEEDS				
Reference Data Feed	Snapshot	53150	224.0.75.13	224.0.75.19
	Incremental	53151		
Depth of Market Feed		53150 53250	224.0.75.14	224.0.75.20
Top Quote Feed		53150 53250	224.0.75.15	224.0.75.21
Trade Feed		53151 53251		
Order Feed		53150 53250	224.0.75.17	224.0.75.23
Spread Feed	Depth	53150 53250	224.0.75.18	224.0.75.24
	Top Quote	53151 53251		
	Order	53152 53252		
	Trade	53153 53253		

6.5.2.2 ISE Gemini Exchange

Table 66: MT1 Multicast Channels (Gemini)

Feed Name		Ports	A Feed Source:207.231.198.199 or 200 or 201	B feed Source:207.231.198.199 or 200 or 201
FAST FEEDS				
Reference Data Feed	Snapshot	53510	224.0.75.1	224.0.75.7
	Incremental	53511		
Depth of Market Feed		53510	224.0.75.2	224.0.75.8
Trade Feed		53511	224.0.75.3	224.0.75.9
Order Feed		53510	224.0.75.5	224.0.75.11
Spread Feed	Depth	53510	224.0.75.6	224.0.75.12
	Top Quote	53511		
	Order	53512		
	Trade	53513		
BINARY FEEDS				
Reference Data Feed	Snapshot	53510	224.0.75.13	224.0.75.19
	Incremental	53511		
Depth of Market Feed		53510	224.0.75.14	224.0.75.20
Top Quote Feed		53510	224.0.75.15	224.0.75.21
Trade Feed		53511		
Pre-Open Feed		53510	224.0.75.16	224.0.75.22
Order Feed		53510	224.0.75.17	224.0.75.23
Spread Feed	Depth	53510	224.0.75.18	224.0.75.24
	Top Quote	53511		
	Order	53512		
	Trade	53513		

6.5.2.1 ISE Mercury Exchange

Table 67: MT1 Multicast Channels (Mercury)

Feed Name		Ports	A Feed Source:207.231.198.199 or 200 or 201	B feed Source:207.231.198.199 or 200 or 201
FAST FEEDS				
Reference Data Feed	Snapshot	51010	224.0.75.1	224.0.75.7
	Incremental	51011		
Depth of Market Feed		51010	224.0.75.2	224.0.75.8
Order Feed		51010	224.0.75.5	224.0.75.11
Spread Feed	Depth	51010	224.0.75.6	224.0.75.12
	Top Quote	51011		
	Order	51012		

Feed Name		Ports	A Feed Source:207.231.198.199 or 200 or 201	B feed
	Trade	51013		
BINARY FEEDS				
Reference Data Feed	Snapshot	51010	224.0.75.13	224.0.75.19
	Incremental	51011		
Depth of Market Feed		51010	224.0.75.14	224.0.75.20
Top Quote Feed		51010	224.0.75.15	224.0.75.21
Trade Feed		51011		
Order Feed		51010	224.0.75.17	224.0.75.23
Spread Feed	Depth	51010	224.0.75.18	224.0.75.24
	Top Quote	51011		
	Order	51012		
	Trade	51013		

6.5.3 Member Test 2 (MT2) Environment

6.5.3.1 ISE Exchange

Table 68: MT2 Multicast Channels (ISE)

Feed Name		Ports	A Feed Source:207.231.198.215 or 216 or 217	B feed
FAST FEEDS				
Reference Data Feed	Snapshot	53150	224.0.75.65	224.0.75.71
	Incremental	53151		
Depth of Market Feed		53150 53250	224.0.75.66	224.0.75.72
Trade Feed		53151 53251	224.0.75.67	224.0.75.73
Order Feed		53150 53250	224.0.75.69	224.0.75.75
Spread Feed	Depth	53150 53250	224.0.75.70	224.0.75.76
	Top Quote	53151 53251		
	Order	53152 53252		
	Trade	53153 53253		
BINARY FEEDS				
Reference Data Feed	Snapshot	53150	224.0.75.77	224.0.75.83
	Incremental	53151		
Depth of Market		53150 53250	224.0.75.78	224.0.75.84
Top Quote Feed		53150 53250	224.0.75.79	224.0.75.85
Trade Feed		53151 53251		
Pre-Open Feed		53150 53250	224.0.75.80	224.0.75.86
Order Feed		53150 53250	224.0.75.81	224.0.75.87
Spread Feed	Depth	53150 53250	224.0.75.82	224.0.75.88
	Top Quote	53151 53251		
	Order	53152 53252		

Feed Name		Ports	A Feed Source:207.231.198.215 or 216 or 217	B feed
	Trade	53153 53253		

6.5.3.1 ISE Gemini Exchange

Table 69: MT2 Multicast Channels (Gemini)

Feed Name		Ports	A Feed Source:207.231.198.215 or 216 or 217	B feed
FAST FEEDS				
Reference Data Feed	Snapshot	53510	224.0.75.65	224.0.75.71
	Incremental	53511		
Depth of Market Feed		53510	224.0.75.66	224.0.75.72
Trade Feed		53511	224.0.75.67	224.0.75.73
Order Feed		53510	224.0.75.69	224.0.75.75
Spread Feed	Depth	53510	224.0.75.70	224.0.75.76
	Top Quote	53511		
	Order	53512		
	Trade	53513		
BINARY FEEDS				
Reference Data Feed	Snapshot	53510	224.0.75.77	224.0.75.83
	Incremental	53511		
Depth of Market		53510	224.0.75.78	224.0.75.84
Top Quote Feed		53510	224.0.75.79	224.0.75.85
Trade Feed		53511		
Pre-Open Feed		53510	224.0.75.80	224.0.75.86
Order Feed		53510	224.0.75.81	224.0.75.87
Spread Feed	Depth	53510	224.0.75.82	224.0.75.88
	Top Quote	53511		
	Order	53512		
	Trade	53513		

6.5.3.1 ISE Mercury Exchange

Table 70: MT2 Multicast Channels (Mercury)

Feed Name		Ports	A Feed Source:207.231.198.215 or 216 or 217	B feed
FAST FEEDS				
Reference Data Feed	Snapshot	51010	224.0.75.65	224.0.75.71

Feed Name		Ports	A Feed Source:207.231.198.215 or 216 or 217	B feed
	Incremental	51011		
Depth of Market Feed		51010	224.0.75.66	224.0.75.72
Order Feed		51010	224.0.75.69	224.0.75.75
Spread Feed	Depth	51010	224.0.75.70	224.0.75.76
	Top Quote	51011		
	Order	51012		
	Trade	51013		
BINARY FEEDS				
Reference Data Feed	Snapshot	51010	224.0.75.77	224.0.75.83
	Incremental	51011		
Depth of Market		51010	224.0.75.78	224.0.75.84
Top Quote Feed		51010	224.0.75.79	224.0.75.85
Trade Feed		51011		
Order Feed		51010	224.0.75.81	224.0.75.87
Spread Feed	Depth	51010	224.0.75.82	224.0.75.88
	Top Quote	51011		
	Order	51012		
	Trade	51013		

List of Appendices

This section provides additional reference information to support the content of this document:

Appendix A: Prod. Multicast Channels — FAST

The following tables detail the T7 production MDI FAST feeds, including IP addresses and ports. The actual assignment of product to market data channel is detailed in the reference data.

Not all channels are active. Active channels are determined by the presence of, at least, the Heartbeat message.

I. ISE Exchange

Table 71: ISE FAST Multicast Configuration Details

Env	Group	Source	Subnet	RP
Primary Data Center (Equinix NY4)	A	224.0.68.0/24	207.231.199.0/26	207.231.198.251/32
	B	224.0.69.0/24	207.231.199.128/26	207.231.198.252/32
Disaster Recovery Data Center (60 Broad St, NY)	A	224.0.68.0/24	74.120.87.0/26	207.231.198.251/32
	B	224.0.69.0/24	74.120.87.128/26	207.231.198.252/32

Table 72: ISE FAST Depth of Market Feed Channels

Depth of Market FAST Feed Channels		
A Group IP Addresses	B Group IP Addresses	Ports
224.0.68.0/27	224.0.69.0/27	
224.0.68.1	224.0.69.1	11001 11002 11003
224.0.68.2	224.0.69.2	11004 11005 11006
224.0.68.3	224.0.69.3	11007 11008 11009
224.0.68.4	224.0.69.4	11010 11011 11012
224.0.68.5	224.0.69.5	11013 11014 11015
224.0.68.6	224.0.69.6	11016 11017 11018
224.0.68.7	224.0.69.7	11019 11020 11021

Depth of Market FAST Feed Channels		
A Group IP Addresses	**B Group IP Addresses**	**Ports**
224.0.68.8	224.0.69.8	11022 11023 11024
224.0.68.9	224.0.69.9	11025 11026 11027
224.0.68.10	224.0.69.10	11028 11029 11030
224.0.68.11	224.0.69.11	11031 11032 11033
224.0.68.12	224.0.69.12	11034 11035 11036
224.0.68.13	224.0.69.13	11037 11038 11039
224.0.68.14	224.0.69.14	11040 11041 11042
224.0.68.15	224.0.69.15	11043 11044 11045
224.0.68.16	224.0.69.16	11046 11047 11048
224.0.68.17	224.0.69.17	11049 11050 11051
224.0.68.18	224.0.69.18	11052 11053 11054
224.0.68.19	224.0.69.19	11055 11056 11057
224.0.68.20	224.0.69.20	11058 11059 11060

Table 73: ISE FAST Order Feed Channels

Order FAST Feed Channels		
A Group IP Addresses	**B Group IP Addresses**	**Ports**
224.0.68.136/29	224.0.69.136/29	
224.0.68.137	224.0.69.137	13001 13002 13003 13004 13005 13006 13007 13008 13009 13010

Table 74: ISE FAST Spread Feed Channels

Spread FAST Feed Channels			
Feed Name	A Group IP Addresses	B Group IP Addresses	Ports
	224.0.68.96/27	224.0.69.96/27	
Order Feed	224.0.68.97	224.0.69.97	14001 14002 14003
	224.0.68.98	224.0.69.98	14004 14005 14006
	224.0.68.99	224.0.69.99	14007 14008 14009
	224.0.68.100	224.0.69.100	14010
Top Quote	224.0.68.100	224.0.69.100	14011 14012
	224.0.68.101	224.0.69.101	14013 14014 14015
	224.0.68.102	224.0.69.102	14016 14017 14018

Spread FAST Feed Channels			
Feed Name	A Group IP Addresses	B Group IP Addresses	Ports
	224.0.68.103	224.0.69.103	14019 14020
Trade Feed	224.0.68.103	224.0.69.103	14021
	224.0.68.104	224.0.69.104	14022 14023 14024
	224.0.68.105	224.0.69.105	14025 14026 14027
	224.0.68.106	224.0.69.106	14028 14029 14030
Depth of Market Feed	224.0.68.107	224.0.69.107	14031 14032 14033
	224.0.68.108	224.0.69.108	14034 14035 14036
	224.0.68.109	224.0.69.109	14037 14038 14039
	224.0.68.110	224.0.69.110	14040

Table 75: ISE FAST Reference Data Feed Channels

Reference Data FAST Feed Channels		
A Group IP Addresses	**B Group IP Addresses**	**Ports**
224.0.68.248/29	224.0.69.248/29	
Snapshot		
224.0.68.249	224.0.69.249	17001
Incremental		
224.0.68.250	224.0.69.250	17004

II. ISE Gemini Exchange

Table 76: Gemini FAST Multicast Configuration Details

Env	Group	Source	Subnet	RP
Primary Data Center (Equinix NY4)	A	224.4.8.0/24	207.231.199.0/26	207.231.198.251/32
	B	224.4.9.0/24	207.231.199.128/26	207.231.198.252/32
Disaster Recovery Data Center (60 Broad st, NY)	A	224.4.8.0/24	74.120.87.0/26	207.231.198.251/32
	B	224.4.9.0/24	74.120.87.128/26	207.231.198.252/32

Table 77: Gemini FAST Depth of Market Feed Channels

Depth of Market Feed Channels		
A Group IP Addresses	B Group IP Addresses	Ports
224.4.8.0/27	224.4.9.0/27	
224.4.8.1	224.4.9.1	11001 11002 11003
224.4.8.2	224.4.9.2	11004 11005 11006
224.4.8.3	224.4.9.3	11007 11008 11009
224.4.8.4	224.4.9.4	11010 11011 11012
224.4.8.5	224.4.9.5	11013 11014 11015
224.4.8.6	224.4.9.6	11016 11017 11018
224.4.8.7	224.4.9.7	11019 11020 11021
224.4.8.8	224.4.9.8	11022 11023 11024
224.4.8.9	224.4.9.9	11025 11026 11027

Depth of Market Feed Channels		
A Group IP Addresses	B Group IP Addresses	Ports
224.4.8.10	224.4.9.10	11028 11029 11030
224.4.8.11	224.4.9.11	11031 11032 11033
224.4.8.12	224.4.9.12	11034 11035 11036
224.4.8.13	224.4.9.13	11037 11038 11039
224.4.8.14	224.4.9.14	11040 11041 11042
224.4.8.15	224.4.9.15	11043 11044 11045
224.4.8.16	224.4.9.16	11046 11047 11048
224.4.8.17	224.4.9.17	11049 11050 11051
224.4.8.18	224.4.9.18	11052 11053 11054
224.4.8.19	224.4.9.19	11055 11056 11057
224.4.8.20	224.4.9.20	11058 11059 11060

Table 78: Gemini FAST Order Feed Channels

Order Feed Channels		
A Group IP Addresses	B Group IP Addresses	Ports
224.4.8.136/29	224.4.9.136/29	

Order Feed Channels		
A Group IP Addresses	**B Group IP Addresses**	**Ports**
224.4.8.137	224.4.9.137	13001 13002 13003 13004 13005 13006 13007 13008 13009 13010

Table 79: Gemini FAST Spread Feed Channels

Spread Feed Channels (NOT ACTIVE)			
Feed Name	A Group IP Addresses	B Group IP Addresses	Ports
	224.4.8.96/27	224.4.9.96/27	
Order Feed	224.4.8.97	224.4.9.97	14001 14002 14003
	224.4.8.98	224.4.9.98	14004 14005 14006
	224.4.8.99	224.4.9.99	14007 14008
Top Quote	224.4.8.100	224.4.9.100	14011 14012
	224.4.8.101	224.4.9.101	14013 14014 14015
	224.4.8.102	224.4.9.102	14016 14017 14018
Trade Feed	224.4.8.103	224.4.9.103	14021
	224.4.8.104	224.4.9.104	14022 14023 14024
	224.4.8.105	224.4.9.105	14025 14026 14027
	224.4.8.106	224.4.9.106	14028

Spread Feed Channels (NOT ACTIVE)			
Feed Name	A Group IP Addresses	B Group IP Addresses	Ports
Depth of Market Feed	224.4.8.107	224.4.9.107	14031 14032 14033
	224.4.8.108	224.4.9.108	14034 14035 14036
	224.4.8.109	224.4.9.109	14037 14038

Table 80: Gemini FAST Reference Data Feed Channels

Reference Data Feed Channels		
A Group IP Addresses	**B Group IP Addresses**	**Ports**
224.4.8.248/29	224.4.9.248/29	
Snapshot		
224.4.8.249	224.4.9.249	17001
Incremental		
224.4.8.250	224.4.9.250	17004

Appendix B: Prod. Multicast Channels — Binary

The following tables detail the T7 production MDI Binary feeds, including IP addresses and ports. The actual assignment of product to market data channel is detailed in the reference data.

Not all channels are active. Active channels are determined by the presence of, at least, the Heartbeat message.

I. ISE Exchange

Table 81: ISE Binary Multicast Configuration Details

Env	Group	Source	Subnet	RP
Primary Data Center (Equinix NY4)	A	224.0.118.0/24	207.231.199.0/26	207.231.198.251/32
	B	224.0.119.0/24	207.231.199.128/26	207.231.198.252/32
Disaster Recovery Data Center (60 Broad st, NY)	A	224.0.118.0/24	74.120.87.0/26	207.231.198.251/32
	B	224.0.119.0/24	74.120.87.128/26	207.231.198.252/32

Table 82: ISE Binary Depth of Market Feed Channels

Depth of Market Binary Feed Channels (NOT ACTIVE)		
A Group IP Addresses	B Group IP Addresses	Ports
224.0.118.0/27	224.0.119.0/27	
224.0.118.1	224.0.119.1	11001 11002 11003
224.0.118.2	224.0.119.2	11004 11005 11006
224.0.118.3	224.0.119.3	11007 11008 11009
224.0.118.4	224.0.119.4	11010 11011 11012
224.0.118.5	224.0.119.5	11013 11014 11015
224.0.118.6	224.0.119.6	11016 11017 11018
224.0.118.7	224.0.119.7	11019 11020 11021

Depth of Market Binary Feed Channels (NOT ACTIVE)		
A Group IP Addresses	B Group IP Addresses	Ports
224.0.118.8	224.0.119.8	11022 11023 11024
224.0.118.9	224.0.119.9	11025 11026 11027
224.0.118.10	224.0.119.10	11028 11029 11030
224.0.118.11	224.0.119.11	11031 11032 11033
224.0.118.12	224.0.119.12	11034 11035 11036
224.0.118.13	224.0.119.13	11037 11038 11039
224.0.118.14	224.0.119.14	11040 11041 11042
224.0.118.15	224.0.119.15	11043 11044 11045
224.0.118.16	224.0.119.16	11046 11047 11048
224.0.118.17	224.0.119.17	11049 11050 11051
224.0.118.18	224.0.119.18	11052 11053 11054
224.0.118.19	224.0.119.19	11055 11056 11057
224.0.118.20	224.0.119.20	11058 11059 11060

Table 83: ISE Binary Top Quote Feed Channels

Top Quote Binary Feed Channels

A Group IP Addresses	B Group IP Addresses	Ports
224.0.118.32/27	224.0.119.32/27	
224.0.118.33	224.0.119.33	12001 12002 12003
224.0.118.34	224.0.119.34	12004 12005 12006
224.0.118.35	224.0.119.35	12007 12008 12009
224.0.118.36	224.0.119.36	12010 12011 12012
224.0.118.37	224.0.119.37	12013 12014 12015
224.0.118.38	224.0.119.38	12016 12017 12018
224.0.118.39	224.0.119.39	12019 12020 12021
224.0.118.40	224.0.119.40	12022 12023 12024
224.0.118.41	224.0.119.41	12025 12026 12027
224.0.118.42	224.0.119.42	12028 12029 12030
224.0.118.43	224.0.119.43	12031 12032 12033
224.0.118.44	224.0.119.44	12034 12035 12036
224.0.118.45	224.0.119.45	12037 12038 12039
224.0.118.46	224.0.119.46	12040 12041 12042

Top Quote Binary Feed Channels		
A Group IP Addresses	**B Group IP Addresses**	**Ports**
224.0.118.47	224.0.119.47	12043 12044 12045
224.0.118.48	224.0.119.48	12046 12047 12048
224.0.118.49	224.0.119.49	12049 12050 12051
224.0.118.50	224.0.119.50	12052 12053 12054
224.0.118.51	224.0.119.51	12055 12056 12057
224.0.118.52	224.0.119.52	12058 12059 12060

Table 84: ISE Binary Order Feed Channels

Order Binary Feed Channels		
A Group IP Addresses	**B Group IP Addresses**	**Ports**
224.0.118.136/29	224.0.119.136/29	
224.0.118.137	224.0.119.137	13001 13002 13003 13004 13005 13006 13007 13008 13009 13010

Table 85: ISE Binary Spread Feed Channels

Spread Binary Feed Channels			
Feed Name	**A Group IP Addresses**	**B Group IP Addresses**	**Ports**
	224.0.118.96/27	224.0.119.96/27	
Order Feed	224.0.118.97	224.0.119.97	14001 14002 14003
	224.0.118.98	224.0.119.98	14004 14005 14006
	224.0.118.99	224.0.119.99	14007 14008 14009
	224.0.118.100	224.0.119.100	14010
Top Quote	224.0.118.100	224.0.119.100	14011 14012
	224.0.118.101	224.0.119.101	14013 14014 14015
	224.0.118.102	224.0.119.102	14016 14017 14018
	224.0.118.103	224.0.119.103	14019 14020
Trade Feed	224.0.118.103	224.0.119.103	14021
	224.0.118.104	224.0.119.104	14022 14023 14024
	224.0.118.105	224.0.119.105	14025 14026 14027
	224.0.118.106	224.0.119.106	14028 14029 14030
Depth of Market Feed	224.0.118.107	224.0.119.107	14031 14032 14033
	224.0.118.108	224.0.119.108	14034 14035 14036
	224.0.118.109	224.0.119.109	14037 14038 14039
	224.0.118.110	224.0.119.110	14040

Table 86: ISE Binary Trade Feed Channels

Trade Binary Feed Channels		
A Group IP Addresses	**B Group IP Addresses**	**Ports**
224.0.118.128/29	224.0.119.128/29	
224.0.118.129	224.0.119.129	16001 16002 16003 16004 16005 16006 16007 16008 16009 16010

Table 87: ISE Binary Reference Data Feed Channels

Reference Data Binary Feed Channels		
A Group IP Addresses	**B Group IP Addresses**	**Ports**
224.0.118.248/29	224.0.119.248/29	
Snapshot Feed		
224.0.118.249	224.0.119.249	17001
Incremental Feed		
224.0.118.250	224.0.119.250	17004

II. ISE Gemini Exchange

Table 88: Gemini Binary Multicast Configuration Details

Env	Group	Source	Subnet	RP
Primary Data Center (Equinix NY4)	A	224.0.79.0/24	207.231.199.0/26	207.231.198.251/32
	B	224.0.80.0/24	207.231.199.128/26	207.231.198.252/32
Disaster Recovery Data Center (60 Broad st, NY)	A	224.0.79.0/24	74.120.87.0/26	207.231.198.251/32
	B	224.0.80.0/24	74.120.87.128/26	207.231.198.252/32

Table 89: Gemini Binary Depth of Market Feed Channels

Depth of Market Feed Channels (NOT ACTIVE)

A Group IP Addresses	B Group IP Addresses	Ports
224.0.79.0/27	224.0.80.0/27	
224.0.79.1	224.0.80.1	11001 11002 11003
224.0.79.2	224.0.80.2	11004 11005
224.0.79.4	224.0.80.4	11011 11012
224.0.79.5	224.0.80.5	11013 11014 11015
224.0.79.7	224.0.80.7	11019 11020 11021
224.0.79.8	224.0.80.8	11022 11023 11024
224.0.79.9	224.0.80.9	11025
224.0.79.11	224.0.80.11	11031 11032 11033
224.0.79.12	224.0.80.12	11034 11035
224.0.79.14	224.0.80.14	11041 11042
224.0.79.15	224.0.80.15	11043 11044 11045
224.0.79.17	224.0.80.17	11051
224.0.79.18	224.0.80.18	11052 11053 11054
224.0.79.19	224.0.80.19	11055

Table 90: Gemini Binary Top Quote Feed Channels

Top Quote Feed Channels		
A Group IP Addresses	**B Group IP Addresses**	**Ports**
224.0.79.32/27	224.0.80.32/27	
224.0.79.33	224.0.80.33	12001 12002 12003
224.0.79.34	224.0.80.34	12004 12005 12006

Top Quote Feed Channels		
A Group IP Addresses	**B Group IP Addresses**	**Ports**
224.0.79.35	224.0.80.35	12007 12008 12009
224.0.79.36	224.0.80.36	12010 12011 12012
224.0.79.37	224.0.80.37	12013 12014 12015
224.0.79.38	224.0.80.38	12016 12017 12018
224.0.79.39	224.0.80.39	12019 12020 12021
224.0.79.40	224.0.80.40	12022 12023 12024
224.0.79.41	224.0.80.41	12025 12026 12027
224.0.79.42	224.0.80.42	12028 12029 12030
224.0.79.43	224.0.80.43	12031 12032 12033
224.0.79.44	224.0.80.44	12034 12035 12036
224.0.79.45	224.0.80.45	12037 12038 12039
224.0.79.46	224.0.80.46	12040 12041 12042
224.0.79.47	224.0.80.47	12043 12044 12045
224.0.79.48	224.0.80.48	12046 12047 12048
224.0.79.49	224.0.80.49	12049 12050 12051

Top Quote Feed Channels		
A Group IP Addresses	**B Group IP Addresses**	**Ports**
224.0.79.50	224.0.80.50	12052 12053 12054
224.0.79.51	224.0.80.51	12055 12056 12057
224.0.79.52	224.0.80.52	12058 12059 12060

Table 91: Gemini Binary Order Feed Channels

Order Feed Channels		
A Group IP Addresses	**B Group IP Addresses**	**Ports**
224.0.79.136/29	224.0.80.136/29	
224.0.79.137	224.0.80.137	13001 13002 13003 13004 13005 13006 13007 13008 13009 13010

Table 92: Gemini Binary Spread Feed Channels

Spread Feed Channels (NOT ACTIVE)			
Feed Name	A Group IP Addresses	B Group IP Addresses	Ports
	224.0.79.96/27	224.0.80.96/27	
Order Feed	224.0.79.97	224.0.80.97	14001 14002 14003
	224.0.79.98	224.0.80.98	14004 14005

Spread Feed Channels (NOT ACTIVE)			
Feed Name	A Group IP Addresses	B Group IP Addresses	Ports
Top Quote	224.0.79.100	224.0.80.100	14011 14012
	224.0.79.101	224.0.80.101	14013 14014 14015
	224.0.79.102	224.0.80.102	14016 14017
Trade Feed	224.0.79.103	224.0.80.103	14021
	224.0.79.104	224.0.80.104	14022 14023 14024
	224.0.79.105	224.0.80.105	14025 14026 14027
Depth of Market Feed	224.0.79.107	224.0.80.107	14031 14032 14033
	224.0.79.108	224.0.80.108	14034 14035

Table 93: Gemini Binary Trade Feed Channels

Trade Feed Channels		
A Group IP Addresses	**B Group IP Addresses**	**Ports**
224.0.79.128/29	224.0.80.128/29	
224.0.79.129	224.0.80.129	16001 16002 16003 16004 16005 16006 16007 16008 16009 16010

Table 94: Gemini Binary Reference Data Feed Channels

Reference Data Feed Channels		
A Group IP Addresses	**B Group IP Addresses**	**Ports**
224.0.79.248/29	224.0.80.248/29	
Snapshot		
224.0.79.249	224.0.80.249	17001
Incremental		
224.0.79.250	224.0.80.250	17004

Appendix C: FIX Field Descriptions

The following table, ordered by field name, describes some of the possible values for some of the FAST FIX fields defined in this document. Additional values for standard FIX fields may be found at fixprotocol.org.

Tag	Name	Type	Possible Values/Notes
8522	AuctionType	String	Simple Instruments: "1"=Block "2"=Directed Order "3"=Flash "6"=Facilitation "9"=Solicitation "12"=PIM "15"=Directed Order PIM --- Complex Instruments: "4"=Exposure "5"=Exposure (with stock) "7"=Facilitation "8"=Facilitation (with stock) "10"=Solicitation "11"=Solicitation (with stock) "13"=PIM "14"=PIM (with stock)
1544	InstrumentScopeProductComplex	String	1=Simple Instrument (default) 2=Standard Combination 3=Stock Combination
1682	MDSecurityTradingStatus	Integer	See SecurityTradingStatus (tag 326)
528	OrderCapacity	String	"C"=Customer "D"=Customer Professional "B"=Broker/Dealer "G"=Proprietary "N"=Away Market Maker "M"=Market Maker
1227	ProductComplex	Integer	1=Simple Instrument (default) 2=Standard Combination 3=Stock Combination
1679	SecurityMassTradingStatus	Integer	See SecurityTradingStatus (tag 326)

Tag	Name	Type	Possible Values/Notes
326	SecurityTradingStatus	Integer	1=Opening Delay (due to imbalance at opening) 2=Halt (Underlying is closed) 17=Regular (Open or Ready-to-trade) 18=Not Available for Trading (Closed) 20=Unknown or Invalid 21=Pre-Open (Pre-Open or intraday Halt) 22=Rotation (ready to open) 23=Fast Market
277	TradeCondition	String	Condition Codes, multi-value string. "U"=Exchange Last "k"=Out of Sequence "AA"=Spread "AC"=Straddle "AH"=Combo "AE"=Stopped "Z"=Intermarket Sweep "AU"=Trade Through Exempt (IAM trade) "3"=Multi Asset Class Multileg Trade "n"=Cancel Last "p"=Cancel Open "s"=Cancel Only "0"=Cancel
6653	UnderlyingSecurityType	String	"CURRENCY" "CURRENCY INDEX" "ENERGY" "ENERGY INDEX" "EQUITY" "ETF" "ETF INDEX" "FX " "FX INDEX " "INDEX" "INTEREST RATE" "INTEREST RATE INDEX" "METAL" "METAL INDEX" "SOFT AND AGRICS" "SOFTS AND AGRICS INDEX" "STOCK" "STOCK INDEX"

Appendix D: RefData Instrument Definitions

The underlying stock deliverable for any regular instrument is defined as the contract multiplier factored with the Delivery Component, if present. (DeliveryComponent repeating group in FAST, and Security Deliverable and Cash Deliverable data vectors in binary.)

In general, all regular instruments are defined with contract multiplier equal to 100 and no delivery component. In other words, one options contract delivers 100 shares of the underlying stock. This is considered the standard deliverable.

Instruments with non-standard deliverables, or subject to corporate actions, can be defined using the standard multiplier (100) with delivery component, or using a non-standard multiplier (≠ 100) with no delivery component. For example, an instrument for a 3:2 stock split can be defined using contract multiplier equal to 100 with underlying factor (within the delivery component) equal to 1.5, or it can be defined using contract multiplier equal to 150 with no delivery component. In other words, one options contract, however defined, delivers 150 shares of the underlying stock. The important difference is that while either definition represents the same deliverable, the latter method (contract multiplier) supports stock-complex instrument creation, and the former (delivery component) does not.

Whenever possible, ISE defines non-standard deliverable and corporate action instruments using contract multiplier with no delivery component. However, corporate actions that result in multiple underlying deliverables (for example, a merger or spin-off) or that contain a cash component (for example, cash in lieu of fractional shares) are always defined with a delivery component.

The following examples detail standard deliverable, non-standard deliverable, and various corporate action instruments. These examples are simplistic and are not meant to be all-encompassing. They are provided simply to help members understand the various ways instruments can be defined in ISE reference data.

Example 1. Regular Option

ISE defines a regular option for ABC Corp. (ABC) delivering 100 shares per contract.

OSI Symbol = “ABC”

Contract Multiplier = 100

No Delivery Component

Stock-complex supported? Yes.

Example 2. Mini Option

ISE defines a mini option for ABC Corp. (ABC) delivering 10 shares per contract.

OSI Symbol = “ABC7”

Contract Multiplier = 10

No Delivery Component

Stock-complex supported? Yes.

Example 3. Cash Distribution

ABC Corp. (ABC) announces an approximate special cash distribution of $1.50 per share.

OSI Symbol = “ABC1”

Contract Multiplier = 100

Delivery Component —

Underlying Symbol = ABC

Underlying Factor = 1

Underlying Cash Amount = 1.5

Stock-complex supported? No (because of delivery component).

Example 4. Merger

ABC Corp. (ABC) announces merger with XYZ Ltd. (XYZ). Option (OSI) symbols XYZ become ABC1 and each share of XYZ is converted into .7 shares of ABC (assumes no fractional shares or cash in the deal).

OSI Symbol = “ABC1”

Contract Multiplier = 70

No Delivery Component

Stock-complex supported? Yes.

Example 5. Merger with Cash

ABC Corp. (ABC) announces merger with XYZ Ltd. (XYZ). Option symbol XYZ becomes ABC1 and each share of XYZ is converted into .7 shares of ABC and $13.50 per share.

OSI Symbol = “ABC1”

Contract Multiplier = 70

Delivery Component —

Underlying Symbol = ABC

Underlying Factor = 1

Underlying Cash Amount = 13.5

Stock-complex supported? No (because of delivery component).

Example 6. Spinoff

ABC Corp. (ABC) announces a spinoff of ZYX Co. (ZYX) with a distribution ratio of .5 XYZ shares for every ABC share.

OSI Symbol = “ABC1”

Contract Multiplier = 100

Delivery Component —

Underlying Symbol[1] = ABC

Underlying Factor[1] = 1

Underlying Symbol[2] = ZYX

Underlying Factor[2] = .5

Stock-complex supported? No (because of delivery component).

Example 7. Odd Stock Split (3:2)

ABC Corp. (ABC) announces a 3 for 2 stock split.

OSI Symbol = "ABC1"

Contract Multiplier = 150

No Delivery Component

Stock-complex supported? Yes.

Example 8. Reverse Stock Split (1:4)

ABC Corp. (ABC) announces a 1 for 4 reverse stock split.

OSI Symbol = "ABC1"

Contract Multiplier = 25

No Delivery Component

Stock-complex supported? Yes.

Example 9. Reverse Stock Split with Cash (1:8)

ABC Corp. (ABC) announces a 1 for 8 reverse stock split with cash in lieu of fractional shares.

OSI Symbol = "ABC1"

Contract Multiplier = 100

Delivery Component —

Underlying Symbol = ABC

Underlying Factor = .12

Underlying Cash Amount = .01

Stock-complex supported? No (because of delivery component).

Example 10. Stock Dividend

ABC Corp. (ABC) declares a 4% stock dividend to shareholders.

OSI Symbol = "ABC1"

Contract Multiplier = 104

No Delivery Component

Stock-complex supported? Yes.

Document Revision Table

Version	Date	Change	Section
1.0	May 11, 2010	Original Version	
1.4.5	July 2011	Updated byte offsets in **Table 25** on **Page 44**.	5.3.2
1.4.6	August 2011	Minor update to tag 276 comment, Table 19	3.6.5.1
1.5	August 2011	Release version update	
1.5.1	August 2011	Updated Trade Feed IP address from 53250 to 53251. Updated Top Quote and Pre-Open Feed bandwidth requirements in Table 30.	Table 38 and Table 39 Table 30
1.5.2	September 2011	Corrected various message layouts to correspond to template.	
1.6.0	September 2011	Release version update	
1.6.1, 1.6.2	October 2011	Minor revisions to connectivity tables	
2.0	October 2011	No changes from previous version.	
2.0.1	November 2011	Updated MT1/2 IP Addresses in Tables 38 and 39	
2.0.2	December 2011	Updated contact information in Tables 31-37. Added contact information for Options IT. Corrected Depth Incremental example.	
3.0	December, 2011	No changes, version update	
3.0.1	March 2012	Updated port info. For member test, Tables 39, 40	Sect. 7.4
3.1.0	April 2012	Increased revision number to include in 3.1 release. No changes.	
3.1.1	May 2012	Renamed Depth Quote to Depth Incremental to correspond with template. Template IDs added to message format tables. Standardized formatting and sub-headings. Switched ordering of chapters 4 and 5 for reading that is more logical. Corrected and clarified various market data examples. Add'l edits and corrections.	

Version	Date	Change	Section
4.0	June 2012	New version. New section for FAST feeds New section for Binary feeds Definitions section moved from 3.1 to 2.3 (3.x sections move up) New fields — Depth Snapshot New fields — Depth Incremental Opening behavior clarified New fields — TOB Full New fields — TOB Quote New fields — TOB Ticker Trade feed snapshot New fields — Order On Book New codes — MDFeedType field New section: Binary Feed Descriptions FAST feed instance IDs and names removed	 Sect 2.1.1 Sect 2.1.2 Sect 2.3 Table 6 Table 8 Sect 3.2.4 Table 12 Table 13 Table 14 Sect 3.4 Table 18 Tables 20, 22 Sect 4 Appendix A
4.0.1	July 2, 2012	Missing TradeTime field added to binary Ticker structure	Table 33
4.0.2	July 12, 2012	Clarified UDP packet size Added Binary spread feed channels Rewrote section Add'l edits and corrections	Various Tables 52, 53 Sect. 6.3
4.0.3	July 24, 2012	Correct data type, Block Header, MarketSegmentID field	Table 30
4.0.4	August 10, 2012	Correct field values, Combo Snapshot, LegSide field Feed names added to FAST Spread Feed channels New Appendix B — Prod. Binary Feed Multicast Addresses Add'l misc. edits/corrections	Table 37 Table 58 Appendix B
5.0.0	October 12, 2012	No changes. Version update.	
5.0.1	October 19, 2012	Add'l misc. edits/corrections	Sect. 3.2.3 Sect. 4.5.3 Sect. 7.0.0

Version	Date	Change	Section
5.1.0	Dec. 18, 2012	New version	
		Changed behavior of Trade Feed Snapshot	Sect. 3.4.3 Sect. 4.6.3
		Addition of binary reference data feed	Sect. 2.1.2 Sect. 4.1 Table 30 Sect. 4.8
		Binary *Linkage* field renamed *StateFlag*	Table 36 Table 40
		Description of binary *StateFlag* field	Sect. 4.5.8 Sect. 4.5.12
		MT2 Binary feed channels reallocated	Table 57
		Add'l misc. edits/corrections	
5.1.1	Jan. 18, 2013	New binary data type: Decimal Binary ContractMultiplier field re-typed as Decimal Binary Factor field re-typed as Decimal Add'l criteria for binary Quote message Sect. 7, FIX Field Descriptions moved to new Appendix C Removed FIX TradeCondition "R" (Opening Price) Add'l misc. edits/corrections	Table 29 Table 46 Table 46 Sect. 4.5.5 Appendix C Appendix C
5.1.2	Feb. 15, 2013	Corrected binary Feed Type vector size Updated MDI Distributors list Add'l misc. edits/corrections	Sect. 4.9.5 Sect. 6.2
5.1.3	Mar. 8, 2013	New Appendix D: RefData Instrument Definitions	Appendix D
5.1.4	Mar. 27, 2013	Updated MDI Distributors list	Sect. 6.2
6.0.0	April 9, 2013	New Release Version MT1 Binary feed channels reallocated Add'l misc. edits/corrections	Table 62
6.0.1	May 1, 2013	Gemini Market Data Channels added for MT1 Gemini exchange identifier changed to GMNI	Sect. 6.5.2 various
6.0.2	May 31, 2013	ISE Gemini production multicast channels defined	Appendix B:
6.0.3	July 5, 2013	Rebranded to T7 Clarified ISE binary spread feed allocations Add'l misc. edits/corrections	various Table 85
6.0.4	July 17,2013	Cleaned up Partition 11-15 from tables & added labeling column for Order and Depth feeds in binary tables	Various tables

Version	Date	Change	Section
6.0.5	Aug. 30, 2013	Clarified auction orders on Order feed. Corrected binary StrikePrice field datatype ("Decimal"). Removed unused Gemini channels. Added shading to table rows for readability. Add'l misc. edits/corrections.	Sect. 3.2 Sect. 4 Appendix A:, Appendix B: Appendix A:, Appendix B:
7.0.0	Sep. 25, 2013	New Release Version Quote message removed from binary pre-open feed (feed uses Long Quote message) Gemini Market Data Channels added for MT2	 Sect. **Error! Reference source not found.** Sect. 6.5.2.1
7.0.1	Oct. 10, 2013	Add'l prod. Gemini market data channels (highlighted in yellow).	Appendix A:, Appendix B:
7.0.2		Corrected *OrderCapacity* field number in examples.	Sect. 3.2.5.2
8.0.0	November 12, 2013	New Release Version Description of Binary Depth and Order feeds.	
8.0.1	Nov. 21, 13	Add'l prod. Gemini market data channels (highlighted in yellow).	Appendix A:, Appendix B:
8.0.2	Dec. 6, 13	Corrected Table 51- Binary Order On Book message	
8.0.3	Jan 3, 2014	Updated field order in binary Order on Book messages. Changed name of InstType field in Binary Snapshot message to ProductComplex.	Sect 4.9.6.1, 4.9.6.2
8.0.4	Jan. 8, 14	Updated binary auction message description. Corrected mislabeled value column in binary auction message tables 53,54. Corrected data type for leg side (uint8) and legratio (uint16) fields in binary combo depth snapshot message.	Sect 4.9.6,4.9.6.1, 4.9.6.2 Sect. 4.8.8
8.0.5	Jan. 15, 14	Add'l prod. Gemini market data channels (highlighted in yellow). Updated ISE Binary Top of book and depth of book channles.	Appendix A:, Appendix B:
8.1.0	Mar 7, 14	Add'l prod. Gemini market data channels (highlighted in yellow). For binary Auction messages, a new field, ExecFlag, has been added. Added new service provider.	Appendix A:, Appendix B: Sect. 4.9.6 Sect. 6.2
8.1.1	Mar 18, 14	Updated Gemini binary order feed channels (highlighted in yellow).	Appendix B:

Version	Date	Change	Section
8.1.2	Apr 02, 14	Add'l prod. Gemini market data channels (highlighted in yellow). Updated New DR Site information	Appendix B: Appendix B: Various tables
9.0.0	April 23, 2014	New Release Version Description of Binary Depth and Order feeds.	
9.0.1	July 28, 2014	Edited PIM auction functionality to align with new PIM rules.No longer support FAST Top Quote, pre-Open, and Trade feeds. Removed FAST feed channel tables for these feeds	Sect. 3.2.3, 3.2.5, 3.2.5.2, **Error! Reference source not found.**, **Error! Reference source not found.**
10.0.0	July 28, 2014	New Release Version	
10.0.1	August 27, 2014	Combined BinaryPre-Open and Top Quote feeds.	
10.0.2	January 23, 2015	Updated Binary Combo Snapshot Message field values	Sect 4.5.9
10.0.3	February 20, 2015	Adding Merury exchange	All
10.0.4	February 27, 2015	Multicast Channels reduction	Various tables.