CBOE FIX Protocol Support

Version 9.0.2

Volume 3D: Market Making and Market Data

Programmer's guide to the CBOE FIX 4.2 Service for making CBOE markets using: Quoting and Mass Quoting selected markets; Market Data subscription including book depth

CONFIDENTIAL
CBOE Proprietary Information

15 July 2011

Document #[FIX-03D]

Production Release

Front Matter

Disclaimer

Portions of this document have been taken from the Fix 4.2 Specification which is property of FIX Protocol Ltd. (http://www.fixprotocol.org). The FIX 4.2 Specification is property of FIX Protocol Ltd.

Change Notices

The following change notices are provided to assist users of the CBOE FIX Services in determining the impact of changes to their applications.

Date	Version	Description of Change
15 July 2011	V9.0.2	No changes
29 Apr 2011	V9.0.1	Modified the Quote Processing section.
14 Jan 2011	V9.0	No changes
08 Jan 2010	V7.0	New UDF, ShortSaleIndicator[20101], to mark short sale positions.
14 Aug 2009	V6.1	No changes
22 May 2009	V6.0	Updated the Subscribe for Current Market section to include CFIX information.
		Updated tags 55, 167, 200, 205, 201 and 202 for “by Name” look up
		Added C2_MAIN to TradingSessionID[336]
11 Feb 2009	V5.3	Added additional information to the Quote Processing section for concurrent quote processing.
25 Nov 2008	V5.3	Added a new section for Quote Processing.
03 Oct 2008	V5.2	Added new UDF 9221 – AuxAuctionInfo to the Quote Request Message for COA
23 July 2008	V5.1	Updated the quote rate limits
29 February 2008	V5.0	AuctionType[9383], STOCK_ODD_LOT=7
18 January 2008	V4.2.4	Referenced FIX-03b for Business Reject message details
02 November 2007	V4.2.3	Added an example for subscribing to STOCK_NBBO_FLASH for AuctionType[9383]
		Updated the “Market Data Types” table for Book Depth to read: “Snapshot query or subscription (*Hybrid: 1 Snapshot once every 1second & no subscriptions)*
26 June 2007	V4.2.2	Add the Block Cancel Requests sub-section to the Mass Quote section
01 June 2007	V4.2.2	Added a new AuctionType[9383], STOCK_NBBO_FLASH = 6
23 Feb 2006	V4.2.1	Updated the User Input Monitor section to include the opening behavior

Date	Version	Description of Change
		Added the CBSX quote limit of 100 quotes per second
15 Dec 2006	V4.2	Added a section: Recommended Market Making Guidelines
		Added a section for "User Input Monitor"
20 Sept 2006	V4.1	No changes
25 May 2006	V4.0	Updated the quote rate limits for the Hybrid session to 40 quotes per second, instead of 20 quotes per second.
30 Dec 2005	V3.2b	Updated AuctionType[9383] to include AuctionType, SAL=5
		Added a new QuoteEntryRejectReason[368], SAL_IN_PROGRESS = 89
		Updated TradeSessionID[336] to include COF_MAIN
12 Aug 2005	V3.2	Updated the expected opening price section.
		Added single acronym restrictions to the Quotes and QRM sections
		Updated Table1: Types of Market Data provided
29 Jul 2005	V3.2	Updated the Market Data Interface section to include more information on recommended subscriptions.
		Added a new section: Repeating Groups
		Updated tag 9383 to include AuctionType, HAL=4
		Included value "PRICE_NOT_IN_BOTR_RANGE = "X" " in tag [277] for expected opening price
08 Apr 2005	3.1a	Documentation Errata Release
17 Dec 2004	3.1	Document updates for internalization and automated auction.
04 Oct 2004	3.0b	Version Changes
20 July 2004	3.0a	Documentation Update
18 June 2004	3.0	API Enhancements
28 April 2004	2.52	Version Updates
06 February 2004	2.63	Support added for the CBOE Futures Exchange (CFE) and Stock Trading On CBOEdirect (STOC)
08 July 2003	2.51	Documentation Errata Release
23 Apr 2003	2.5	Support for Hybrid Trading
24 Jan 2003	2.1	Support for P orders
07 Oct 2002	2.01	Errata release for CBOEdirect® 2.0 that includes specifications for order routing through the Options Linkage Authority.
22 Apr 2002	2.0	Production Release
15 Feb 2002	2.0a	Retitled document to "Market making using FIX"
		Added an introduction that presents a simplified market making process.
		Restructured document to follow general market maker usage.
		Added details on the definition of complex securities and futures.

Date	Version	Description of Change
		Simplified the format of the document.

Support and Questions Regarding This Document

Questions regarding this document can be directed to The Chicago Board Options Exchange at 312.786.7300 or via e-mail: api@cboe.com.

The latest version of this document can be found at the CBOE web site http://systems.cboe.com.

Table of Contents

Tables & Figures

List of Tables

List of Figures

About This Document

Purpose

This document is intended to provide information and guidance on how to connect to the CBOE FIX 4.2 service to access CBOE markets to perform market-making (two-sided quoting) and market data.

Intended Audience

Management requiring a deeper technical understanding of CBOE's support for FIX 4.2 in making decisions on how best to participate in CBOE markets and developers of applications that will use the FIX 4.2 service to communicate with CBOE markets.

Related Documents

Document Number	Document Description
FIX-RELNOTES	CBOE FIX Release Notes Version 2.0
FIX-ROADMAP	CBOE FIX Document Road Map
FIX-01	CBOE FIX Volume 1: Overview & Concepts
FIX-03A	CBOE FIX Volume 3A: FIX 4.2 Programmer's Guide: FIX Session Layer
FIX-03B	CBOE FIX Volume 3B: FIX 4.2 Programmer' Guide: Application Layer: Fundamentals and Field (Tag) Dictionary
FIX-03C	CBOE FIX Volume 3C: FIX 4.2 Programmer' Guide: Order Routing
FIX-03D	CBOE FIX Volume 3D: FIX 4.2 Programmer' Guide: Market making
FIX-06	CBOE FIX Volume 6: FIX 4.2 Certification Guide
FIX-07	CBOE FIX Volume 7: FIX 4.2 CBOE Market Data FIX Engine (CFIX)
NET-01	CBOE Network Connectivity Guide
	Financial Information Exchange Protocol (FIX) Version 4.2 (http://www.fixprotocol.org)
	Financial Information Exchange Protocol (FIX) Version 4.3 (http://www.fixprotocol.org)

Usage and Conventions

The FIX 4.2 Specification contains definitions for all standard FIX messages and tags. With the exception of the Tag Dictionary, the standard definitions for FIX messages have been omitted from this document. This was done for brevity and to not obscure the text describing CBOE's particular implementation of a message or a tag. In the Tag Dictionary, the standard definition is provided next to CBOE's usage.

Fields that follow the FIX standard and are not subject to any CBOE specific constraints are denoted with the phrase "**Per standard.**"

Fragments of FIX messages are shown in the courier new font. The ^ is used to represent the FIX field separator (ASCII 01).

```
55=IBM^48=1237^167=OPT^200=200010^201=0^202=105.00^207=W^
```

FIX Tags are shown are presented by name in italics followed by the tag number in brackets [].

SecurityType[167]

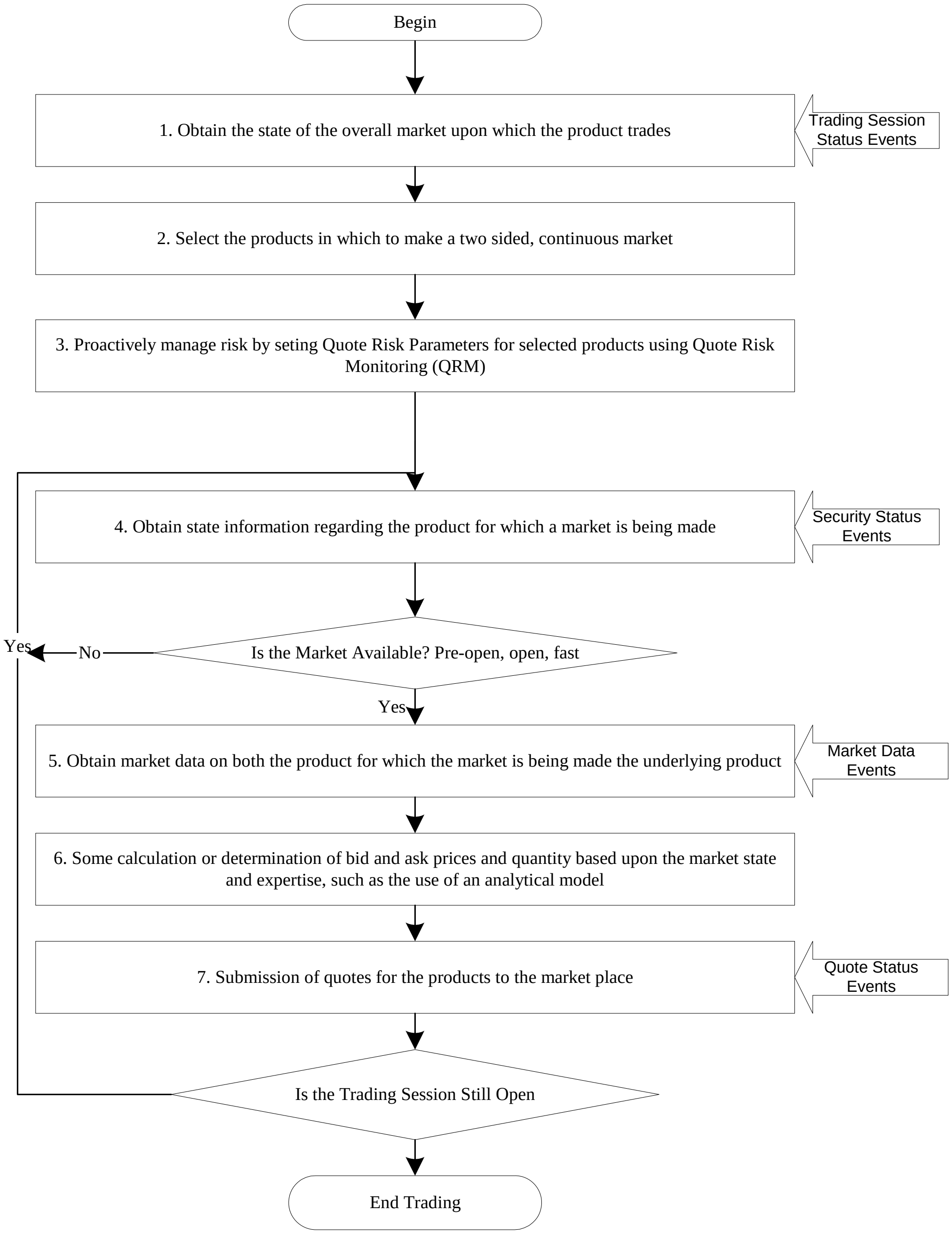
Symbol[55]

Firms should put the repeating group tags in exactly the same order as they appear in the FIX Specification. This will become a requirement for FIX 4.3.

Market making using FIX

The CBOE FIX 4.2 Service supports market making for electronic trading sessions. It is useful to first identify a simplified set of steps involved in making markets to understand how to use the service for market making. Once we have identified these steps, we can then explain how to use the service for each step. The steps involved in a simplified market making process are provided in the following flow chart. Each of the steps is then mapped to the CBOE FIX 4.2 Service.

Figure 1 Simplified Market Making Process


Begin
1. Obtain the state of the overall market upon which the product trades
Trading Session Status Events
2. Select the products in which to make a two sided, continuous market
3. Proactively manage risk by seting Quote Risk Parameters for selected products using Quote Risk Monitoring (QRM)
4. Obtain state information regarding the product for which a market is being made
Security Status Events
Yes
No
Is the Market Available? Pre-open, open, fast
Yes
5. Obtain market data on both the product for which the market is being made the underlying product
Market Data Events
6. Some calculation or determination of bid and ask prices and quantity based upon the market state and expertise, such as the use of an analytical model
7. Submission of quotes for the products to the market place
Quote Status Events
Is the Trading Session Still Open
End Trading

Using FIX to implement the Market making model

These steps map into the CBOE FIX 4.2 Service messages as follows:

Obtain the state of the overall market place (Step 1)

CBOEdirect segregates markets into trading sessions. Trading Session Status Request – retrieve the current state of a trading session (market) and subsequent changes in the state of the trading session. Refer to **Volume 3B: Application Layer: Fundamentals and Field (Tag) Dictionary (FIX-03B)** for details on the usage of the Trading Session Status Request and Trading Session Status messages.

Select products in which to make markets (Step 2)

The Security Definition Request is used to retrieve information on products that trade during a trading session. Unfortunately, in FIX 4.2 the Security Definition Request and the Security Definition messages are very overloaded. Each message is employed to provide a number of product information requests. Refer to **Volume 3B: Application Layer: Fundamentals and Field (Tag) Dictionary (FIX-03B)** for details on the usage of the Security Definition Request and Security Definition messages.

Proactively institute risk monitoring for markets (Step 3)

Inherent in an electronic market is exposure to the market maker to having quotes filled excessively on one side of the market due to unusual market conditions, such as large moves of the underlying product. CBOEdirect provides a proprietary feature for controlling risk inherent in continuously quoted markets, this feature is referred to as Quote Risk Monitoring. The use of QRM is documented in this volume.

Obtain state information for products in which markets will be made (Step 4)

The Security Status Request and Security Status messages are used to subscribe for changes in the state of a market for a specific product trading in a specific trading session. Refer to **Volume 3B: Application Layer: Fundamentals and Field (Tag) Dictionary (FIX-03B)** for details on the usage of the Security Status Request and Security Status messages.

Obtain market data for products in which markets are to be made and for the underlying product (Step 5)

In order to ascertain the state of the product that is trading, and in the case of a derivative product market, the product underlying the trading product, you subscribe for market data using the Market Data Request message. CBOEdirect responds with Market Data messages in response. If an invalid request is received, CBOEdirect will respond with a Market Data Request Reject message. Subscribing for market data is documented in this volume.

Determine pricing and size of market for products to be traded (Step 6)

This is the “you are on your own” step.

Submit quotes to the market place (Step 7)

Quoting:

The Quote and Mass Quote messages are used to make markets within CBOEdirect. The Quote Status Request is used to configure your quoting session, to subscribe for Quote Requests (RFQs), and to retrieve the status of your quote. The Quote Cancel message is used to cancel quotes. CBOEdirect responds with Quote and Quote Acknowledgement messages to report the status of your quote and Execution Reports when those quotes are filled. Quoting and Mass quoting are documented in this volume.

Recommended Market Making Guidelines

It is important to follow the market making guildelines listed below to effectively interact with CBOE’s Hybrid Trading System.

1) Waiting on acknowledgements. For FIX users, orders and quotes in particular, please obey the protocol of ***receiving the ack*** from the prior transaction ***before*** sending in another transaction for the same class. Quoters or order flow providers who do not wait for an ack prior to sending another transaction are, in effect, exacerbating perceived slowness within the system and may be exposing themselves to processing stale quotes.
2) Logout implications. The logout is an expensive function. We have observed traders logging out of the system in what appears to be a panic. A less severe method of removing quotes would be to send a ***cancel all quotes*** message.
3) Excessive quoting/thrashing. We have observed firms sending in mass quotes, then single cancels, followed by mass quotes again. If there is a perceived reason validating the need for cancels, then they should not be immediately followed by mass quotes. If the intent is to update the quote, then the cancel is superfluous.
4) There are many single quote blocks entered. It is more efficient to block multiple quotes together within a single message, when possible.

User Input Monitor

This feature is awaiting regulatory approval and is not currently enabled.

CBOE’s User Input Monitor (UIM) is a service that runs in the CBOE trade server to monitor user quote times. The UIM service will limit the CBOE’s liability for a user who is unable, due to exchange system issues, to update or cancel their market quotes. An additional benefit is to protect Market Makers when their own system problems prohibit them from quoting.

The UIM service is intended to assist in the detection of problems with a user's quote input stream. The CBOEDirect system services discrete collections of underlying symbols per business server group. If one of these server group detects no inbound quotes for a user for any of the group's serviced underlying symbols during the configured time period, it will declare an error condition for the user. This will result in all of the user's quotes in that group to be removed by the systems. If a user is actively quoting in classes serviced by one business server group and stops streaming quotes in another business group there will be no impact to their quotes in the active group. Cancel reports will be sent to the user in the *Quote Acknowledgement message [MsgType = b]* with *QuoteRejectReason[300]*=99. Refer to the Quote Update Control ID (also called Quote Token) section below for details on how CBOE handles messages that may cross in flight.

Behavior at the open

If a user has system problems before (or seconds after) the open, their quotes will be at risk and UIM will not save them. UIM is only effective during the trading hours (*n* seconds after the open and *m* seconds before the close, currently *n*=*m*=30sec).

Details:

The basic UIM behavior is to record a timestamp for each inbound quote or quote block per user per business server group. A background thread will cancel orders and quotes per business server group for users whose most recent business server timestamp is older than *m* seconds.

The timestamp for inbound quotes is *not* recorded for UIM if the current time is less than "*n* seconds after the first series in the class opens". This is to avoid cancelling quotes entered during preopen, when market makers may not be quoting very often.

The side effect of this is that UIM protection is not activated on a business server group for a user until that user has entered a quote for a class that has at least one open series on the business server.

Example:

- User JIM enters an IBM Jul 07 quote at 8:25 (Because this is before any IBM series is open, UIM does not record his quote time.)

- Due to system problems, he is unable to quote but has not logged out.
- IBM opens, and JIM's quote is at risk until he is logged out, *or* until he enters a quote on a class that has an open series.

Custom Defined Tags (User or CBOE Defined Fields)

FIX firms should never reject messages sent from the CBOE to the firm. If the firm receives messages from CBOE containing tags that the firm does not recognize (CBOE User Defined Tags or other), the firm should ignore the tag. The firm should not try to validate the unknown tag and reject CBOE's message. CBOE reserves the right to force logout any user that rejects a message sent from the CBOE to the firm.

Repeating Groups

It is important to stress that when repeating groups are involved, the order of the tags is significant. Additionally, only those tags documented as belonging to a repeating group should be used. If these guidelines are not followed, it is likely that the message will not be interpreted as desired; error responses may be misleading, and it is possible that some attribute values will not be processed.

Market Data Interface

CBOE uses the *Market Data Request* and *Market Data – Full Refresh* messages to provide the following types of market data. CBOE does not require that a user utilize market data to become a certified user of CBOEdirect.

Most market data is provided on a subscription basis. An *initial Market Data Request* message is transmitted to subscribe for market data. *The Market Data Request* message specifies the type of information and the securities for which the market data is to be provided.

CBOE suggests that you subscribe to market data by class for recap and current market. Note that book depth must be by product (not class) and ticker should be subscribed by class for futures and options, by class or by product for underlying. Also note that CBOE currently does not permit book depth subscriptions for options.

In order to optimize CBOE's market data delivery mechanism to improve bandwidth and CPU performance, market data will only be available via a separate FIX 4.2 connection to a CBOE FIX 4.2 market data server located on CBOE premises. Firms will no longer be able to obtain market data over the same FIX 4.2 connection that it uses to submit orders and quotes. A separate document FIX Volume 7: FIX 4.2 CBOE Market Data FIX Engine (CFIX) describes the CBOE FIX4.2 Market Data Service and provides network and server configuration.

If a firm plans to subscribe to, or query, Hybrid book depth in W_MAIN, the firm must read and implement the information contained in document FIX-07.

Table 1 Types of Market Data provided

Market Data Type	Description	Trading Sessions	Request Type	Subscription Type	When Published
Current Market	Top of the book for products trading on CBOE markets. Includes: • Best bid price • Best ask price • Best bid size • Best ask size	W_MAIN, ONE_MAIN, CFE_MAIN, W_STOCK	Subscription plus initial snapshot at time of subscription	Class Level (recommended) or Product Level ** A mix of class level and product level subscriptions is not permitted.	Anytime there is a change in Best bid price Best ask price Best bid size Best ask size

Market Data Type	Description	Trading Sessions	Request Type	Subscription Type	When Published
Recap	**Options and Futures:** Last trade (last sale) including: • Last sale price • Last sale quantity • Low price • High price • Total quantity traded on the day **Underlying stock or index:** • Best bid price • Best ask price • Best bid size • Best ask size • Last sale price • Last sale quantity • Low price • High price • Total quantity traded on the day	W_MAIN, ONE_MAIN, CFE_MAIN, W_STOCK	Subscription plus initial snapshot at time of subscription	Class Level (recommended) for options and futures or Product Level for options, futures, or underlying ****** A mix of class level and product level subscriptions in one class is not permitted.	**Options and Futures:** Published when there is a trade **Underlying stock or index:** Published when there is a trade *or* when any of the following change: • Best bid price • Best ask price • Best bid size • Best ask size
Ticker (Last Sale)	Last Sale for underlying and derivative products traded on CBOE markets. Ticker is a subset of Recap. Includes: • Last trade price • Last trade quantity	W_MAIN, ONE_MAIN, CFE_MAIN	Subscription	Product Level Only	Each trade (last sale)
Expected Opening Price	The expected opening price for products traded on CBOE markets. This also announces when order imbalances occur.	W_MAIN, ONE_MAIN, CFE_MAIN	Subscription	Class Level or Product Level****** A mix of class level and product level subscriptions is not permitted.	Published on a timer basis before market open
Book Depth	Query or subscription of the book depth for a product traded on a CBOE market	W_MAIN & W_STOCK via FIX, and ONE_MAIN & CFE_MAIN (via CFN only)	Snapshot query or subscription (*Hybrid: 1 Snapshot once every 1second & no subscriptions*)	Product Level Only	Each Time a change occurs to the book

Market Data Type	Description	Trading Sessions	Request Type	Subscription Type	When Published
NBBO	*National Best Bid and Offer – best of consolidated options markets*	*Not published at this time*	*N/A*	*N/A*	*N/A*

Underlying Market Data

For underlying market data, if any of the items in a recap message change, INCLUDING the bidPrice, bidSize, askPrice, and askSize, in addition to lastSale price and quantity, a recap is published. There will be no currentMarket messages published for underlying market data. If a firm cares about last sales, it should also subscribe to the ticker, which contains the underlying last sale information that is published when a trade takes place. Occasionally, if a trade is out of sequence, a trade will be published in the form of a ticker and no recap will be published. Indexes that are not actually traded do not have bid/asks, only last sale prices. CBOE publishes the underlying index market data for index classes (such as OEX, S&P500, Dow Jones, NDX etc.). CBOE does not publish the market data for the futures contract that mirrors the index. For example, CBOE does publish the underlying market data for the S&P 500 *index*, but CBOE does not publish the market data for the S&P 500 *futures* contract. Note: underlying ticker and recap are published following industry standard practices (CTS/CQS).

****** A mix of class level and product level market data subscriptions is not permitted on an individual class. If a user is subscribed to receive IBM market data at the class level and would like to receive market data at the IBM product level, the user must unsubscribe from receiving market data at the class level before subscribing to receive market data at the product level.

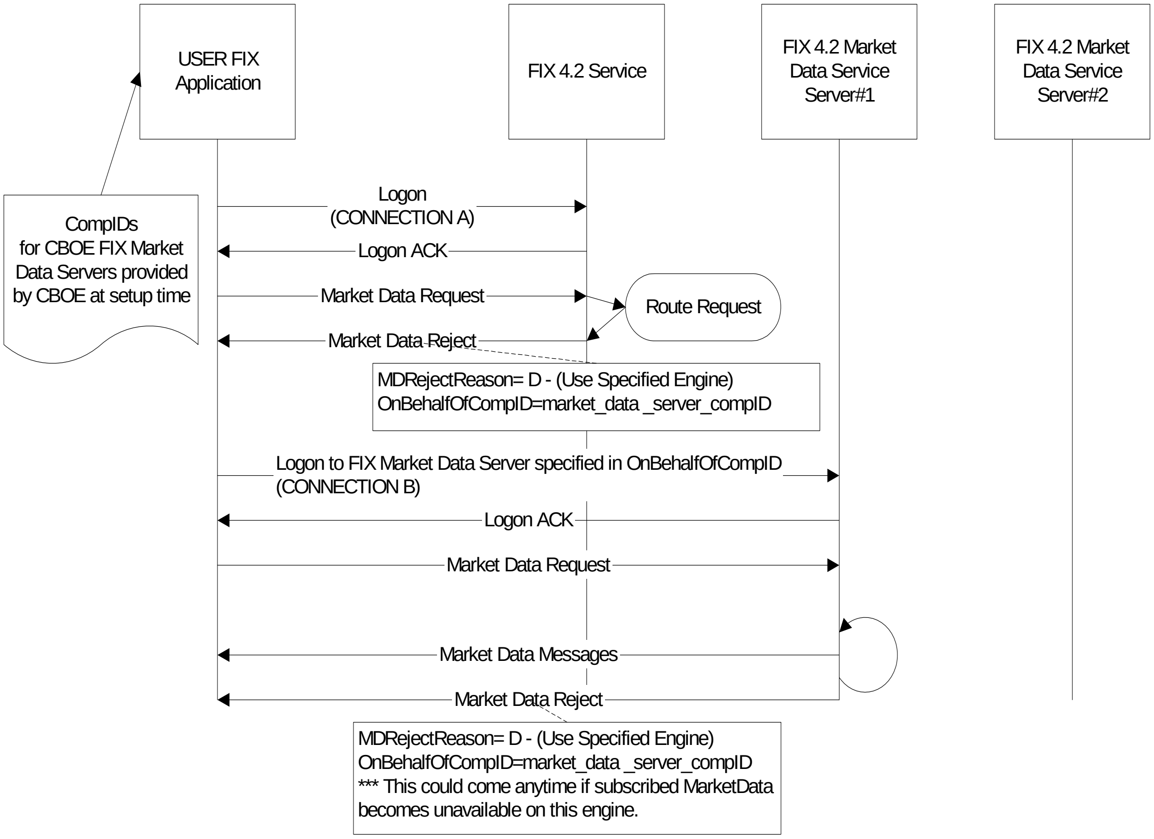
For Hybrid market data, the firm must open up a range of IP addresses (subnet) on the firm’s firewall and routers as described in the Hybrid Port document available for download at the API web site. This applies in the test and production environments.

General Message Flow for Market Data

The general message flow in CBOE FIX 4.2 for market data consists of a Market Data Request being sent from firm to CBOE. CBOE will respond with one or more Market Data – Full Refresh messages. For subscription requests, CBOE will send *Market Data – Full Refresh* messages each time the requested market data changes. This will continue until the market data is no longer produced or you transmit a Market Data Request to unsubscribe for data. Market data publication also stops when the market closes for a product.

In order to support higher-throughput market data, CBOE changed some FIX engines to the Market Data FIX Engines, which will now force midday rerouting to alternate FIX engines to support internal load-balancing or error recovery.

Figure 2 Market Data Usage - Sequence Diagram


USER FIX Application
FIX 4.2 Service
FIX 4.2 Market Data Service Server#1
FIX 4.2 Market Data Service Server#2
CompIDs for CBOE FIX Market Data Servers provided by CBOE at setup time
Logon (CONNECTION A)
Logon ACK
Market Data Request
Route Request
Market Data Reject
MDRejectReason= D - (Use Specified Engine)
OnBehalfOfCompID=market_data _server_compID
Logon to FIX Market Data Server specified in OnBehalfOfCompID (CONNECTION B)
Logon ACK
Market Data Request
Market Data Messages
Market Data Reject
MDRejectReason= D - (Use Specified Engine)
OnBehalfOfCompID=market_data _server_compID
*** This could come anytime if subscribed MarketData becomes unavailable on this engine.

In this model, the initial MarketDataRequest on Connection A is potentially rejected, specifying an alternative "TargetCompID" in the MarketDataRequestReject's OnBehalfOfCompID field. The firm's engine then has to establish Connection B (unless already established) to the specified Target, and resubmit the Market Data Request. All the Market Data Incremental responses to that request are sent back on Connection B.

If the firm attempts to make the initial market data connection on the same connection as Orders and Quotes, every Market Data Request will be rejected with MDReqRejReason equal to 'D' and OnBehalfOfCompID equal to desired "TargetCompID". The returned ID can be looked up in a specified table of TargetCompIDs to find the connection IP address information.

Upon the receipt of an unrequested Market Data Reject with MDReqRejReason='D', the Market Data Reject message will contain a list of automatically unsubscribed MDReqIDs – the firm should resubscribe them on the new market data connection to "TargetCompID".

Specifying a market data request

You specify the selection criteria using tags contained in the Market Data Request message. The combination of selection tags defines the subscription for market data. It is important for you to understand how to request each type of CBOE market data provided.

FIX 4.2 uses the following tags to specify the request for market data. CBOE introduced minor extensions to the specification to support electronic derivatives markets. These extensions were made in consultation and with the approval of the FIX Technical Committee. It is likely, but not certain, that these extensions will be included in a future version of the specification.

The fields are used in combination to retrieve specific market data offerings available via CBOEdirect. You must properly specify the proper combination of request tags to successfully subscribe for CBOE market data.

You must specify a market data request identifier *MDReqID[262]* on each request. This identifier will be returned on each market data response that corresponds to the request.

Table 2 Required Fields Used To Specify Market Data Request Criteria

Tag	Field Name	Usage within CBOE
264	MarketDepth	Used to specify book depth. CBOE supports the following enumerations: 0-Full book 1-Top of book
265	MDUpdateType	CBOE only supports 0-Full Refresh. Any other enumeration will be ignored.
266	AggregatedBook	At this time CBOE does not provide aggregated book – multiple entries per side per price will be provided when available.
267	NoMDEntryTypes	Specifies the number of market data entries to be provided.
269	MDEntryType	CBOE supports the following MDEntryTypes at this time: 0-Bid (Same for MLEG products) 1-Offer (Opposite for MLEG products) 2-Trade 3-Index Value 4-Opening Price 5-Closing Price 7-Session High Price 8-Session Low Price
336	TradingSessionID	Specifies the trading session for which market data is being requested. W_MAIN, ONE_MAIN, CFE_MAIN, COF_MAIN, W_STOCK, UNDERLYING
Extensions used in the CBOE FIX 4.2 Service		
9315	*MDScope*	*MDScope is currently a user defined tag that will likely be added to a future version of the FIX specification. MDScope is used to specify the scope of the market data in terms of what markets are covered. The possible enumerations for MDScope are:* 1-Local (Exchange/ECN/ATS) 2-National 3-Global *CBOE uses 1 to specify the CBOE best market (or top of the book)* *CBOE uses 2 to specify the National Best Bid – or best quote across all US equity option exchanges.* *Not Part of FIX standard until 4.3*

Tag	Field Name	Usage within CBOE
286	*OpenCloseSettleFlag*	*OpenCloseSettleFlag is used on the Market Data Request message to indicate the type of price being requested. This tag is not part of Market Data Request message in FIX 4.2. This is an extension that will likely be included in a future release of the FIX specification.* *CBOE has added an enumeration for OpenCloseSettleFlag for expected opening price. This is used to request expected opening prices for electronic markets.* *Settlement Price is not supported by CBOEdirect or OneChicago and can be obtained from the Options Clearing Corporation (OCC) or CBOE Financial Network (CFN).* *0-Daily Open / Close Price* *1-Session Open / Close Price* *3-Expected Opening Price* *Both 0 and 1 will return the open / close prices for the session specified on the request.* *NOT Part of FIX standard until 4.3*
6700	*ApplicationQueueActionRequest*	*Optional user defined field that indicates the action that should be taken to resolve the Application queue depth (backlog).* *0- No action taken* *1- Queue flushed* *2- Overlay last* *3- End subscription* *This tag will be applicable ONLY on Market Data subscription request messages. It will be rejected on all the other messages. For additional information regarding this tag, please review document FIX-03b.*

Market Data Request

The following table lists the Market Data Request message, as implemented by CBOE. Notice that there is a user defined tag *MDScope[9315]* that specifies the scope of market data (local, national, global) – which will likely be included in a future version of FIX. The *OpenCloseSettleFlag[286]* is included in the Market Data Request as an extension in order to specify the type of price being requested. An additional enumeration for *OpenCloseSettleFlag[286]* = 3 has been added that represents the expected opening price for a product. The use of *OpenCloseSettleFlag[286]* will likely be included in a future version of FIX as a standard field.

Table 3 Market Data Request

Tag	Field Name	FIX Req'd	CBOE Req'd	Comments
	Standard Header	Y	Y	*MsgType* = V
262	MDReqID	Y	Y	Will be used to track subscriptions – must be unique per request
263	SubscriptionRequestType	Y	Y	0 = Snapshot 1 = Snapshot + Update (Subscribe) 2 = Disable previous Snapshot+Update Request (Unsubscribe)
264	MarketDepth	Y	Y	Valid values: 0 – Full book 1 – Top of book
265	MDUpdateType	N	Y	This value must = 0 (full refresh). Any other enumeration will be ignored.
266	AggregatedBook	N	N	CBOE recommends that you don't use this field. At this time CBOE does not provide aggregated book – multiple entries per side per price will be provided when available.
267	NoMDEntryTypes	Y	Y	Number of MDEntryType fields requested.
Begin Repeating Group for NoMDEntryTypes				
269	→ MDEntryType	Y	Y	CBOE supports the following MDEntryTypes at this time: 0-Bid (Same for MLEG products) 1-Offer (Opposite for MLEG products) 2-Trade 3-Index Value 4-Opening Price 5-Closing Price 7-Session High Price 8-Session Low Price
End Repeating Group				
146	NoRelatedSym	Y	Y	The system only supports one security per request. This can either be a class or a product(option series or futures contract) – depending on the request type. **Must = 1**
Begin Repeating Group for NoRelatedSym				

Tag	Field Name	FIX Req'd	CBOE Req'd	Comments
55	→ Symbol	Y	Y	For **options,** this tag would contain the full underlying stock or index symbol. Examples: IBM, AOL, MSFT, DELL, OEX, DJX. For **OneChicago futures** this is a one to four character ticker symbol, combined with a contract specification designation between 0 and 9 inclusive, followed by one character that represents OneChicago, for example: MSFT1C, IBM1C. For **CBOE Futures Exchange (CFE)** products, this would be the class symbol, for example: VIX. For **underlying, or equity-based** products it is the ticker symbol for the underlying product, such as IBM, T, MSFT, OEX, SPX.
48	→ SecurityID	N	N	CBOEdirect product key
22	→ IDSource	N	N	Ignored by CBOEdirect FIX 4.2 Service on messages received from firms.
167	→ SecurityType	N	Y	OPT, CS, FUT, INDX, MLEG, USTB. Required for "by name" look up of product.
200	→ MaturityMonthYear	N	Y	Format: YYYYMM (e.g., 200209)
205	→ MaturityDay	N	Y	MaturityDay of the product. Required for "by Name" look up of product.
201	→ PutOrCall	N	Y	For options only. 0 = Put, 1 = Call. Required for "by name" look up of product.
202	→ StrikePrice	N	Y	For options only. Required for "by name" look up of product.
336	→ TradingSessionID	N	N	W_MAIN, ONE_MAIN, CFE_MAIN, COF_MAIN, W_STOCK, UNDERLYING, C2_MAIN
End Repeating Group				
Extensions				
9315	***MDScope***	***N***	***N***	***User Defined Tag – will be adopted in FIX 4.3***
286	***OpenCloseSettleFlag***	***N***	***N***	***Extension to FIX 4.2 – will be adopted in FIX 4.3***
6700	*ApplicationQueueActionRequest*	*N*	*N*	*Optional user defined field that indicates the action that should be taken to resolve the Application queue depth (backlog).* *0- No action taken* *1- Queue flushed* *2- Overlay last* *3- End subscription* *This tag will be applicable ONLY on Market Data subscription request messages. It will be rejected on all the other messages.*
	Standard Trailer	Y	Y	

Queue Action

User defined tag *ApplicationQueueActionRequest[6700]* provides four actions that should be take to resolve the application queue depth (backlog). Below is an example of how overlay works in FIX 4.2.

Overlay Mode Example

Let's say there are 10 products for a given class.

1) The firm gets an update from CBOE for one of those products.

2) Before the firm finishes processing that one update, the other nine tick.

3) The next call the firm gets from us will be the remaining nine ticks on the remaining nine products all at once.

So, one of the features of overlay is to "group" up Current Market into portions. The portions will be small if the firm processes them fast and potentially larger as the firm's processing becomes slower.

The second feature of overlay is to only give the very latest Current Market.

1) The firm gets an update from CBOE for product X

2) While the firm is processing that update, Product X ticks eight more times.

3) CBOE would then send only the eighth of those updates for Product X to the firm. CBOE would not send the first seven messages in step #2 above.

The only messages that will ever be dropped in overlay mode are "old" current market messages that are no longer "current" for one particular product. Those old current markets will be overlaid on a product by product basis. Also, there will never be queuing in overlay because of this, thus it is impossible to disconnect because of large queues if the firm is using overlay mode.

Specifying products in market data requests

The CBOE FIX 4.2 Service permits subscribing to market data at both the class level and the product level (option series or futures products) for all types of market data except ticker (last sale) and book depth, which both must be subscribed to at the product level only.

Only one class or product can be specified on a market data request.

Selection by class will subscribe for market data for all products belonging to that class. You are required to specify the *Symbol[55]* and the *SecurityType[167]* for classes.

Selection by product will subscribe market data for an individual product. Your are required to specify the *Symbol[55]* and the *SecurityType[167]* =CS for equities.

For option series, you must specify *Symbol[55]* + *SecurityType[167]* + *MaturityMonthYear[200]* + *PutOrCall[201]* + *StrikePrice[202]* or the *SecurityID[48]*.

For an individual class, only one kind of subscription is permitted. A mix of class level and product level market data subscriptions is not allowed. If the user is subscribed to receive IBM market data at the class level and would like to receive market data at the IBM product level, the user must unsubscribe from receiving market data at the class level before subscribing to receive market data at the product level.

Obtaining Underlying Market Data

CBOE does not leave underlying market data turned on at all times in the test environment due to log space capacity consequences. The following is an explanation of how a firm should obtain underlying market data. This information applies to the underlying that will be published for W_MAIN, ONE_MAIN, and CFE_MAIN.

In options and futures, the recap message callback is only sent to the firm when a trade takes place. In this case, the recap contains last sale price, last sale volume, high price, low price, total volume, etc. For options and futures, the bid price, bid size, ask price, and ask size (also known as currentMarket or *top* of book information) fields in the recap message should be ignored. For a firm to receive updated bid price, bid size, ask price, and ask size when any of these change, the firm needs to subscribe to current market. This is also stated in the Market Data Test Plan.

However, underlying market data works a little bit differently. If any of the items in a recap message change, *including* the bid price, bid size, ask price, and ask size, a recap is published. There will be no current market messages published for underlying market data. If a firm cares only about last sales, it should subscribe to the ticker, instead of the recap, which is the underlying last sale that is published when a trade takes place. Occasionally, if a trade is out of sequence, a trade will be published in the form of a ticker and no recap will be published. In addition, indexes that are not physically traded do not have bid / asks, only last sale prices.

FIX 4.2 firms would perform a Market Data Request for recap and ticker specifying the underlying ticker symbol for the underlying product, such as IBM, T, MSFT, OEX, SPX.

CBOE Futures Exchange (CFE) Market Data

CBOE requires that firms must use a Class Display (CD) role in a separate user session to get CFE current market (top of book), recap, and ticker market data. CBOE will not allow a market-maker or broker-dealer role to obtain CFE market data. The CD role would logon to the CFE FIX trading engines, not the CFIX engines (for Hybrid users). The CD role needs to have a SenderCompId set up on the CFE FIX trading engines.

Market Data Responses

All CBOE FIX 4.2 Service market data is returned using the *Market Data – Snapshot / Full Refresh* message.

Market Data – Snapshot / Full Refresh

Table 4 Market Data - Snapshot / Full Refresh

Tag	Field Name	FIX Req'd	CBOE Req'd	Comments
	Standard Header	Y	Y	*MsgType[35]* = W
262	MDReqID	N	N	Will correspond to the MDReqID provided on the Market Data Request
55	Symbol	Y	Y	Will be the same value as provided as on the Market Data Request.
48	SecurityID	N	N	Will be the same value as provided as on the Market Data Request.
22	IDSource	N	N	This field will be set to “8” (Exchange Symbol)
167	SecurityType	N	N	Will be the same value as provided as on the Market Data Request.
200	MaturityMonthYear	N	N	Will be the same value as provided as on the Market Data Request.
205	MaturityDay	N	N	Will be the same value as provided as on the Market Data Request.
201	PutOrCall	N	N	Will be the same value as provided as on the Market Data Request.
202	StrikePrice	N	N	Will be the same value as provided as on the Market Data Request.
207	SecurityExchange	N	N	W = CBOE
387	TotalVolumeTraded	N	N	Total volume traded in this trading session for this security.
268	NoMDEntries	Y	Y	Number of entries following.
Begin Repeating Group for NoMDEntries				
269	→ MDEntryType	Y	Y	CBOE supports the following MDEntryTypes at this time: 0-Bid (Same for MLEG products) 1-Offer (Opposite for MLEG products) 2-Trade 3-Index Value 4-Opening Price 5-Closing Price 7-Session High Price 8-Session Low Price
270	→ MDEntryPx	Y	Y	Price of the Market Data Entry
271	→ MDEntrySize	N	N	Conditionally required if MDEntryType = Bid(0), Offer(1), or Trade(2)
272	→ MDEntryDate	N	N	Date of Market Data Entry
273	→ MDEntryTime	N	N	Time of Market Data Entry

274	→ TickDirection	N	N	Direction of the "tick” Valid values: 0 = Plus Tick 1 = Zero-Plus Tick 2 = Minus Tick 3 = Zero-Minus Tick
275	→ MDMkt	N	N	Market posting quote / trade. Valid values: See Appendix C of the fix protocol specification.
336	→ TradingSessionID	N	N	W_MAIN, ONE_MAIN, CFE_MAIN, W_STOCK, UNDERLYING

276	→ QuoteCondition	N	N	Space-delimited list of conditions describing a quote.
				Space-delimited list of conditions describing a quote.
				Valid values:
				A = Open / Active
				B = Closed / Inactive
				C = Exchange Best
				D = Consolidated Best
				E = Locked
				F = Crossed
				G = Depth
				H = Fast Trading
				I = Non-Firm

277	→ TradeCondition	N	N	Space-delimited list of conditions describing a trade
283	→ LocationID	N	N	This field will be blank.
284	→ DeskID	N	N	This field will be blank.
286	→ OpenCloseSettleFlag	N	N	Used if MDEntryType = Opening Price(4) or Closing Price(5).
59	→ TimeInForce	N	N	Absence of this field indicates Day order. Refer to Contingency Mapping Table. FIX 4.2 supports the following values for TimeInForce: 0 = Day 1 = Good Till Cancel (GTC) 2 = At the Opening (OPG) 3 = Immediate or Cancel (IOC) 4 = Fill or Kill (FOK)
18	→ ExecInst	N	N	Can contain multiple instructions, space delimited. Refer to Contingency Mapping Table. The following values are supported in FIX 4.2 and are mapped to ORS contingencies: 1 = Not Held G = All or None (AON)
287	→ SellerDays	N	N	This field will be blank.
346	→ NumberOfOrders	N	N	In an Aggregated Book, used to show how many individual orders make up an MDEntry
290	→ MDEntryPositionNo	N	N	Display position of a bid or offer, numbered from most competitive to least competitive, per market side, beginning with 1
58	→ Text	N	N	Text to describe the Market Data Entry. Part of repeating group.

End Repeating Group				
Extensions used in CBOE FIX 4.2 Service				
9316	***LegalMarket***	***N***	***N***	***User Defined Tag***
9314	***OpenInterest***	***N***	***N***	OpenInterest is not supported by CBOEdirect or OneChicago and can be obtained from the Options Clearing Corporation (OCC) or CBOE Financial Network (CFN).
6699	*ApplicationQueueDepth*	*N*	*N*	*A User defined field that provides the number of application level events that are queued for processing behind this current message. For instance, when ApplicationQueueDepth > 0 on a corresponding application response message sent from CBOE to the firm, this indicates that there are still ApplicationQueueDepth # of reports that have yet to be generated and transmitted to the user. This information is provided to help counter parties manage throughput and backlog issues.*
6701	*ApplicationQueueActionTaken*	*N*	*N*	*Optional application message field that indicates the action taken to resolve the Application queue depth (backlog).* *0- No action taken* *1- Queue flushed* *2- Overlaid last* *3- Subscription Ended* *This tag will be applicable ONLY on Market Data response messages.*
	Standard Trailer	Y	Y	

Market Data Request Reject

If an invalid market data request is sent to CBOE– a Market Data Request Reject message will be sent as the response. There will be no market data generated when an invalid request is received. Refer to the *MDReqRejReason[281]* field for the reject reason. Additional error description and diagnostic information is provided in the *Text[58]* field.

Market Data Reject Reason Codes

CBOE added four new enumerations to the MDReqRejReason[281] codes. This is provided in the Market Data Request Reject message

D	Use Specified Engine
E	Already Subscribed
F	Subscription Replaced
G	Forced Unsubscribe

D) Use Specified Engine
Used to "reroute" a specific MarketData request to another FIX engine that can handle it.

- It can be in direct response to a MarketDataRequest -- no subscription takes place on the original engine
- Can be sent in a subsequent "loss-of-data-feed" scenario -- with the original engine automatically unsubscribing the affected MDRefReqIDs.

In lieu of a new tag for the field containing the new "rerouted" connection information, the new routing FIX engine will be in the OnBehalfOfCompID tag. Upon receipt, firms should look up this routing information in the provided table, and establish a connection to the new FIX engine.

E) Already Subscribed

- Prevents duplicate subscriptions for the same product

- Prevents duplicate subscriptions for a product already subscribed for as a part of a larger subscription (option class, etc) containing that product.

F) Subscription Replaced

- Removes duplicate subscriptions for a product upon a larger subscription (option class, etc.) containing that product

G) Forced Unsubscribe

- Allows the engine to remove subscriptions, for throughput, bandwidth, or other reasons, but not for "Use Specified Engine" reasons. For example, if the firm had a backlog of market data messages, the firm would be unsubscribed to that particular type of market data and notified by the Market Data Reject message.

Table 5 Market Data Request Reject

Tag	Field Name	Comments
	Standard Header	*MsgType[35]* = Y
262	MDReqID	Will refer to the MDReqID of the invalid request.
281	MDReqRejReason	This will state the reason for the reject of the market data request. Reason for the rejection of a Market Data request. Valid values: 0 = Unknown symbol 1 = Duplicate MDReqID 2 = Insufficient Bandwidth 3 = Insufficient Permissions 4 = Unsupported SubscriptionRequestType 5 = Unsupported MarketDepth 6 = Unsupported MDUpdateType 7 = Unsupported AggregatedBook 8 = Unsupported MDEntryType D = Use Specified Engine E = Already Subscribed F = Subscription Replaced G = Forced Unsubscribe
58	Text	This field may contain an error message from CBOEdirect
115	OnBehalfOfCompID	In the event of a Market Data Reject with tag MDReqRejReason[281] set to = ‘D’, the new routing FIX engine name will be in this tag. Upon receipt, firms should look up this routing information in the provided table, and establish a connection to the new FIX engine.
	Standard Trailer	

Current Market (Top of Book)

Subscribe for Current Market Data

In addition to the standard fields required for a market data request as specified above, the fields below must be set to request current market data for a product.

Table 6 Current Market (Top of Book) Market Data Request

Tag	Field Name	Value you must specify
264	MarketDepth	1-Top of book
265	MDUpdateType	0-Full Refresh
9315	MDScope	1-Local
267	NoMDEntryTypes	2
269	MDEntryType	0-Bid (Same for MLEG products)
269	MDEntryType	1-Offer (Opposite for MLEG products)

CFIX users that want to receive current market information for Customer and Professional size need to indicate the appropriate UDF on their logon message to CFIX via tag 9003. Users need to set tag 9003 to either a value of 2 or 4 (see Logon Message fix-03a for further details). A value of "2" indicates that users will support UDF on all FIX messages. A value of "4" indicates support for UDF on *Market Data MsgType[35] = W*. CFIX will then provide Customer and Professional information on the current market messages going out to the users. Customer quantity will be provided on ***CustomerSize[tag9004]*** and Professional information will be on ***ProfessionalSize[tag9005]***. Users need to support the presence of these tags on *Market Data MsgType[35] = W* message types, and be able to interpret the information on these tags.

Current Market (Top of book) response

The top of the book information provided by CBOE is separated by order contingency type. Instead of providing one value and quantity for the CBOE best bid and offer – there are multiple quantities returned corresponding to the contingency types of orders.

Table 7 Current market (top of book) response

Tag	Field Name	Comments
	Standard Header	*MsgType[35]* = W
262	MDReqID	Corresponds to MDReqID provided on the Market Data Request
55	Symbol	Corresponds to Symbol provided on the Market Data Request
48	SecurityID	Corresponds to SecurityID provided on the Market Data Request
22	IDSource	“8”
167	SecurityType	Corresponds to SecurityType provided on the Market Data Request
200	MaturityMonthYear	Corresponds to MaturityMonthYear provided on the Market Data Request
205	MaturityDay	Corresponds to MaturityDay provided on the Market Data Request
201	PutOrCall	Corresponds to PutOrCall provided on the Market Data Request
202	StrikePrice	Corresponds to StrikePrice provided on the Market Data Request
207	SecurityExchange	W = CBOE
268	NoMDEntries	Number of entries following.
Begin Repeating Group for NoMDEntries		
269	***MDEntryType***	One or more Bid(0) price and size (Same for MLEG products), One or more Offer(1) price and sizes (Opposite for MLEG products)
270	***MDEntryPx***	Price of Bid or Offer (or Same or Opposite)
271	***MDEntrySize***	Quantity of this particular bid or offer (or Same or Opposite)

276	***QuoteCondition***	This denotes if the bid or offer is NBBO.
		D = CONSOLIDATED_BEST (NBBO)
		C = EXCHANGE_BEST (not NBBO)

273	***MDEntryTime***	Time entry was created
336	***TradingSessionID***	W_MAIN, ONE_MAIN, CFE_MAIN, W_STOCK, UNDERLYING
346	***NumberOfOrders***	In an Aggregated Book, used to show how many individual orders make up an MDEntry
59	***TimeInForce***	Absence of this field indicates Day order. Refer to Contingency Mapping Table. FIX 4.2 supports the following values for TimeInForce: 0 = Day 1 = Good Till Cancel (GTC) 2 = At the Opening (OPG) 3 = Immediate or Cancel (IOC) 4 = Fill or Kill (FOK)

18	***ExecInst***	Can contain multiple instructions, space delimited. Refer to Contingency Mapping Table. The following values are supported in FIX 4.2 and are mapped to ORS contingencies: 1 = Not Held G = All or None (AON)
58	***Text***	
9004	***CustomerSize***	This indicates the quantity of the bid or offer that is of customer origin. This size + ProfessionalSize = total bid or offer size. Only applies to the W_MAIN session. This tag will not appear for messages in the CFE_MAIN or ONE_MAIN sessions.
9005	***ProfessionalSize***	This indicates the quantity of the bid or offer that is of professional origin. This size + CustomerSize = total bid or offer size. Only applies to the W_MAIN session. This tag will not appear for messages in the CFE_MAIN or ONE_MAIN sessions.
End Repeating Group		
	Standard Trailer	

Current Market Examples

Subscribe Current Market

```
8=FIX.4.2^49=LLO^56=TESTFIX201^35=V^262=CURMKT_CLASS^263=1^ 264=1^ 265=0^ 9315=1^ 267=2^
269=0^ 269=1^ 146=1^ 55=OEX^ 167=OPT^ 336=W_MAIN^
```

Current Market Response

```
8=FIX.4.2^9=0234^35=W^49=TESTFIX201^56=LLO^34=1118^52=20010505-20:16:45^
262=CURMKT_CLASS^55=OEZ^48=1245419^22=8^167=OPT^200=200109^205=22^202=780.0^207=W^268=2^
269=0^270=0.0^271=0.0^273=0:0:0.0^336=W_MAIN^276=C^346=1^269=1^270=0.0^271=0.0^276=C^
346=1^
```

Unsubscribe Current Market

```
8=FIX.4.2^ 49=LLO^ 56=TESTFIX201^ 35=V^ 262=CURMKT_CLASS^ 263=2^ 264=1^ 265=0^ 9315=1^
267=2^ 269=0^ 269=1^ 146=1^ 55=OEX^ 167=OPT^ 336=W_MAIN^
```

Ticker Data (Last Sale)

Ticker data is available for both underlying products and products traded on CBOEdirect markets. However, ticker data is not supported in the CBSX session.

Subscribe for Ticker

In addition to the standard fields required for a market data request as specified above, the following fields must be specified to request the ticker (last sale) for an underlying or a CBOEdirect traded product.

Table 8 Ticker (Last Sale) Market Data Request

Tag	Field Name	Value you must specify
264	MarketDepth	1-Top of book
265	MDUpdateType	0-Full Refresh
267	NoMDEntryTypes	1
269	MDEntryType	2-Trade
Extensions		
9315	MDScope	1-Local

Ticker response

There is a single market data entry published for each price change that occurs.

Table 9 Market Data – Ticker

Tag	Field Name	Comments
	Standard Header	*MsgType[35]* = W
262	MDReqID	Corresponds to the MDReqID provided on the Market Data Request
55	Symbol	Corresponds to the Symbol provided on the Market Data Request
48	SecurityID	Corresponds to the SecurityID provided on the Market Data Request
22	IDSource	“8”
167	SecurityType	Corresponds to the SecurityType provided on the Market Data Request
200	MaturityMonthYear	Corresponds to the MaturityMonthYear provided on the Market Data Request
205	MaturityDay	Corresponds to the MaturityDay provided on the Market Data Request
201	PutOrCall	Corresponds to the PutOrCall provided on the Market Data Request
202	StrikePrice	Corresponds to the StrikePrice provided on the Market Data Request

Tag	Field Name	Comments
207	SecurityExchange	W = CBOE
268	NoMDEntries	**1**
Begin Repeating Group for NoMDEntries		
269	***→MDEntryType***	**2**-Trade
270	***→MDEntryPx***	Trade price
271	***→MDEntrySize***	Trade size
273	***→MDEntryTime***	Trade time
275	***→MDMkt***	W = CBOE
336	***→TradingSessionID***	W_MAIN, ONE_MAIN, CFE_MAIN, W_STOCK, UNDERLYING
277	***→TradeCondition***	Space-delimited list of conditions describing a trade
58	***→Text***	This field will be blank.
End Repeating Groups		
	Standard Trailer	

Ticker Example for CBOE Traded Product

Subscribe Ticker for CBOE traded product

```
8=FIX.4.2^ 49=LLO^ 56=TESTFIX201^ 35=V^ 262=TICKER_PRODUCT^ 263=1^ 264=1^ 265=0^ 9315=1^
267=1^ 269=2^ 146=1^ 55=OEZ^ 167=OPT^ 200=200109^ 201=0^202=760^336=W_MAIN^
```

Ticker Response for CBOE traded product

```
8=FIX.4.2^ 9=0157^ 35=W^ 49=TESTFIX201^ 56=LLO^ 34=237^ 262=TICKER_PRODUCT^ 52=20010505-
20:15:13^ 55=OEZ^48=1245417^22=8^167=OPT^200=200109^ 205=22^ 202=760.0^ 207=W^ 268=1^
269=2^ 270=0.0^ 271=0.0^ 336=W_MAIN^
```

Unsubscribe Ticker for CBOE traded product

```
8=FIX.4.2^49=LLO^56=TESTFIX201^35=V^262=TICKER_PRODUCT^263=2^264=1^ 265=0^9315=1 ^267=1
^269=2^ 146=1^ 55=OEZ^ 167=OPT^200=200109^201=0^ 202=760^336=W_MAIN^
```

Ticker Example for Underlying Product

To subscribe for ticker for an underlying product you must designate the underlying symbol, the underlying security type, and the underlying trading session. The underlying trading session is the literal string “Underlying”.

Subscribe Ticker for underlying product

```
8=FIX.4.2^49=LLO^56=TESTFIX201^35=V^262=TICKER_UNDERLYING^263=1^
264=1^265=0^9315=1^267=1^269=2^146=1^55=OEX^167=INDX^200=200109^
201=0^202=760^336=Underlying^
```

Ticker Response for underlying product

```
8=FIX.4.2^9=0157^35=W^49=TESTFIX201^56=LLO^34=237^
262=TICKER_UNDERLYING^52=20010505-20:15:13^
55=OEX^48=1245417^22=8^167=INDX^200=200109^205=22^
202=760.0^207=W^268=1^269=2^270=0.0^271=0.0^336=Underlying^
```

Unsubscribe Ticker for underlying product

```
8=FIX.4.2^49=LLO^56=TESTFIX201^35=V^262=TICKER_UNDERLYING^263=2^
264=1^265=0^9315=1^267=1^269=2^146=1^55=OEX^167=INDX^200=200109^
201=0^202=760^336=Underlying^
```

Recap Data

Request Recap

In addition to the standard fields required for a market data request as specified above, the following fields must be specified to request the recap for an underlying or a CBOEdirect traded product.

Table 10 Recap Market Data Request

Tag	Field Name	Value you must specify
264	MarketDepth	0-Full Book
265	MDUpdateType	0-Full Refresh
9315	MDScope	1-Local
267	NoMDEntryTypes	7
269	MDEntryType	0-Bid (only used for Underlying products)
269	MDEntryType	1-Offer (only used for Underlying products)
269	MDEntryType	2-Trade
269	MDEntryType	4-Opening Price
269	MDEntryType	5-Closing Price
269	MDEntryType	7-Trading Session High Price
269	MDEntryType	8-Trading Session Low Price

Recap response

For underlying products, CBOEdirect provides a recap message that contains last sale, session high and low, current bid and offer (top of the book), open price, and previous closing price.

For options and security futures products, ignore the current bid and offer (top of book) information that is published at the time of recap because at this time it will *not* contain usable values.

Table 11 Recap response

Tag	Field Name	Comments
	Standard Header	*MsgType[35]* = W
262	MDReqID	Corresponds to the MDReqID provided on the Market Data Request
55	Symbol	Corresponds to the Symbol of the product on which the recap is based.
48	SecurityID	Corresponds to the SecurityID of the product on which the recap is based.
22	IDSource	“8”

Tag	Field Name	Comments
167	SecurityType	Corresponds to the SecurityType of the product on which the recap is based.
200	MaturityMonthYear	Corresponds to the MaturityMonthYear of the product on which the recap is based.
205	MaturityDay	Corresponds to the MaturityDay of the product on which the recap is based.
201	PutOrCall	Corresponds to the PutOrCall of the product on which the recap is based.
202	StrikePrice	Corresponds to the StrikePrice of the product on which the recap is based.
207	SecurityExchange	W = CBOE
387	TotalVolumeTraded	
9314	OpenInterest	OpenInterest is not supported by CBOEdirect or OneChicago and can be obtained from the Options Clearing Corporation (OCC) or CBOE Financial Network (CFN).
268	NoMDEntries	7
Begin Repeating Group		
269	***→MDEntryType***	One entry for each of the following 0-Bid 1-Offer 2-Trade 4-Opening Price 5-Closing Price 7-Session High Price 8-Session Low Price
270	***→MDEntryPx***	Price of the bid, offer, etc. as specified in *MDEntryType[269]*
271	***→MDEntrySize***	Provided forBid(0), Offer(1), or Trade(2)
272	***→MDEntryDate***	Date of the recap.
273	***→MDEntryTime***	Time of the recap.
274	***→TickDirection***	
275	***→MDMkt***	W = CBOE
336	***→TradingSessionID***	W_MAIN, ONE_MAIN, CFE_MAIN, W_STOCK, UNDERLYING
276	***→QuoteCondition***	
277	***→TradeCondition***	
58	***→Text***	
End Repeating Group		

Tag	Field Name	Comments
	Standard Trailer	

Recap Example for Class

Subscribe Recap for class

8=FIX.4.2^49=LLO^56=TESTFIX201^**35=V**^262=*RECAP_CLASS*^263=1^264=0^
265=0^9315=1^267=7^269=2^269=0^269=1^269=4^269=5^269=7^269=8^146=1^
55=OEX^167=OPT^336=W_MAIN^

Recap Response for class

8=FIX.4.2^9=0333^**35=W**^49=TESTFIX201^56=LLO^34=245^52=20010505-20:15:37^
262=*RECAP_CLASS*^55=OEZ^48=1245419^22=8^167=OPT^200=200109^205=22^
202=780.0^207=W^268=8^269=2^270=0.0^271=0.0^273=1:0:0.0^336=W_MAIN^269=0^
270=0.0^271=0.0^273=0:0:0.0^269=1^270=0.0^271=0.0^273=0:0:0.0^
269=8^270=0.0^269=7^270=0.0^269=4^270=0.0^269=5^270=0.0^269=6^270=0.0^9314=0^

Unsubscribe Recap for class

8=FIX.4.2^49=LLO^56=TESTFIX201^35=V^262=RECAP_CLASS^263=2^264=0^
265=0^9315=1^267=7^269=2^269=0^269=1^269=4^269=5^269=7^269=8^146=1^
55=OEX^167=OPT^336=W_MAIN^

Expected Opening Price

Subscribe for Expected Opening Price

In addition to the standard fields required for a market data request as specified above, the following fields must be specified to request the expected opening price for a CBOE traded product

Table 12 Expected Opening Price Market Data Request

Tag	Field Name	Value you must specify
264	MarketDepth	0-Full Book Depth
265	MDUpdateType	0-Full Refresh
286	OpenCloseSettleFlag	3=Expected Opening Price
267	NoMDEntryTypes	1
269	MDEntryType	4-Opening Price

Expected Opening Price (EOP) response

The expected opening price contains the price at which a product is expected to open. Expected opening messages are sent prior to opening in addition to the current opening process. The messages will be sent at pre-determined intervals, currently planned for every 30 seconds.

Table 13 Expected Opening Price response

Tag	Field Name	Comments
	Standard Header	*MsgType[35]* = W
262	MDReqID	Corresponds to the MDReqID provided on the Market Data Request
55	Symbol	Corresponds to the Symbol of the product on which the EOP is based.
48	SecurityID	Corresponds to the SecurityID of the product on which the EOP is based.
22	IDSource	“8”
167	SecurityType	Corresponds to the SecurityType of the product on which the EOP is based.
200	MaturityMonthYear	Corresponds to the MaturityMonthYear of the product on which the EOP is based.
205	MaturityDay	Corresponds to the MaturityDay of the product on which the EOP is based.
201	PutOrCall	Corresponds to the PutOrCall of the product on which the EOP is based.

202	StrikePrice	Corresponds to the StrikePrice of the product on which the EOP is based.
207	SecurityExchange	W = CBOE
9316	LegalMarket	*"Y" if legal market*
268	NoMDEntries	Number of entries in the following repeating group.
	Begin Repeating Group	
269	***→MDEntryType***	Must be the first field in this repeating group.
270	***→MDEntryPx***	Expected opening price
271	***→MDEntrySize***	Imbalance quantity if one exists
273	***→MDEntryTime***	Time that the EOP is published.
275	***→MDMkt***	W = CBOE
336	***→TradingSessionID***	W_MAIN, ONE_MAIN, CFE_MAIN, W_STOCK, UNDERLYING

277	***→TradeCondition***	Can contain one of the following values:
		Imbalance_More_Buyers = "P";
		This EOP message type indicates that the current market is imbalanced and will need more sellers' contracts (n) at the opening price. The EOP message will provide users with the potential opening price (bid price) and the imbalanced quantity (n).
		Imbalance_More_Sellers = "Q";
		This EOP message type indicates that the current market is imbalanced and will need more buyers' contracts (m) at the opening price. The EOP message will provide users with the potential opening price (ask price) and the imbalanced quantity (m).
		Opening_Price = "R";
		No_Opening_Trade = "S";
		Need_Quote_To_Open = "U";
		Price_Not_In_Quote_Range = "V";
		Need_DPM_Quote_To_Open = "W"
		PRICE_NOT_IN_BOTR_RANGE = "X"
		This EOP message type indicates that the potential opening price is not in the valid Best of the Rest (BOTR) range based on the current BOTR price, current potential opening quantity and the system configuration. The EOP message will provide users with the potential opening price and opening quantity.

286	***→OpenCloseSettleFlag***	3 – expected opening price
58	***→Text***	Text to describe the Market Data Entry. Part of repeating group.
	End Repeating Group	
	Standard Trailer	

The Hybrid openings have changed to calculate the opening range based on the midpoint of the Best quote (or I order) bid and the best quote (or I order) ask, minus/plus half the spread width for the price range. (Ex. Best bid 1.00, best ask 1.10

Opening Spread width .25 for non-LEAP option will calculate an opening range of .925 - 1.175 which will round to .90 - 1.20)

Example Price Not in BOTR range:

Opening quote range: 1.00 - 1.10 20 x 20

Order to sell 10 at the market; BOTR Range = 1.05 - 1.10

Message X sent to users will include the CBOE calculated opening price of 1.00 and the size of 10.

Example Need more buyers:

Opening quote range: 1.00 - 1.10 20 x 20

Order to sell 50 at the market

Message P sent to users will include the Bid price of 1.00 and the size of 30.

Example Need more sellers:

Opening quote range: 1.00 - 1.10 20 x 20

Order to buy 50 at the market

Message Q sent to users will include the Ask price of 1.10 and the size of 30.

Note: Tag 270 [MDEntryPx] will be populated with the potential opening price when TradeConditon equals P or Q. Firms should verify that their application is programmed to handle this tag and make use of this value.

Expected Opening Price Example

Subscribe for Expected Opening Price

8=FIX.4.2^49=LLO^56=TESTFIX201^**35=V**^262=*EOP_FOR_OEX*^**263=1**^264=0^265=0^
286=3^267=1^269=4^146=1^55=OEX^167=OPT^336=W_MAIN^

Expected Opening Price Response

Please ask your API Client Relations representative for EOP Response examples.

Unsubscribe for Expected Opening Price

8=FIX.4.2^49=LLO^56=TESTFIX201^**35=V**^262=*EOP_FOR_OEX*^**263=2**^264=0^265=0^
286=3^267=1^269=4^146=1^55=OEX^167=OPT^336=W_MAIN^

Book Depth

In addition to the standard fields required for a market data request as specified above, the following fields must be specified to query or subscribe to the book depth of a CBOEdirect traded product. Book depth can be accessed for a product either by querying *SubscriptionRequestType[263]*=0-SNAPSHOT or via subscription using *SubscriptionRequestType[263]*=1-Subscribe. Your firm may query or subscribe to book depth by product only. CBOE does not permit querying or subscribing to book depth by class.

W_MAIN or Hybrid Book Depth

Users can query book depth snapshots in the W_MAIN session only once per user for all products every 10 minutes. CBOE will not support book depth subscriptions in W_MAIN. W_MAIN book depth subscriptions will be accepted, but never published, as CBOE currently has a book depth level set to “0” for subscriptions.

OneChicago and CBOE Futures Exchange

OneChicago book depth will be available via the CMi, FIX, and CBOE Financial Network (CFN) interfaces. CBOE Futures Exchange (CFE) book depth will be available only through the CFN interface (specification available at API web site).

Query Book Depth

This is only available once every ten minutes per user in the W_MAIN session. This is not allowed for new development in CBOE Futures Exchange and OneChicago.

Table 14 Recap Market Data Request

Tag	Field Name	Value you must specify
263	SubscriptionRequestType	0-Snapshot
264	MarketDepth	0-Full Book Depth
265	MDUpdateType	0-Full Refresh
9315	MDScope	0-Local
267	NoMDEntryTypes	2
269	MDEntryType	0-Bid (Same for MLEG products)
269	MDEntryType	1-Offer (Opposite for MLEG products)

Subscribe for Book Depth

This is not allowed for new development in any session.

Table 15 Recap Market Data Request

Tag	Field Name	Value you must specify
263	SubscriptionRequestType	1-Snapshot+Update (Subscribe)
264	MarketDepth	0-Full Book Depth
265	MDUpdateType	0-Full Refresh
9315	MDScope	0-Local

Table 15 Recap Market Data Request

Tag	Field Name	Value you must specify
267	NoMDEntryTypes	2
269	MDEntryType	0-Bid (Same for MLEG products)
269	MDEntryType	1-Offer (Opposite for MLEG products)

Book Depth Response

Book depth contains two MDEntries for each price level one for the bid volume and one for the ask volume. Either the bid or ask can volume can possibly be zero. The contingency volume (integer) is placed in the Text[58] field. FIX 4.2 does not support a separate tag in release 4.2 for contingency volume – but will in FIX 4.3 – so the Text[58] field was used as it is part of the repeating group.[1]

Table 16 Book depth response

Tag	Field Name	Comments
	Standard Header	*MsgType[35]* = W
262	MDReqID	Will correspond to the MDReqID provided on the Market Data Request
55	Symbol	Symbol of the product whose book is being published
48	SecurityID	CBOEdirect product key of the product whose book is being published
22	IDSource	8
167	SecurityType	SecurityType of the product whose book is being published
200	MaturityMonthYear	MaturityMonthYear of the product whose book is being published
205	MaturityDay	MaturityDay of the product whose book is being published
201	PutOrCall	PutOrCall of the product whose book is being published
202	StrikePrice	StrikePrice of the product whose book is being published
207	SecurityExchange	W = CBOE
268	NoMDEntries	0 - n(5) There may be as few as zero (0) entries in the book. CBOEdirect plans to restrict the maximum number of prices (an aggregate of all orders and quotes at that price) deep that book depth will be published. CBOE plans to publish up to five (5) prices deep, but that number is subject to change.

[1] The Text[58] field was used instead of a user defined tag for ContingencyVolume because many FIX engines cannot process user defined tags in repeating groups.

Table 16 Book depth response

Tag	Field Name	Comments
	Begin Repeating Group	
269	***→MDEntryType***	One entry for each bid and offer pair in the book. This value will be "0" when it is for a bid and "1" for an offer.
270	***→MDEntryPx***	The price of the book entry that is being published
271	***→MDEntrySize***	The aggregate quantity of all orders (including contingency quantity orders) and quotes that are in the CBOEdirect book at the published book price.
272	***→MDEntryDate***	Date that the book entry is published.
273	***→MDEntryTime***	Time that the book entry is published.
275	***→MDMkt***	W = CBOE
336	***→TradingSessionID***	W_MAIN, ONE_MAIN, CFE_MAIN, W_STOCK
58	***→Text***	The aggregate contingency volume (integer) of all orders at the price of the book entry is placed in this field.
	End Repeating Group	
	Standard Trailer	

Book Depth Example

Book Depth Subscription Request (by product only)

```
8=FIX.4.2^49=TEST11^56=DFIX201^35=V^262=0002BOOK_DEPTH10^263=1^264=0^265=0^267
=2^269=0^269=1^146=1^55=IBM1C^200=200209^167=FUT^336=ONE_MAIN^
```

National Best Bid and Offer (NBBO)

NBBO is not implemented at this time.

Requesting NBBO

In addition to the standard fields required for a market data request as specified above, the following fields must be specified to request the national best bid and offer for a product. NBBO is not implemented at this time.

Table 17 National Best Bid and Offer Market Data Request

Tag	Field Name	Value you must specify
264	MarketDepth	1-Top of book
265	MDUpdateType	0-Full Refresh
9315	MDScope	1-National
267	NoMDEntryTypes	2
269	MDEntryType	0-Bid
269	MDEntryType	1-Offer

National Best Bid and Offer (NBBO) response

The national best bid and offer (NBBO) provides the best bid and offer across all options exchanges. There is one market data entry for each exchange that has a bid/offer at the given best price. NBBO is not implemented at this time.

Table 18 National best bid and offer (NBBO) response

Tag	Field Name	Comments
	Standard Header	*MsgType[35]* = W
262	MDReqID	Will correspond to the MDReqID provided on the Market Data Request
55	Symbol	Symbol whose NBBO is being published
48	SecurityID	CBOEdirect product key of the product whose NBBO is being published
22	IDSource	8
167	SecurityType	SecurityType of the product whose NBBO is being published
200	MaturityMonthYear	MaturityMonthYear of the product whose NBBO is being published
205	MaturityDay	MaturityDay of the product whose NBBO is being published
201	PutOrCall	PutOrCall of the product whose NBBO is being published
202	StrikePrice	StrikePrice of the product whose NBBO is being published
207	SecurityExchange	W = CBOE
268	NoMDEntries	Number of entries following.
Repeating Group repeated NoMDEntries[268] times		

Tag	Field Name	Comments
269	**→MDEntryType**	One or more Bid(0) price and size, One or more Offer(1) price and sizes
270	**→MDEntryPx**	Price of bid or offer
271	**→MDEntrySize**	Quantity of this particular bid or offer
273	**→MDEntryTime**	Time entry was created
275	**→MDMkt**	Exchange code for the entry
336	**→TradingSessionID**	W_MAIN, ONE_MAIN, CFE_MAIN, W_STOCK
276	**→QuoteCondition**	“D”
58	**→Text**	
	Standard Trailer	

Quoting and Mass Quoting Interface

The FIX *Mass Quote* and the *Quote* messages can be used to submit quotes to CBOE electronic markets. The CBOE requires the Lead Market-Maker (LMM or DPM) and Market-Maker (MM) CBOEdirect roles to use the Mass Quote message to quote multiple options or futures products. The Quote message is more likely to be used to respond to a Quote Request (Request For Quote, or RFQ) on a particular product that may not have a current quote. The *Quote Acknowledgement* message is used in response to a Mass Quote, Quote, *Quote Status Requests*, and *Quote Cancel* submission from the user. The *Quote Status Request* message is used to query and to subscribe for status on quotes that have been submitted to CBOEdirect electronic markets. A slightly enhanced version of the *Quote* message is used as the reply to a Quote Status Request.

In the Mass Quote Message, the number of products per message will be restricted to a value (n), which is currently set to 400 in the W_MAIN session, and 12 in the ONE_MAIN session (see section on Quote Throttling). The steps involved in quoting are:

- Subscribe for Request for Quotes (transmitted as *Quote Request Messages*) for product classes you are obligated to quote using the *Quote Status Request* message.
- Submit multiple quotes using the Mass Quote message or a single quote using the Quote Message.
- Each Mass Quote message [35=i] from the user is acknowledged with a Quote Acknowledgement Message[35=b] from CBOE.
- Each Quote message [35=S] is acknowledged with a Quote Message [35=S] from CBOE.
- Each fill that occurs against a quote is acknowledged using an Execution Report.[2]
- Any trade busted by CBOE personnel (after consultation with both counterparties) is acknowledged using an Execution Report with *ExecTransType[20]* = ***CANCEL***. The *ExecID[17]* of the fill that is being busted is provided in the *ExecRefID[19]* field. The CBOE requires that all LMMs and market-makers who quote are able to receive complete and partial quote trade busts.
- Only the FIX user that sends a quote to CBOE can receive a copy of the Quote Execution Report. If another user queries quote status, or tries to modify or cancel the quote, CBOE will reject that request. Firms can obtain duplicate copies of their execution reports via the RTC interface. RTC documents are available on the CBOE API web site.
- Execution Reports for quotes will have the *OrderID[37]* set to the *QuoteID[117]* of the quote for which the fill is being reported.

Lifetime and disposition of quotes

Quotes are only active while the user is connected to the FIX session. Quotes will be removed from the system if the user is logged off the system for any reason. This is done to make sure that the user has control over their markets. Quotes will remain active while the connection is maintained until the end of the trading session or until the quote has been fully traded.

There is only one quote active per user per product at a time. Submission of a new quote overwrites the previous quote. There is not a notion of quote history – a quote does not keep an accumulated quantity. You must keep track of this using execution reports that are generated as a result of trades against your quotes.

[2] Clearing information (member firm, CMTA, Q Account) for the resultant trade is configured by user out of band by CBOE help desk personnel. Contact the API relations group for more specifics.

Quote Throttling

CBOE regulates the number of quotes that can be generated at one time. When a quote or mass quote message is not accepted due to the quote limit being exceeded, you will receive a Quote Acknowledgement response that indicates that quote(s) has not been accepted. The most common recovery action is to resubmit the quote. CBOEdirect will reject the entire block of quotes that puts the user over the threshold.

Quote rate limits are currently set as follows, but are configurable and subject to change.

Hybrid (in session W_MAIN)

400 Quotes (products) per Mass Quote message **(SEQUENCE_SIZE_EXCEEDED)**
133 Mass Quote or Quote message calls per user per 1 second period **(RATE_EXCEEDED)**
4000 total quotes (products) per user per 3 second period **(QUOTE_RATE_EXCEEDED)**

CFE_MAIN (CBOE Futures Exchange) **and ONE_MAIN** (OneChicago)

12 Quotes (products) per Mass Quote message
250 Mass Quote or Quote message calls per user per five (5) second period
1000 total quotes (products) per user per five (5) second period

W_STOCK (CBOE Stock Exchange)

100 Quote message calls per second

Quote Processing

The FIX order and quote processing mechanism is multi-threaded on a class basis. All orders and quotes entered through FIX belonging to a specific class are queued up one behind the other on the same thread. This architecture is efficient for users who send in orders and quotes on a large number of classes. It is not efficient for a user who is quoting with orders or quotes on a few classes.

CBOE has improved its quote processing time by allowing quotes to be submitted concurrently even for a given class. The redefined quote process allows multiple concurrent quote messages up to a configurable limit. FIX users no longer have to choose to use the concurrent thread model through the user defined field, *ConcurrentOrder/QuoteIndicator [9192]* in the *Logon Message (MsgType=A)*. The concurrent thread model is now the standard. The following behavior should be considered by the quoting Firm:

A quote method is any of the following requests: SingleQuote, MassQuote, cancelQuote, cancelQuotesByClass and cancelAllQuotes. Due to the asynchronous nature of the concurrent quote message calls; it is very important that the same series not be included in multiple quote blocks. While CBOE will not enforce any restriction with regard to this check, the order of processing of the individual calls is ***not*** guaranteed. The quote processing changes are as follows:

1. The fix engine will allow multiple concurrent quote messages up to a configurable limit. The limit at this time is 10.

2. Immediately before the quote message is to be dispatched the quote rate limits are checked. If the call or quote rate limits are not exceeded the message is dispatched.

3. When the number of concurrent quotes in-flight equals the maximum allowed, any new Quote message calls will be **rejected** with a NotAcceptedException. The error code in this case will be *EXCEEDS_CONCURRENT_QUOTE_LIMIT (tag 368=89*).

4. The cancelQuotesByClass and cancelAllQuotes requests will be forwarded to the server immediately without regard to the number of concurrent quote messages currently in progress.

5. While any cancelQuotesByClass is in flight to the server, any new quote request calls (includes *SingleQuote, MassQuote and cancelQuote*) will be **rejected** with a NotAcceptedException. The error code in this case will be *QUOTE_CANCEL_IN_PROGRESS (tag 368=88*).

6. Block quote cancels (block of 0-0 quotes coming through the MassQuote calls) are considered to be a MassQuote calls and could be rejected if the number of in-flight quote requests to the server exceed the limit or if a cancelQuotesByClass is in-flight.

7. A cancelAllQuotes request is always forwarded to the server. Due to the asynchronous nature of the cancel all quotes request, the fix engine will not prevent any new quotes message or quote cancel message (including another cancelAllQuotes) from being forwarded to server while a previous cancel all quotes request is in progress.

Quote Status

The CBOE FIX 4.2 Service will automatically subscribe your application for Quote Status upon transmission of the initial Mass Quote or Quote message. There is no manual way to subscribe for quote status at this time. Individual Quote Status messages (fill, bust, cancel, etc.) for Mass Quotes have been eliminated to reduce bandwidth requirements. Firms will now only receive one Quote Status message per Mass Quote message. If backlogs in receiving Quote Status and / or Execution Reports occur, the CBOE will remove the user's subscription to Quote Status and notify the user by sending a Business Reject Message with reason text. The user would have to re-register for Quote Status.

Subscribing for Quote Requests (RFQs)

Users who want to initiate quoting will first want to subscribe for *Quote Requests* (RFQs) for each class for which they plan to submit quotes. Subscription for quote status and quote requests is accomplished using the *Quote Status Request* message. The CBOE Fix 4.2 Service has extended the FIX 4.2 *Quote Status Request* message to include the *SubscriptionRequestType[263]*. To subscribe for quote requests, set the *SubscriptionRequestType[263]* to ***1***. To unsubscribe for quote requests – set the *SubscriptionRequestType[263]* to ***2***. CBOE requires that all LMMs and market-makers subscribe to receive Quote Requests.

CBOE requires futures market-makers and DPMs to subscribe to Quote Requests (RFQs). RFQs are not supported for options (Hybrid W_MAIN).

Using the Quote Status Request Message to Subscribe for Auctions

Firms must indicate interest in receiving solicitations to join auctions by sending a subscribe message. Subscription for auctions is performed using the *Quote Status Request* message. To subscribe for auction participation, set the *SubscriptionRequestType[9463]* to **500**. Only one Auction Type will be allowed per subscription. To unsubscribe from auction participation, set the *SubscriptionRequestType[9463]* to **501**. Subscribing for auction events for a session and class does not guarantee delivery of events for products in the class; business rules may dictate that auction notification be limited to certain users.

Table 19 Subscribe for Auctions using Quote Status Request message

Tag	Field Name	FIX Req'd	CBOE Req'd	Comments
	Standard Header	Y	Y	*MsgType[35]* = a (lowercase)
336	TradingSessionID	N	Y	**This tag is required for Auctions** **Options and Futures**: No more than one TradingSessionID must be specified: CBOE trading floor: 'W_MAIN' CBOE Futures Exchange: 'CFE_MAIN' OneChicago: 'ONE_MAIN' Stock Trading On CBOEdirect: 'W_STOCK' CBOE 2: "C2_MAIN"
55	Symbol	Y	Y	Symbol of the options or futures class for which quote status should be subscribed
167	SecurityType	N	Y	Security Type of the class for which quote status is being requested. Possible values at this time are `OPT, FUT, MLEG, CS`
	Extensions to Quote Status Request Message			
9463	SubscriptionRequestType	N	Y	This tag is required for auction subscription. Values 500 and 501 are used for auctions. 500= subscribe to auctions 501=unsubscribe from auction
9383	AuctionType	N	Y	This field is required for auctions. Values from cmiConstants::AuctionTypes: 1=internalization 2=strategy 3=regular_single d 4= HAL 5=SAL 6=STOCK_NBBO_FLASH – This is not a new auction type. This value is used as an NBBO indicator in the W_STOCK session. It is being used as an AuctionType in order to share the auction event channel. If the user ties to respond with an auction, a reject message will be received. 7=STOCK_ODD_LOT 8=AUCTION_DAIM Only one Auction Type will be allowed per subscription.
	Standard Trailer	Y	Y	

Examples

STOCK_NBBO_FLASH example

Subscribe for NBBO flash

```
[8=FIX.4.2^A9=0094^A35=a^A34=17^A49=TEST301^A56=DFIX301^A52=20071005-
15:18:47^A55=C^A167=CS^A336=W_STOCK^A9463=500^A9383=6^A10=210^A]
```

Put in a quote

```
[8=FIX.4.2^A9=0156^A35=S^A34=18^A49=TEST301^A56=DFIX301^A52=20071005-15:18:57^A117=RBJ0001-
20071005^A336=W_STOCK^A55=C^A167=CS^A132=49.0^A133=50.5^A134=1000^A135=1000^A60=20000302-
12:00:00^A10=093^A]
```

```
[8=FIX.4.2^A9=0142^A35=S^A34=18^A49=DFIX301^A56=TEST301^A52=20071005-15:18:57^A117=RBJ0001-
20071005^A336=W_STOCK^A55=C^A167=CS^A132=49^A133=50.5^A134=1000^A135=1000^A9312=New^A10=066^A
]
```

Get RFQ to better the Quote

```
[8=FIX.4.2^A9=0194^A35=R^A34=20^A49=DFIX301^A56=TEST301^A52=20071005-
15:19:39^A131=0:0^A146=1^A55=C^A48=69208659^A22=8^A167=CS^A201=0^A202=0^A207=W^A336=W_STOCK^A
54=1^A38=100^A60=00010709-16:19:39^A9384=1^A9302=50.2500^A9743=1^A9383=6^A10=008^A]
```

STOCK_ODD_LOT example

Odd lot Subscription Request

```
REQUEST:8=FIX.4.2^35=a^49=TEST302^50=SUN:SUN^56=DFIX301^57=TEST^55=C^167=CS^336=W_STOCK^9463=
500^9383=7^
```

Odd lot Response

```
RESPONSE:8=FIX.4.2^9=0203^35=R^34=53^49=DFIX301^56=TEST302^52=20080131-
17:20:22^131=65036:41022211^146=1^55=C^48=69208659^22=8^167=CS^201=0^202=0^207=W^336=W_STOCK
^54=1^38=10^60=00270731-17:20:21^9384=1^9302=1.2000^9743=1^9383=7^10=180^
```

Auction Subscription Example

Firm subscribes for auction solicitation

```
35=a^55=AOO^167=OPT^336=W_MAIN^9463=500^
```

Quote Identification

Users should provide a quote identifier QuoteID[117] on each quote. A firm can have only one two sided quote in effect for a product at a time. Because of this, QuoteIDs do not have to be unique. You have the choice to keep the same QuoteID across multiple submissions of quotes for a specific product.

Compliance with FIX 4.2

The CBOE quoting interface requires extension to the FIX 4.2 specification. CBOE worked with the FIX Protocol Limited Group (FPL) to make these enhancements part of the FIX 4.3 specification.[3]

Specific area of non-compliance:

Addition of *SubscriptionRequestType[263]* to the Quote Status Request message

The addition of QuoteStatus user defined tag to the Quote message.

The use of a Quote message as a response to a Quote Status Request.

[3] FPL, in FIX4.3, has issued a revised approach for both tradeable quoting models (as used by CBOE where quotes become orders) and indicative quotes (where quotes are used to indicate a market and do not become orders) that address the limitation in FIX 4.2. Specifically, FIX 4.3 has added a Quote Status message used a response to Quote Status Requests and a Request for RFQ to subscribe or request RFQs from a market place.

Table 20 Subscribe for Quote Requests using Quote Status Request message

Tag	Field Name	FIX Req'd	CBOE Req'd	Comments
	Standard Header	Y	Y	*MsgType[35]* = a (lowercase)
55	Symbol	Y	Y	Symbol of the options or futures class for which quote status should be subscribed
167	SecurityType	N	Y	Security Type of the class for which quote status is being requested. Possible values at this time are `OPT, FUT, MLEG, CS`
Extensions to Quote Status Request Message				
263	***SubscriptionRequestType***	***N***	***N***	***If specified as 0 or 1 will be used to register interest in receiving Quote Requests for the security identified in the security block.*** ***NOTE: This is not a FIX 4.2 Standard Field.***
	Standard Trailer	Y	Y	

Subscription Example – subscribe for quote requests for SPX options:

```
55=SPX^263=1^
```

Unsubscribing Example – unsubscribe for quote requests for SPX options:

```
55=SPX^263=2^
```

NOTE: The CBOE FIX 4.2 Service will automatically subscribe you for quote status upon transmission of the initial Mass Quote or Quote message. You must explicitly subscribe to receive quote requests (RFQs) using the Quote Status Request.

Using the Quote Message to submit a quote for a product to CBOEdirect

Users can submit quotes on individual products (such as futures or option series or options on futures) using the Quote Message. This is most useful for responding to Quote Requests, whereas the Mass Quote Message is required for quoting multiple products at one time (autoquoting).

A particular user session in CBOEdirect may only access one exchange at a time. For example, a single user may not send quotes to CBOE options and CFE at the same time. The firm must use two separate user IDs: one will send quotes to CBOE and one will send quotes to CFE.

In addition, two users sharing the same acronym cannot quote the same class. For example,

- User ID1 and user ID2 share the same acronym
- User ID1 is quoting class, IBM. User ID2 tries to quote IBM but the quote is rejected.
- If user ID1 logs off and all quotes are cancelled, user ID2 will be able to quote IBM.
- If user ID1 cancels all his/her quotes, without logging off, user ID2 will be able to quote IBM

Acknowledgement of the quote is done via transmission of a slightly enhanced version of the *Quote* Message from CBOEdirect.

Table 21 Quote Message used to enter a quote into CBOEdirect

Tag	Field Name	FIX Req'd	CBOE Req'd	Comments
	Standard Header	Y	Y	*MsgType[35]* = S
131	QuoteReqID	N	N	Required when quote is in response to a Quote Request message
117	QuoteID	Y	Y	Users should provide a quote identifier in this field on each quote sent to CBOEdirect. QuoteIDs do not have to be unique. You have the choice to keep the same QuoteID across multiple submissions of quotes for a specific product.
301	QuoteResponseLevel	N	N	CBOE does not support this tag. CBOE will always provide a response. This field is ignored on input.
336	TradingSessionID	N	Y	Trading Session that the product is in for which the quote is being entered W_MAIN, ONE_MAIN, CFE_MAIN, COF_MAIN, W_STOCK
55	Symbol	Y	Y	Symbol of the product for which the quote is being entered
48	SecurityID	N	N	SecurityID of the product for which the quote is being entered
22	IDSource	N	N	This value will be ingnored.
167	SecurityType	N	Y	OPT, FUT, MLEG, CS --Required for “by Name” lookup of product.
200	MaturityMonthYear	N	Y	MaturityMonthYear of the product for which the quote is being entered. Required for “by Name” look up of product.
205	MaturityDay	N	Y	MaturityDay of the product for which the quote is being entered. Required for “by Name” look up of product.
201	PutOrCall	N	Y	PutOrCall of the product for which the quote is being entered. Required for “by Name” look up of product.
202	StrikePrice	N	Y	StrikePrice of the product for which the quote is being entered. Required for “by Name” look up of product.

132	BidPx	N	N	Bid Price of the quote (Same Price of quote for MLEG products)
133	OfferPx	N	N	Offer Price of the quote (Opposite Price of quote for MLEG products)
134	BidSize	N	N	Bid Quantity of the quote (Same Quantity of quote for MLEG products)
135	OfferSize	N	N	Offer Quantity of the quote (Opposite Price of quote for MLEG products)
60	TransactTime	N	N	Time that the Quote Message is sent
20101	ShortSaleIndicator	N	Y	Used to indicate short sale positions for the CBSX session. Valid values for CBSX are: 1 = Buy 2 = Sell 5 = Sell_Short 6 = Sell_Short_Exemp
	Standard Trailer	Y	Y	

Quote Example

The following example shows a quote for the DJV Sep 2001 96.00 option.

```
35=S^117=MY_QUOTE_ID^336=W_MAIN^55=DJV^167=OPT^200=200109^201=1^202=96^132=7.9^133=8
.2^134=75^135=75^60="20000302-12:00:00"^
```

A Quote message will be issued in response to quote submission in the event that contains a user defined QuoteStatus[9312] with the status of the quote.

Quote Message outbound from CBOE to report quote status

The Quote Message is also used to communicate the status of a quote for the user. The quotes belonging to the current user are transmitted back to the user.

Table 22 Quote status using Quote Message

Tag	Field Name	Comments
	Standard Header	*MsgType[35]* = S
117	QuoteID	Corresponds to the QuoteID of the quote that was entered
336	TradingSessionID	Corresponds to the Trading Session of the product for which the quote was entered W_MAIN, ONE_MAIN, CFE_MAIN, COF_MAIN, W_STOCK
55	Symbol	Corresponds to the product Symbol in which the quote that was entered
48	SecurityID	Corresponds to the SecurityID (CBOEdirect product key) in which the quote that was entered
22	IDSource	8
167	SecurityType	Corresponds to the Security Type of the product for which the quote was entered
200	MaturityMonthYear	Corresponds to the MaturityMonthYear of the product for which the quote was entered
205	MaturityDay	Corresponds to the MaturityDay of the product for which the quote was entered
201	PutOrCall	Corresponds to the PutOrCall of the product for which the quote was entered
202	StrikePrice	Corresponds to the StrikePrice of the product for which the quote was entered
207	SecurityExchange	W = CBOE

Tag	Field Name	Comments
132	BidPx	Corresponds to the Bid/Same Price of the quote that was entered
133	OfferPx	Corresponds to the Offer/Opposite Price of the quote that was entered
134	BidSize	Corresponds to the Bid/Same Size of the quote that was entered
135	OfferSize	Corresponds to the Offer/Opposite Size of the quote that was entered
60	TransactTime	Corresponds to the QuoteID of the quote that was entered
Extensions to the Quote Message Outbound from CBOE		
9312	***QuoteStatus***	*This is a custom field that is used on the Quote message outbound from CBOE. The field is used to communicate the status of the quote back to the quote issuer. See possible values below.*
9008	*QuoteText*	**Quote Locked Messages** *When Quote Bid of one market-maker = Quote Ask of another market-maker in W_MAIN, a Quote Locked Notification status message will be sent out and will contain a string of buy side acronyms and sell side acronyms. When a broker-dealer user enters an order on behalf of an In-Crowd Market-Maker ("ICM") of order of origin "I", the Quote Locked message will report the acronym of the market-maker, not the broker-dealer.* *Example:* `CBOE:BUYER1,CBOE:BUYER2;CBOE:SELLER1,CBOE:SELLER2` **Quotes Canceled By QRM** *When a market-maker's quotes are canceled due to QRM, this tag will be set to:* *"QRM REMOVED"* **Failover Scenario** If the CBOE's servers are in failover mode, the CBOE will cancel the user's quotes, and then send out a message with this tag containing the value of: `FAILOVER`
6699	*ApplicationQueueDepth*	*A User defined field that provides the number of application level events that are queued for processing behind this current message. For instance, when ApplicationQueueDepth > 0 on a corresponding application response message sent from CBOE to the firm, this indicates that there are still ApplicationQueueDepth # of reports that have yet to be generated and transmitted to the user. This information is provided to help counter parties manage throughput and backlog issues.*
	Standard Trailer	

Quote Status Example

The following is an example of Quote Status Report:

```
35=S^117=MY_QUOTE_ID^336=W_MAIN^55=DJV^167=OPT^200=200109^201=1^202=96^132=7
.9^133=8.2^134=75^135=75^60="20000302-12:00:00"^9312=New
```

Quote Status [9312] Values

CBOEdirect will send Quote messages in response to changes in the status of a quote. The following values are returned in the QuoteStatus[9312] user defined tag field.

QuoteStatus[9312] value	Description
New	[Indicates the quote has been received by the market]
Buy Quote Trigger/Sell Quote Trigger	[Indicates a quote trigger generated by a quote]

QuoteStatus[9312] value	Description
Buy Quote Trigger Order/Sell Quote Trigger Order	[Indicates a quote trigger generated as a result of an "I" order]
Cancelled	[Indicates quote was cancelled at user request]
Removed	[Indicates quote was removed because user logged off]
Possible Resend	[PosResend[97] will be set = "Y"]
Locked (W_MAIN only)	["Locked" indicates that the Quote Bid = Quote Ask for Hybrid classes in W_MAIN.]
Busted	
Filled	
Open Outcry	
Query	
Reinstated	
Updated	

NOTE: The CBOE FIX 4.2 Quote interface is non-standard as extensions are used to implement electronic market style quoting. CBOE is working to add these extensions into FIX 4.3.

Quote Request (RFQ)

CBOE permits market participants to submit quote requests or requests for quotes (RFQ's) for any products in the ONE_MAIN session. RFQs are not supported in Hybrid W_MAIN at this time. Users responsible for making markets (market makers, LMMs, DPMs) are required to receive and respond to these RFQs within a ONE prescribed timeframe. Request for Quotes (RFQ's) will not be available via the CBOE FIX 4.2 market data server. RFQ's will only be available via connections used for quoting / order flow.

Market makers subscribe to receive Quote Requests (RFQs) using *the Quote Status Request* message with the *SubscriptionRequestType[263]* = 1.

NOTE: CBOE FIX 4.2 Service only permits one product per *Quote Request* message. Repeating groups subscribing to multiple products in one Quote Request message are not allowed

Table 23 Quote Request

Tag	Field Name	FIX Req'd	CBOE Req'd	Comments
	Standard Header	Y		*MsgType[35]* = R
131	QuoteReqID	Y		Unique identifier for quote request
146	NoRelatedSym	Y		**Must always = 1**
55	Symbol	Y	Y	Class symbol of the product to which RFQs are being subscribed
48	SecurityID	N		SecurityID of the product to which RFQs are being subscribed
22	IDSource	N		Ignored by CBOEdirect FIX 4.2 Service on messages received from firms.
167	SecurityType	N	Y	SecurityType of the product to which RFQs are being subscribed. Required for "by Name" look up of product.
200	MaturityMonthYear	N	Y	MaturityMonthYear of the product to which RFQs are being subscribed. Required for "by Name" look up of product.
205	MaturityDay	N	Y	MaturityDay of the product to which RFQs are being subscribed. Required for "by Name" look up of product.
201	PutOrCall	N	Y	PutOrCall of the product to which RFQs are being subscribed. Required for "by Name" look up of product.
202	StrikePrice	N	Y	StrikePrice of the product to which RFQs are being subscribed. Required for "by Name" look up of product.
336	TradingSessionID	N		W_MAIN, ONE_MAIN, CFE_MAIN, COF_MAIN, W_STOCK
54	Side	N		This should be left blank.
38	OrderQty	N		This should be left blank
60	TransactTime	N		Time of the subscription request
	Standard Trailer	Y		

RFQ Example

The following requests a quote for the Dow Jones DJV Sep 96.00 Call Option. The quote request id is “740”.

```
8=FIX.4.2^49=CCC^56=TESTFIX201^35=R^131=740^146=1^55=DJV^167=OPT^200=200109^201=1^202=96^3
36=W_MAIN^38=0^60="20000302-12:00:00"
```

CBOE Solicits Auction Participation

Firms that have subscribed for auction solicitation using *the Quote Status Request* message will receive solicitation from CBOE to participate in auctions. CBOE sends out a message for auction participation using *the Quote Request* message with additional fields. Users will respond to auction solicitations using the *New Order – Single* message. Firms should review document FIX-03c, which describes internalization and automated auction.

Table 24 CBOE solicits auction participation using the Quote Request message

Tag	Field Name	FIX Req'd	Comments
	Standard Header	Y	*MsgType[35]* = R
131	QuoteReqID	Y	Unique identifier for quote request
146	NoRelatedSym	Y	**Must always = 1**
55	Symbol	Y	Class symbol of the product to which RFQs are being subscribed
48	SecurityID	N	SecurityID of the product to which RFQs are being subscribed
22	IDSource	N	Ignored by CBOEdirect FIX 4.2 Service on messages received from firms.
167	SecurityType	N	SecurityType of the product to which RFQs are being subscribed
200	MaturityMonthYear	N	MaturityMonthYear of the product to which RFQs are being subscribed
205	MaturityDay	N	MaturityDay of the product to which RFQs are being subscribed
201	PutOrCall	N	PutOrCall of the product to which RFQs are being subscribed
202	StrikePrice	N	StrikePrice of the product to which RFQs are being subscribed
336	TradingSessionID	N	W_MAIN, ONE_MAIN, CFE_MAIN, W_STOCK, COF_MAIN
54	Side	N	The values to be used to solicit auction participation are: 1=Buy or 2=Sell.
38	OrderQty	N	This field must be the size of the order being auctioned.
60	TransactTime	N	Time of the subscription request
Extensions to the Quote Request Message Outbound from CBOE			

Tag	Field Name	FIX Req'd	Comments
9221	AuxAuctionInfo	N	This field may not always be present, and may contain multiple data values concerning the auction. Specifically, in the case of a COA auction, the NBBO Bid and NBBO Ask values at the time of the auction solicitation may be provided. For COA and other auction types; each of the executing firm, correspondent firm, and CMTA firm values of the order(s) being auctioned may be provided (these settings are configurable by firm). In order to present these new data values, AuxAucionInfo will be formatted as a comma-delimited series of name=value pairs, where the names will be: • NBBOBID • NBBOASK • FIRM • CORRESFIRM • CMTAFIRM The comma-delimited list will be surrounded by "{" "}" characters
9383	AuctionType	N	This field takes its value from cmiConstants::AuctionTypes; 1=internalization 2=strategy 3=regular_single 4= HAL 5=SAL 6=STOCK_NBBO_FLASH – This is not a new auction type. This value is used as an NBBO indicator in the W_STOCK session. It is being used as an AuctionType in order to share the auction event channel. If the user ties to respond with an auction, a reject message will be received 7=STOCK_ODD_LOT 8=AUCTION_DAIM
9302	OrderPrice	N	This is the starting price for the auction*
15	Currency	N	The currency of the OrderPrice. This message is optional.
9384	AuctionContingency	N	Same values as cmiConstants::ContingencyTypes 1=none, 2=AON, 3=FOK, 4=IOC,…
9743	PIPManagementType	N	1=solicitation to participate in an auction*
	Standard Trailer	Y	

* Different meaning than the UDF list published at fixprotocol.org

Example of auction solicitation from CBOE

```
8=FIX.4.2^49=HFIX2^56=CLNT^35=R^131=384027385:58734932^146=1^55=AOO^167=OPT^200=200107^201=1^
202=50^336=W_MAIN^54=1^38=10^60=20001102-15:12:03^9383=1^9302=.45^9384=1^9743=1^
```

Examples of auction soliciation from CBOE using UDF AuxAuctionInfo [9221]

```
8=FIX.4.2^A9=0277^A35=R^A34=292^A49=DFIX501^A56=TEST501^A52=20080630-
19:25:36^A131=65044:436400020^A146=1^A55=C^A48=499151539^A22=8^A167=MLEG^A201=0^A202=0^A207=W^A336=W_MA
IN^A38=100^A60=00260805-20:25:36^A9383=2^A9384=1^A9743=1^A9302=0.0000^A

9221={NBBOBID=0.2000,NBBOASK=0.2000,FIRM=325,CORRESFIRM=PAX,CMTAFIRM=671}^A10=166^A

8=FIX.4.2^A9=0249^A35=R^A34=209^A49=DFIX501^A56=TEST501^A52=20080630-
18:41:58^A131=65044:436400017^A146=1^A55=C^A48=499151539^A22=8^A167=MLEG^A201=0^A202=0^A207=W^A336=W_MA
IN^A38=100^A60=00260805-19:41:52^A9383=2^A9384=1^A9743=1^A9302=0.0000^A

9221={NBBOBID=0.2000,NBBOASK=0.2000,FIRM=325}^A10=006^A
```

```
8=FIX.4.2^A9=0264^A35=R^A34=401^A49=DFIX501^A56=TEST501^A52=20080630-
20:19:57^A131=65044:436400025^A146=1^A55=C^A48=499151539^A22=8^A167=MLEG^A201=0^A202=0^A207=W^A336=W_MA
IN^A38=100^A60=00260805-21:19:57^A9383=2^A9384=1^A9743=1^A9302=0.0000^A

9221={NBBOBID=0.2000,NBBOASK=0.2000,FIRM=325,CORRESFIRM=PAX}^A10=074^A

8=FIX.4.2^A9=0262^A35=R^A34=454^A49=DFIX501^A56=TEST501^A52=20080630-
20:46:27^A131=65044:436400028^A146=1^A55=C^A48=499151539^A22=8^A167=MLEG^A201=0^A202=0^A207=W^A336=W_MA
IN^A38=100^A60=00260805-21:46:27^A9383=2^A9384=1^A9743=1^A9302=0.0000^A

9221={NBBOBID=0.2000,NBBOASK=0.2000,FIRM=325,CMTAFIRM=671}^A10=089^A

8=FIX.4.2^A9=0240^A35=R^A34=11^A49=DFIX501^A56=TEST501^A52=20080701-
15:05:08^A131=65044:480871748^A146=1^A55=C^A48=477236851^A22=8^A167=OPT^A200=200801^A205=19^A201=1^A202=15^
A207=W^A336=W_MAIN^A54=1^A38=10^A60=00040806-16:05:06^A9383=1^A9384=1^A9743=1^A9302=1.1000^A

9221={FIRM=325}^A10=173^A


8=FIX.4.2^A9=0253^A35=R^A34=33^A49=DFIX501^A56=TEST501^A52=20080701-
15:15:42^A131=65044:480871759^A146=1^A55=C^A48=477236851^A22=8^A167=OPT^A200=200801^A205=19^A201=1^A202=15^
A207=W^A336=W_MAIN^A54=1^A38=10^A60=00040806-16:15:42^A9383=1^A9384=1^A9743=1^A9302=1.1000^A

9221={FIRM=325,CMTAFIRM=671}^A10=017^A
```

Mass Quote

CBOE requires Lead Market-Makers (LMMs), Designated Primary Market-Makers (DPMs) & Market-Makers to use Mass Quote message when quoting multiple options or futures products.

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

A QuoteSet specifies the first level of repeating tags for the Mass Quote message. It represents a group of related quotes and can, for example, represent an option or futures class.

Each QuoteSet contains an optional repeating group of QuoteEntries which can represent an option series or a futures month.

It is possible the number of Quote Entries for a Quote Set (option or futures class) could exceed one's physical or practical message size (see section on Quote Throttling above). It may be necessary to fragment a message across multiple quote messages. Message size limits must be mutually agreed to with one's counterparties.

The grouping of quotes is as follows:

NoQuoteSets – specifies the number of sets of quotes contained in the message

QuoteSetID – Is a unique ID given to the quote set

Information regarding the security to which all of the quotes belong

TotQuoteEntries – defines the number of quotes for the quote set across all messages

NoQuoteEntries – defines the number of quotes contained within this message for this quote set

QuoteEntryID – Is a unique ID given to a specific quote entry

Information regarding the specific quote (bid/ask size and price)

If there are too many Quote Entries for a Quote Set to fit into one physical message, then the quotes can be continued in another Mass Quote message by repeating all of the QuoteSet information and then specifying the number of Quote Entries (related symbols) in the continued message. The TotQuoteEntries is provided to optionally indicate to the counterparty the total number of Quote Entries for a Quote Set in multiple quote messages. This permits, but does not require, a receiving application to react in a stateful manner where it can determine if it has received all quotes for a Quote Set before carrying out some action. However, the overall approach to fragmentation is to permit each mass quote message to be processed in a stateless manner as it is received. Each mass quote message should contain enough information to have the Quote Entries applied to a market without requiring the next message if fragmentation has occurred. Also, a continued message should not require any information from the previous message.

Maximum message size for fragmentation purposes can be determined by using the optional MaxMessageSize field in the Logon message or by mutual agreement between counterparties.

Requesting Acknowledgement for Mass Quotes

CBOE does not support the *QuoteResponseLevel[301]* tag.

Quote Update Control ID (also called Quote Token)

A new field has been added to the mass quote message to indicate that a new quote submission should be overridden. This is to solve the case when a market maker's quote system is in the process of sending in new quotes at the same time that CBOEdirect is removing the old quotes. The messages may cross in flight and the market maker will have replaced his old quotes. One particular example is QRM. The QRM is intended to protect a market maker from being hit multiple times across a large number of series. In one example, the market maker would be streaming its quotes into the market without knowing that the QRM has triggered. The Quote Update Control ID field will provide the user with the ability to protect against new quotes being processed before the user acts upon the QRM notice.

1. The user must first enable the quote control ID feature at login by allowing for UDF's in repeating groups. This is done by setting tag 9003=3 (see Logon Message fix-03a for further details).
2. If the Quote Update Control ID field is set to zero (0), then the Quote Update Control function will be turned off. Using this new method will have the exact same behavior as the old mass quote method, i.e. CBOE would not provide any protection against the new quotes when such race condition occurs.
3. When a QRM event or quote cancellation occurs, either user or system initiated, CBOEdirect will look at the value in the control field. CBOE will not accept any new quotes from the user as long as the new Control ID value remains the same as it was before the cancel event.
4. Once the user changes the value on a new quote, that quote will be accepted and that value becomes the new test condition.

CBOE recommends that if the market-maker uses the Quote Update Control ID, then the market-maker should increment the Quote Update Control ID every time the market-maker receives a quote cancel acknowledgement from CBOE (intentional or unintentional cancellation). CBOE discourages the market-maker from incrementing the Quote Update Control ID every time a quote cancel request is sent *and* every time a quote cancel acknowledgement is received from CBOE. The reason for this is that the market-maker would not be protected from the race condition.

In addition to incrementing the Quote Update Control ID when receiving a regular quote cancel confirmation from CBOE, the market-maker must increment the Quote Update Control ID every time a quote entry to CBOE rejects or is canceled with the following error codes:

1. Quote rejected
   User will get Quote Acknowledgement with error code:
   QuoteRejectReason[300] = QUOTE_UPDATE_CONTROL = 95
2. Quote Cancel Report
   User will get a cancel report with reason:
   const cmiUtil::ActivityReason QUOTE_UPDATE_CONTROL =10;

The Quote Update Control ID is stored at CBOE in a short (16 bit signed integer, -32768 to 32767). Since the ID can be reused during a trading day by the same market-maker, the firm may rollover their numbers in any feasible range desired (1-99, 1-999, etc.).

The Quote Update Control ID must be submitted on a product-by-product basis (one ID per individual option or future, not one ID per class). It is likely that a market-maker will have different IDs for quotes for different options or futures products within the same class at the same time.

Blocked Cancel Requests

In order to allow users to cancel quotes more efficiently the CBOE has modified the accept quote for class processing to allow for blocked cancel requests. The feature is used much in the same way multiple series quotes for a class are submitted. Instead of sending a product key, price and quantity (for both bid and ask) the user sends in a product key and zero values for **both** the bid and ask quantity. The price for the bid and ask is ignored if the quantities are both zero. It is recommended that the price be set to zero as well but it is not required. This modification will allow the user to send in cancel requests for up to 40 series per method call. For example, using Mass Quote *MsgType[35]* = i for a zero bid and zero offer quantity, *BidSize[134] tag* is set to zero and *OfferSize[135] tag* is set to zero and up to 40 series in the mass quote block.

While you can mix quotes with cancels (for different products) in the same message, it is not recommended. If you exceed a quote rate limit the entire quote block message is rejected and no cancels will make it to the market. An application release at the end of August 2007 will remove the quote rate limit from this feature. There is also no reason to send a cancel and a quote for the same product.

Table 25 Mass Quote

Tag	Field Name	FIX Req'd	CBOE Req'd	Comments
	Standard Header	Y	Y	*MsgType[35]* = i (lowercase)
117	QuoteID	Y	Y	Users should provide a quote identifier in this field on each quote sent to CBOEdirect. QuoteIDs do not have to be unique. You have the choice to keep the same QuoteID across multiple submissions of quotes for a specific product.
301	QuoteResponseLevel	N	N	CBOE does not support this tag. CBOE will always provide a response. This field is ignored on input.
296	NoQuoteSets	Y	Y	The number of sets of quotes in the message
Begin Repeating Group for NoQuoteSets				
302	→QuoteSetID	Y	Y	Sequential number for the Quote Set. For a given QuoteID – assumed to start at 1. Must be used if NoQuoteSets is used.
311	→UnderlyingSymbol	Y	Y	FIX 4.2 Specification requires this tag, but the CBOE ignores the value that the firm sends. The firm should use the Symbol[55] tag to specify the symbol. All products in a Mass Quote Message must be in the same underlying.
309	→UnderlyingSecurityID	N	N	The product key of the symbol of the underlying product. Examples: IBM, MSFT, T, OEX, DJX
305	→UnderlyingIDSource	N	N	
310	→UnderlyingSecurityType	N	Y	`CS, INDX, USTB--` Required for “by Name” look up of product.
313	→UnderlyingMaturityMonthYear	N	Y	Required for “by Name” look up of product.
314	→UnderlyingMaturityDay	N	Y	Required for “by Name” look up of product.
315	→UnderlyingPutOrCall	N	Y	Required for “by Name” look up of product.
316	→UnderlyingStrikePrice	N	Y	Required for “by Name” look up of product.
308	→UnderlyingSecurityExchange	N	N	The exchange on which the underlying security trades.
307	→UnderlyingSecurityDesc	N	N	
304	→TotQuoteEntries	Y	Y	Total number of quotes for the quote set across all messages. Should be the sum of all NoQuoteEntries in each message that has repeating quotes that are part of the same quote set. Required if NoQuoteEntries > 0
295	→NoQuoteEntries	Y	Y	The number of quotes for this Symbol (QuoteSet) that follow in this message.
Begin Nested Repeating Group for NoQuoteEntries				
299	→→QuoteEntryID	Y	Y	
55	→→Symbol	N	Y	Product class symbol Examples: DLY, DLQ, OEX, OEZ, MSFT1C, IBMIC All products in a Mass Quote Message must be in the same underlying.

48	→→SecurityID	N	N	
22	→→IDSource	N	N	Ignored by CBOEdirect FIX 4.2 Service on messages received from firms.
167	→→SecurityType	N	Y	OPT, FUT, MLEG-- Required for “by Name” look up of product.
200	→→MaturityMonthYear	N	Y	MaturityMonthYear of the product. Required for “by Name” look up of product.
205	→→MaturityDay	N	Y	MaturityMonthYear of the product. Required for “by Name” look up of product.
201	→→PutOrCall	N	Y	MaturityMonthYear of the product. Required for “by Name” look up of product.
202	→→StrikePrice	N	Y	MaturityMonthYear of the product. Required for “by Name” look up of product.
106	→→Issuer	N	N	Company name of security issuer (e.g. International Business Machines)
132	→→BidPx	N	N	Bid Price of the quote (Same Price of quote for MLEG products)
133	→→OfferPx	N	N	Offer Price of the quote (Opposite Price of quote for MLEG products)
134	→→BidSize	N	N	Bid Quantity of the quote (Same Quantity of quote for MLEG products)
135	→→OfferSize	N	N	Offer Quantity of the quote (Opposite Quantity of quote for MLEG products)
60	→→TransactTime	N	N	Time that the quote is sent to CBOEdirect
336	→→TradingSessionID	N	N	W_MAIN, ONE_MAIN, CFE_MAIN, W_STOCK, COF_MAIN
9006	→→QuoteUpdateControlId	N	Y	This tag is strongly encouraged. If the firm sets this tag to zero (0), then this feature will be turned off. Firms should send any value between 1 – 32767 (-32767 to -1 is also allowable) and should increment this value any time the firm receives a quote cancel report, or receives a quote reject with error code tag QuoteRejectReason[300] set to QUOTE_UPDATE_CONTROL = 95. The QuoteUpdateControlId value can be reused during the day. A Mass Quote with invalid quote update control (which cannot be mapped onto the system internally as short such as values above 32767 and below –32767) would be rejected with reject code INVALID_QUOTE_UPDATE_CONTROL_ID (tag 300 = 93). Even if one quote has this invalid quote update token then the whole quote block will be rejected.
20101	→→ShortSaleIndicator	N	Y	This tag is used for the CBSX session to indicate short sale positions. Valid values are: 1 = Buy 2 = Sell 5 = Sell_Short 6 = Sell_Short_Exempt
End Nested Repeating Group for NoQuoteEntries				

End Repeating Group for NoQuoteSets				
	Standard Trailer	Y	Y	

Mass Quote Example

Note that CBOEdirect Hybrid treats the Mass Quote Message differently from futures

CBOEdirect Hybrid does NOT generate a separate individual Quote Message for Mass Quotes.

Mass Quote Request (CBOEdirect Hybrid)

This Mass Quote message contains 5 simple quotes.

```
8=FIX.4.2^ 9=0565^ 35=i^ 34=2^ 49=AATG1^ 56=HFIX201^ 52=20030602-21:04:17^
117=QT_12345_OK^ 296=1^ 302=1^ 311=SUQ^ 304=5^ 295=5^ 299=ONE^ 55=SUQ^
167=OPT^ 200=200306^ 201=0^ 202=2.50^ 132=1.00^ 133=1.10^ 134=10^ 135=10^
336=W_MAIN^ 299=TWO^ 55=SUQ^ 167=OPT^ 200=200306^ 201=0^ 202=5.00^ 132=2.00^
133=2.10^ 134=20^ 135=20^ 336=W_MAIN^ 299=THREE^ 55=SUQ^ 167=OPT^ 200=200306^
201=0^ 202=7.50^ 132=3.00^ 133=3.10^ 134=30^ 135=30^ 336=W_MAIN^ 299=FOUR^
55=SUQ^ 167=OPT^ 200=200306^ 201=0^ 202=10.00^ 132=4.00^ 133=4.10^ 134=40^
135=50^ 336=W_MAIN^ 299=FIVE^ 55=SUQ^ 167=OPT^ 200=200306^ 201=0^ 202=12.50^
132=5.00^ 133=5.10^ 134=50^ 135=50^ 336=W_MAIN^ 10=184^
```

Mass Quote Response (CBOEdirect Hybrid)

A Quote Acknowledgement Message for the Mass Quote as a Whole. Note that it contains an entry for each of the original Mass Quote Message's 5 simple quotes.

```
8=FIX.4.2^ 9=0394^ 35=b^ 34=2^ 49=HFIX201^ 56=AATG1^ 52=20030602-21:04:16^
117=QT_12345_OK^ 297=0^ 300=0^ 336=W_MAIN^ 296=1^ 302=1^ 311=SUQ^ 304=5^
295=5^ 299=ONE^ 55=SUQ^ 167=OPT^ 200=200306^ 201=0^ 202=2.5^ 368=0^ 299=TWO^
55=SUQ^ 167=OPT^ 200=200306^ 201=0^ 202=5^ 368=0^ 299=THREE^ 55=SUQ^ 167=OPT^
200=200306^ 201=0^ 202=7.5^ 368=0^ 299=FOUR^ 55=SUQ^ 167=OPT^ 200=200306^
201=0^ 202=10^ 368=0^ 299=FIVE^ 55=SUQ^ 167=OPT^ 200=200306^ 201=0^ 202=12.5^
368=0^ 10=112^
```

Mass Quote Request with Quote Update Control:

```
8=FIX.4.2^A9=0324^A35=i^A57=TEST^A34=471^A49=AAS^A56=AFIX201^A52=20040614-
20:15:00^A117=BBM0011-
20040614^A296=1^A302=1^A311=MOT^A304=2^A295=2^A299=1^A55=MSQ^A22=8^A167=OPT^A2
00=200404^A201=0^A202=35.0^A132=1.3^A133=1.5^A134=200.0^A135=200.0^A336=W_MAIN
^A9006=2^A299=2^A55=MSQ^A22=8^A167=OPT^A200=200404^A201=0^A202=35.0^A132=3.2^A
133=3.5^A134=150.0^A135=150.0^A336=W_MAIN^A9006=3^A10=015^A
```

Mass Quote Ack with Quote Update Control Id:

```
8=FIX.4.2^A9=0253^A35=b^A34=503^A49=AFIX201^A56=AAS^A52=20040614-2

0:15:00^A117=BBM0011-
20040614^A297=0^A300=0^A336=W_MAIN^A296=1^A302=1^A311=MOT^A304=2^A295=2^A299=1
^A55=MSQ^A22=8^A167=OPT^A200=200404^A201=0^A202=35^A368=0^A9006=2^A299=2^A55=M
SQ^A22=8^A167=OPT^A200=200404^A201=0^A202=35^A368=0^A9006=3^A10=022^A
```

Hybrid Quote Trigger

Hybrid quote trigger takes place when a Hybrid market-maker's quote in a Hybrid class is in the process of being traded. This process of "being traded" can be n(1) second or more (check with the CBOE Production Help Desk or Trading Operations department for the current value which can vary by class). While a Hybrid quote is being traded, CBOE places restrictions on cancellations and modifications to the quote. The following scenarios only apply to options that trade on the CBOEdirect Hybrid system.

Scenario 1:

The market-maker requests to cancel the individual quote, cancel quotes by class, or cancel all quotes during a quote trigger.

In this case, the cancel request will immediately successfully complete, CBOE will immediately deliver a cancel report and then once the trigger completes, the fill report will be delivered. All cancels confirmations will be delivered with a cancel quote reason stated in the QuoteText tag (9008) as "USER".

Scenario 2:

The market-maker submits a Mass Quote or Quote message during a quote trigger.

When this happens, the quote message is rejected, but upon the trigger completing, the server will cancel the market-maker's quote with a cancel quote reason stated in the QuoteText tag (9008) as "SYSTEM". Quotes rejected as a result of the quote being in trigger will have the tag 300 reject code = QUOTES_IN_PROGRESS=97. This prevents a market-maker from modifying his or her quote during the trigger. This also prevents the market-maker from having a stale quote left in the marketplace after a quote trigger has completed.

Quote Trigger Notifications

CBOE will automatically subscribe the market-maker to Quote Trigger notifications. CBOE will publish Quote Trigger notifications via the Quote Message with tag 9008 having all the information of the Quote Trigger.

Quote Trigger in response to a Quote:

```
8=FIX.4.2^A9=0208^A35=S^A34=498^A49=AFIX201^A56=AAS^A52=20040614-
20:12:45^A117=BBM0010-
20040614^A336=W_MAIN^A55=MSQ^A48=99359119^A22=8^A167=OPT^A200=200404^A205=17^A
201=0^A202=35^A207=W^A132=1.3^A133=1.5^A134=200^A135=200^A9312=Sell Quote
Trigger^A10=001^A
```

Quote Cancel

The Quote Cancel message is used by an originator of quotes to cancel quotes that the user has made during the current session.

The Quote Cancel message supports cancellation of:

All quotes for a particular user

Quotes for a specific symbol or (Class) for a particular user

Quotes for a specific product (Series) for a particular user

Canceling a Quote is accomplished by indicating the type of cancellation in the *QuoteCancelType[298]* field.

Table 26 Supported quote cancel types

QuoteCancelType[298] Enumeration	Usage
1	Cancel for Symbol (cancel specific product → REQUIRED for LMMs and market-makers)
3	Cancel for Underlying Symbol (cancel class → REQUIRED for LMMs and market-makers)
4	Cancel All Quotes

NOTE: CBOE does not permit cancellation by Security Type QuoteCancelType[298]=2.

The Quote Cancellation only applies to quotes made by the current FIX user.

Table 27 Quote Cancel

Tag	Field Name	FIX Req'd	CBOE Req'd	Comments
	Standard Header	Y	Y	*MsgType[35]* = Z
131	QuoteReqID	N	N	Required when quote is in response to a Quote Request message
117	QuoteID	Y	Y	Corresponds to the QuoteID(s) of the mass quote or quote that the user wishes to cancel.
298	QuoteCancelType	Y	Y	Identifies the type of Quote Cancel request. 1 = Cancel specific product (cancel for Symbol → REQUIRED) 3 = Cancel all quotes in a class (cancel for Underlying Symbol → REQUIRED) 4 = Cancel all quotes for the user
301	QuoteResponseLevel	N	N	CBOE does not support this field in the Quote Cancel message. Level of Response requested from receiver of quote messages. A Business Reject (see FIX-03b for Business Reject message details) will be sent in response to invalid Quote Cancel requests. A successful Quote Cancel request will receive Quote Status message(s) indicating the status of the quote after processing.
336	TradingSessionID	N	N	Trading Session of the product in which the user wishes to cancel quotes W_MAIN, ONE_MAIN, CFE_MAIN, COF_MAIN, C2_MAIN or W_STOCK
295	NoQuoteEntries	Y	Y	The number of securities whose quotes are to be canceled
Begin Repeating Group for NoQuoteEntries				
55	→Symbol	Y	Y	Must be the first field in the repeating group. Class symbol of the product class for the quotes the user is canceling. If canceling all quotes enter "NA" for the symbol
48	→SecurityID	N	N	Class key of the product class for the quotes the user is canceling.
22	→IDSource	N	N	Ignored by CBOEdirect FIX 4.2 Service on messages received from firms.
167	→SecurityType	N	Y	Security Type of the products that the user is canceling quotes in. Required for "by name" look up of product.
200	→MaturityMonthYear	N	Y	MaturityMonthYear of the products that the user is canceling quotes in. Required for "by name" look up of product.
205	→MaturityDay	N	Y	MaturityDay of the products that the user is canceling quotes in. Required for "by name" look up of product.
201	→PutOrCall	N	Y	PutOrCall of the products that the user is canceling quotes in. Required for "by name" look up of product.
202	→StrikePrice	N	Y	StrikePrice of the products that the user is canceling quotes in. Required for "by name" look up of product.
311	→UnderlyingSymbol	N	N	The symbol of the underlying security of options that should be canceled.
End Repeating Group				
	Standard Trailer	Y	Y	

Figure 3 Successful Quote Cancel sequence diagram

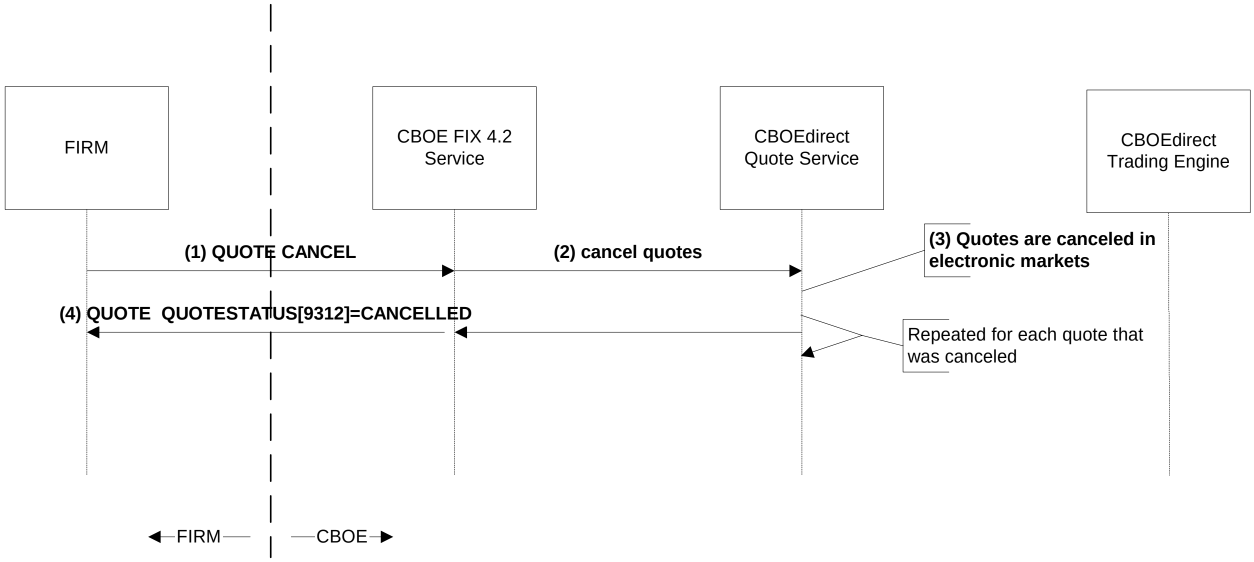
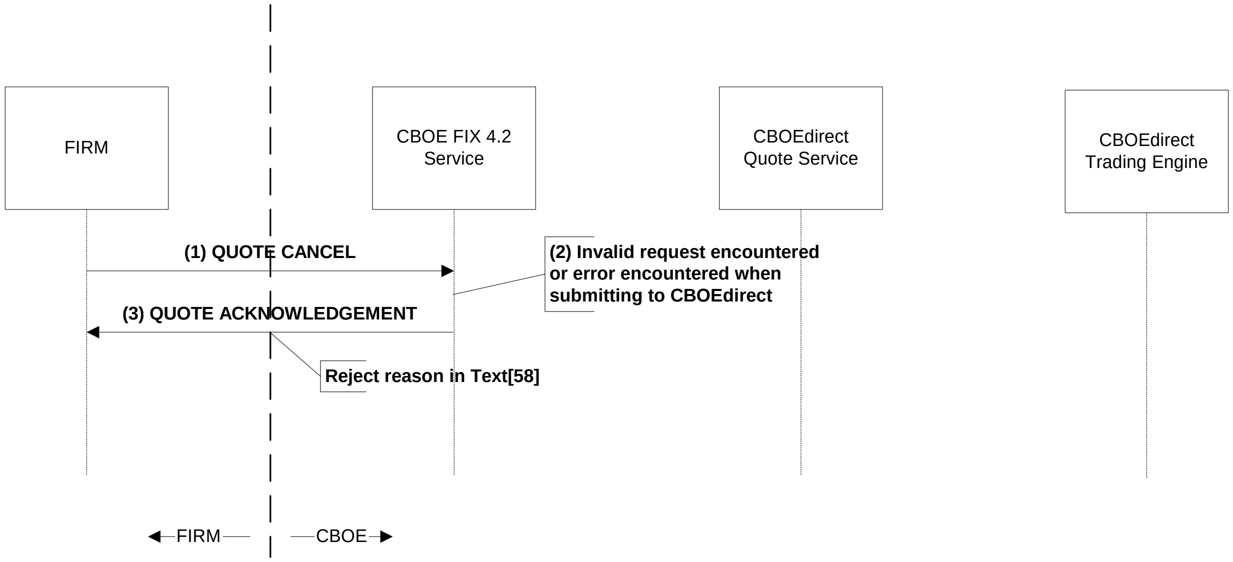

FIRM
CBOE FIX 4.2 Service
CBOEdirect Quote Service
CBOEdirect Trading Engine
(1) QUOTE CANCEL
(2) cancel quotes
(3) Quotes are canceled in electronic markets
(4) QUOTE QUOTESTATUS[9312]=CANCELLED
Repeated for each quote that was canceled
FIRM
CBOE

Figure 4 Unsuccessful Quote Cancel sequence diagram


FIRM
CBOE FIX 4.2 Service
CBOEdirect Quote Service
CBOEdirect Trading Engine
(1) QUOTE CANCEL
(2) Invalid request encountered or error encountered when submitting to CBOEdirect
(3) QUOTE ACKNOWLEDGEMENT
Reject reason in Text[58]
FIRM
CBOE

Important Considerations for concurrent thread model users

For Firms that use the concurrent thread model feature for quote processing, there is a possibility of the user’s cancel request getting rejected when a quote is in flight. Firms should make changes to their software so that they resubmit the cancel request if they think they still have an open quote in the market which is not completely filled or cancelled. Refer to the Quote Processing section in this document for details.

Quote Cancel Examples

Cancel quote for an individual series

REQUEST

```
8=FIX.4.2^49=CCC^56=TESTFIX201^35=Z^117=AAA0017-
20010504^298=1^336=W_MAIN^295=1^55=DJV^167=OPT^200=200109^201=0^202=100^
```

RESPONSE

```
8=FIX.4.2^9=0200^35=S^49=TESTFIX201^56=ABC^34=2182^52=20010504-17:00:56^117=AAA0016-
20010504^336=W_MAIN^55=DJV^48=2818387^22=8^167=OPT^200=200109^205=22^202=100.0^207=W^132=3
.0^133=3.3^134=50.0^135=50.0^9312=Cancelled^
```

Cancel Quotes for an Underlying (Class)

REQUEST

```
8=FIX.4.2^49=CCC^56=TESTFIX201^35=Z^117=AAA0086-
20010504^298=3^336=W_MAIN^295=1^55=DJX^167=OPT^
```

RESPONSE

```
8=FIX.4.2^9=0210^35=S^49=TESTFIX201^56=ABC^34=4480^52=20010504-17:24:33^117=AAA0077-
20010504^336=W_MAIN^55=DJV^48=2818404^22=8^167=OPT^200=200112^205=22^201=1^202=100.0^207=W
^132=12.0^133=12.3^134=101.0^135=101.0^9312=Cancelled^
```

Cancel All Quotes

REQUEST

```
8=FIX.4.2^49=CCC^56=TESTFIX201^35=Z^117=MY_CANCEL_ALL^298=4^336=W_MAIN^295=1^55=NA^200=200
109^201=1^202=92^
```

RESPONSE

```
8=FIX.4.2^9=0200^35=S^49=TESTFIX201^56=CCC^34=2247^52=20010504-
17:12:59^117=MY_CANCEL_ALL^336=W_MAIN^55=NA^9312=Cancelled^
```

Using the Quote Status Request message to query quotes

The *Quote Status Request* message can be used to query the current quotes for a series of products. This is not normally required after you subscribe for quote status and quote requests using the Quote Status Request. A Quote Status Request will only returns quotes for the current user.

A *Quote* message is returned for each valid quote matching the security information provided in the *Quote Status Request* message.

A *Quote Acknowledgement* message is returned for each invalid quote encountered.

Table 28 Quote Status Request

Tag	Field Name	FIX Req'd	CBOE Req'd	Comments
	Standard Header	Y	Y	*MsgType[35]* = a (lowercase)
37	QuoteID	N	N	NOT USED
55	Symbol	Y	Y	Class Symbol of the product whose quote the user is querying
48	SecurityID	N	N	CBOEdirect product key of the product whose quote the user is querying
22	IDSource	N	N	Ignored by CBOEdirect on messages received from firms.
167	SecurityType	N	Y	Security Type of the product whose quote the user is querying
200	MaturityMonthYear	N	Y	MaturityMonthYear of the product whose quote the user is querying. “Required for “by Name” look up of product.”
205	MaturityDay	N	Y	MaturityDay of the product whose quote the user is querying. “Required for “by Name” look up of product.”
201	PutOrCall	N	Y	PutOrCall of the product whose quote the user is querying. Required for “by Name” look up of product.
202	StrikePrice	N	Y	StrikePrice of the product whose quote the user is querying. Required for “by Name” look up of product.
263	SubscriptionRequestType	N	N	
	Standard Trailer	Y	Y	

Figure 5 Valid Quote Status Request sequence diagram

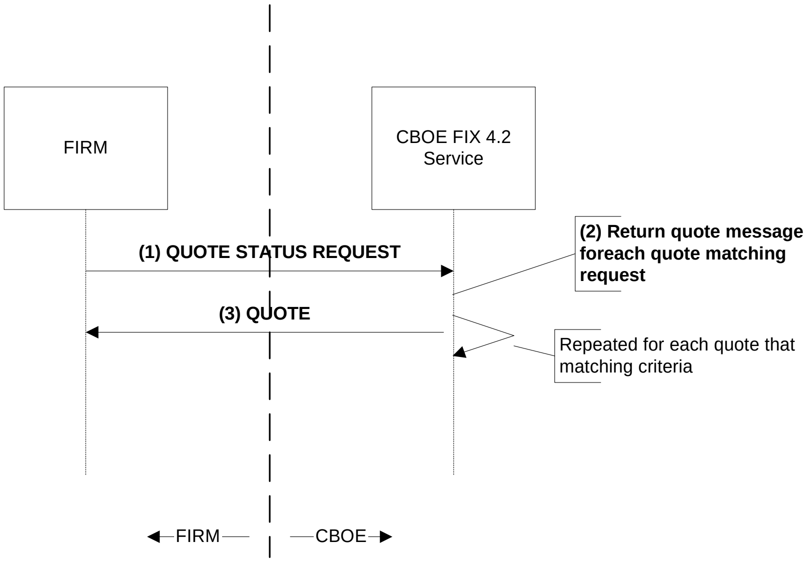
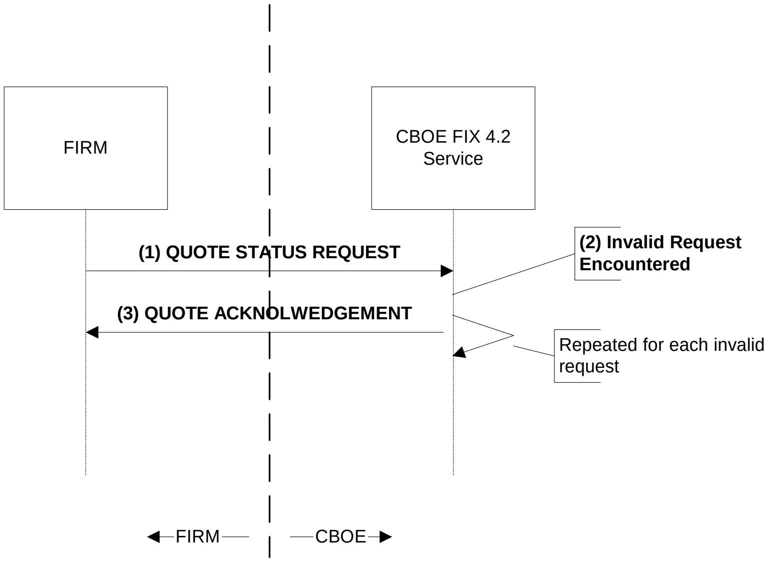

FIRM
CBOE FIX 4.2 Service
(1) QUOTE STATUS REQUEST
(2) Return quote message foreach quote matching request
(3) QUOTE
Repeated for each quote that matching criteria
FIRM
CBOE

Figure 6 Invalid Quote Status Request sequence diagram


FIRM
CBOE FIX 4.2 Service
(1) QUOTE STATUS REQUEST
(2) Invalid Request Encountered
(3) QUOTE ACKNOLWEDGEMENT
Repeated for each invalid request
FIRM
CBOE

Reporting Quote Fills

This Execution Report is used to report quote fills. If backlogs in receiving Quote Status and / or Execution Reports occur, the CBOE will remove the user's subscription to Quote Status and notify the user by sending a Business Reject Message (see FIX-03b for Business Reject message details) with reason text. The user would have to re-register for Quote Status.

Table 29 Execution Report for quote fills

Tag	Field Name	Comments
	Standard Header	*MsgType[35]* = 8
1	Account	The account on the trade is not published by CBOEdirect at this time.
6	AvgPx	Always 0.0
11	ClOrdID	QuoteID provided when the quote was submitted
14	CumQty	Always 0.0
17	ExecID	Each execution report in Hybrid and OneChicago will contain a unique ExecID to identify the message. The ExecID will contain the CBOE Quote ID (HighId and LowId), CBOE Trade ID (HighId and LowId), and a TransactionSequenceNumber. Ignore the TransactionSequenceNumber as it is used only for orders, not quotes. **ExecID Format: QuoteIdHigh:QuoteIdLow.TradeIdHigh:TradeIdLow.TransactionSequenceNumber** **(Example: 123:12345.987:98765.0)** The Trade ID (High and Low) must be given to the CBOEdirect Administrator at the time the user requests a trade bust (currently supported in OneChicago only). **ExecID Format for Market Maker Trade Notification:** Represented by 0:0 (e.g. `17=0:0.65432:2104808741.0`) For Purposes of the DelFloorTrade Message, either the ExecID, as reported in the MMTN Message, or the TradeID portion of it (e.g `17=65432:2104808741`) may be used. The TradeID-only format will be returned by the DelFloorTrade Message outbound on success.
20	ExecTransType	Always "0" (New).
31	LastPx	Fill price.
32	LastShares	Quantity of the trade.
37	OrderID	Will be set to the CBOEdirect internal Quote key
38	OrderQty	Always 0.0
39	OrdStatus	This field will be blank
40	OrdType	This field will be blank
44	Price	This value will be the price of the fill report.
48	SecurityID	Product key of the product that is being traded

54	Side	Corresponds to the side of the trade or bust. 1 = Buy 2 = Sell
55	Symbol	Class Symbol of the product that was traded or busted
58	Text	This field will be blank
60	TransactTime	UTC time the execution report was created.
76	ExecBroker	Executing or Give up firm
77	OpenClose	This field will be blank
150	ExecType	
151	LeavesQty	Populated with the quantity remaining on the quote
167	SecurityType	Security Type of the product that is being traded
200	MaturityMonthYear	Expiration year and month of the product that is being traded
205	MaturityDay	Expiration day of the month of the product that is being traded
201	PutOrCall	Put or Call value of the product that is being traded
202	StrikePrice	StrikePrice of the product that is being traded
207	SecurityExchange	W = CBOE
336	TradingSessionID	TradingSessionID where execution occurred.
439	ClearingFirm	This field will be blank as CMTA values are not allowed on quotes
440	ClearingAccount	This field is not published at this time.
382	NoContraBrokers	This field will be blank as CBOEdirect does not publish contra-party information at this time.
Begin Repeating group for NoContraBrokers		
375	→ContraBroker	This field will be blank as CBOEdirect does not publish contra-party information at this time.
337	→ContraTrader	This field will be blank as CBOEdirect does not publish contra-party information at this time.
437	→ContraTradeQty	This field will be blank as CBOEdirect does not publish contra-party information at this time.
438	→ContraTradeTime	This field will be blank as CBOEdirect does not publish contra-party information at this time.
End Repeating group for NoContraBrokers		
The following custom field(s) are used by CBOE (note: they are not part of the repeating group)		

6699	*ApplicationQueueDepth*	*A User defined field that provides the number of application level events that are queued for processing behind this current message. For instance, when ApplicationQueueDepth > 0 on a corresponding application response message sent from CBOE to the firm, this indicates that there are still ApplicationQueueDepth # of reports that have yet to be generated and transmitted to the user. This information is provided to help counter parties manage throughput and backlog issues. This tag is available for Order Status, Quote Status, Order Execution Reports, Quote Execution Reports, and all related Market Data messages.*
	Standard Trailer	

Reporting Quote Busts

CBOEdirect supports the ability for the CBOE help desk to bust a trade under certain circumstances. The initiation of a trade bust is done out of band via contact to the CBOE help desk over telephone or email / text messaging. The ability to request the bust of a trade is not supported by CBOE application programming interfaces (CMi or FIX 4.2). CBOE does not support busts through the FIX 4.2 API for the W_MAIN session.

Because trade busts are initiated out of band, trade bust reports are sent to the counterparties without notice through the API. Within FIX, trade busts are reported using an unsolicited Execution Report. An Execution Report indicating an occurrence of trade bust is identified by the field: ExecTransType[20]=1 (Cancel). Only trade busts are reported using the ExecTransType=CANCEL.

Each Execution Report that reports a fill (or partial fill) is uniquely identified using an Execution ID in the ExecID[17] field. In the Execution Report of a trade bust, FIX provides a field, ExecRefID[19], which is used to refer to the Execution ID of the previous fill Execution Report that is being busted (canceled / deleted). This provides you with a programmatic or manual way to identify the fill report that was busted.

Other relevant fields on a Trade Bust Execution Report

ExecType[150] = CANCEL because a trade bust not only reduces the CumQty (total filled quantity), it cancels part of the original order quantity (because the cancelled fill quantity is not added back to the leaves quantity).

CBOE has added an additional custom field to help make the action of the trade bust more explicit. The **LastBustShares[9368]** custom field is used to report the fill quantity to be busted.

The **Text[58]** field is also set to the string "Trade Bust Report" as a visual reference. Note: The human readable explanation is being provided to help identify the trade report. The Text[58] field should not be accessed or tested programmatically. CBOE does not consider these text descriptions of errors as part of the API. These text values are subject to change without notice.

Developers are recommended to use the ExecTransType[20]=1 (Cancel) to identify the occurrence of a trade bust.

A firm must be able to handle Trade Bust messages to become a certified CBOEdirect user.

Trade bust requests transmitted to the floor may require a response from floor personnel under certain conditions. This means that there could be a delay in the receipt of an Execution Report on the bust.

In addition to the ***Common Execution Report fields*** defined above, the following applies for this type of Execution Report

Table 30 Execution Report—Bust Report from CBOEdirect

Tag	Field Name	Comments
19	ExecRefID	Used to refer to the Execution ID of the previous fill Execution Report that is being busted (canceled, or deleted)
20	ExecTransType	1 (Cancel) to identify the occurrence of a trade bust
58	Text	"Trade Bust Report"

Tag	Field Name	Comments
150	ExecType	Cancel
151	LeavesQty	Leaves Quantity (0.0)
9368	LastBustShares	CBOE custom field that is used to report the quantity to be busted. Bust quantity will be greater than or equal to 1 and less than or equal to the trade quantity.
11	ClOrdID	NONE
	Standard Trailer	

Quote Acknowledgement

The Quote Acknowledgement message is used to respond to the *Quote*, *Mass Quote*, *Quote Status Requests*, and *Quote Cancel* messages.

Table 31 Quote Acknowledgement

Tag	Field Name	Comments
	Standard Header	MsgType = b (lowercase)
131	QuoteReqID	Required when acknowledgment is in response to a Quote Request message
117	QuoteID	Required when acknowledgment is in response to a Quote message
297	QuoteAckStatus	Status of the quote acknowledgement.

300	QuoteRejectReason	If this is set to "0", then the quote was accepted properly by CBOE. Any other value indicates an error. Reason Quote was rejected:
		(Quote was accepted) = 0
		UNKNOWN_SYMBOL = 1;
		EXCHANGE_OR_SECURITY_CLOSED = 2;
		ORDER_EXCEEDS_LIMIT = 3;
		TOO_LATE_TO_ENTER = 4;
		UNKNOWN_ORDER = 5;
		DUPLICATE_ORDER = 6;
		INVALID_BID_ASK_SPREAD = 7;
		INVALID_PRICE = 8;
		NOT_AUTHORIZED_TO_QUOTE_SECURITY = 9;
		INSUFFICIENT_QUANTITY = 11;
		INCOMPLETE_QUOTE = 12;
		TWO_SIDED_QUOTE_REQUIRED = 13;
		INVALID_SIDE = 14;
		INVALID_TRADING_SESSION = 15;
		EXCHANGE_OR_BROKER_OPTION = 16;
		QUOTE_CANCEL_IN_PROGRESS = 88;
		EXCEEDS_CONCURRENT_QUOTE_LIMIT = 89;
		QUOTE_RATE_EXCEEDED= 91;
		SEQUENCE_LIMIT_EXCEEDED= 92;
		INVALID_QUOTE_UPDATE_CONTROL_ID= 93;
		QUOTE_TRIGGER= 94; - Not currently used
		QUOTE_UPDATE_CONTROL = 95;
		SERVER_NOT_AVAILABLE = 96;
		QUOTE_BEING_PROCESSED = 97;
		CALL_LIMIT_EXCEEDED = 98;
		UNSPECIFIED_REASON = 99;

301	QuoteResponseLevel	CBOE does not support this tag.
336	TradingSessionID	Trading session of the product that the quote is in
58	Text	May be blank or may contain an error condition
296	NoQuoteSets	The number of sets of quotes in the message

Begin Repeating Group for NoQuoteSets		
302	→QuoteSetID	First field in repeating group. Required if NoQuoteSets > 0
311	→UnderlyingSymbol	Required if NoQuoteSets > 0
312	→UnderlyingSymbolSfx	
309	→UnderlyingSecurityID	
305	→UnderlyingIDSource	
310	→UnderlyingSecurityType	CS, INDX, USTB
313	→UnderlyingMaturityMonthYear	
314	→UnderlyingMaturityDay	
315	→UnderlyingPutOrCall	
316	→UnderlyingStrikePrice	
308	→UnderlyingSecurityExchange	
304	→TotQuoteEntries	Total number of quotes for the quote set across all messages. Should be the sum of all NoQuoteEntries in each message that has repeating quotes that are part of the same quote set. Required if NoQuoteEntries > 0
295	→NoQuoteEntries	The number of quotes for this Symbol (QuoteSet) that follow in this message.
Begin Nested Repeating Group for NoQuoteEntries		
299	***→→→→QuoteEntryID***	Uniquely identifies the quote as part of a QuoteSet. First field in repeating group. Required if NoQuoteEntries > 0.
55	***→→→→Symbol***	Product Class Symbol of the product for the quote
48	***→→→→SecurityID***	SecurityID of the product for the quote
22	***→→→→IDSource***	8
167	***→→→→SecurityType***	Security Type of the product for the quote
200	***→→→→MaturityMonthYear***	MaturityMonthYear of the product for the quote
205	***→→→→MaturityDay***	MaturityDay of the product for the quote
201	***→→→→PutOrCall***	PutOrCall of the product for the quote
202	***→→→→StrikePrice***	Strike Price of the product for the quote
207	***→→→→SecurityExchange***	W = CBOE

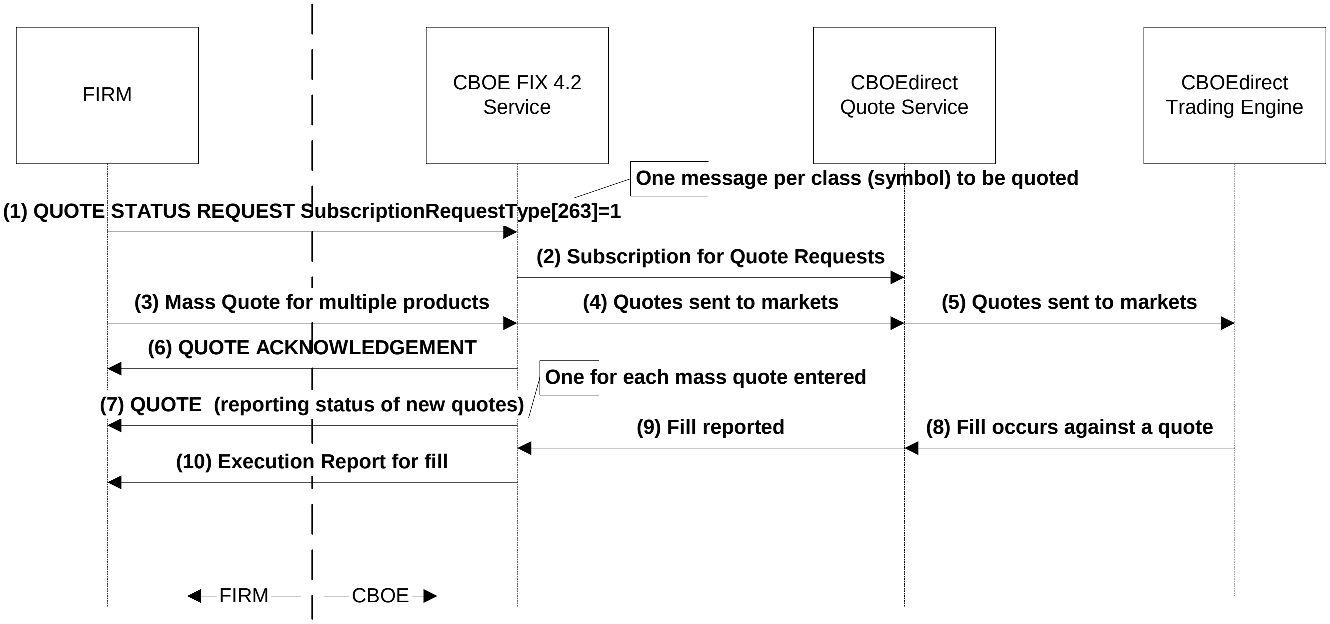
368	***→→→→QuoteEntryRejectReason***	If this is set to "0", then the quote was accepted properly by CBOE. Any other value indicates an error. Reason Quote was rejected:
		UNKNOWN_SYMBOL = 1;
		EXCHANGE_OR_SECURITY_CLOSED = 2;
		ORDER_EXCEEDS_LIMIT = 3;
		TOO_LATE_TO_ENTER = 4;
		UNKNOWN_ORDER = 5;
		DUPLICATE_ORDER = 6;
		INVALID_BID_ASK_SPREAD = 7;
		INVALID_PRICE = 8;
		NOT_AUTHORIZED_TO_QUOTE_SECURITY = 9;
		INSUFFICIENT_QUANTITY = 11;
		INCOMPLETE_QUOTE = 12;
		TWO_SIDED_QUOTE_REQUIRED = 13;
		INVALID_SIDE = 14;
		INVALID_TRADING_SESSION = 15;
		EXCHANGE_OR_BROKER_OPTION = 16;
		SAL_IN_PROGRESS = 89;
		QUOTE_RATE_EXCEEDED= 91;
		SEQUENCE_LIMIT_EXCEEDED= 92;
		INVALID_QUOTE_UPDATE_CONTROL_ID= 93;
		QUOTE_TRIGGER= 94; - Not currently used
		QUOTE_UPDATE_CONTROL = 95;
		SERVER_NOT_AVAILABLE = 96;
		QUOTE_BEING_PROCESSED = 97;
		CALL_LIMIT_EXCEEDED = 98;
		UNSPECIFIED_REASON = 99;

End Nested Repeating Group NoQuoteEntries		
End Repeating Group NoQuoteSets		
The following custom field(s) are used by CBOE (note: they are not part of the repeating group)		
6699	*ApplicationQueueDepth*	*A User defined field that provides the number of application level events that are queued for processing behind this current message. For instance, when ApplicationQueueDepth > 0 on a corresponding application response message sent from CBOE to the firm, this indicates that there are still ApplicationQueueDepth # of reports that have yet to be generated and transmitted to the user. This information is provided to help counter parties manage throughput and backlog issues. This tag is available for Order Status, Quote Status, Order Execution Reports, Quote Execution Reports, and all related Market Data messages.*
	Standard Trailer	

Typical Quote Usage

The following sequence diagram shows a typical quoting application using FIX.

Figure 7 Typical quoting usage sequence diagram


FIRM
CBOE FIX 4.2 Service
CBOEdirect Quote Service
CBOEdirect Trading Engine
One message per class (symbol) to be quoted
(1) QUOTE STATUS REQUEST SubscriptionRequestType[263]=1
(2) Subscription for Quote Requests
(3) Mass Quote for multiple products
(4) Quotes sent to markets
(5) Quotes sent to markets
(6) QUOTE ACKNOWLEDGEMENT
One for each mass quote entered
(7) QUOTE (reporting status of new quotes)
(9) Fill reported
(8) Fill occurs against a quote
(10) Execution Report for fill
FIRM
CBOE

Quote Risk Monitoring

Quote Risk Monitoring ("QRM") is a proprietary feature designed by the CBOE in order to permit market makers to automatically cancel quotes based upon market action. CBOE highly recommends use of QRM for LMMs and market-makers, but it is not required.

QRM is implemented in the CBOE FIX 4.2 Service by embedding XML messages within Quote Status Request and Quote Acknowledgement messages. XML was chosen as CBOE for Quote Risk Monitoring for these reasons:

- The FIX 4.2 specification did not support quote risk monitoring.
- Quote risk monitoring is a proprietary feature of the CBOEdirect Trading System and, as such, will not be offered as an industry standard – as it is viewed as a market place differentiator.
- XML is an emerging standard across all information technology and will increasingly be used as an interface. Since FIX 4.2 did not support QRM, it was decided to implement the messages using another widely available standard.

The FIX 4.2 standard permits enveloping XML within the header of any FIX 4.2 message – using the following tags.

The XmlDataLen[212] and XmlData[213] tags are optional data tags available in the FIX standard header message. These tags are used to contain QRM XML information.

The QRM XML Schema is similar in concept to SOAP (Simple Object Access Protocol) – a World Wide Web (W3C) standard for describing objects in XML.

This section is taken from the CBOEdirect functional specifications for quote risk monitoring.

QRM Functionality

This function is intended to provide benefits to both the customers and the market-makers. For the customer, the markets would tend to be deeper and more liquid --- bigger size quotes and more market-makers providing quotes --- because market-makers would have better control of their risk and therefore, more willing to quote. For the market-maker, it would allow them to control their risk after they have traded and actually taken risk.

There is common agreement that market-makers who provide quotes in SBT would be exposed to greater risks than in open outcry. For example, in SBT a single market-maker's bids in several series, if his bids were the best, could be hit by a coordinated set of incoming orders within a few seconds, saddling him with a large position before he can react and change his quotes. In open outcry, he has less risk because he can change his market after a predetermined number of transactions have occurred.

To encourage market-makers to provide deep and liquid markets, the system has to provide a way for market-makers to control their risk after they have already taken trades and assumed risk. What is suggested is a system feature that would pull a market-maker's quotes from a class when the system determines that his resting quotes have been filled in n (60) seconds or less to a net total of n (200) contracts long or short. In other words, a number of his quotes got filled within a short period of time and the system would pull his other quotes in the same class and notify him to give him a chance to react. Note that the system would cancel his remaining quotes based on this function, not because of cancel quotes sent by the market-maker himself. It is then up to him to assess the risk he has taken so far in that class. He may provide replenishment quotes according to his quote obligations.

This function would only consider trades with the market-maker's resting quotes, not trades that the market-maker initiates by entering new orders to hit a bid or take an offer. The function would also take effect even if the incoming orders were uncoordinated, coming from one or more sources. The time period within which the trade takes place and the net contract volume would be configurable by the market-maker for each class.

QRM is setup at the acronym level. Therefore, multiple users that share the same acronym and exchange will be sharing the same QRM values. For example,

- User ID1 and user ID2 are sharing acronym ABC

- User ID1 is quoting IBM
- User ID2 changes QRM on his/her CBOE*direct* trading system for IBM
- User ID1's QRM values will change to reflect the changes made by User ID2

Quote Risk Monitoring extensions to FIX 4.2

QRM Requests and Responses are XML messages that contain the name of an operation and arguments to that operation. The QRM Requests are contained with the tags specifying the operation to be performed:

<*methodname***>...</***methodname***>**

An example of this is the QRM operation to set the risk profile for the user:

<setQuoteRiskProfile> *arguments for this operation go here* **</setQuoteRiskProfile>**

All QRM Requests are entered via the *Quote Status Request* message. The XML message is contained in the XmlData[213] tag that is part of the standard FIX 4.2 header.

An example of this is the QRM operation to set the risk profile for the user:

^A212=*length of XML message*

213=<setQuoteRiskProfile> *arguments for this operation go here* **</setQuoteRiskProfile>^A**

All QRM Responses to QRM Requests are returned within a *Quote Acknowledgement* message. Quote Responses follow the following format:

<*methodname***Response>** *Attributes returned with the response*

</*methodname***Response>**

An example of this response is:

XmlData =

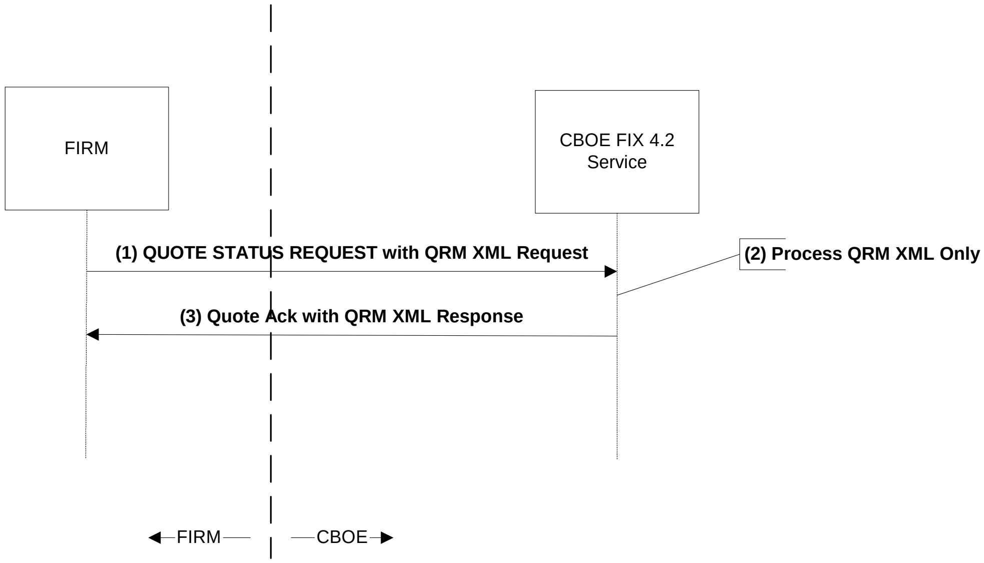
213=

```
<getAllQuoteRiskProfilesResponse>
  <UserQuoteRiskManagementProfileStruct>
    <defaultQuoteRiskProfile>
      <classKey>0</classKey>
      <Symbol>N/A</Symbol>
      <SecurityType>OPT</SecurityType>
      <quoteRiskManagementEnabled>false
      </quoteRiskManagementEnabled>
      <timeWindow>120000</timeWindow>
      <volumeThreshold>1001</volumeThreshold>
   </defaultQuoteRiskProfile>
   <globalQuoteRiskManagementEnabled>false
   </globalQuoteRiskManagementEnabled>
   <quoteRiskProfiles>
    </quoteRiskProfiles>
  </UserQuoteRiskManagementProfileStruct>
</getAllQuoteRiskProfilesResponse>^A
```

In case of error, the Quote Acknowledgement will contain an error message.

The response to a QRM Instruction is placed in a *Quote Acknowledgement*.

Figure 8 QRM Quote Status Request sequence diagram


FIRM
CBOE FIX 4.2 Service
(1) QUOTE STATUS REQUEST with QRM XML Request
(2) Process QRM XML Only
(3) Quote Ack with QRM XML Response
FIRM
CBOE

Using Quote Status Request to submit QRM Request

The Quote Status Request is used to submit a QRM request. Only a small subset of the Quote Status Request tags are used. Most important is that the QRM XML Request is placed in the XMLData[213] tag in the FIX 4.2 header.

Table 32 Quote Status Request message for submitting Quote Risk Monitoring Requests

Tag	Field Name	FIX Req'd	CBOE Req'd	Comments
	Standard Header	Y	Y	*MsgType[35]* = a (lowercase)
212	XmlDataLen	Y	Y	This field must contain the exact length of the XML string contained in tag 213. NOTE: This field is part of the FIX 4.2 standard header.
213	XmlData	Y	Y	This field will contain the QRM XML Request string. NOTE: This field is part of the FIX 4.2 standard header.
117	QuoteID	Y	Y	QuoteID – when used with QRM – this is the identifier that is returned in the QuoteReqID[131] in the Quote Acknowledgement. You should use this value to match the response back to the request.
55	Symbol	Y	Y	Symbol for the class, which the QRM Request applies. Use “NONE” if the QRM Request does not require a Class Symbol.
167	SecurityType	Y	Y	Type of security
	Standard Trailer	Y	Y	

QRM Responses contained in the Quote Acknowledgement Messages

The Quote Acknowledgement is used to send the QRM XML response.

Table 33 Quote Acknowledgement for Quote Risk Monitoring

Tag	Field Name	Comments
	Standard Header	MsgType = b (lowercase)
212	XmlDataLen	This field will contain the exact length of the XML string contained in tag 213. NOTE: This field is part of the FIX 4.2 standard header.
213	XmlData	This field will contain the QRM XML Response string. NOTE: This field is part of the FIX 4.2 standard header.art of the
131	QuoteReqID	Required when acknowledgment is in response to a Quote Request message
297	QuoteAckStatus	Status of the quote acknowledgement. Will be = 6 (CBOE defined value) that indicates this is a QRM response.
58	Text	Will be set to “QRM Response Only” if the request succeeded. Will contain an error otherwise.
	Standard Trailer	

getAllQuoteRiskProfiles

Obtain all quote risk profiles for the user.

Note: The symbol is required by FIX so use “NONE” if no symbol is needed for the particular request.

Request

<getAllQuoteRiskProfiles/>

Arguments: None.

Example:

```
8=FIX.4.2^49=JIM^56=DFIX202^35=a^212=26^
213=<getAllQuoteRiskProfiles/>
^117=QRM.2^55=NONE^167=OPT^10=137^
```

Response

```
8=FIX.4.2^9=0613^35=b^49=DFIX202^56=JIM^34=4^
52=20010803-00:28:20^212=514^
213=<getAllQuoteRiskProfilesResponse>
  <UserQuoteRiskManagementProfileStruct>
       <defaultQuoteRiskProfile>
         <classKey>0</classKey>
         <Symbol>N/A</Symbol>
         <SecurityType>OPT</SecurityType>
         <quoteRiskManagementEnabled>false
         </quoteRiskManagementEnabled>
         <timeWindow>120000</timeWindow>
         <volumeThreshold>1001</volumeThreshold>
        </defaultQuoteRiskProfile>
    <globalQuoteRiskManagementEnabled>false
    </globalQuoteRiskManagementEnabled>
    <quoteRiskProfiles>
    </quoteRiskProfiles>
  </UserQuoteRiskManagementProfileStruct>
</getAllQuoteRiskProfilesResponse>^
131=QRM.2^297=6^58=QRM Response only^10=179^
```

NOTE: The message comes back as a continuous string – the message is indented with line breaks for readability in the documentation. This applies to all examples in this document.

getDefaultQuoteRiskProfile

Obtain the default quote risk profile for the user. There is default risk profile available for each market maker. By default the default risk profile is disabled.

Request

<getDefaultQuoteRiskProfile/>

Arguments: None.

Example:

```
8=FIX.4.2^49=JIM^56=DFIX202^35=a^212=29^
213=<getDefaultQuoteRiskProfile/>^
117=QRM.0^55=NONE^167=OPT^10=137^
```

Response

```
8=FIX.4.2^9=0403^35=b^49=DFIX202^56=JIM^34=2^
52=20010803-00:28:11^212=304^
213=<getDefaultQuoteRiskProfileResponse>
 <QRMProfile>
   <classKey>0</classKey>
   <Symbol>N/A</Symbol>
   <SecurityType>OPT</SecurityType>
   <quoteRiskManagementEnabled>false
   </quoteRiskManagementEnabled>
   <timeWindow>120000</timeWindow>
   <volumeThreshold>1001</volumeThreshold>
 </QRMProfile>
</getDefaultQuoteRiskProfileResponse>
^131=QRM.0^297=6^58=QRM Response only^10=167^
```

getQuoteRiskManagementEnabledStatus

Get the quote risk management enabled status. Returns true if quote risk management is enabled.

Request

<getQuoteRiskManagementEnabledStatus/>

Arguments: None.

Example:

```
8=FIX.4.2^49=JIM^56=DFIX202^35=a^212=29^
213=<getDefaultQuoteRiskProfile/>^
117=QRM.0^55=NONE^167=OPT^10=137^
```

Response

```
8=FIX.4.2^9=0212^35=b^49=DFIX202^56=JIM^34=3^
52=20010807-18:25:11^212=113^
213=<getQuoteRiskManagementEnabledStatusResponse>
<status>false</status>
</getQuoteRiskManagementEnabledStatusResponse>^
131=QRM.1^297=6^58=QRM Response only^10=245^
```

getQuoteRiskManagementProfileByClass

Return the quote risk management profile for a specific class.

Request

<getQuoteRiskManagementProfileByClass/>

Arguments: None.

Symbol[55] Required

SecurityType[167] Required

Example:

```
8=FIX.4.2^49=JIM^56=DFIX202^35=a^212=39^
213=<getQuoteRiskManagementProfileByClass/>^117=QRM.4^
55=DJX^167=OPT^10=137^
```

Response

```
8=FIX.4.2^9=0422^35=b^49=DFIX202^56=JIM^34=6^
52=20010803-00:28:25^212=323^
213=<getQuoteRiskManagementProfileByClassResponse>
    <QRMProfile>
     <classKey>9175087</classKey>
     <Symbol>DJX</Symbol>
     <SecurityType>OPT</SecurityType>
     <quoteRiskManagementEnabled>true
     </quoteRiskManagementEnabled>
     <timeWindow>90</timeWindow>
     <volumeThreshold>50</volumeThreshold>
    </QRMProfile>
</getQuoteRiskManagementProfileByClassResponse>
^131=QRM.4^297=6^58=QRM Response only^10=147^
```

removeAllQuoteRiskProfiles

Remove all risk management profiles for the user.

Request

<removeAllQuoteRiskProfiles/>

Arguments: None.

Example:

```
8=FIX.4.2^49=JIM^56=DFIX202^35=a^212=29^
213=<removeAllQuoteRiskProfiles/>^
117=QRM.17^55=NA^167=OPT^10=137^
```

Response

```
8=FIX.4.2^9=0173^35=b^49=DFIX202^56=JIM^34=19^52=20010803-00:28:51^212=73^
213=<removeAllQuoteRiskProfilesResponse>
</removeAllQuoteRiskProfilesResponse>^
131=QRM.17^297=6^58=QRM Response only^10=192^
```

removeQuoteRiskProfile

Remove a quote risk management profile for a specific class for the user.

Request

<removeQuoteRiskProfile/>

Arguments: None.

Symbol[55] Required

SecurityType[167] Required

Example:

```
8=FIX.4.2^49=JIM^56=DFIX202^35=a^212=25^
213=<removeQuoteRiskProfile/>^
117=QRM.10^55=DJX^167=OPT^10=137^
```

Response

```
8=FIX.4.2^9=0165^35=b^49=DFIX202^56=JIM^34=12^
52=20010803-00:28:39^
212=65^213=<removeQuoteRiskProfileResponse>
</removeQuoteRiskProfileResponse>^
131=QRM.10^297=6^58=QRM Response only^10=162^
```

setQuoteRiskManagementEnabledStatus

Turn quote risk monitoring off or on.

Request

<setQuoteRiskManagementEnabledStatus>

<status>

true | false

</status>

</setQuoteRiskManagementEnabledStatus>^

Arguments: True or false

Class Symbol Required: No

Example:

```
8=FIX.4.2^49=JIM^56=DFIX202^35=a^212=96^
213=<setQuoteRiskManagementEnabledStatus>
<status>true</status>
</setQuoteRiskManagementEnabledStatus>^
117=QRM.7^55=NONE^167=OPT^10=137^
```

Response

```
8=FIX.4.2^9=0190^35=b^49=DFIX202^56=JIM^34=13^
52=20010807-23:41:18^212=91^
213=<setQuoteRiskManagementEnabledStatusResponse>
</setQuoteRiskManagementEnabledStatusResponse>^
131=QRM.7^297=6^58=QRM Response only^10=169^
```

setQuoteRiskProfile

Set a quote risk management profile for a specific class for the user.

Request

```
<setQuoteRiskProfile>
<QRMStruct>
    <quoteRiskManagementEnabled>
          true  |  false
     </quoteRiskManagementEnabled>
     <timeWindow>
         Integer Value - Time in milliseconds
     </timeWindow>
     <volumeThreshold>
         Integer Value – quantity of product
     </volumeThreshold>
  </QRMStruct>
</setQuoteRiskProfile>
```

Arguments: Required <QRMStruct>

Symbol[55] Required.

SecurityType[167] Required.

Example:

```
8=FIX.4.2^49=JIM^56=DFIX202^35=a^212=194^213=
<setQuoteRiskProfile><QRMStruct><quoteRiskManagementEnabled>true</quoteRiskManagementEnabl
ed><timeWindow>90000</timeWindow><volumeThreshold>50</volumeThreshold></QRMStruct>
</setQuoteRiskProfile>^117=QRM.3^55=DJX^167=OPT^10=137^
```

Response

```
8=FIX.4.2^9=0157^35=b^49=DFIX202^56=JIM^34=5^52=20010803-
00:28:23^212=59^213=<setQuoteRiskProfileResponse>
</setQuoteRiskProfileResponse>^131=QRM.3^297=6^
58=QRM Response only^10=191^
```