



FIX Protocol Limited
Market Data Working Group

Recommended Practices for Book Management

Version 2.00
January, 2007

CONTENTS

1	INTRODUCTION	4
1.1	Background	4
1.2	Purpose and Scope.....	4
1.3	Document Structure and Format	4
1.4	Acknowledgements.....	5
1.5	Related Documentation	5
1.6	Disclaimer.....	5
2	OVERVIEW OF FIX MARKET DATA MESSAGING	6
2.1	Introduction	6
2.2	Top-Of-Book	7
2.3	Moved to Tag Definition section below Price-Depth	9
2.3	Price-Depth	10
2.4	Use of the Tags PriceLevel and MDEntryID	11
2.5	Order-Depth.....	13
2.6	Trading and Security Status.....	14
2.7	Quote Markets	16
2.8	Optional Tags	17
2.8.1	Sequencing the Book.....	17
2.8.2	Book Type.....	17
2.8.3	Tags Referenced in the Document.....	17
3	EXAMPLES	19
3.1	Initial Market	19
3.2	An Order is added to the book	20
3.3	An Order is added Below Top Of Book	22
3.4	Change an order.....	24

3.5	Delete Order which deletes the Price level.	26
3.6	A Trade occurs over multiple price levels.....	28
3.7	A new price causes the bottom price level to be deleted	32
3.8	Use of MDEntryID for Top-Of-Book and Price-Depth	36
4	MARKET STATISTICS.....	38
4.1	Market Statistics	38
5	SNAPSHOTS AND RECOVERY.....	41
5.1	Requesting Snapshots	41
5.2	A Market Data Snapshot for Top-Of-Book.....	43
5.3	A Market Data Snapshot for Price-Depth:	43
5.4	A Market Data Snapshot for Order-Depth.....	45
5.5	Start Of Day Snapshot	47
5.6	Issues with Snapshot Functionality	47
5.7	Synchronization of Snapshots and Data Feed.....	47
5.8	Retransmission Capability	48
5.9	Market Data Request.....	51
5.10	Considerations for Data Retransmissions.....	52
5.11	Continuous Refresh.....	52

1 Introduction

1.1 Background

The FIX (Financial Information eXchange) protocol has been developed to allow participants in the financial services industry transmit certain types of information in a common format. It was originally developed for pre-trade and trade messaging and has since been extended to include various market data messages.

The protocol is administrated by FIX Protocol Ltd., whose aims are to ensure the continuing development and improvement of the protocol, to market its existence and to assist the industry in its uptake. The FIX Market Data Optimization Working Group (MDOWG) operates within this framework and exists to fulfil these three objectives in particular with regard to Market data messaging.

This document is a product of the FIX Market Data Optimization Working Group. These recommendations are intended to be supported with FIX 5.0 which contains extensions developed by the Market Data Optimization Work Group. However, it is also possible to implement on a limited basis using earlier versions of the FIX specification such as 4.4.

1.2 Purpose and Scope

Objective of this Document

This document aims to be a guide to implementing Market Data instructions for equities, listed derivatives, foreign exchange and fixed income transactions using version 5.0 of the FIX protocol. It does not seek to replace the FIX 5.0 Protocol Specification, rather to complement it by providing specific instructions and examples relating to best-practice implementations of the messages contained within the Specification.

Why This Document Exists

The FIX 5.0 Protocol Specification provides, through its message definitions (volume 3) and data dictionary (volume 6) details as to how FIX messages should be constructed and formatted. It also contains some details as to how messages should be used. It has been kept deliberately flexible, in order to ensure that it is able to support as many business scenarios as possible. The drawback of this from an implementation point of view is that the Specification does not dictate a single way of implementing specific functionality.

The purpose of this document is to define an agreed sub-set of permissible message implementations and, in doing so, provide a definitive statement as to how these messages should be implemented in practice. Adherence to the standards laid out in this document should result in implementations that are wholly compatible with each other.

Who Should Read It

This document is intended for the following audience:

- Anybody involved in or considering an implementation of FIX Market Data using version 5.0 of the FIX protocol. Note that the majority of recommendation may also be applied to FIX 4.4.
- Suppliers of market data processing technology.

1.3 Document Structure and Format

Nomenclature

To avoid any ambiguity, the following words shall be used throughout this document to take the following meanings. This may differ from convention in certain markets or organisations, but has been done for consistency and to ensure complete understanding of the document's content.

The *Vendor* is the owner of the orderbook and disseminator of the Data Feed. The *Client* is the recipient of the data feed and is recreating a copy of the orderbook from the messages sent by the Vendor.

The FIX Market Data message has a FIX header followed by a number of items. Each item is one instruction such as an orderbook update, a quote or a trade. This document refers to the items as *instructions* when describing how the orderbook is maintained.

A single FIX Market Data message can contain many instructions such as New bid/ask, Change bid/ask; Delete bid/ask and Trades across multiple instruments.

1.4 Acknowledgements

Some entities/trademarks are referred to in this document. Further details together with contact points can be obtained from the respective web-sites:

FIX Protocol Limited at www.FIXprotocol.org

Many thanks to all that have helped to review and provide input to this document. If you have comments or questions on this document please raise them through the FIX 5.0 forum which can be accessed at <http://www.fixprotocol.org/discuss/47..> Contact details are available on the FIX Protocol website under 'FPL Organization', 'Market Data Optimization Working Group'.

1.5 Related Documentation

FIX 4.4 Protocol Specification, available on the FIX Protocol website (www.FIXprotocol.org) under 'Technical Specifications'. This is itself made up of 7 volumes – 1 Introduction, 2 Session, 3 Pre-Trade Messages, 4 Order/Execution Messages, 5 Post-Trade Messages, 6 Data Dictionary, 7 Usage by Product. For further information regarding the FIX post-trade message set, please refer to volumes 5 (for the messages) and 6 (for the field definitions). For information regarding the creation of a FIX session, refer to volume 2. For other product-specific information, refer to volume 7.

A variety of other supporting and reference documents are available on the FIX protocol website. The Allocations Working Group's page (under 'Working Groups', 'Allocations') contains a number of documents relating specifically to the post-trade area, together with a listing of the members of the Working Group.

1.6 Disclaimer

This document is intended as a guide to implementing certain FIX messages to perform certain functions. It makes no representation as to the legality or otherwise, or conformance to market practice, local regulations or any organisational rules of any of the processes referred to in this document. Before undertaking any implementation of these messages, your own organisation's relevant control departments (e.g. compliance, legal, risk management/control) should be made aware of the functionality being supported by these messages and be confident that all necessary internal and external controls, data auditing requirements and similar are being met.

2 Overview of FIX Market Data Messaging

2.1 Introduction

The FIX Market Data Working Group has defined a set of messages and protocols with the objective of creating a standard that allows vendors and clients to connect quickly and reliably across a range of markets.

The Working Group recognizes two general categories of markets. The first category is the style of market where executable orders and quotes are maintained in a central limit order book and is typical of the automated equity markets. The other style is a Quote market where quotes are streamed from many dealers and cannot be consolidated. The latter is typical of the bond and currency markets.

Please note that all materials included herein are recommended guidelines with respect to implementing a book management paradigm, not hard and fast rules. The specification provides flexibility in applying different techniques to manage and organize book-related market data.

The use of a Central Orderbook is typical of electronic equity markets where executable orders are placed into a book, sorted by price, then time. A client can expect to execute at the price of the best order in the book or better. The markets are usually anonymous, where the displayed price is firm and available to all parties.

Precision and Depth

A Central Orderbook is characterized by two fundamental concepts; **Precision** and **Depth**. Precision refers to the level of detail at which a book is kept and may be summarized by price level or comprised of individual orders. These are referred to as **Price-Depth (Aggregate Order Book)** and **Order-Depth (Non-aggregated Order Book Detail)** respectively.

Price-Depth is used to refer to the case where the aggregated quantity is provided for each side of each price level.

Order-Depth is used to refer to the case where every order is described and the client can recreate a copy of the orderbook. There is limited information about each order. Details such as Identity, Account and FreeText are not available, except in a hit and take market where identity of the opposite party may be divulged.

The second aspect of book management is Depth which refers to the number of entries that will be kept in a book. Book Depth may range from **Top-Of-Book** which represents only the best price level to Full Book Depth which represents all orders currently open in the market.

Note that different markets refer to the views with different names. The working group has selected a name that is generally acceptable so that the different views are not confused. A common name for the aggregate view is Price-Depth while the common name for the orderbook Detail is Order-Depth. Alternate names for Top-Of-Book are Top Of File and Best Bid Offer or BBO.

The Working Group has defined a standard for these views that can be used across multiple markets. Given the fundamental characteristics of Aggregation and Depth, the following permutations are possible and are recommended as the most common book management practices:

- Top-Of-Book
- Partial Price-Depth
- Partial Order-Depth
- Full Price-Depth
- Full Order-Depth

Trades are sent as a separate instructions within a FIX Market Data Message (MDEntryType = "Trade"). Trades are only used to update the Ticker, or any display where Last, High, Low, Open, Close, Volume, and

Cumulative Volume are displayed. **The Trade instruction is not used to update the view of the orderbook.** A trade may not have executed fully against the orderbook depending upon the priority rules of the market. When a trade executes against the book then a Change or Delete instruction is used to update the orderbook.

A FIFO market could require that the client use the Trade Instruction to update the Order Book so that corresponding Orderbook changes do not need to be sent; however, this is not the recommended practice. Additionally, when privately negotiated trades (Trade Capture Reports) are also published as market data, using trade instructions to update Order Book data is again not recommended.

FIX Message Structures

FIX provides two message formats that will be referred to within this document.

The Market Data Incremental Refresh message is used to apply instructions to a book in an incremental manner. This message is best suited for the maintenance of price-depth and order-depth books but may also be used for top-of-book.

The Market Data Snapshot Full Refresh is used to provide a full snapshot of the current book. For this reason it is appropriate for top-of-book updates in which both sides of the book are sent. The message may also be used in a recovery situation where a number of order books need to be reestablished.

2.2 Top-Of-Book

The Vendor may choose to provide the current best bid and offer in the market. Trade details and Instrument Status are provided with separate Market Data Instructions.

Top-Of-Book (Best Bid Offer)

Bid	Ask
100 @ \$2.90	400 @ \$3.10

The vendor may distribute other information beside Top-Of-Book

Trade Information

Last	Qty	Open	High	Low	Chg	Tick	CumQty
\$2.95	2000	\$3.10	\$3.10	\$2.90	- 0.15	+	10,500,000

Trading Session Status

Status
Open

The Vendor maintains the Top-Of-Book view with the OVERLAY instruction.

Overlay, to update the quantity and Price

The incremental instruction approach assumes the use of the Market Data Incremental Refresh message. The Bid and Ask Sides are updated independently with separate instructions. Many existing BBO (Best Bid Offer) feeds have fixed formats that provide both the bid and offer sides in all updates. The practice of sending separate instructions can provide efficiencies by allowing only the bid or ask to be sent, based on which side has changed, rather than both sides. Top-Of-Book only has one price level so there is no need to use multiple instructions when the best price changes. An action of *overlay* is used to indicate a new price or a change in quantity at an existing price. When the best price changes the vendor does not send a delete of one price level followed by a New for the next price level.

A Trade instruction does not update the Top-Of-Book. A separate *overlay* instruction is sent to update the book as a result of the trade. The *overlay* instruction does not necessarily follow the trade. The Trade instruction and the instruction to update the book may be separated by a period of time and the change could be sent before or after the trade.

Trades

The Trade instruction that is sent within the Market Data message is used to provide information about two sided trades. One party could enter an order that executes against several counterparties over multiple prices. Some Vendors will send a consolidated trade for the total quantity at each price level. Other vendors will send one trade instruction for each counterparty pair. Either method is acceptable. One concept that is prohibited is to show the Bid and Ask sides as separate trades as this results in double counting. Individual Fill execution reports are sent to each party in the trade, but not through the Market Data feed.

A Trade can update the trade statistics (Last, Volume, Open, High, Low, Chg, Direction) for both the local exchange and the national market. The trade may have been an off-market trade in which case it does not update any of the trade statistics. There are two approaches commonly used to update the trade statistics. Some vendors send Open, High, Low and Last with each trade and some send them only if they are changed. Proper formatting of Trades, Volume, Open, High, and Low will be discussed further below.

A Basic Instruction of Top-of_Book

A FIX message sent to update the top of book will update one side only. The tags normally sent with Top Of Book are:

```
SecurityDesc/SecurityID/Symbol=security identifier (note that the there  
are several ways to identify the security)MDBBookType=Top-Of-Book,  
MDUpdateAction=Overlay  
MDEntryType=Bid/Ask  
MDEntrySize=quantity available at bid/ask MDEntryPx=price.  
MDEntryPx=price
```


Indexing the Entry

Top-of-Book entries can be referenced for maintenance purposes using several techniques. The first is to create a composite index which consists of the Security Identifier (see above), MDEntryType (bid/ask) and MDEntryPx. This set of fields acts as a composite key that allows the entry (bid or ask) to be accessed and subsequently deleted.

The other approach is to use MDEntryID and assign a unique identifier to act as a key. Every active entry carries a unique id which can be used for reference purposes. When the entry is no longer active, the ID can be re-used and assigned to the next active entry.

Both approaches have their advantages. The 'composite key' approach does not require that an external identifier to be carried on the initial Add or Delete instruction nor does it require the sending system to maintain a matrix of external identifiers. The MDEntryID approach provides a single field for referencing an entry which may improve system performance.

Note that use of MDEntryID does not mean that MDEntryType, MDEntryPx, or MDPriceLevel can be dropped on a Delete instruction. These fields are necessary to ensure the integrity of the book when applying the instruction.

2.3 Moved to Tag Definition section below

Price-Depth

The Vendor may choose to provide the aggregate quantity available at each price level. This view can be visualized as a number of rows in a table for each of the bid and offer sides. On each side there are a number of rows showing the quantity available at a number of price levels.

Example:

Bid	Ask
100 @ \$2.90	400 @ \$3.10
65 @ \$2.85	5000 @ \$3.20
450 @ \$2.80	200 @ \$3.30

The Price-Depth view is an expansion of the Top-Of-Book view. The trade and status windows are independent and can be shown beside the Price-Depth view in the same way as they are shown beside the Top-Of-Book view.

A Price-Depth Book is sequenced by Price, descending for bid and ascending for ask.

Another common representation is to show the Market in a Vertical fashion.

Bid	Ask
	200 @ \$3.30
	5000 @ \$3.20
	400 @ \$3.10
100 @ \$2.90	
65 @ \$2.85	
450 @ \$2.80	

The Vendor must maintain the Price-Depth view with the following entries

New, to create/insert a new price level

Delete, to remove a price level

Change, to update quantity at a price level

The Bid and Ask Sides are updated with separate instructions.

Most markets do not show the Price-Depth view for the entire book. The Price-Depth view is limited to a number of levels, typically three to five. Equity Markets that trade in pennies tend to show at least ten price levels. The Recipient must know how many price levels are being supplied by the vendor and must delete the bottom row when the number of rows is exceeded. The vendor will not send a delete instruction when the number of rows is exceeded. The vendor will send the bottom row again when a higher level row is deleted.

If a trade occurs in the market then the vendor will send Delete or Change instructions to update the Price-Depth. The trade instruction itself is not used to update the order book as the order may not have executed against the book.

2.4 Use of the Tags PriceLevel and MDEntryID

MDPriceLevel

The addition of the tag PriceLevel is used for markets that are not Price priority or in any situation where there can be confusion on the sequencing of the rows. PriceLevel is also beneficial for recovery after a gap in the data feed. If PriceLevel is present then the client can tell if a new price is the top price level and can delete higher price levels. Exhibit 1 shows how PriceLevel can be useful in detecting a book made inaccurate due to dropped message.

Indexing the Entry

Price-Depth entries can be referenced for maintenance purposes using several techniques. The first is to create a composite index which consists of the Security Identifier (see above), MDEntryType (bid/ask) and MDEntryPx. This set of fields acts as a composite key that allows the entry to be accessed and subsequently deleted.

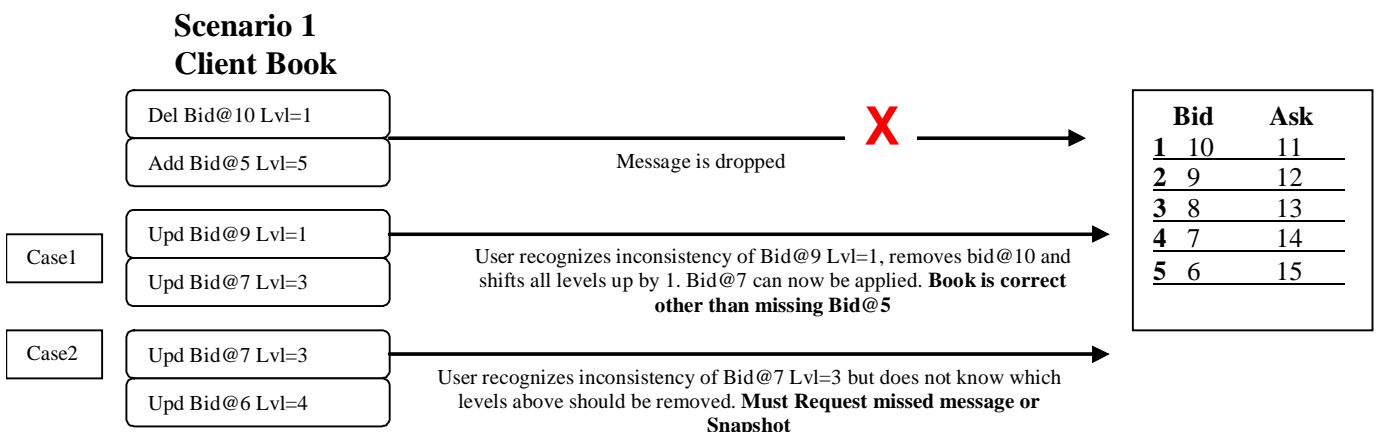
The other approach is to use MDEntryID and assign a unique identifier to act as a key. Every active entry carries a unique ID which can be used for reference purposes. When the entry is no longer active, the ID can be re-used and assigned to the next active entry.

Both approaches have their advantages. The 'composite key' approach does not require that an external identifier to be carried on the initial Add or Delete instruction nor does it require the sending system to maintain a matrix of external identifiers. The MDEntryID approach provides a single field for referencing an entry which may improve system performance.

Note that use of MDEntryID does not mean that MDEntryType, MDEntryPx, or MDPriceLevel can be dropped on an Update or Delete instruction. These fields are necessary to ensure the integrity of the book when applying the instruction.

Exhibit 1

Price Level is a technique for conveying the position to which a given incremental instruction should be applied. The purpose is to reinforce the integrity of the instruction being received in relation to the current state of the book.



Note that with both the Price and Order Depth views there is a possibility that some orders are available for unconditional execution while others are not. Examples include various quantity conditions as "minimum fill

quantity”, “minimum execution increment”, and “AON”. Currently, this should be supported via the QuoteCondition field, tag 276 and enumerations that are yet to be defined.

2.5 Order-Depth

The vendor may provide limited information about each individual orders at each price level. Information is sent about each order at the price level rather than stating that a total quantity is available at each price level. The Client can summarize the information to create the Price-Depth view if desired.

It is possible in some markets to assume that orders can be sorted by price then time, however many markets do not prioritize all orders by time. The Orderbook can be viewed as a table with multiple orders at each price level. Each order has a unique Entry ID. When adding orders to the book the Add instruction contains the EntryPositionNumber within the PriceLevel at which the order is to be inserted. Change and Delete instructions will reference the order by its MEntryID. The MEntryID is not to be confused with the ClOrdID which is the confidential order reference assigned by the client when the order was entered. MEntryID is an identifier assigned by the vendor to track this order in the market data feed and may be unique across the entire feed. FIX requires that the MEntryID be unique among all other active entries,

Note that MEntryID is constant for an entry until that entry is removed from the book due either to deletion or dropping out of the stated market depth.

Order-Depth	
BID	
9740	ID=213, Qty=2
9730	ID=223, Qty=4; ID=227, Qty=1
9720	ID=973, Qty=1
ASK	
9760	ID=230, Qty=5
9770	ID=231, Qty=3
9780	ID=232, Qty=5
9790	ID=233, Qty=4; ID=234, Qty=3

The vendor must send the following instructions to maintain the Orderbook:

New, to insert a new order

Delete, to delete an order

Change, to alter the details of an order

New is used to insert an order, identified by MEntryID, into a position within a price level. The new entry is sequenced by Price Level Number within the book and then Entry Position Number within that PriceLevel.

Change is used to change the quantity or other information related to an order. Change should not be used to alter the PriceLevel or EntryPositionNumber of an entry as this is the responsibility of the client maintaining the book. The trading function of Cancel & Replace that changes the price of an order would result in a delete of the old order and an insert of the updated order into the new price. 'Change' should not be used when the price of the order is changing due to a cancel/replace or when the order is being re-ranked for any reason. Any action which causes the order to move to a different price level or position should be accomplished through use of a 'delete' followed by a 'new'.

Delete and Change Instructions use MDEntryID as the key to reference the order. They can optionally include the PriceLevel and EntryPositionNumber of the order. This can aid client systems when the market has deep orderbooks as it narrows the search for the subject order. It also provides a constant check regarding the integrity of the book.

The Recipient must know how many price levels are being supplied by the vendor and must delete the bottom row when the number of rows is exceeded. The vendor will not send a delete instruction when the number of rows is exceeded. The vendor will send the bottom row again when a higher level row is deleted.

Delete-Thru is an instruction that can be used to reduce the number of instructions necessary to maintain a book and further improve efficiency. The "Delete Thru" instruction causes the book to be deleted from the top down to the specified Price and PriceLevel. The empty price levels are then filled with Add instructions.

Indexing the Entry

Order-depth books should use MDEntryID as the key to reference the order for Delete and Change Instructions. They can optionally include the PriceLevel and EntryPositionNumber of the order. This can aid client systems when the market has deep orderbooks as it narrows the search for the subject order. It also provides a constant check regarding the integrity of the book. The 'composite key' approach using a Security Identifier, MDEntryType (bid/ask) and MDEntryPx is not sufficiently detailed to reference a specific order in the book.

A Trade instruction is not used to update the book. An execution against the book may not occur in Time priority. Following an execution a trade instruction is sent along with Change/Delete instructions to update the book as a result of the trade.

Summary:

Each order has an MDEntryID that is a unique reference within a feed or order book. The MDEntryID must be unique across the feed when active but can be reused once when no longer in use.

Price is used to sequence the order book unless PriceLevel is provided.

MDEntryID is used to update and delete an order. Price may be included to assist in finding the order within the book.

EntryPositionNumber is used to sequence the orders within a given price level.

The order-level view has the option of showing a limited number of price levels. When adding an order at a new price level it is implied that all orders at the bottom price level are to be removed. If an order is removed that clears an entire price level then the bottom price level must be resent by the vendor.

2.6 Trading and Security Status

The status of each instrument is normally sent as part of the data feed. When the state of an instrument is changed, a FIX Trading Session Status or Security Status message is sent with the instrument name and the new state. The current quotes are not sent again. The state is associated with the instrument, not the Bid or the Offer.

The trading status of a security should be sent on the FIX Security Status message (35=h).

The trading status of a market should be sent on the FIX Trading Session Status message (35=f).

In some markets, there is the concept that the instruments belong to various markets and that the markets open and close at different times. The trading state of an instrument is the lowest state of either its state or the owning market state. When the market opens then all instruments in an open state that are in that market will open as well. If the market closes then all instruments in that market will close.

Some Derivative markets will open/close all instruments in the market at the open and the close. This can be a very large number of messages which can create capacity issues for other instruments that are still trading. Another method is to use wildcards to indicate that all instruments of a particular class have changed state.

2.7 Quote Markets

In a Quote Market there are dealers that provide quotes. There is no central market and the concept of a Central Orderbook does not exist. The prices displayed are not necessarily firm or available to all participants. A customer must contact the dealer to trade and the price is often negotiated as a points discount to the published price. Orders are sent directly to the dealer. In some equity OTC markets, the orders can be sent to a central routing hub which then routes the orders to the end dealers.

In the debt markets the dealers provide quotes for different products based upon the popularity of the product. Some markets provide unsolicited quotes while others are solicited and require a call to the dealer to obtain a quote. The dealer may provide discounts to the quoted rate depending upon the rating of the client or other negotiated arrangements.

The dealer in a quote market will publish its prices to multiple vendors. The vendors seek to pass the dealer quotes to the client as a montage of prices showing the price and quantity available from each dealer. Some Vendors will take consolidated feeds from other vendors and further consolidate the quotes into higher level montages such as adding markets from multiple countries.

A standard to cover this style of market has the same requirements as the Price-Depth view on the orderbook concept. The vendor will pass on quotes that have size and price however there can be many rows that have the same price, representing the different dealers. An argument can be made that it is also like an orderbook view with multiple orders at each price level. It is considered to be closer to a Price-Depth view than an Order Depth view since the number of quotes in the market is static with the prices constantly changing rather than the number of orders changing.

The Vendor must issue instructions to maintain the view of each table:

New, to create a new price level

Delete, to remove a price level

Change, to update quantity at a price level

The Bid and Ask Sides are updated independently. A message may contain updates to many books and can update the bid and ask sides in the same message.

Each Dealer will be represented as its own price-level row and there can be many at the same price level.

Each price-level row must be assigned a unique MDEntryID, price-level, a price and a price-position along with the quantity available at that position. Change and Delete must then reference MDEntryID and Dealer reference to update the book. With this approach it is possible to show multiple rows at the same price, show the rows in any sequence and minimize errors.

2.8 Optional Tags

2.8.1 Sequencing the Book

There are a number of cases where you cannot use price to sort the rows.

Some orders can sit in the book with no price, such as Market orders prior to the open or Cabinet orders. A Market Order is generally shown with a null price at the top of the book. Cabinet orders are generally placed at the bottom of the book with an undefined price.

Some debt markets show price inverse to the yield. The best bid is that order with the largest price, resulting in the priorities being reversed. Others quote relative to a standard rate which results in zero and negative prices.

Prices can be consolidated from several markets that are at the same price.

If any of these situations lead to confusion, the vendor should use the optional tag PriceLevel in the Add instruction to specify which row in the table this new price level is to be inserted. When this convention is used, the row numbers start at one as the best price level.

Example:

PriceLevel	Bid	Ask	PriceLevel
1	100 @ \$2.90	400 @ \$3.10	1
2	65 @ \$2.85	5000 @ \$3.20	2
3	450 @ \$2.80	200 @ \$3.30	3

If a SELL order is entered onto the book at 3.15 then it would be Added as PriceLevel 2. Changes and deletes will refer to the price of 3.15.

Market Orders with no price could be shown at the top of the book by inserting PriceLevel 1 with a null price.

Multiple Markets or Multiple Dealers at the same price must use the market ID or Dealer ID to provide a unique key for each row.

2.8.2 Book Type

Many trading systems provide multiple views of an orderbook at the same time. The Market data update messages must specify which Book Type they are to be applied. In the event that the vendor is providing multiple views in parallel then the optional tag MDBookType should be used to indicate which view the instruction is to be applied.

2.8.3 Tags Referenced in the Document

The emphasis of this document is to provide a secure and reliable protocol for orderbook maintenance. Depending upon the application the Vendor may choose not to implement these recommendations in favor on increased throughput. A number of new tags are proposed:

MDBookType (tag is 1021): Used to allow multiple views over a common network session. Values are **Top-Of-Book**, Price-Depth, Order-Depth.

MDPriceLevel (tag is 1023): Used to specify sequencing of prices when a market is not Price priority. **MDPriceLevel** contains the row number at which a price is to be displayed, where the best price in the market is level 1. Also used to resynchronize a book if there are missing blocks in the data feed.

TradeCondition (tag 277): Specifies the conditions which qualify a trade that is being reported as part of the market data stream. Common values are Opening (E), Cancel (O), Exchange Last (U).

To support Reg-NMS, quotes must be marked as manual/Slow if they are manual, ie, not executed automatically.

Manual quotes are indicated with tag 276 QuoteCondition= L (Manual/slow quote)

Trades resulting from manual/slow quotes are also indicated as follows:

277 TradeCondition Valid values:

Y = Trade resulting from manual/slow quote

Z = Trade resulting from intermarket sweep

MDFeedType (tag 1022): Describes a class of service for given data feed

MDEntryPositionNo (tag 290): Used to identify the position of an order at a given PriceLevel in an order depth book.

MDEntryID (tag 278): Used to uniquely identify each entry in a book for more efficient update and deletion of that entry. Note that MDEntryID may also be used as an index price levels for subsequent updates and deletions and may improve the performance of client systems under certain conditions. Each side is assigned a unique integer that can be used as an index to that row for future updates. The index should start at one and re-use freed entries. The maximum number should be the maximum number of bid/ask rows.

It may be useful to assign an MDEntryID series per security (order book) to avoid problems when processes are partitioned. It may not be practical for MDEntryID to be unique across the feed.

Example:

The first BID disseminated for stock ZZZ may have an MDEntryID of 333 assigned. From then on, all further updates to the ZZZ BID should include MDEntryID of 333. This will allow a client system to create a reference table and index directly into row to update or delete that row. If the bid is deleted then MDEntryID 333 is now free and can be reused for another row.

3 Examples

3.1 Initial Market

This example shows how the different views are updated for the same events. In this example we are using a Futures Market where the price is very large and the quantity is expressed in contracts and tend to be small.

The following representation shows Top-Of-Book, Price-Depth and Order-Depth. In the Top-Of-Book and Price-Depth the cells show total quantity and price. On the bid-side you see qty/price and on the Ask side you see price/qty. In the Order-Depth you see the orders at each price level. For each order you see the reference ID and the quantity. The vendor may make additional information available but this is the minimum required.

Top-Of-Book – BBO		Price-Depth		Order-Depth	
BID	ASK	BID	ASK	BID	
2 / 9740	9760 / 5	2 / 9740	9760 / 5	9740	ID=213, Qty=2
		5 / 9730	9770 / 3	9730	ID=223, Qty=4; ID=227, Qty=1
		1 / 9720	9780 / 5	9720	ID=973, Qty=1
			9790 / 7	ASK	
				9760	ID=230, Qty=5
				9770	ID=231, Qty=3
				9780	ID=232, Qty=5
				9790	ID=233, Qty=4; ID=234, Qty=3

3.2 An Order is added to the book

An order is added to the orderbook creating a new best price level. It will cause all updates to occur to all three orderbook views.

Buy 3 @ 9750

Update Top-Of-Book

Tag number	Tag name	Value	Description
1021	MDBookType	1	Top-Of-Book
279	MDUpdateAction	0	New
269	MDEntryType	0	Bid
271	MDEntrySize	3	Quantity
270	MDEntryPx	9750	Price
107	SecurityDesc	GEZ5	Instrument

Update Price-Depth

Tag number	Tag name	Value	Description
1021	MDBookType	2	Price-Depth
1023	MDPriceLevel	1	Price Level
279	MDUpdateAction	0	New
269	MDEntryType	0	Bid
107	SecurityDesc	GEZ5	Instrument
271	MDEntrySize	3	Quantity
270	MDEntryPx	9750	Price

Update Order-Depth

Tag number	Tag name	Value	Description
1021	MDBookType	3	Order Depth
1023	MDPriceLevel	1	Price Level
279	MDUpdateAction	0	New
269	MDEntryType	0	Bid
107	SecurityDesc	GEZ5	Instrument
271	MDEntrySize	3	Quantity
270	MDEntryPx	9750	Price
290	MDEntryPositionNo	1	Position at this price level
278	MDEntryID	274	Order Reference

Result

Top-Of-Book		Price-Depth		Order-Depth	
BID	ASK	BID	ASK	BID	
3 / 9750	9760 / 5	3 / 9750	9760 / 5	9750	ID=274, Qty=3
		2 / 9740	9770 / 3	9740	ID=213, Qty=2
		5 / 9730	9780 / 5	9730	ID=223, Qty=4; ID=227, Qty=1
		1 / 9720	9790 / 7	9720	ID=973, Qty=1
				ASK	
				9760	ID=230, Qty=5
				9770	ID=231, Qty=3
				9780	ID=232, Qty=5
				9790	ID=233, Qty=4; ID=234, Qty=3

3.3 An Order is added Below Top Of Book

An order is added to a lower price level. The Top-of-book is not affected. The Price-Depth will update the size being displayed.

BUY 9 @ 9730

No change to Top-Of-Book

Update Price-Depth

Tag number	Tag name	Value	Description
1021	MDBookType	2	Price Depth
1023	MDPriceLevel	3	Price Level
279	MDUpdateAction	1	Change
269	MDEntryType	0	Bid
107	SecurityDesc	GEZ5	Instrument
271	MDEntrySize	14	Quantity
270	MDEntryPx	9730	Price

Update Order-Depth

Tag number	Tag name	Value	Description
1021	MDBookType	3	Order Depth
1023	MDPriceLevel	3	Price Level
279	MDUpdateAction	0	New
269	MDEntryType	0	Bid
107	SecurityDesc	GEZ5	Instrument
271	MDEntrySize	9	Quantity
270	MDEntryPx	9730	Price
290	MDEntryPositionNo	3	Position at this price level
278	MDEntryID	275	Order Reference

Result

Top-Of-Book BBO		Price-Depth		Order-Depth	
BID	ASK	BID	ASK	BID	
3 / 9750	9760 / 5	3 / 9750	9760 / 5	9750	ID=274, Qty=3
		2 / 9740	9770 / 3	9740	ID=213, Qty=2
		14 / 9730	9780 / 5	9730	ID=223, Qty=4; ID=227, Qty=1; ID=275, Qty=9
		1 / 9720	9790 / 7	9720	ID=973, Qty=1
				ASK	
				9760	ID=230, Qty=5
				9770	ID=231, Qty=3
				9780	ID=232, Qty=5
				9790	ID=233, Qty=4; ID=234, Qty=3

3.4 Change an order

An order is changed to increase its size. This causes the order to be updated in the Order-Depth and changes to the size in the Top-Of-Book and Price-Depth.

EntryID 274 changes Qty from 3 to 5.

Update Top-Of-Book

Tag number	Tag name	Value	Description
1021	MDBookType	1	Top-Of-Book
279	MDUpdateAction	0	New
269	MDEntryType	0	Bid
271	MDEntrySize	5	Quantity
270	MDEntryPx	9750	Price
107	SecurityDesc	GEZ5	Instrument

Update Price-Depth

Tag number	Tag name	Value	Description
1021	MDBookType	2	Price Depth
1023	MDPriceLevel	1	Price Level
279	MDUpdateAction	1	Change
269	MDEntryType	0	Bid
107	SecurityDesc	GEZ5	Instrument
271	MDEntrySize	5	Quantity
270	MDEntryPx	9750	Price

Update Order-Depth

Tag number	Tag name	Value	Description
1021	MDBookType	3	Order Depth
1023	MDPriceLevel	1	Price Level
279	MDUpdateAction	1	Change
269	MDEntryType	0	Bid
107	SecurityDesc	GEZ5	Instrument
271	MDEntrySize	5	Quantity
270	MDEntryPx	9750	Price,
290	MDEntryPositionNo	1	Position at this price level
278	MDEntryID	274	Order Reference

Result

Top-Of-Book, BBO		Price-Depth		Order-Depth	
BID	ASK	BID	ASK	BID	
5 / 9750	9760 / 5	5 / 9750	9760 / 5	9750	ID=274, Qty=5
		2 / 9740	9770 / 3	9740	ID=213, Qty=2
		14 / 9730	9780 / 5	9730	ID=223, Qty=4; ID=227, Qty=1; ID=275, Qty=9
		1 / 9720	9790 / 7	9720	ID=973, Qty=1
				ASK	
				9760	ID=230, Qty=5
				9770	ID=231, Qty=3
				9780	ID=232, Qty=5
				9790	ID=233, Qty=4; ID=234, Qty=3

Note that the trading function Cancel & Replace where an order changes price will cause two market data instructions in each of the Aggregate and Detail views. One instruction is required to delete the order from one price level and another to add a new order to the new price level.

Resequencing Entries at a PriceLevel

In this example the order remained at position one. If an order was changed such that it lost priority then a change instruction is sent with the new position. An Insert causes the new order to be inserted at the position stated in the instruction. A change will cause the order to be removed from the book then inserted at the position specified in the instruction. You must remove the order from the book before determining the new position for the order.

3.5 Delete Order which deletes the Price level.

An order is deleted at a lower price level and it is the only order at that price level. It causes the Price-Depth to have the entire price-level removed.

Delete Order with EntryID=213

No change to Top-Of-Book

Update Price-Depth

Tag number	Tag name	Value	Description
1021	MDBookType	2	Price Depth
1023	MDPriceLevel	2	Price Level
279	MDUpdateAction	2	Delete
269	MDEntryType	0	Bid
107	SecurityDesc	GEZ5	Instrument
270	MDEntryPx	9740	Price

Update Order-Depth

Tag number	Tag name	Value	Description
1021	MDBookType	3	Order Depth
1023	MDPriceLevel	2	Price Level
290	MDEntryPositionNo	1	Position at this price level
279	MDUpdateAction	2	Delete
269	MDEntryType	0	Bid
107	SecurityDesc	GEZ5	Instrument
270	MDEntryPx	9740	Price,
278	MDEntryID	213	Order Reference

Result

Top-Of-Book, BBO		Price-Depth		Order-Depth	
BID	ASK	BID	ASK	BID	
5 / 9750	9760 / 5	5 / 9750	9760 / 5	9750	ID=274, Qty=5
		14 / 9730	9770 / 3	9730	ID=223, Qty=4; ID=227, Qty=1; ID=275, Qty=9
		1 / 9720	9780 / 5	9720	ID=973, Qty=1
			9790 / 7	ASK	
				9760	ID=230, Qty=5
				9770	ID=231, Qty=3
				9780	ID=232, Qty=5
				9790	ID=233, Qty=4; ID=234, Qty=3

3.6 A Trade occurs over multiple price levels.

An order is entered to SELL 12 contracts at 9730.

It traded 5 @ 9750 then 7 @ 9730.

At 9730 there are three counterparties for quantities 4, 1 and 9.

The trade instructions do NOT update the view of the book. The trade instructions are followed by change and delete instructions to update the book.

Trade Instructions

Tag number	Tag name	Value	Description
279	MDUpdateAction	0	New
269	MDEntryType	2	Trade
271	MDEntrySize	5	Quantity
270	MDEntryPx	9750	Price
107	SecurityDesc	GEZ5	Instrument
277	TradeCondition	Last, Opening, Block, Average Price	Trade condition codes, Indicators

Tag number	Tag name	Value	Description
279	MDUpdateAction	0	New
269	MDEntryType	2	Trade
271	MDEntrySize	7	Quantity
270	MDEntryPx	9730	Price
107	SecurityDesc	GEZ5	Instrument

Update Top-Of-Book

Tag number	Tag name	Value	Description
1021	MDBookType	1	Top-Of-Book
279	MDUpdateAction	0	New
269	MDEntryType	0	Bid
271	MDEntrySize	7	Quantity
270	MDEntryPx	9730	Price
107	SecurityDesc	GEZ5	Instrument

Update Price-Depth

Tag number	Tag name	Value	Description
1021	MDBookType	2	Price Depth
1023	MDPriceLevel	1	Price Level
279	MDUpdateAction	2	Delete
269	MDEntryType	0	Bid
107	SecurityDesc	GEZ5	Instrument
270	MDEntryPx	9750	Price

Tag number	Tag name	Value	Description
1021	MDBookType	2	Price Depth
1023	MDPriceLevel	1	Price Level
279	MDUpdateAction	1	Change
269	MDEntryType	0	Bid
107	SecurityDesc	GEZ5	Instrument
271	MDEntrySize	7	Quantity
270	MDEntryPx	9730	Price

Update Order-Depth

Tag number	Tag name	Value	Description
1021	MDBookType	3	Order Depth
1023	MDPriceLevel	1	Price Level
290	MDEntryPositionNo	1	Position at this price level
279	MDUpdateAction	2	Delete
269	MDEntryType	0	Bid
107	SecurityDesc	GEZ5	Instrument
270	MDEntryPx	9750	Price,
278	MDEntryID	274	Order Reference

Tag number	Tag name	Value	Description
1021	MDBookType	3	Order Depth
1023	MDFPriceLevel	1	Price Level
290	MDEntryPositionNo	1	Position at this price level
279	MDUpdateAction	2	Delete
269	MDEntryType	0	Bid
107	SecurityDesc	GEZ5	Instrument
270	MDEntryPx	9730	Price
278	MDEntryID	223	Order Reference

Tag number	Tag name	Value	Description
1021	MDBookType	3	Order Depth
1023	MDFPriceLevel	1	Price Level
290	MDEntryPositionNo	1	Position at this price level
279	MDUpdateAction	2	Delete
269	MDEntryType	0	Bid
107	SecurityDesc	GEZ5	Instrument
270	MDEntryPx	9730	Price
278	MDEntryID	227	Order Reference

Tag number	Tag name	Value	Description
1021	MDBookType	3	Order Depth
1023	MDFPriceLevel	1	Price Level
290	MDEntryPositionNo	1	Position at this price level
279	MDUpdateAction	1	Change
269	MDEntryType	0	Bid
107	SecurityDesc	GEZ5	Instrument
270	MDEntryPx	9730	Price
271	MDEntrySize	7	Quantity
278	MDEntryID	275	Order Reference

Result

Top-Of-Book, BBO		Price-Depth		Order-Depth	
BID	ASK	BID	ASK	BID	
7 / 9730	9760 / 5	7 / 9730	9760 / 5	9730	ID=275, Qty=7
		1 / 9720	9770 / 3	9720	ID=973, Qty=1
			9780 / 5		
			9790 / 7	ASK	
				9760	ID=230, Qty=5
				9770	ID=231, Qty=3
				9780	ID=232, Qty=5
				9790	ID=233, Qty=4; ID=234, Qty=3

3.7 A new price causes the bottom price level to be deleted

If the Price-Depth and Order-Depth are only maintained to 5 price levels, then the addition of two more price levels to the example above will cause the bottom row to be deleted.

Sell 8 @ 9750, then Sell 6 @ 9740

Update Top-Of-Book for first order, Sell 8 @ 9750

Tag number	Tag name	Value	Description
1021	MDBookType	1	Top-Of-Book
279	MDUpdateAction	0	New
269	MDEntryType	1	Ask
271	MDEntrySize	8	Quantity
270	MDEntryPx	9750	Price
107	SecurityDesc	GEZ5	Instrument

Update Price-Depth

Tag number	Tag name	Value	Description
1021	MDBookType	2	Price Depth
1023	MDPriceLevel	1	Price Level
279	MDUpdateAction	0	New
269	MDEntryType	1	Ask
107	SecurityDesc	GEZ5	Instrument
271	MDEntrySize	8	Quantity
270	MDEntryPx	9750	Price

Update Order-Depth

Tag number	Tag name	Value	Description
1021	MDBookType	3	Order Depth
1023	MDPriceLevel	1	Price Level
279	MDUpdateAction	0	New
269	MDEntryType	1	Bid
107	SecurityDesc	GEZ5	Instrument, it's the key to the book
271	MDEntrySize	8	Quantity
270	MDEntryPx	9750	Price, it's the key to insert the entry
290	MDEntryPositionNo	1	Position at this price level
278	MDEntryID	296	Order Reference

Result

Top-Of-Book, BBO		Price-Depth		Order-Depth	
BID	ASK	BID	ASK	BID	
7 / 9730	9750 / 8	7 / 9730	9750 / 8	9730	ID=275, Qty=7
		1 / 9720	9760 / 5	9720	ID=973, Qty=1
			9770 / 3	ASK	
			9780 / 5	9750	ID=296, Qty=8
			9790 / 7	9760	ID=230, Qty=5
				9770	ID=231, Qty=3
				9780	ID=232, Qty=5
				9790	ID=233, Qty=4; ID=234, Qty=3

Price-Depth and Order-Depth are now at the maximum number of levels on the Ask side of the book. The addition of another price level will cause the bottom row to be deleted.

Update the Top-Of-Book for second order, 6 @ 9740

Tag number	Tag name	Value	Description
1021	MDBookType	1	Top-Of-Book
279	MDUpdateAction	0	New
269	MDEntryType	1	Ask
271	MDEntrySize	6	Quantity
270	MDEntryPx	9740	Price
107	SecurityDesc	GEZ5	Instrument

Update Price-Depth

Tag number	Tag name	Value	Description
1021	MDBookType	2	Price Depth
1023	MDPriceLevel	1	Price Level
279	MDUpdateAction	0	New
269	MDEntryType	1	Ask
107	SecurityDesc	GEZ5	Instrument
271	MDEntrySize	6	Quantity
270	MDEntryPx	9740	Price

Update Order-Depth

Tag number	Tag name	Value	Description
1021	MDBookType	3	Order Depth
1023	MDPriceLevel	1	Price Level
279	MDUpdateAction	0	New
269	MDEntryType	1	Bid
107	SecurityDesc	GEZ5	Instrument
271	MDEntrySize	6	Quantity
270	MDEntryPx	9740	Price
290	MDEntryPositionNo	1	Position at this price level
278	MDEntryID	297	Order Reference

Result

Top-Of-Book, BBO		Price-Depth		Order-Depth	
BID	ASK	BID	ASK	BID	
7 / 9730	9740 / 6	7 / 9730	9740 / 6	9730	ID=275, Qty=7
		1 / 9720	9750 / 8	9720	ID=973, Qty=1
			9760 / 5	ASK	
			9770 / 3	9740	ID=297, Qty=6
			9780 / 5	9750	ID=296, Qty=8
				9760	ID=230, Qty=5
				9770	ID=231, Qty=3
				9780	ID=232, Qty=5
				Orders ID=233, Qty=4; ID=234, Qty=3 are deleted by client.	
				9790 / 7 is deleted by client	

3.8 Use of MDEntryID for Top-Of-Book and Price-Depth

A vendor can include a the Tag MDEntryID on the Top-Of-Book and Price-Depth feeds to improve performance of client systems.

Sell 8 @ 9750

Update Top-Of-Book

Tag number	Tag name	Value	Description
1021	MDBookType	1	Top-Of-Book
279	MDUpdateAction	0	New
269	MDEntryType	1	Ask
271	MDEntrySize	8	Quantity
270	MDEntryPx	9750	Price
107	SecurityDesc	GEZ5	Instrument
297	MDEntryID	2254	Unique key for this instrument/side

Update Price-Depth

Tag number	Tag name	Value	Description
1021	MDBookType	2	Price Depth
1023	MDPriceLevel	1	Price Level
279	MDUpdateAction	0	New
269	MDEntryType	1	Ask
107	SecurityDesc	GEZ5	Instrument
271	MDEntrySize	8	Quantity
270	MDEntryPx	9750	Price
297	MDEntryID	25244	Unique Key for this instr/price/side

The size changes to 10.

Update Top-Of-Book

Tag number	Tag name	Value	Description
1021	MDBookType	1	Top-Of-Book
279	MDUpdateAction	0	New
269	MDEntryType	1	Ask
271	MDEntrySize	10	Quantity
270	MDEntryPx	9750	Price
107	SecurityDesc	GEZ5	Instrument

297	MDEntryID	2254	Unique key for this instrument/side
-----	-----------	------	-------------------------------------

Update Price-Depth

Tag number	Tag name	Value	Description
1021	MDBookType	2	Price Depth
1023	MDPriceLevel	1	Price Level
279	MDUpdateAction	0	New
269	MDEntryType	1	Ask
107	SecurityDesc	GEZ5	Instrument
271	MDEntrySize	10	Quantity
270	MDEntryPx	9750	Price
297	MDEntryID	25244	Unique Key for this instr/price/side

The Client does not need to find the commodity, price-level then side. The use of MDEntryID allows the client firm to index directly to the line to be changed.



4 Market Statistics

4.1 Market Statistics

There are a number of statistics often included in market data feeds which are related to changes in a book but which are not used to update the book. These include the Open High Low, Last trade prices. Also included are Highest Bid/ Lowest Offer and Last Best Bid/Last Best Offer.

Last Trade

The Last Trade indicates that the last trade has occurred at a given price level and provides consolidated time and sales information to a client. The Last Trade summarizes trade activity at a price level and can be used as an alternative or complement to individual trades that are generated as a result of a number of partial fills.

FIX Syntax for Last Trade

Tag number	Tag name	Value	Description
279	MDUpdateAction	0	New
269	MDEntryType	2	Trade
271	MDEntrySize	5	Quantity
270	MDEntryPx	9740	Price
107	SecurityDesc	GEZ5	Instrument
277	TradeCondition	U	ExchangeLast

The Opening Price is the first sale price for the trading session (day). The High Trade Price pertains to a trade event that has produced the highest trade price for the current session. Likewise, the Low Trade Price indicates that a trade produced the lowest trade price for a given day. In most countries a trading session equates to a 24 hour period but It might start at a different time to midnight. Some countries cease trading for lunch and have multiple trading sessions within a day but this is not common.

A vendor may choose to provide high and low prices only as a response to the event that produces them, or may also send the high and low prices on every trade.

Last Best Quote Price

The Last Best Price indicates that a new best bid or new best ask price has occurred.

In certain cases, such as an order submission immediately followed by an order cancellation, the top of the book can be bettered without resulting in a new top of book instruction being sent. This is due to the transitory nature of the event. However, Last Best Price can be used to capture this information and convey it to the market.

FIX Syntax for Last Best Price (Bid or Ask)

Tag number	Tag name	Value	Description
279	MDUpdateAction	0	New
269	MDEntryType	0 or 1	0 = Bid 1 = Ask
271	MDEntrySize	5	Quantity
270	MDEntryPx	9740	Price
107	SecurityDesc	GEZ5	Instrument
276	QuoteCondition	C	Exchange Best

High and Low Trade

The High Trade Price pertains to a trade event that has produced the highest trade price for the current session. Likewise, the Low Trade Price indicates that a trade event has produced the lowest trade price for a given session. High and Low Trade Prices are helpful in tracking market trends. They also provide historical information for the current session regarding market behavior.

A vendor may choose to provide high and low prices only as a response to the event that produces them, or may also send the high and low prices on every trade.

FIX Syntax for Session High and Low Trade Price

Tag number	Tag name	Value	Description
279	MDUpdateAction	0	New
269	MDEntryType	7 8	7 = Session High Trade 8 = Session Low Trade
270	MDEntryPx	9740	Price
107	SecurityDesc	GEZ5	Instrument
336	TradingSessionID	2	Normal

Best High Bid and Best Low Ask

The Best High Bid and Best Low Ask prices are used to indicate the highest bid and lowest ask quote prices of the session. These prices are useful in tracking market behavior. A Best High Bid and Best Low Ask may be different from the High Trade Price described above if an arriving order is given a price better than the specified limit order. An order to buy at 9751 may trade at 9750 if a resting sell order is already on the book at 9750. In this case, the Best High Bid Price would be higher than High Trade Price. A better priced order may not trade!

A vendor may choose to provide Best High Bid and Best Low Ask prices only as a response to the event that produces them, or may also send the prices on every trade.

FIX Syntax for Best High Bid and Best Low Ask

Tag number	Tag name	Value	Description
------------	----------	-------	-------------

279	MUpdateAction	0	New
269	MEntryType	N O	N = Session High Bid O = Session Low Ask
270	MEntryPx	9751	Price
107	SecurityDesc	GEZ5	Instrument
336	TradingSessionID	2	Normal

5 Snapshots and Recovery

5.1 Requesting Snapshots

The Market Data specification provides for a Client application to request a snapshot of the orderbook for selected instruments. The Client can specify the type of information and the number of levels required.

The FIX Market Data Request message is used for making either a “snapshot” or an “incremental” request in order to reset the book. The table below shows the basic request for returning a snapshot of the book and the corresponding response.

Guidelines for Requesting Snapshots and/or Updates

A market data feed may consist of both Market Data Snapshot Full Refresh messages and Market Data Incremental Refresh messages. Previously, the specification required that a feed consist of one message type or the other.

The Market Data Request message is used to request a static book snapshot or subscribe to a stream of snapshots and updates.

Market Data Snapshot Full Refresh should be used to provide a snapshot of the market when Snapshot is requested using SubscriptionRequestType (263). Use of Market Data Incremental Refresh is being discouraged for this purpose.

Market Data Snapshot Full Refresh will be used to provide initial snapshot when Snapshot + Updates are requested using SubscriptionRequestType (263)

The Market Data Request scenarios that will be supported are as follows:

Market Data Request - Request Type	SubscriptionRequest Type Tag 263	MDUpdateType Tag 265	Messages Received by Subscriber
User requests state of the book and receives one and only one snapshot for each request	0=Snapshot	Not Provided	Full Refresh
User requests state of the book + updates and specifies that only Full Refresh Message is used	1=Snapshot+Updates	0=Full Refresh	Full Refresh only
User requests state of the book + updates and specifies that updates are to be sent using Incremental Refresh Message	1=Snapshot+Updates	1=Incremental Refresh	Full Refresh + Incremental Refresh

Snapshot Request Format

Example of a Market Data Snapshot Request:

Tag number	Tag name	Value	Description
Header		V	Snapshot request
263	SubscriptionRequest Type	0	0 = Snapshot requested
265	MDUpdateType	1	1 = Full Refresh
266	AggregatedBook	Y	For Aggregate / Order-Depth
264	MarketDepth	3	Number of price levels
262	MDReqID	123	My request ID
146	NoRelatedSym	1	
107	SecurityDesc	IBM	Security name

The tags MarketDepth and Aggregate are used to specify the type of information and depth of information in the request. The Vendor must specify the choices that are available to the Client application.

When choices are available then the following settings are used:

Orderbook type	Tag values	Comment
Top-Of-Book	Aggregate=Y MarketDepth=1	
Price-Depth, n levels	Aggregate=Y MarketDepth=n	When ths is the only choice then also add MarketDepth=0
Price-Depth, all levels	Aggregate=Y MarketDepth=0	
Order-Depth, n levels	Aggregate=N MarketDepth=n	When this is the only choice then also add MarketDepth=0
Order-Depth, all levels	Aggregate=N MarketDepth=0	

If the vendor offers 5 level Depth and the client requests 3 levels, the Vendor should provide 3 levels. If the Vendor offers three and the client requests 5 then the vendor should provide 3 levels. A request for MarketDepth=0 will get the deepest view available. The vendor should confirm the outcome when the requested level of depth is not available.

Multiple instruments per request can be made. The Client must list the instruments for which the data is requested. Some vendors may allow wildcards to be specified or place limits on the maximum number of instruments that can be listed in the snapshot request.

Snapshot response

The response to a Snapshot is a list of Snapshot items. The FIX Market Data Snapshot Full Refresh message will be sent in response. The content of each message is determined by the type of data requested.

5.2 A Market Data Snapshot for Top-Of-Book

The Top-Of-Book Snapshot contains two items for best bid and best offer. Within each item are the quantity and price. Note that there is no need for price level.

The following table is an example of a Top-Of-Book Snapshot using the FIX Market Data Snapshot Full Refresh message (35=W).

Tag number	Tag name	Value	Description
Header		W	Snapshot Full Refresh
262	MDReqID	123	My request ID
107	SecurityDesc	IBM	Security name
1021	MDBookType	1	Top-Of-Book
	NoMDEntries	2	Number of repeating entries
269	MDEntryType	0	A Buy Side Snapshot
271	MDEntrySize	3	Quantity
270	MDEntryPx	9730	Price
269	MDEntryType	1	A Sell Side Snapshot
271	MDEntrySize	5	Quantity
270	MDEntryPx	9750	Price
	Trailer		

5.3 A Market Data Snapshot for Price-Depth:

The Security description appears once at the head of the snapshot.

MDBookType describes the type of information contained within the snapshot. It could be Top-Of-Book, Price-Depth or Order-Depth and describes the type of data included in each item.

This is followed by a number of items then the list of items. Each item begins with the MDEntryType tag and describes one side of a price level including Price and total quantity at that price. The requester is expected to order the book using bid or ask and the price.

In a Multicast environment, MDReqID would be sent back on a separate Market Data Request Reject message only if there was a problem with the request. Otherwise, it should not be provided. If provided on the data, then competitors could determine who was requesting and take advantage of the situation. The Client must know that the snapshot relates to its request from the security code.

The following table is an example of a Price-Depth snapshot using the FIX Market Data Snapshot Full Refresh message (35=W)

Tag number	Tag name	Value	Description
Header		W	Snapshot Full Refresh
262	MReqID	123	My request ID
107	SecurityDesc	IBM	Security name
1021	MDBookType	2	Price-Depth
	NoMDEntries	5	Number of repeating entries
269	MDEntryType	0	A Buy Side Snapshot
271	MDEntrySize	3	Quantity
270	MDEntryPx	9730	Price
1023	<i>PriceLevel</i>	1	<i>RowNumber</i>
269	MDEntryType	0	A Buy Side Snapshot
271	MDEntrySize	10	Quantity
270	MDEntryPx	9720	Price
1023	<i>PriceLevel</i>	2	<i>RowNumber</i>
269	MDEntryType	0	A Buy Side Snapshot
271	MDEntrySize	15	Quantity
270	MDEntryPx	9710	Price
1023	<i>PriceLevel</i>	3	<i>RowNumber</i>
269	MDEntryType	0	A Buy Side Snapshot
271	MDEntrySize	5	Quantity
270	MDEntryPx	9700	Price
1023	<i>PriceLevel</i>	4	<i>RowNumber</i>
269	MDEntryType	1	A Sell Side Snapshot
271	MDEntrySize	25	Quantity
270	MDEntryPx	9760	Price
1023	<i>PriceLevel</i>	1	<i>RowNumber</i>
	Trailer		

5.4 A Market Data Snapshot for Order-Depth

The Order-Depth snapshot contains one item for each order. The data includes the side, quantity and price of each order along with the MDEntryID. The MDEntryID is generated by the vendor to uniquely identify this order. It is NOT the ClOrdID of the order that was entered by the owner of the order. The Client can assume that the sequence in the response is the sequence in the book.

The first order at each price level also has an optional PriceLevel tag. This is used in situations where there could be confusion in the ranking of price levels. The following table is an example of an Order-Depth snapshot using the FIX Market Data Snapshot Full Refresh message (35=W)

Tag number	Tag name	Value	Description
Header		W	Snapshot Full Refresh
262	MDEntryID	123	My request ID
107	SecurityDesc	IBM	Security name
TBD	MDEntryType	3	Order-Depth
268	NoMDEntries	9	Number of Orders
269	MDEntryType	0	A Buy Side Order
271	MDEntrySize	2	Quantity Order 1
37	MDEntryID	123	
270	MDEntryPx	9730	Price
290	<i>MDEntryPositionNo</i>	1	<i>Position at PriceLevel</i>
1023	<i>PriceLevel</i>	1	<i>RowNumber</i>
269	MDEntryType	0	A Buy Side Snapshot
271	MDEntrySize	1	Quantity – Order 2
37	MDEntryID	124	
270	MDEntryPx	9730	Price
290	<i>MDEntryPositionNo</i>	2	<i>Position at PriceLevel</i>
1023	<i>PriceLevel</i>	1	<i>RowNumber</i>
269	MDEntryType	0	A Buy Side Order
271	MDEntrySize	6	Quantity Order 3
37	MDEntryID	125	
270	MDEntryPx	9720	Price
290	<i>MDEntryPositionNo</i>	1	<i>Position at PriceLevel</i>
1023	<i>PriceLevel</i>	2	<i>RowNumber</i>
269	MDEntryType	0	269
271	MDEntrySize	4	Quantity Order 3
37	MDEntryID	126	

270	MDEntryPx	9720	Price
290	<i>MDEntryPositionNo</i>	2	<i>Position at PriceLevel</i>
1023	<i>PriceLevel</i>	2	<i>RowNumber</i>
269	MDEntryType	0	A Buy Side Order
271	MDEntrySize	1	Quantity
37	MDEntryID	127	
270	MDEntryPx	9710	Price
290	<i>MDEntryPositionNo</i>	1	<i>Position at PriceLevel</i>
1023	<i>PriceLevel</i>	3	<i>RowNumber</i>
269	MDEntryType	0	269
271	MDEntrySize	5	Quantity
37	MDEntryID	128	
270	MDEntryPx	9700	Price
290	<i>MDEntryPositionNo</i>	1	<i>Position at PriceLevel</i>
1023	<i>PriceLevel</i>	4	<i>RowNumber</i>
269	MDEntryType	0	269
271	MDEntrySize	10	Quantity
37	MDEntryID	131	
270	MDEntryPx	9690	Price
290	<i>MDEntryPositionNo</i>	1	<i>Position at PriceLevel</i>
1023	<i>PriceLevel</i>	5	<i>RowNumber</i>
269	MDEntryType	0	269
271	MDEntrySize	3	Quantity
37	MDEntryID	132	
270	MDEntryPx	9680	Price
290	<i>MDEntryPositionNo</i>	1	<i>Position at PriceLevel</i>
1023	<i>PriceLevel</i>	5	<i>RowNumber</i>
	Trailer		

5.5 Start Of Day Snapshot

The trading day normally begins with a snapshot of each product. The snapshot provides client systems with all product codes and provides the initial book in markets that support long orders. Long Orders are orders that stay in the book for more than one day, also known as Good-Till-Cancelled or GTC orders.

The snapshot also verifies that the network is working to its full capacity. Markets normally accept orders prior to the opening of the market. Once the snapshot is complete, the vendor can begin to update the book with Pre-open orders, leading up to the opening of the market.

5.6 Issues with Snapshot Functionality

The snapshot function allows a Client to request the orderbook for a single instrument or a list of instruments. This function is ideally suited to the workstation in a trading systems environment where it is moving between instruments, requesting a snapshot, then updating that book with the broadcasts. When the trader moves to the next security the workstation will suspend the broadcast of the first instrument, request a snapshot and broadcasts for the next security.

Vendors must be cautious with the use of snapshots to reconnect clients when using multicast networks. During market hours, the time to provide a snapshot of the entire market could be considerable, depending upon the available bandwidth.

There may also be synchronization issues between the snapshot responses and the data feed. Are the snapshots in the correct position in relation to the data feed updates? Clients may be required to cache real time messages while waiting for a snapshot. When the snapshot arrives clients may be expected to apply all cached messages to snapshot in order to create a current book.

Another possible solution is that the Client system subscribe for broadcasts to a limited number of instruments, then request snapshots for a limited number of instruments and apply the updates until the data feed has caught up. Then repeat this process until all instruments have been loaded.

5.7 Synchronization of Snapshots and Data Feed

These are the approaches to the synchronization of snapshots that are sent through a broadcast mechanism and updates that are being disseminated.

Synchronous Snapshots. Some Trading systems will send the snapshot request to the orderbook server and the response will be inserted into the data feed. The network must be designed to assure that the response and subsequent updates are delivered in the correct sequence to the requestor. This approach places a significant burden upon the network design to assure that every update and snapshot are delivered in the correct sequence to all clients. With this solution, clients should be able to rely on snapshots being properly sequenced within the data feed. Synchronous snapshots may be supported in either a broadcast or point-to-point feed.

Asynchronous Snapshots. Some vendors provide out-of-band snapshots which the client must synchronize with the real-time data feed. Out-of-band snapshots can either be provided as an unsolicited stream which is subscribed to by the client or be provided in response to a client request as described in Section 4.1 above. In both cases, snapshots will carry the sequence number of the last real-time message that was applied and clients will be expected to synchronize the snapshot based on this information. Sequence numbers are described in greater detail below.

Sequence Numbers. Some systems provide a sequence number on the snapshot. This sequence number represents the last message in the feed that is reflected in the snapshot. The client system must already be receiving the data feed when it requests a snapshot. When the Client system receives the snapshot it must go back and apply those messages with sequence numbers greater than the sequence number on the snapshot. These messages must be applied to the snapshot in order to ensure that it reflects the latest state of the book. This approach moves the responsibility to the Client system to manage synchronization issues.

Subscribe first and apply all updates. Some systems manage to synchronize snapshots and updates without sequence numbers. The client must subscribe to the updates for an instrument, store those updates while the snapshot is requested. Once the snapshot is received then the updates are applied. Some updates will have been received prior to the snapshot but by applying all updates, the view is correct by the time you get to the last update. This method works well for Top-Of-Book but the vendor must specify synchronization rules for Price-Depth and Order-Depth. For example, a Delete of an order that does not exist is to be ignored. A change of an order that does not exist is to be created. The insertion of an order that is already in the book is to be ignored.

5.8 Retransmission Capability

A typical configuration for data feeds is to broadcast the data on a multicast address. This is a cost effective configuration as the vendor does not to manage each client connection. An issue to be considered is the recovery of missing data, whether it is due to network errors or system problems at the client site.

One solution is where client systems connect to the vendor with a separate two-way line. The Client can request retransmissions of missing messages and have those missing messages retransmitted over the multicast high-speed data line.

à	62	TradingSessionSubID	N	
Tag	Field Name	Req'd		Comments
815	StandardHeader	N		MsgType take if application level queuing exists
862	AppQueueMax	N		Must have application level of previous Market Data Request for disabling operation send data if SubscriptionRequestType = Disable previous
TBD	StartTime	N		SubscriptionRequestType = Disable previous if Snapshot + Updates Request (2) or Subscription Resend Request (3), respectively.
TBD	SubscriptionRequestType	N Y		SubscriptionRequestType indicates to the other party what type of response is expected. A Snapshot request only asks for current information. A subscribe request asks for updates as the status changes. Unsubscribe will cancel any future update messages from the Subscription Resend Request (3)
TBD	StartRptSeq	N		Can be used to specify the start of a reporting sequence range if SubscriptionRequestType = Resend (3)
TBD	EndRptSeq	N		Can be used to specify the end of a reporting sequence range if SubscriptionRequestType = Resend (3)
264	MarketDepth	Y		Can be used to specify the end of a reporting sequence range if SubscriptionRequestType = Resend (3)
265	MDUpdateType	N		Used to qualify the type of feed being requested
1022	MDFeedType	N		Used to qualify the type of feed being requested
266	AggregatedBook	N		for resend. For example, the category (index), can be used to clarify our requests. Defined here
286	OpenCloseSettlFlag	N		Used with MDFeedType to identify the specific Settlement Price (e) channel 10, the index type, defines the scope (s) of the request
TBD	MDRequestInterval	N		Indicates the interval (in seconds) in which updates are to be delivered to receive
546	Scope	N		MDUpdateType = Incremental Refresh(1).
547	MDImplicitDelete	N		
	Standard Trailer			
267	NoMDEntryTypes	Y		Number of MDEntryType fields requested.
à	269	MDEntryType	Y	Must be the first field in this repeating group. This is a list of all the types of Market Data Entries that the firm requesting the Market Data is interested in receiving.
146	NoRelatedSym	Y		Number of symbols (instruments) requested.
à	component block <Instrument>		Y	Insert here the set of "Instrument" (symbology) fields defined in "COMMON COMPONENTS OF APPLICATION MESSAGES"
à	711	NoUnderlyings	N	Number of underlyings
à	à	component block <UnderlyingInstrument>	N	Must be provided if Number of underlyings > 0
à	555	NoLegs	N	Required for multileg quotes
à	à	component block <InstrumentLeg>	N	Required for multileg quotes For Swaps one leg is Buy and other leg is Sell
386	NoTradingSessions	N		Number of trading sessions for which the request is valid.
à	336	TradingSessionID	N	

Market Data Request for Application-level Resends

The FIX Market Data Request message can be used to request missed messages in a number of ways. Market Data recipients may send one or more Market Data Request messages on reconnection with SubscriptionRequestType = "Subscription Resend Request" (3).

The Market Data sender might honor these requests immediately, might send current data and resend old data when bandwidth allows, or might choose to reject the request with a Market Data Request Reject message. When sending old Market Data to fill such gaps, the sending application should set PossResend = Y on these messages to differentiate between old and new data. Recipients of Market Data may use the fields StartTime, EndTime, xxx, and yyy to specify the beginning and/or the end of the range of data they wish resent. The Market Data Request message format is shown below and allows the following types of resend requests:

Time Range – specify a valid StartTime (tag TBD) to a valid EndTime (tag TBD)

Message Sequence – specify a valid StartRptSeq (tag TBD) to a valid EndRptSeq (tag TBD)

Feed Type – optionally used to qualify a resend request for a specific feed type

5.9 Market Data Request

Example of a Market Data Resend Request by Time Interval

Tag number	Tag name	Value	Description
Header		V	Market Data Request
263	SubscriptionRequest Type	0	3 = Resend Request
265	MDUpdateType	1	1 = Full Refresh
266	AggregatedBook	Y	For Aggregate / Order-Depth
264	MarketDepth	3	Number of price levels
262	MDReqID	123	My request ID
TBD	StartTime	08:00:00	Start time for which data is to be resent
TBD	EndTime	08:10:00	End time for which data is to be resent

Example of a Market Data Resend Request by Message Sequence Number and Feed Type

Tag number	Tag name	Value	Description
Header		V	Market Data Request
263	SubscriptionRequest Type	0	3 = Resend Request
265	MDUpdateType	1	1 = Full Refresh
266	AggregatedBook	Y	For Aggregate / Order-Depth
264	MarketDepth	3	Number of price levels
262	MDReqID	123	My request ID
TBD	StartRptSeq	100000	Starting message sequence number for which data is to be resent
TBD	EndRptSeq	105000	Ending message sequence number for which data is to be resent
1022	MDFeedType	Channel	Feed type descriptor qualifying feed
TBD	MDFeedTypeCode	10	Feed type code specifying the feed type

5.10 Considerations for Data Retransmissions

There are practical issues to be considered when providing the ability for Client systems to automatically request retransmissions. Depending on the amount of data and the available capacity on the network, it may not be practical for recipients to recover large amounts of data through the multicast network.

If a client should connect after the opening and request a retransmission of all missed data, it could consume all available capacity on the multicast network for the remainder of the trading session. A second client attempting to connect may not be able to get its recovery data for an extended period. For this concept to work there must be a significant amount of available bandwidth for retransmission on the multicast network.

In many cases it should be possible for a client application to connect to a data feed and start processing from that point. The following suggestions are offered to assist vendors in providing some recovery mechanisms within the data feed.

Top-Of-Book: the next instruction will overwrite the previous prices. A regular refresh of all market prices can be included in the data stream.

Price-Depth: The inclusion of the PriceLevel tag will allow the client to correct the view as the price is changed. Upon receipt of a new PriceLevel of 1, the Client should delete any existing rows at the same or better price and accept the new row as the best row.

Order-Depth: It is difficult to recover a missed block for Order-Depth. The vendor should attempt to recover the book or at least provide some mechanism for the client to determine that the orderbook is no longer consistent. Some suggestions:

- Use PriceLevel And PositionNumber. The client is to delete rows above row 1.
- Regularly refresh orderbooks as part of the broadcast stream
- Add a tag containing total number or some checksum of orders at each price level so the Client can detect orderbooks that contain errors.
- Add a sequence number for each orderbook so that missing messages can be traced to individual securities.

5.11 Continuous Refresh

One recovery method is to send continuous snapshot messages for all symbols. This solution is particularly suited to high volume data feeds. The benefit is that recipients can connect to the data feed and within a relatively short time have a complete picture of the market. Missed data can be obtained through other channels.