ACTIVETICK MARKETIF

MULTICAST INTERFACE SPECIFICATION

REVISION 2.0

1 Contents

1.0 Introduction

1.1 Background

ActiveTick Market Interface (MarketIf) is a software service that offers normalization of exchange's data into a proprietary messaging format. Normalized data produced by MarketIf standardizes data across all supported exchanges into a single unified format. This feed specification outlines the format of output data produced by MarketIfs.

In addition to normalization functionality, MarketIf also provides efficient transport of data feed from point A to point B using proprietary real-time compression techniques without any added latency.

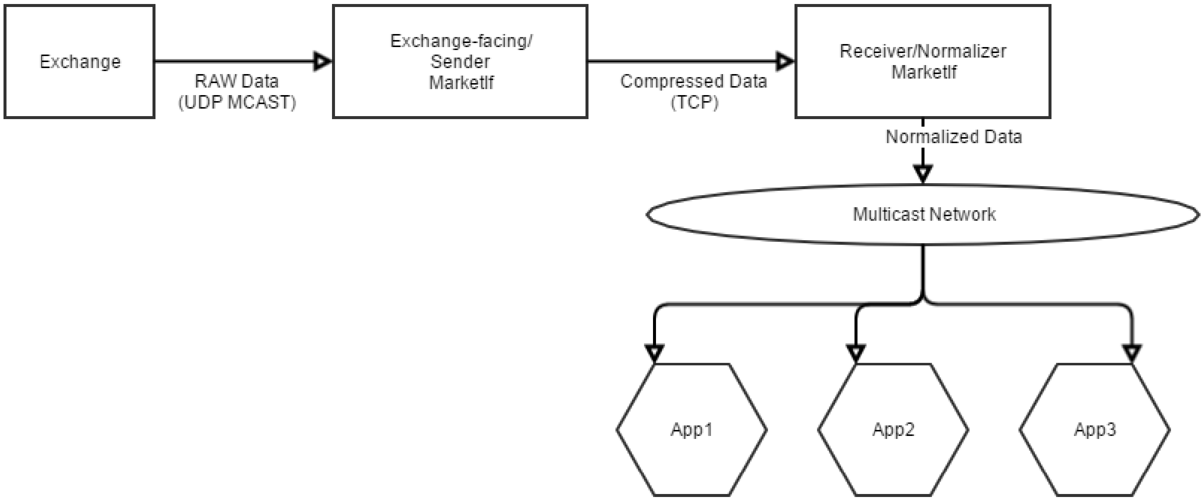
MarketIf can function in three modes, as exchange-facing/remote sender, as remote receiver/normalizer, or as exchange-facing/normalizer.

1.2 Exchange-facing/Sender Mode

Exchange-facing side of MarketIf receives original RAW data from exchange, typically by subscribing to exchange's multicast channels, and consumes the data into internal format. After each message is consumed, the data is compressed and transported to a remote or local MarketIf running in Receiver/Normalizer mode.

1.3 Receiver/Normalizer Mode

MarketIf configured as Receiver/Normalizer accepts data forwarded to it by MarketIf running as Exchange-facing/Sender. Forwarded data received by MarketIf gets uncompressed, validated, and then normalized into standard ActiveTick's proprietary messaging format. After normalization, the data gets published onto predefined set of multicast channels, from where it can be picked up by multicast subscribers.

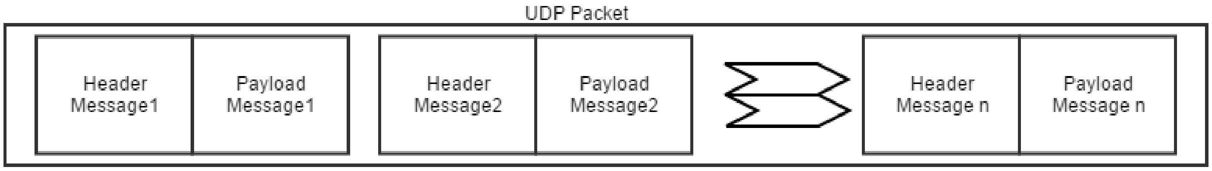


2.0 Transmission Characteristics

2.1 Transmission Block

Each normalized UDP packet send by MarketIf contains one or more messages. Messages are stacked next to each other within the packet, up to maximum allowed payload size of 1420 bytes.

Each individual message always contains a message header, followed by a message payload.



2.2 Data Endianness

All normalized data is sent with Little Endian byte order.

3.0 Message Header

Message header is placed in front of the message, and contains information about a message payload that follows it. Message header is used as a lower-level transmission header. The size of the header is 7 bytes, and consists of the following data fields:

Field Name	Type	Length (bytes)
Message Id	UInt8	1
Message Size	UInt16	2
Sequence Number	UInt32	4

3.1 Message Id

The Message Id identifies type of message that follows the message header, and consists of the following IDs:

Message Id	Value
Feed Message Top BBO Quote	0
Feed Message Trade	1
Feed Message Refresh	2
Feed Message Refresh Instrument Definition	3
Feed Message Refresh Instrument Definition Continuation	4
Feed Message Fundamental	6
Feed Message Volume	7
Feed Message Book Insert Quote	51
Feed Message Book Update Quote	52
Feed Message Book Delete Quote	53
Feed Message Book Delete Quote Range	54
Feed Message Book Trade Execution	55
Feed Message Book Reset	56
Feed Message Book Order Add	57
Feed Message Book Order Fill	58
Feed Message Book Order Cancel	59
Feed Message Book Order Delete	60
Feed Message Book Order Replace	61
Feed Message Book Order Break	62
Feed Message Long Top BBO Quote	100
Feed Message Long Trade	101
Feed Message Long Refresh	102
Feed Message Long Refresh Instrument Definition	103
Feed Message Long Refresh Instrument Definition Continuation	104

Feed Message Long Fundamental	105
Feed Message Long Volume	106
Feed Message Long Book Insert Quote	150
Feed Message Long Book Update Quote	151
Feed Message Long Book Delete Quote	152
Feed Message Long Book Delete Quote Range	153
Feed Message Long Book Trade Execution	154
Feed Message Long Book Reset	155

3.2 Message Size

The message size field indicates the size of the payload that follows the message header, excluding size of the message header. For example, if payload is 10 bytes, message size will be set to 10.

3.3 Sequence Number

The sequence number incrementally increases with each message sent by MarketIf. The number gets reset to 1 when it reaches 0xffffffff. A special value of 0 indicates that all consumers should reset their sequence numbers to 1.

4.0 Feed Messages

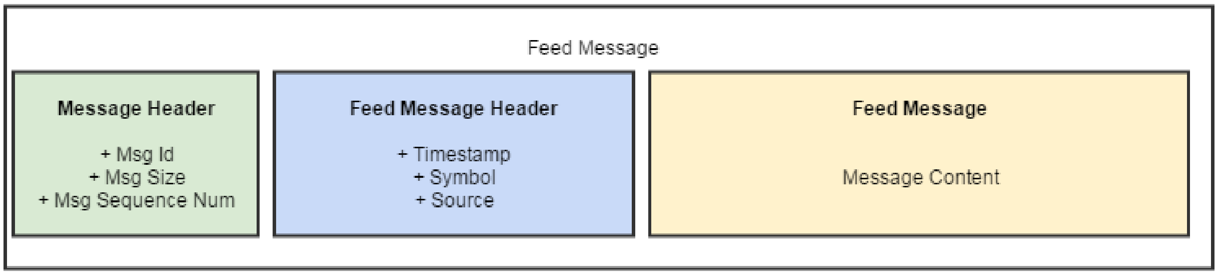
All feed messages generated by MarketIf contain a higher-level feed message header, which follows right after lower-level transmission message header. This header contains common data fields which are present in all data feed messages. In the context of a message, combined feed message header and message content is what constitutes a single message payload.

There are two versions of a feed message header, one that uses standard length symbol and another that uses a long length symbol. In order to efficiently manage bandwidth, MarketIf determines whether a message needs standard or long versions of the feed message header. Message content that follows feed message header is always the same length.

Consumers should always process both types of feed message headers.

4.1 Feed Message Header

There are two versions of feed message headers, short and long version. The short version of the feed message header is used for messages with symbol size up to 11 characters in length, while the long version is used with symbol sizes up to 32 characters.



Standard feed message header length is 23 bytes, whereas long feed message header length is 44 bytes. The header consists of the following data fields:

Field Name	Type	Length (bytes)
Timestamp	DATETIME	8
Symbol	SYMBOL or LONG SYMBOL	14 or 35
Source	UInt8	1

4.1.1 Timestamp

Timestamp is a 64-bit unsigned integer that represents time in nanoseconds since epoch, i.e. 00:00 Jan 1 1970. The time is represented in UTC. For example, timestamp value of 1513204919123456000 would be: 2017-12-13 22:41:59.123456000.

4.1.2 Symbol

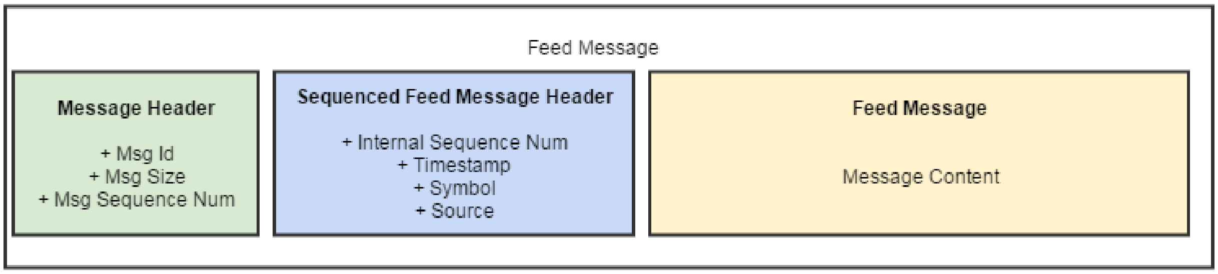
Symbol is used to identify the instrument for which the message is intended for. Based on length of the symbol, there are two versions of the symbol, standard and long. For detailed information, see SYMBOL under Field Descriptions.

4.1.3 Source

Source field identifies which MarketIf generated the message. For detailed information, see SOURCE under Field Descriptions.

4.2 Sequenced Message Header

Sequenced feed message headers are used when there is a strict requirement for feed message related sequencing, such as book messages.



The header contains a total of 31 or 52 bytes, and consists of the following data fields:

Field Name	Type	Length (bytes)
Sequence Number	UInt64	8
Timestamp	DATETIME	8
Symbol	SYMBOL or LONG SYMBOL	14 or 35
Source	UInt8	1

4.3 Feed Message Top BBO Quote

Message Id	Value
Feed Message Top BBO Quote	0

Top of the book best bid/offer quote message is sent whenever there is a change in BBO pricing information, contains a total of 52 or 73 bytes, and consists of the following data fields:

Field Name	Type	Length (bytes)
Header	SHORT or LONG HEADER	23 or 44
Condition	UInt8	1
Bid Exchange	UInt8	1
Ask Exchange	UInt8	1
Bid Price	PRICE	5
Ask Price	PRICE	5
Bid Size	UInt32	4
Ask Size	UInt32	4
Reserved Field	UInt8	8

4.4 Feed Message Trade

Message Id	Value
Feed Message Trade	1

Last sale message is sent when there is a sale for a given security, contains a total of 54 or 75 bytes, and consists of the following data fields:

Field Name	Type	Length (bytes)
Header	SHORT or LONG HEADER	23 or 44
Flags	UInt32	4
Condition1	UInt8	1
Condition2	UInt8	1
Condition3	UInt8	1
Condition4	UInt8	1
Last Exchange	UInt8	1
Last Price	PRICE	5
Last Size	UInt32	4
Reserved Field	UInt8	13

4.5 Feed Message Volume

Message Id	Value
Feed Message Volume	7

Volume message is sent when a trade happens outside normal trading conditions. This type of message contains last traded size and flags.

Field Name	Type	Length (bytes)
Header	SHORT or LONG HEADER	23 or 44
Volume Flags	UInt32	4
Last Size	UInt32	4

4.6 Feed Message Refresh

Message Id	Value
Feed Message Refresh	2

Refresh message is sent during a special event at the exchange, or within ActiveTick's Ticker Plant processing loop. Refresh messages should use Message Header's size to indicate the length of the feed message.

There are several refresh messages, including:

Refresh Message Type	Value
Refresh Message Intraday	0
Refresh Message Option End of Day	1
Refresh Message Option Open Interest	2
Refresh Message Exchange Session State	3
Refresh Message Exchange Session Data	4
Refresh Message Equity Split	5

4.6.1 Refresh Exchange Session State

Refresh Message Type	Value
Refresh Message Exchange Session State	3

Session state is generated by MarketIf whenever exchange state change event occurs for the whole market, or for individual instrument. The state change typically occurs whenever a session is opened or closed, but can also occur intraday, whenever a new instrument begins trading on the exchange. In the latter case, MarketIf sends out a new instrument definition, followed by Session Data refresh message, followed by Session State message.

If symbol is empty for this message, then message is applied to all symbols based on message's source. Otherwise, the session state message is applicable only to the symbol provided by this message.

The message contains a total of 3 bytes, and consists of the following data fields:

Field Name	Type	Length (bytes)
Refresh Message Type	UInt8	1
Session Type	UInt8	1
Session State	UInt8	1

4.6.2 Refresh Exchange Session Data

Refresh Message Type	Value
Refresh Message Session Data	4

Exchange Session Data message is typically sent at the beginning of each exchange session to indicate information pertaining to current session. This message is also generated for a new instrument that begins trading on the exchange for the first time.

Session Data message can contain up to 10 different session events. For example, a typical equity instrument with extended hours trading would contain pre-market event, regular market event, and after-hours event, for a total of 3 events.

If symbol is empty for this message, then message is applied to all symbols based on message's source. Otherwise, the session state message is applicable only to the symbol provided by this message.

The message contains a total of 192 bytes, and consists of the header and 10 session events following the header:

Field Name	Type	Length (bytes)
Refresh Message Type	UInt8	1
Events Count	UInt8	1

Session event data contains the following fields:

Field Name	Type	Length (bytes)
Session Type	UInt8	1
Begin Time	DATETIME	8
End Time	DATETIME	8
Begin Session State	UInt8	1
End Session State	UInt8	1

4.6.3 Refresh Intraday Data

Refresh Message Type	Value
Refresh Message Intraday	0

Refresh Intraday message contains data fields which have to be used to overwrite any stored value. This message is used to roll previous day's closing price. Flags field is used in a bitwise AND operation to check for existence of a Refresh Intraday Flag, and if it set, appropriate value should be used from the data structure. The message contains a total of 87 bytes, and consists of the following fields:

Field Name	Type	Length (bytes)
Refresh Message Type	UInt8	1
Flags	UInt32	4
Last Trade Condition1	UInt8	1
Last Trade Condition2	UInt8	1
Last Trade Condition3	UInt8	1
Last Trade Condition4	UInt8	1
Quote Condition	UInt8	1
Pre Market Open Price	PRICE	5
Open Price	PRICE	5
Last Price	PRICE	5
High Price	PRICE	5
Low Price	PRICE	5
Close Price	PRICE	5
Previous Close Price	PRICE	5
After Market Close Price	PRICE	5
Bid Price	PRICE	5
Ask Price	PRICE	5
Last Exchange	UInt8	1
Bid Exchange	UInt8	1
Ask Exchange	UInt8	1
Bid Size	UInt32	4
Ask Size	UInt32	4
Last Size	UInt32	4
Volume	UInt64	8
Reserved Field	UInt8	4

4.6.4 Refresh Option End Of Day

Refresh Message Type	Value
Refresh Message End Of Day	1

Open EOD message is sent to summarize today's session information. The message contains a total of 41 bytes, and consists of the following fields:

Field Name	Type	Length (bytes)
Refresh Message Type	UInt8	1
Volume	UInt32	4
Open Interest Volume	UInt32	1
Open Price	PRICE	1
High Price	PRICE	1
Low Price	PRICE	1
Last Price	PRICE	1
Underlying Price	PRICE	5
Bid Price	PRICE	5
Ask Price	PRICE	5
Reserved Field	UInt8	16

4.6.5 Refresh Option Open Interest

Refresh Message Type	Value
Refresh Message Open Interest	2

Option Open Interest message is sent to update open interest. The message contains a total of 21 bytes, and consists of the following fields:

Field Name	Type	Length (bytes)
Refresh Message Type	UInt8	1
Open Interest Volume	UInt32	4
Reserved Field	UInt8	16

4.6.6 Refresh Equity Split

Refresh Message Type	Value
Refresh Message Equity Split	5

Equity Split message is sent when there is a split within equity. The message contains a total of 58 bytes, and consists of the following fields:

Field Name	Type	Length (bytes)
Refresh Message Type	UInt8	1
Previous Amount	UInt64	8
New Amount	UInt64	8
Announce Date	UInt64	8
Pay Date	UInt64	8
Ex Date	UInt64	8

Is Optionable	UInt8	1
Reserved Field	UInt8	16

4.7 Feed Message Refresh Instrument Definition

Message Id	Value
Feed Message Refresh Instrument Definition	3

Instrument Definition message is sent whenever there is a new instrument added at the exchange level, or is modified or deleted. In case if message is larger than maximum allowed UDP packet, the message is followed by an Instrument Definition Continuation message. The message content is unique for different types of instruments; currently MarketIf definitions support the following instruments:

Instrument Type	Value
Equity Instrument Type	0
Option Equity Instrument Type	1
Future Instrument Type	2
Future Option Instrument Type	3
Future Spread Instrument Type	4

4.7.1 Equity Instrument Definition

The message contains a total of 332 or 353 bytes, and consists of the following fields:

Field Name	Type	Length (bytes)
Header	SHORT or LONG HEADER	23 or 44
Instrument Type	UInt8	1
Instrument Definition Action Type	UInt8	1
Flags	UInt32	4
Primary Exchange	UInt8	1
Short Description	UInt8	50
Long Description	UInt8	100
Sector	UInt8	50
Industry	UInt8	50
SIC	UInt8	10
CIK	UInt8	10
Reserved Field	UInt8	32

4.7.2 Option Equity Instrument Definition

The message contains a total of 102 or 123 bytes, and consists of the following fields:

Field Name	Type	Length (bytes)
Header	SHORT or LONG HEADER	23 or 44
Instrument Type	UInt8	1
Instrument Definition Action Type	UInt8	1
Underlying Symbol	LONG_SYMBOL	35
Expiration Date	UInt32	4
Strike Price	PRICE	5
Is Call	UInt8	1
Reserved Field	UInt8	32

4.7.3 Future Instrument Definition

The message contains a total of 161 or 182 bytes along with variable appendages, and consists of the following fields:

Field Name	Type	Length (bytes)
Header	SHORT or LONG HEADER	23 or 44
Instrument Type	UInt8	1
Instrument Definition Action Type	UInt8	1
Market Segment Id	UInt8	1
Security Id	UInt32	4
Security Id Source	UInt8	1
Maturity Year	UInt16	2
Maturity Month	UInt8	1
Maturity Day	UInt8	1
Security Group	UInt8	6
Asset	UInt8	6
Security Type	UInt8	6
CFI Code	UInt8	6
Underlying Product	UInt8	1
Security Exchange	UInt8	1
Security Trading Status	UInt8	1
Currency	UInt8	3
Is User Defined Instrument	UInt8	1
Match Algorithm	UInt8	1
Minimum Trading Volume	UInt32	4

Maximum Trading Volume	UInt32	4
Minimum Price Increment	PRICE	5
Minimum Price Increment Amount	PRICE	5
Display Factor	UInt64	8
Main Fraction	UInt8	1
Sub Fraction	UInt8	1
Price Display Format	UInt8	1
Contract Multiplier Unit	UInt8	1
Flow Schedule Type	UInt8	1
Contract Multiplier	UInt32	4
Unit Of Measure	UInt8	30
Unit Of Measure Quantity	PRICE	5
Decay Quantity	UInt32	4
Decay Start Date	UInt32	4
Original Contract Size	UInt32	4
Reserved Field	UInt8	8
Number Of Event Types	UInt8	1
Number Of Feed Types	UInt8	1
Number Of Instrument Attributes	UInt8	1
Number of Lot Type Rules	UInt8	1

4.7.4 Future Option Instrument Definition

The message contains a total of 160 or 181 bytes along with variable appendages, and consists of the following fields:

Field Name	Type	Length (bytes)
Header	SHORT or LONG HEADER	23 or 44
Instrument Type	UInt8	1
Instrument Definition Action Type	UInt8	1
Market Segment Id	UInt8	1
Security Id	UInt32	4
Security Id Source	UInt8	1
Maturity Year	UInt16	2
Maturity Month	UInt8	1
Maturity Day	UInt8	1
Security Group	UInt8	6
Asset	UInt8	6
Security Type	UInt8	6
CFI Code	UInt8	6
Underlying Product	UInt8	1
Security Exchange	UInt8	1

Security Trading Status	UInt8	1
Strike Price	PRICE	5
Min Cab Price	PRICE	5
Strike Currency	UInt8	3
Currency	UInt8	3
Settle Currency	UInt8	3
Match Algorithm	UInt8	1
Minimum Trading Volume	UInt32	4
Maximum Trading Volume	UInt32	4
Minimum Price Increment	PRICE	5
Minimum Price Increment Amount	PRICE	5
Display Factor	UInt64	8
Tick Rule	UInt8	1
Main Fraction	UInt8	1
Sub Fraction	UInt8	1
Price Display Format	UInt8	1
Unit Of Measure	UInt8	30
Unit Of Measure Quantity	PRICE	5
Reserved Field	UInt8	8
Number Of Event Types	UInt8	1
Number Of Feed Types	UInt8	1
Number Of Instrument Attributes	UInt8	1
Number of Lot Type Rules	UInt8	1
Number Of Underlyings	UInt8	1

4.7.5 Future Spread Instrument Definition

The message contains a total of 145 or 166 bytes along with variable appendages, and consists of the following fields:

Field Name	Type	Length (bytes)
Header	SHORT or LONG HEADER	23 or 44
Instrument Type	UInt8	1
Instrument Definition Action Type	UInt8	1
Market Segment Id	UInt8	1
Security Id	UInt32	4
Security Id Source	UInt8	1
Maturity Year	UInt16	2
Maturity Month	UInt8	1
Maturity Day	UInt8	1
Security Group	UInt8	6
Asset	UInt8	6

Security Type	UInt8	6
Security Sub Type	UInt8	5
CFI Code	UInt8	6
Underlying Product	UInt8	1
Security Exchange	UInt8	1
Security Trading Status	UInt8	1
Currency	UInt8	3
Is User Defined Instrument	UInt8	1
Match Algorithm	UInt8	1
Minimum Trading Volume	UInt32	4
Maximum Trading Volume	UInt32	4
Minimum Price Increment	PRICE	5
Minimum Price Increment Amount	PRICE	5
Display Factor	UInt64	8
Tick Rule	UInt8	1
Main Fraction	UInt8	1
Sub Fraction	UInt8	1
Price Display Format	UInt8	1
Unit Of Measure	UInt8	30
Reserved Field	UInt8	8
Number Of Event Types	UInt8	1
Number Of Feed Types	UInt8	1
Number Of Instrument Attributes	UInt8	1
Number of Lot Type Rules	UInt8	1
Number Of Legs	UInt8	1

4.8 Feed Message Refresh Instrument Definition Continuation

Message Id	Value
Feed Message Refresh Instrument Definition Continuation	4

Instrument Definition Continuation message is sent whenever there is a new instrument added at the exchange level, or is modified or deleted, and its size was not enough to fit onto a single UDP message packet. Only appendages are included with the continuation message.

The message contains a total of 32 or 53 bytes along with variable appendages, and consists of the following fields:

Field Name	Type	Length (bytes)
Header	SHORT or LONG HEADER	23 or 44

Instrument Type	UInt8	1
Instrument Definition Action Type	UInt8	1
Number Of Event Types	UInt8	1
Number Of Feed Types	UInt8	1
Number Of Instrument Attributes	UInt8	1
Number Of Lot Type Rules	UInt8	1
Number Of Underlyings	UInt8	1
Number Of Legs	UInt8	1
Is Last	UInt8	1

4.9 Refresh Instrument Definition Appendages

4.9.1 Instrument Definition Event Type Appendage

The appendage contains a total of 9 bytes, and consists of the following data fields:

Field Name	Type	Length (bytes)
Event Type	UInt8	1
Event Time	UInt64	8

4.9.2 Instrument Definition Feed Type Appendage

The appendage contains a total of 4 bytes, and consists of the following data fields:

Field Name	Type	Length (bytes)
Feed Type	UInt8	3
Market Depth	UInt8	1

4.9.3 Instrument Definition Attribute Appendage

The appendage contains a total of 5 bytes, and consists of the following data fields:

Field Name	Type	Length (bytes)
Attribute Type	UInt8	1
Attribute Value	UInt32	4

4.9.4 Instrument Definition Lot Type Appendage

The appendage contains a total of 9 bytes, and consists of the following data fields:

Field Name	Type	Length (bytes)
Lot Type	UInt8	1
Minimum Lot Size	Double	8

4.9.5 Instrument Definition Underlying Appendage

The appendage contains a total of 40 bytes, and consists of the following data fields:

Field Name	Type	Length (bytes)
Underlying Symbol	LONG_SYMBOL	35
Underlying Security Id	UInt32	4
Underlying Security Id Source	UInt8	1

4.9.6 Instrument Definition Leg Appendage

The appendage contains a total of 55 bytes, and consists of the following data fields:

Field Name	Type	Length (bytes)
Leg Security Symbol	LONG_SYMBOL	35
Leg Security Id	UInt32	4
Leg Security Id Source	UInt8	1
Leg Side	UInt8	1
Leg Ratio Quantity	UInt8	1
Leg Price	PRICE	5
Leg Option Delta	Double	8

4.10 Feed Message Fundamental

Coming Soon.

4.11 Aggregate Books

Aggregate books are comprised of orders sent to exchanges, where exchange groups these orders at same price levels, and sends out the data when it changed for each price level. Aggregated books allow the display of depth of market for all orders in an instrument broken down by price level.

4.11.1 Aggregated Book Add Item

Message Id	Value
Aggregated Book Add Item	51

Aggregated Book Add Item message is sent when a new entry needs to be inserted into order book at specified index. The message is typically generated when a new order is added to the book, and the price level needs to be updated. The message contains a total of 57 or 78 bytes.

Field Name	Type	Length (bytes)
Header	SHORT or LONG SEQUENCED HEADER	31 or 52
Side	UInt8	1
Index	Uint32	4
Flags	UInt32	4
Quantity	UInt32	4
Number of Orders	UInt32	4
Price	PRICE	5
Attribution	UInt8	4

4.11.2 Aggregated Book Change Item

Message Id	Value
Aggregated Book Change Item	52

Aggregated Book Change Item message is sent when existing book entry needs to be updated with newer information. The message contains an index at which information must be overwritten with contents of the update message. The message contains a total of 57 or 78 bytes.

Field Name	Type	Length (bytes)
Header	SHORT or LONG SEQUENCED HEADER	31 or 52

Side	UInt8	1
Index	Uint32	4
Flags	UInt32	4
Quantity	UInt32	4
Number of Orders	UInt32	4
Price	PRICE	5
Attribution	UInt8	4

4.11.3 Aggregated Book Delete Item

Message Id	Value
Aggregated Book Delete Item	53

Aggregated Book Delete Item message is sent when existing entry at specified index needs to be deleted from order book. The message contains a total of 40 or 61 bytes.

Field Name	Type	Length (bytes)
Header	SHORT or LONG SEQUENCED HEADER	31 or 52
Side	UInt8	1
Index	UInt32	4
Flags	UInt32	4

4.11.4 Aggregated Book Delete Range

Message Id	Value
Aggregated Book Delete Range	54

Aggregated Book Delete Quote Range is sent when a range of existing entries need to be removed from the book. The message contains a total of 44 or 65 bytes.

Field Name	Type	Length (bytes)
Header	SHORT or LONG SEQUENCED HEADER	31 or 52
Side	UInt8	1
Index From	UInt32	4
Index To	UInt32	4
Flags	UInt32	4

4.11.5 Aggregated Book Trade

Message Id	Value
Aggregated Book Trade	55

Aggregated Book Trade is sent when a trade executes and modifies one of the sides of the book. The message contains a total of 49 or 68 bytes.

Field Name	Type	Length (bytes)
Header	SHORT or LONG SEQUENCED HEADER	31 or 52
Flags	UInt32	4
Quantity	UInt32	4
Number of Orders	UInt32	4
Aggressor Side	UInt8	1
Price	PRICE	5

4.11.6 Feed Message Book Reset

Message Id	Value
Feed Message Book Reset	56

Reset message is sent to clear the book for specified symbol and feed source. If the symbol is sent empty, the whole book should be cleared.

Field Name	Type	Length (bytes)
Header	SHORT or LONG SEQUENCED HEADER	31 or 52

4.12 Order Books

Order books are similar in nature to aggregate books, with the main difference being how orders are stored inside the book. With aggregate books, exchange groups orders with same prices into price levels, whereas order books store each order individually. To track orders within the book, each order is referenced by its original order id, rather than an index location in aggregate book.

Messages that modify existing orders will always include referenced order id of the Order Add message. To determine current quantity for each order, subscriber must deduct the modifying message quantity from current order quantity. Exchanges may send multiple modifying orders for the same order id and the effects are cumulative. When quantity reaches zero, the order should be removed from the book.

4.12.1 Book Order Add

Message Id	Value
Book Order Add Message	57

Book Order Add message is sent when a new order is sent to exchange and needs to be inserted into order book.

Field Name	Type	Length (bytes)
Header	SHORT SEQUENCED HEADER	31
Side	UInt8	1
Flags	UInt32	4
Quantity	UInt32	4
Order ID	UInt64	8
Price	PRICE	5
Attribution	UInt8	4

4.12.2 Book Order Fill

Message Id	Value
Book Order Fill	58

Book Order Fill is sent whenever an order is executed in whole or in part. It is possible to receive multiple order fill messages for the same order id. There are multiple possible order fill flags that can be included in the order fill.

When order is filled at the same price as the order, the price will be set empty.

It is possible to receive multiple order fills at different prices than specified by the order. In this case the price field will be set to the fill price. These executions might be marked with non-printable fill flag. If the flag is set, it would mean that the quantity will be included in the later bulk trade print.

Field Name	Type	Length (bytes)
Header	SHORT SEQUENCED HEADER	31
Flags	UInt32	4
Quantity	UInt32	4
Match ID	UInt64	8
Order ID	UInt64	8
Price	PRICE	5

4.12.3 Book Order Cancel

Message Id	Value
Book Order Cancel	59

Book Order Cancel is sent whenever an order is being modified as a result of a partial cancelation.

Field Name	Type	Length (bytes)
Header	SHORT SEQUENCED HEADER	31
Flags	UInt32	4
Quantity	UInt32	4
Order ID	UInt64	8

4.12.4 Book Order Delete

Message Id	Value
Book Order Delete	60

Book Order Delete message is sent whenever an order on the book is being cancelled. All remaining quantity is no longer accessible and must be removed from the book.

Field Name	Type	Length (bytes)
Header	SHORT SEQUENCED HEADER	31
Flags	UInt32	4
Order ID	UInt64	8

4.12.5 Book Order Replace

Message Id	Value
Book Order Replace	61

Book Order Replace message is sent whenever an order in the book is being canceled-replaced. All remaining quantity from original order is no longer accessible and must be removed. The new order details are provided in the message, with side, symbol and attribution staying the same.

Field Name	Type	Length (bytes)
Header	SHORT SEQUENCED HEADER	31
Flags	UInt32	4
Original Order ID	UInt64	8
New Order ID	UInt64	8
New Quantity	UInt32	4
New Price	PRICE	5

4.12.6 Book Order Break

Message Id	Value
Book Order Replace	62

Book Order Break message is sent whenever a fill is broken. A fill may be broken if it's been made erroneously. A trade break is final, and once broken, it cannot be reinstated.

Flags	**UInt32**	**4**
Match ID	UInt64	8

5.0 Field Descriptions

<table>
<tr><th>Field Name</th><th>Length (bytes)</th><th>Description</th></tr>
<tr><td><b>AGGRESSOR SIDE</b></td><td>1</td><td>Aggressor side consists of the following values:<table><tr><th>Aggressor Side Type</th><th>Value</th></tr></table></td></tr>
</table>

<table>
<tr><td></td><td></td><td><table>
<tr><td>Aggressor Side None</td><td>0</td></tr>
<tr><td>Aggressor Side Buy</td><td>1</td></tr>
<tr><td>Aggressor Side Sell</td><td>2</td></tr>
</table></td></tr>
<tr><td>BOOK ATTRIBUTION</td><td>4</td><td>Book Attribution field contains 4-character identifier that is quoting the price. If no identifier exists, the field is empty.</td></tr>
<tr><td>BOOK ORDERS</td><td>4</td><td>Number of orders at current price level. Used by aggregate books.</td></tr>
<tr><td>BOOK FILL FLAGS</td><td>4</td><td>An order fill could have multiple flags. The flags values are OR’ed into flags field, and have to be checked for presence of bits.
<table>
<tr><th>Order Fill Flags</th><th>Value</th></tr>
<tr><td>Is Printable</td><td>0x1</td></tr>
<tr><td>Is Non-Displayable</td><td>0x2</td></tr>
<tr><td>Opening Cross</td><td>0x4</td></tr>
<tr><td>Closing Cross</td><td>0x8</td></tr>
<tr><td>Halting Cross</td><td>0x10</td></tr>
</table></td></tr>
<tr><td>BOOK MATCH ID</td><td>8</td><td>Unique id assigned by exchange for order fills. In case a fill must be broken, match id is used to identify the fill.</td></tr>
<tr><td>BOOK ORDERID</td><td>8</td><td>Unique order tracking number. Most of exchanges recycle these numbers for each trading session, and are guaranteed to be unique within a trading session.</td></tr>
<tr><td>BOOK QUANTITY</td><td>4</td><td>When used by aggregate books, the quantity represents total combined order quantities at current price level. When used by order books, the quantity represents quantity of a single order.</td></tr>
<tr><td>BOOK SIDE</td><td>1</td><td>Book side type consists of the following values:
<table>
<tr><th>Side Type</th><th>Value</th></tr>
<tr><td>Side None</td><td>0</td></tr>
<tr><td>Side Bid</td><td>1</td></tr>
<tr><td>Side Ask</td><td>2</td></tr>
<tr><td>Side Implied Bid</td><td>3</td></tr>
<tr><td>Side Implied Ask</td><td>4</td></tr>
</table></td></tr>
<tr><td>DATETIME</td><td>8</td><td>Date time is a 64-bit unsigned integer and represents time in nanoseconds since epoch, ie, 00:00 Jan 1 1970.</td></tr>
<tr><td>EXCHANGE TYPE</td><td>1</td><td>Exchange Types consist of the following alphanumeric types:
<table>
<tr><th>Exchange Type</th><th>Value</th></tr>
<tr><td>CONSOLIDATED</td><td>Space</td></tr>
<tr><td></td><td></td></tr>
<tr><td>USA NYSE AMERICAN</td><td>A</td></tr>
<tr><td>USA NASDAQ OMX BX</td><td>B</td></tr>
<tr><td>USA NYSE NATIONAL</td><td>C</td></tr>
<tr><td>USA FINRA ADF</td><td>D</td></tr>
<tr><td>USA MIAX PEARL</td><td>H</td></tr>
</table></td></tr>
</table>

USA ISE	I
USA CBOE EDGA	J
USA CBOE EDGX	K
USA LONG TERM	L
USA NYSE CHICAGO	M
USA NYSE	N
USA NYSE ARCA	P
USA NASDAQ	Q
USA CTS	S
USA MEMBERS MEMX	U
USA CBOE STOCK	W
USA INVESTORS IEX	V
USA NASDAQ OMX PSX	X
USA CBOE BYX	Y
USA CBOE BZX	Z
CANADA TSX	T
CANADA VENTURE	V
USA OPTION NYSE AMERICAN	A
USA OPTION BOSTON	B
USA OPTION CBOE	C
USA OPTION MIAX EMERALD	D
USA OPTION CBOE EDGX	E
USA OPTION NASDAQ GMEX	H
USA OPTION NASDAQ ISE	I
USA OPTION NASDAQ ISE MERCURY	J
USA OPTION MIAMI	M
USA OPTION NYSE ARCA	N
USA OPTION OPRA	O
USA OPTION MIAX PEARL	P
USA OPTION NASDAQ	Q
USA OPTION NASDAQ OMX BX	T
USA OPTION CBOE C2	W
USA OPTION NASDAQ PHLX	X
USA OPTION CBOE BZX	Z
USA FUTURE CBOT	O
USA FUTURE CME	C
USA FUTURE NYMEX	N
USA FUTURE COMEX	X
USA FUTURE KCBT	K
FOREX	F

QUOTE CONDITION	1	Quote conditions map to exchange-specific quote conditions. The conditions consist of the following values:

Quote Condition	Value
Regular Quote	0
Regular Two Sided Open Quote	1
Regular One Sided Open Quote	2
Slow Ask Quote	3
Slow Bid Quote	4
Slow Bid Ask Quote	5
Slow Due LRP Bid Quote	6
Slow Due LRP Ask Quote	7
Slow Due NYSE LRP Quote	8
Slow Due Set Slow List Bid Ask Quote	9
Manual Ask Automatic Bid Quote	10
Manual Bid Automatic Ask Quote	11
Manual Bid and Ask Quote	12
Opening Quote	13
Closing Quote	14
Closed Quote	15
Resume Quote	16
Fast Trading Quote	17
Trading Range Indication Quote	18
Market Maker Quotes Closed Quote	19
Non Firm Quote	20
News Dissemination Quote	21
Order Influx Quote	22
Order Imbalance Quote	23
Due To Related Security News Dissemination Quote	24
Due To Related Security News Pending Quote	25
Additional Information Quote	26
News Pending Quote	27
Additional information Due To Related Security Quote	28
Due To Related Security Quote	29
In View of Common Quote	30
Equipment Changeover Quote	31
No Open No Resume Quote	32
Sub Penny Trading Quote	33
Automated Bid No Offer Quote	34
Luld Price Band Quote	35
Market Wide Circuit Breaker Level 1	36
Market Wide Circuit Breaker Level 2	37
Market Wide Circuit Breaker Level 3	38

<table>
<tr><td></td><td></td><td><table><tr><td><b>Republished Luld Price Band Quote</b></td><td>39</td></tr></table></td></tr>
<tr><td><b>PRICE</b></td><td>5</td><td>Price data structure contains information pertaining to price and decimal location. The Price field contains whole price multiplied by the 10 to the power of precision. For example, for the price of 50.25, the Price field will be set to 5025, and Decimal Precision to 2.
The data consists of the following types:<table><tr><th>Field Name</th><th>Type</th><th>Length (bytes)</th></tr><tr><td><b>Price</b></td><td>UInt32</td><td>4</td></tr><tr><td><b>Price Decimal Precision</b></td><td>UInt8</td><td>1</td></tr></table></td></tr>
<tr><td><b>INSTRUMENT ATTRIBUTE FLAGS</b></td><td>4</td><td>Instrument Attribute Flags are OR'ed into flags value, and consist of the following flags:<table><tr><th>Flag</th><th>Value</th></tr><tr><td><b>Electronic Match Eligible</b></td><td>0x0</td></tr><tr><td><b>Order Cross Eligible</b></td><td>0x1</td></tr><tr><td><b>Block Trade Eligible</b></td><td>0x2</td></tr><tr><td><b>EFP Eligible</b></td><td>0x4</td></tr><tr><td><b>EBF Eligible</b></td><td>0x8</td></tr><tr><td><b>EFS Eligible</b></td><td>0x10</td></tr><tr><td><b>EFR Eligible</b></td><td>0x20</td></tr><tr><td><b>OTC Eligible</b></td><td>0x40</td></tr><tr><td><b>Link Mass Quoting Eligible</b></td><td>0x80</td></tr><tr><td><b>Negative Strike Eligible</b></td><td>0x100</td></tr><tr><td><b>Negative Price Eligible</b></td><td>0x200</td></tr><tr><td><b>Is Fractional</b></td><td>0x400</td></tr><tr><td><b>RFQ Cross Eligible</b></td><td>0x1000</td></tr><tr><td><b>Zero Price Eligible</b></td><td>0x2000</td></tr><tr><td><b>Decaying Product Eligible</b></td><td>0x4000</td></tr><tr><td><b>Variable Product Eligible</b></td><td>0x8000</td></tr><tr><td><b>Daily Product Eligible</b></td><td>0x10000</td></tr><tr><td><b>Implied Matching Eligible</b></td><td>0x20000</td></tr></table></td></tr>
<tr><td><b>INSTRUMENT EVENT TYPE</b></td><td>1</td><td>Instrument Event Type contains the following values:<table><tr><th>Event Type</th><th>Value</th></tr><tr><td><b>Event Activation</b></td><td>0</td></tr><tr><td><b>Event Last Eligible Trade Date</b></td><td>1</td></tr></table></td></tr>
<tr><td><b>INSTRUMENT DEFINITION ACTION TYPE</b></td><td>1</td><td>Instrument Definition Action Type contains the type of action that should be performed on the instrument, and consists of the following values:<table><tr><th>Action Type</th><th>Value</th></tr></table></td></tr>
</table>

<table>
<tr><td></td><td></td><td><table><tr><td><b>Action Add</b></td><td>0</td></tr><tr><td><b>Action Modify</b></td><td>1</td></tr><tr><td><b>Action Delete</b></td><td>2</td></tr></table></td></tr>
<tr><td><b>INSTRUMENT LEG SIDE TYPE</b></td><td>1</td><td>Leg Side Type contains the following values:<table><tr><th>Leg Side Type</th><th>Value</th></tr><tr><td><b>Side Buy</b></td><td>0</td></tr><tr><td><b>Side Sell</b></td><td>1</td></tr><tr><td><b>Side NULL</b></td><td>2</td></tr></table></td></tr>
<tr><td><b>INSTRUMENT LOT TYPE</b></td><td>1</td><td>Instrument Lot Type contains the following values:<table><tr><th>Lot Type</th><th>Value</th></tr><tr><td><b>Minimum Order Entry Quantity</b></td><td>0</td></tr><tr><td><b>Minimum Block Trade Quantity</b></td><td>1</td></tr><tr><td><b>Round Lot</b></td><td>2</td></tr></table></td></tr>
<tr><td><b>INSTRUMENT SECURITY TRADING STATUS</b></td><td>1</td><td>Instrument Security Trading Status contains the following values:<table><tr><th>Trading Status</th><th>Value</th></tr><tr><td><b>Status Unknown</b></td><td>0</td></tr><tr><td><b>Status Trading Halt</b></td><td>1</td></tr><tr><td><b>Status Close</b></td><td>2</td></tr><tr><td><b>Status New Price Indication</b></td><td>3</td></tr><tr><td><b>Status Ready To Trade</b></td><td>4</td></tr><tr><td><b>Status Not Available for Trading</b></td><td>5</td></tr><tr><td><b>Status Pre Open</b></td><td>6</td></tr><tr><td><b>Status Pre Cross</b></td><td>7</td></tr><tr><td><b>Status Cross</b></td><td>8</td></tr><tr><td><b>Status Post Close</b></td><td>9</td></tr><tr><td><b>Status No Change</b></td><td>10</td></tr></table></td></tr>
<tr><td><b>REFRESH INTRADAY FLAGS</b></td><td>4</td><td>The flags indicate which data within the Refresh Intraday data structure is present. The flags values are OR'ed into flags field, and have to be checked for presence of data.

The flags consist of the following values:<table><tr><th>Flag</th><th>Value</th></tr><tr><td><b>Last Condition</b></td><td>0x1</td></tr><tr><td><b>Quote Condition</b></td><td>0x2</td></tr><tr><td><b>Open Price</b></td><td>0x4</td></tr><tr><td><b>Last Price</b></td><td>0x8</td></tr><tr><td><b>High Price</b></td><td>0x10</td></tr><tr><td><b>Low Price</b></td><td>0x20</td></tr></table></td></tr>
</table>

<table>
<tr><td></td><td></td><td>
<table>
<tr><td><b>Close Price</b></td><td>0x40</td></tr>
<tr><td><b>Previous Close Price</b></td><td>0x80</td></tr>
<tr><td><b>After Market Close Price</b></td><td>0x100</td></tr>
<tr><td><b>Bid Price</b></td><td>0x200</td></tr>
<tr><td><b>Ask Price</b></td><td>0x400</td></tr>
<tr><td><b>Last Exchange</b></td><td>0x800</td></tr>
<tr><td><b>Bid Exchange</b></td><td>0x1000</td></tr>
<tr><td><b>Ask Exchange</b></td><td>0x2000</td></tr>
<tr><td><b>Bid Size</b></td><td>0x4000</td></tr>
<tr><td><b>Ask Size</b></td><td>0x8000</td></tr>
<tr><td><b>Last Size</b></td><td>0x10000</td></tr>
<tr><td><b>Volume</b></td><td>0x20000</td></tr>
<tr><td><b>Pre Market Open Price</b></td><td>0x40000</td></tr>
</table>
</td></tr>
<tr><td><b>SESSION TYPE</b></td><td>1</td><td>Session type consists of the following types:
<table>
<tr><th>Session Type</th><th>Value</th></tr>
<tr><td><b>Regular Market Session</b></td><td>0</td></tr>
<tr><td><b>Pre Market Session</b></td><td>1</td></tr>
<tr><td><b>After Market Session</b></td><td>2</td></tr>
<tr><td><b>Unknown</b></td><td>255</td></tr>
</table>
</td></tr>
<tr><td><b>SESSION STATE</b></td><td>1</td><td>Session state consists of the following values:
<table>
<tr><th>Session State</th><th>Value</th></tr>
<tr><td><b>Day Open State</b></td><td>0</td></tr>
<tr><td><b>Day Close State</b></td><td>1</td></tr>
<tr><td><b>Open State</b></td><td>2</td></tr>
<tr><td><b>Close State</b></td><td>3</td></tr>
<tr><td><b>Halt State</b></td><td>4</td></tr>
<tr><td><b>Resume State</b></td><td>5</td></tr>
<tr><td><b>Unknown</b></td><td>255</td></tr>
</table>
</td></tr>
</table>

SOURCE 1

Source of originating market, consists of the following codes:

Source Code	Value
Unknown	0
CME	1
CME NYMEX	2
CME COMEX	3
CME CBOT	4
CME KCBT	5
CTA	20
CTA NYSE	21
CTA NYSE MKT	22
UTP NASDAQ	23
OTC MARKETS	24
OPRA	26
CBOE CSMI INDEX	50
NASDAQ ITCH	70
FOREX	80
CALC INDEX	100
S&P DJI INDEX	101
NASDAQ GIDS INDEX	102

SYMBOL or LONG_SYMBOL 14 or 35

Identifies the instrument symbol. Symbol consists of the following data fields:

Field Name	Type	Length (bytes)
Symbol Body	UInt8	11 or 32
Symbol Type	UInt8	1
Exchange Type	UInt8	1
Country Type	UInt8	1

Symbol Body contains different data fields for different symbol types. The symbol body is left-aligned and padded with 0's to the right. Some numeric data fields use partial bits within the same data type, reducing footprint of the data structure.

1. **Equity/Index/Bond/Mutual Fund/Currency/Future Spread Instrument**

Field Name	Type	Length (bytes)
Symbol Body	UInt8	11 or 32

Alphanumeric text specific to an instrument simply fills the symbol body for these types of instruments.

2. Option Instrument

Field Name	Type	Length
Symbol Body	UInt8	5 or 26 bytes
Strike Price	UInt32 (1)	27 bits
Expiration Value	UInt32 (1)	5 bits
Expiration Year	UInt8 (2)	6 bits
Option Type	UInt8 (2)	1 bit
Expiration Value Type	UInt8 (2)	1 bit
Expiration Month	UInt8 (3)	4 bits
Strike Price Precision	UInt8 (3)	4 bits

Option Strike Price Precision indicates the decimal location for the Strike Price, and consists of the following codes:

Strike Price Precision	Value
Whole	0
Decimal 1	1
Decimal 2	2
Decimal 3	3
Decimal 4	4
Decimal 5	5
Decimal 6	6
Decimal 7	7
Negative Whole	8
Negative Decimal 1	9
Negative Decimal 2	10
Negative Decimal 3	11
Negative Decimal 4	12
Negative Decimal 5	13
Negative Decimal 6	14
Negative Decimal 7	15

Option Type indicates whether the option is a call or a put, and consists of the following values:

Option Type	Value
Call	0
Put	1

Expiration Value indicates how to use Expiration Value, and consists of Day or Week types. If Expiration Value Type is Day, then Expiration Value field should be used as a day. In case if Expiration Value Type is a Week, then Expiration Value field should indicate week number, ie, week 1, week 2, etc., with maximum week number 5.

Expiration Value Type	Value
Day	0
Week	1

Expiration Year contains year starting from 2000. For example, year 2017 is set to 17.

Expiration Month contains months with range of 1-12.

3. **Future Instrument**

Field Name	Type	Length
Symbol Body	UInt8	9 or 30 bytes
Expiration Day	UInt16 (1)	6 bits
Expiration Year	UInt16 (1)	4 bits
Expiration Month	UInt16 (1)	5 bits

Expiration Year contains year starting from 2000. For example, year 2017 is set to 17.

Expiration Month contains months with range of 1-12.

Expiration Day is set to expiry date, or 0 if the exchange does not supply one.

Symbol Type is an alphanumeric value, and consists of the following types:

Symbol Type	Value
Equity	S
Index	I
Equity Option	O
Bond	B
Mutual Fund	M
Currency	C
Future	F
Future Option	P
Future Spread	D

See Exchange Type field description for Exchange Type.

Country Type is an alphanumeric value, and consists of the following types:

Country Type	Value
United States	U
Canada	C
International	I

TRADE CONDITINONS	1	Trade conditions map to exchange-specific trade conditions. The conditions consist of the following values:

Trade Condition	Value
Regular Trade	0
Acquisition Trade	1
Average Price Trade	2
Automatic Execution Trade	3
Bunched Trade	4
Bunch Sold Trade	5
CAP Election Trade	6
Cash Trade	7
Closing Trade	8
Cross Trade	9
Derivatively Price Trade	10
Distribution Trade	11
Form T Trade	12
Form T Out of Sequence Trade	13
Inter Market Sweep Trade	14
Market Center Official Close Trade	15
Market Center Official Open Trade	16
Market Center Opening Trade	17
Market Center Re Opening Trade	18
Market Center Closing Trade	19
Next Day Trade	20
Price Variation Trade	21
Prior Reference Price Trade	22
NYSE Rule 155 Trade	23
NYSE Rule 127 Trade	24
Opening Trade	25
Opened Trade	26
Regular Stopped Stock Trade	27
Re Opening Trade	28
Seller Trade	29
Sold Last Trade	30
Sold Last Stopped Stock Trade	31

Sold Out of Sequence Trade	32
Sold Out of Sequence Stopped Stock Trade	33
Split Trade	34
Stock Option Trade	35
Yellow Flag Trade	36
Odd Lot Trade	37
Corrected Consolidated Close Price	38
Unknown	39

For Equity Options, trade conditions map to the following values:

Option Trade Condition	Value
Regular	0
Canc	1
Oseq	2
Cncl	3
Late	4
Cnco	5
Open	6
Cnol	7
Opnl	8
Auto	9
Reop	10
Ajst	11
Sprd	12
Stdl	13
Stpd	14
Cstp	15
Bwrt	16
Cmbo	17
Spim	18
Isoi	19
Bnmt	20
Xmpt	21

TRADE FLAGS	4	Trade flags are processed and set by MarketIfs regardless of exchanges' flags, and indicate a presence of a certain condition within a trade message. All trade flags are OR'ed into the flag. The flag consists of the following values:

Trade Flag	Value
Regular Market Last Price	0x1
Regular Market Volume	0x2
Regular Market High	0x4
Regular Market Low	0x8

Day Market High	0x10
Day Market Low	0x20
Extended Market Last Price	0x40
Pre-Market Volume	0x80
After Market Volume	0x100
Close Price	0x200
Open Price	0x400

VOLUME FLAGS	4	Volume flags indicate conditions whether the volume can be used or not in specific trading sessions. Volume flags are individual bits set by MarketIfs, and should be checked by performing a bitwise AND. The flag consists of the following values:

Volume Flag	Value
Regular Market Volume	0x1
Pre-Market Volume	0x2
After Market Volume	0x4