

MARKET DATA Programmer's Guide

Programmers Guide – Market Data Specification v2.0



**FUNDAMENTAL
INTERACTIONS**

2019 July

CONTENTS

1	Context.....	1
1.1	Intended Audience.....	1
1.2	Business Context.....	1
1.3	Feed Connectivity Requirements.....	1
1.4	Retransmission and Spin Messages.....	1
2	Fundamental Interactions – Data Types.....	2
3	Protocol Message Framing.....	3
3.1	Sequence Number.....	3
3.2	Retransmission and Spin Messages.....	3
4	Message Header.....	4
4.1	Messages Sub Headers.....	4
4.2	Message Header and Sub Message.....	4
5	Fundamental Interactions Message Sets.....	5
6	Fundamental Interactions Messages.....	6
6.1	Time Message (“T”).....	6
6.2	System Event Message (“S”).....	6
6.3	Trading Action (“H”).....	7
6.4	Add Order Message (“A”).....	7
6.5	Order Executed (“E”).....	8
6.6	Order Executed with Price (“C”).....	9
6.7	Trade Message (“P”).....	9
6.8	Broken Trade Message (“B”).....	11
6.9	Order Delete Message (“D”).....	11
6.10	Order Update Message (“U”).....	12

CONTENTS

6.11 Indicative Price/Quantity Message (“I”).....	12
6.12 Best Bid Offer (“Q”).....	13
6.13 Closing Price (“F”).....	13
7 Retransmission.....	14
7.1 Login.....	14
7.2 Login Request Message (“l”).....	14
7.3 Login Response Message (“a”).....	14
7.4 Retransmission Request Message (“r”).....	15
7.5 Retransmission Response Message (“b”).....	15
7.6 Spin Request Message (“s”).....	16
7.7 Spin Response Message (“c”).....	16
7.8 Spin Usage Example.....	16
7.9 Heartbeat Message (“h”).....	18
8 Field Explanation.....	19
8.1 System Event Codes.....	19
8.2 Trading Action.....	19
8.3 Broken Trade Reasons.....	19
8.4 Message Event Flow.....	20

TABLES

Table 1 – Fundamental Interactions Data Types.....	2
Table 2 – Message Header.....	4
Table 3 – Sub Headers.....	4
Table 4 – Message Header and Sub Message.....	4
Table 5 – Available Messages.....	5
Table 6 – Time Message.....	6
Table 7 – System Event Message.....	6
Table 10 -Trading Action.....	7
Table 11 – Add Order Message.....	7
Table 12 – Order Executed.....	8
Table 13 – Order Executed with Price.....	9
Table 14 – Trade Message.....	9
Table 15 – Broken Trade Message.....	11
Table 16 – Order Delete.....	11
Table 17 – Order Replace.....	12
Table 18 – Indicative Price.....	12
Table 19 – Best Bid Offer.....	13
Table 20 – Closing Price.....	13
Table 21 - Login Request.....	14
Table 22 – Login Response.....	14
Table 23 – Retransmission Request.....	15
Table 24 – Retransmission Response.....	15
Table 25 – Spin Request.....	16
Table 26 – Spin Response.....	16
Table 27 - Heartbeat.....	18
Table 28 - System Event.....	19
Table 29 – Trading Action.....	19
Table 30 – Broken Trade.....	19
Table 31 – Message Event Flow.....	20



DISCLAIMER

Copyright © 2019 Fundamental Interactions

All Rights Reserved.

This document may not be reproduced, modified, published or distributed, in whole or in part, in any form without the prior written permission of Fundamental Interactions.

No warranty of accuracy is given, express or implied, concerning the information contained in this publication.

Product and company names herein may be the trademarks of their respective owners.

Contact Information:

Fundamental Interactions

147 West 26th Street, 3rd Floor

New York, NY 10001

support@fundamentalinteractions.com

1 Context

1.1 Intended Audience

- Client technical staff
- Fundamental Interactions technical staff

1.2 Business Context

Fundamental Interactions provides support for the standard multicast protocols. The document will cover market data dissemination. Multicast is an efficient way of distributing market data in terms of bandwidth required.

1.3 Feed Connectivity Requirements

Connectivity via extranet market data provider, cross connects at the Fundamental Interactions Inc hosting facility or dedicated circuits.

The multicast real-time feed is delivered using a published range of multicast addresses divided by the symbol ranges. One or more multicast channels may be configured to accommodate expected and future volumes.

Dropped messages can be retransmitted by requesting a range of messages to be replayed. Replayed messages are transmitted on a separate TCP/IP connection.

1.4 Retransmission and Spin Messages

Missed sequenced data can be retrieved on demand via TCP/IP based retransmission channel. Clients will be able to reconcile their state or request a range of messages for retransmission. The TCP/IP connection is dropped upon successful transmission of the replay data.

The retransmission requests are limited in, message count, frequency, and age. Retransmission requests are only available if they were within a defined sequence range of the current multicast sequence number.

2 Fundamental Interactions – Data Types

Table 1 – Fundamental Interactions Data Types

DATA TYPE	NOTES
Alphanumeric	Fields are left justified ASCII fields and space padded on the right
Integer	Unsigned big-endian binary encoded, 1byte, 2 bytes, 4 bytes or 8 bytes Price is expressed as Integer with implied 4 decimal points. For example, for price field, if you get 123400, the price value is 12.34
Null-Terminated Alphanumeric	Left justified null terminated, i.e. variable length. Maximum length includes the null character

3 Protocol Message Framing

Whenever possible, the system will combine multiple messages into one single UDP frame. Each UDP frame will contain one message header in the beginning. The count of the number of messages in a UDP frame will be communicated using the Message Header.

3.1 Sequence Number

The Message Header is used for all multicast messages as well as messages to and from the Retransmission Request Server and Spin Server.

Sequenced and un-sequenced data may be delivered using the Message Header. An un-sequenced header will have the same sequence number as the last business message for the sequence field.

To reduce overall bandwidth usage, the sequence field specifies the first message in the frame. If there are multiple messages in the frame, each message should have an implied sequence number with one greater than the previous message before.

Sequenced and un-sequenced messages cannot be sent within the same time frame.

3.2 Retransmission and Spin Messages

The client needs to login to request retransmission and spin messages.

Communication between client and Retransmission and Spin Server is done via TCP. Client will need to send in a Login request, then request for retransmission and spin via TCP.

The server will send back the login, retransmission and spin responses to client via the same TCP socket. For spin messages, the server will send those back via same TCP socket.

In addition, the TCP socket connection for the server will send a heartbeat message back to the client every certain period of time (1 minute), for line integrity testing. The client is expected to send back a heartbeat message to the server upon receipt of the heartbeat.

For messages being transmitted via TCP, the sequence number will have different definitions. The specific message only available in TCP, such as login request, login response, heartbeat and others, there is no requirement to send the sequence number, it can be just set to 0. For spin message results traveling through TCP, the sequence number is the corresponding sequence number that would be transmitted over multicast channels.

4 Message Header

Table 2 – Message Header

NAME	OFFSET	LENGTH	VALUE	NOTES
Length	0	2	Binary	Length of entire block of messages includes this header
Number Of Messages	2	1	Binary	Number of messages to follow this header
Start Sequence	3	4	Binary	Sequence of first message to follow this header
Total Length = 7 bytes				

4.1 Messages Sub Headers

After Message Header, each sub message starts with 2-byte sub header that has length (1 byte) and message type (1 byte).

It is very important to note that Fundamental Interactions may add additional fields at the end of each message to expand functionality, so do not program your application expecting each message length must be equal to a certain length. Client application must use message type field to identify message types. If there are additional bytes that exceed your current application expectation, it should just be ignored.

Each of the messages differs depending upon the various message types.

Table 3 – Sub Headers

NAME	OFFSET	LENGTH	VALUE	NOTES
Sub Length	0	1	Binary	Length of this message including this field
Message Type	1	1	Alphanumeric	Message type

4.2 Message Header and Sub Message

Table 4 – Message Header and Sub Message

MESSAGE HEADER (7 BYTES)			SUB MESSAGE 1			SUB MESSAGE 2		
Message Length (2 bytes)	Number Of Messages (1 byte)	Sequence Number (4 bytes)	Sub Message Length (1 byte)	Message Type (1 byte)	Message Body	Sub Message Length (1 byte)	Message Type (1 byte)	Message Body

5 Fundamental Interactions Message Sets

Table 5 – Available Messages

MESSAGE TYPE	NOTES
T	Timestamp, number of seconds since midnight (Eastern Time) of the system start
S	System event
R	Symbol directory, static properties of s symbol
O	Symbol dynamics, values/rules that can be updated intra-day
H	Trading action
A	Add order
E	Order executed
C	Order executed with price
P	Trade
B	Broken trade
D	Delete order
U	Update order
I	Indicative price/quantity
Q	Best Bid Offer
l	Login Request, via TCP socket
a	Login Response, via TCP socket
r	Retransmission Request via TCP socket
b	Retransmission Response, via TCP socket
s	Spin Request, via TCP socket
c	Spin Response, via TCP socket
h	Heartbeat message, via TCP socket

6 Fundamental Interactions Messages

This section describes the messages transmitted by the Fundamental Interactions market data feed.

6.1 Time Message (“T”)

Table 6 – Time Message

NAME	OFFSET	LENGTH	VALUE	NOTES
Length	0	1		Binary, Length of message including this field
Type	1	1	“T”	Timestamp-Seconds Message Id (unsequenced)
Second	2	4	Integer	Number of seconds since midnight (Eastern Time) of the first day of the system cycle
Millisecond	6	2	Integer	Millisecond portion of the time when this time message is generated

6.2 System Event Message (“S”)

Table 7 – System Event Message

NAME	OFFSET	LENGTH	VALUE	NOTES
Length	0	1		Binary, Length of message including this field
Type	1	1	“S”	System Event Message Id
Timestamp	2	4	Integer	Nanoseconds since last Timestamp seconds message
Group	6	8	Integer	Id for symbol grouping
Event Code	14	1	Alpha	Event Code, refer to System Event codes in table 28
Exchange timestamp	15	4	Integer	Official exchange time Milliseconds since midnight

6.3 Trading Action (“H”)

Table 10 -Trading Action

NAME	OFFSET	LENGTH	VALUE	NOTES
Length	0	1		Binary, Length of message including this field
Type	1	1	“H”	Symbol Trading Action Message Id
Timestamp	2	4	Integer	Nanoseconds since last Timestamp seconds message
Symbol	6	12	Alpha	Symbol
Trading State	18	1	Alpha	Current trading state for symbol Suspended intra-day with “V” then unsuspended with “N” (See table 28)
Exchange timestamp	19	4	Integer	Official exchange time Milliseconds since midnight

6.4 Add Order Message (“A”)

Table 11 – Add Order Message

NAME	OFFSET	LENGTH	VALUE	NOTES
Length	0	1		Binary, Length of message including this field
Type	1	1	“A”	Add Order Message Id
Timestamp	2	4	Integer	Nanoseconds since last Timestamp seconds message
Order Number	6	18	Alphanumeric	The unique reference number assigned to the new order
Order Verb	24	1	Alpha	“B” buy “S” sell
Quantity	25	4	Integer	The total quantity of the order being added to the book
Symbol	29	12	Alpha	Symbol
Price	41	4	Integer	The price of the new order
FirmId	45	3	Alpha	3 digits integers FirmId ‘001’ means anonymous

NAME	OFFSET	LENGTH	VALUE	NOTES
Settlement Term	48	8	Alpha	Override Settlement Term "T" = Cash Today (T+0) "C" = Cash (T+1) "N" = Non-net "R" = Non-Resident "" = Special Settlement Date (see below) Special settlement date for the order, in the form of YYYYMMDD will be set Left blank " "if default settlement
OddLot Flag	56	1	Alpha	"Y" = Odd lot order "N" = Not an odd lot
Exchange timestamp	57	4	Integer	Official exchange time Milliseconds since midnight

6.5 Order Executed ("E")

Table 12 – Order Executed

NAME	OFFSET	LENGTH	VALUE	NOTES
Length	0	1		Binary, Length of message including this field
Type	1	1	"E"	Order Executed Message Id
Timestamp	2	4	Integer	Nanoseconds since last Timestamp seconds message
Order Number	6	18	Alphanumeric	The unique executed order reference number
Executed Quantity	24	4	Integer	The number of shares executed
Match Number	28	18	Alpha	The unique match identifier
Exchange timestamp	46	4	Integer	Official exchange time Milliseconds since midnight

6.6 Order Executed with Price (“C”)

Table 13 – Order Executed with Price

NAME	OFFSET	LENGTH	VALUE	NOTES
Length	0	1		Binary, Length of message including this field
Type	1	1	“C”	Order Executed with Price Message Id
Timestamp	2	4	Integer	Nanoseconds since last Timestamp seconds message
Order Number	6	18	Alphanumeric	The unique executed order reference number
Executed Quantity	24	4	Integer	The number of shares executed
Match Number	28	18	Alpha	The unique match identifier
Execution Price	46	4	Integer	The price at which the execution occurred
Exchange timestamp	50	4	Integer	Official exchange time Milliseconds since midnight

6.7 Trade Message (“P”)

Table 14 – Trade Message

NAME	OFFSET	LENGTH	VALUE	NOTES
Length	0	1		Binary, Length of message including this field
Type	1	1	“P”	Trade Message Id
Timestamp	2	4	Integer	Nanoseconds since last Timestamp seconds message
Executed Quantity	6	4	Integer	The number of shares executed
Symbol	10	12	Alpha	Symbol
Execution Price	22	4	Integer	The price at which the execution occurred

NAME	OFFSET	LENGTH	VALUE	NOTES
Match Number	26	18	Alpha	The unique match identifier
FirmId	44	3	Alpha	3 digits integers FirmId '001' means anonymous
ContraFirmId	47	3	Alpha	3 digits integers FirmId '001' means anonymous
Settlement Term	50	8	Alpha	Override Settlement Term "T" = Cash Today (T+0) "C" = Cash (T+1) "N" = Non-net "R" = Non-Resident "" = Special Settlement Date (see below) Special settlement date for the order, in the form of YYYYMMDD will be set Left blank "" if default settlement
Trade Attribute	58	1	Alpha	"B" for ByPass order "" default
Cross Type	59	1	Alpha	"R" = Regular (on new TE only) "I" = Internal "C" = Contingent "B" = Basis "V" = VWAP "" = Otherwise empty string (not specified)
OddLot Flag	60	1	Alpha	"Y" = Odd lot order "N" = Not an odd lot
Sets the Last Price	61	1	Alpha	"Y" = Last sale "N" = Not last sale
Exchange timestamp	62	4	Integer	Official exchange time Milliseconds since midnight

6.8 Broken Trade Message (“B”)

Table 15 – Broken Trade Message

NAME	OFFSET	LENGTH	VALUE	NOTES
Length	0	1		Binary, Length of message including this field
Type	1	1	“B”	Broken Trade Message Id
Timestamp	2	4	Integer	Nanoseconds since last Timestamp seconds message
Match Number	6	18	Alpha	Match number of the execution that was broken. This refers to a match number from a previously transmitted Order Executed Message or Order Executed with Price Message
Reason	24	1	Alpha	The reason the trade was broken. See currently supported BrokenTrade Reasons table 30
Exchange timestamp	25	4	Integer	Official exchange time Milliseconds since midnight

6.9 Order Delete Message (“D”)

Table 16 – Order Delete

NAME	OFFSET	LENGTH	VALUE	NOTES
Length	0	1		Binary, Length of message including this field
Type	1	1	“D”	Order Delete Message Id
Timestamp	2	4	Integer	Nanoseconds since last Timestamp seconds message
Order Number	6	18	Alphanumeric	The reference number of the order being cancelled
Exchange timestamp	24	4	Integer	Official exchange time Milliseconds since midnight

6.10 Order Update Message (“U”)

Table 17 – Order Replace

NAME	OFFSET	LENGTH	VALUE	NOTES
Length	0	1		Binary, Length of message including this field
Type	1	1	“U”	Order Replace Message Id
Timestamp	2	4	Integer	Nanoseconds since last Timestamp seconds message
Order Number	6	18	Alphanumeric	The original order number of the order being updated
Quantity	24	4	Integer	The new total open quantity, i.e. the balance of the new order
Price	28	4	Integer	The new price for the order
Exchange timestamp	32	4	Integer	Official exchange time Milliseconds since midnight

6.11 Indicative Price/Quantity Message (“I”)

Table 18 – Indicative Price

NAME	OFFSET	LENGTH	VALUE	NOTES
Length	0	1		Binary, Length of message including this field
Type	1	1	“I”	Indicative Price/Quantity Message Id
Timestamp	2	4	Integer	Nanoseconds since last Timestamp seconds message
Theoretical Opening Quantity	6	8	Integer	The total quantity eligible to be matched at the current Theoretical Opening Price
Symbol	14	12	Alpha	Symbol
Theoretical Opening Price	26	4	Integer	The current opening price for this symbol
Exchange timestamp	30	4	Integer	Official exchange time Milliseconds since midnight

6.12 Best Bid Offer (“Q”)

Table 19 – Best Bid Offer

NAME	OFFSET	LENGTH	VALUE	NOTES
Length	0	1		Binary, Length of message including this field
Type	1	1	“Q”	Best Bid Offer Quotation Message Id
Timestamp	2	4	Integer	Nanoseconds since last Timestamp seconds message
Symbol	6	12	Alpha	Symbol
Best Bid	18	4	Integer	The best buy price. For an auction such as Pre-Open, Delayed Open or Halt, when indicative price is set, we overwrite the Best Bid with indicative price
Best Bid Size	22	4	Integer	Total quantity at the best buy price
Best Offer	26	4	Integer	The best sell price. For an auction such as Pre-Open, Delayed Open or Halt, when indicative price is set, we overwrite the Best Offer with indicative price
Best Offer Size	30	4	Integer	Total quantity at the best sell price
Exchange timestamp	34	4	Integer	Official exchange time Milliseconds since midnight

6.13 Closing Price (“F”)

Table 20 – Closing Price

NAME	OFFSET	LENGTH	VALUE	NOTES
Length	0	1		Binary, Length of message including this field
Type	1	1	“F”	Closing price
Timestamp	2	4	Integer	Nanoseconds since last Timestamp seconds message
Symbol	6	12	Alpha	Symbol
Price	18	4	Integer	The closing price

7 Retransmission

7.1 Login

Logon message is the first message required to send to the server once a connection is established. At minimum, the firm and user id in the logon request should be provided.

The Login Response Message is sent to a client’s response to a TCP/IP Login Message. The status field is used to reflect an accepted login or the reason the session was not accepted. If login fails the TCP/IP connection will be dropped.

Additionally a few parameters can be set in Logon data.

- HeartbeatInterval, in seconds, set this field if you want server to send you heartbeat
- Heartbeat messages are expected every 5 seconds. Failure to receive 2 consecutive heartbeat messages will result in termination of the client connection.

7.2 Login Request Message (“l”)

Table 21 - Login Request

NAME	OFFSET	LENGTH	VALUE	NOTES
Length	0	1		Binary, Length of message including this field
Type	1	1	“l”	Login request
UserId	2	16	Alpha	User id assigned by Fundamental Interactions
Password	18	16	Alpha	Password assigned by Fundamental Interactions

7.3 Login Response Message (“a”)

Table 22 – Login Response

NAME	OFFSET	LENGTH	VALUE	NOTES
Length	0	1		Binary, Length of message including this field
Type	1	1	“a”	Login response

NAME	OFFSET	LENGTH	VALUE	NOTES
UserId	2	16	Alpha	User id assigned by Fundamental Interactions
Status	18	1	"Y"	Status Always "Y", if it is an invalid userid/ password, Fundamental Interactions will close socket after 5 seconds without sending back a login response

7.4 Retransmission Request Message ("r")

Table 23 – Retransmission Request

NAME	OFFSET	LENGTH	VALUE	NOTES
Length	0	1		Binary, Length of message including this field
Type	1	1	"r"	Retransmission request
Start Sequence	2	4	Integer	Start sequence number
Number Of Messages	6	4	Integer	Number of messages requested -1 to indicate all messages up to date

7.5 Retransmission Response Message ("b")

Table 24 – Retransmission Response

NAME	OFFSET	LENGTH	VALUE	NOTES
Length	0	1		Binary, Length of message including this field
Type	1	1	"b"	Retransmission response
Start Sequence	2	4	Integer	Start sequence number
Number Of Messages	6	4	Integer	Number of messages requested
Status	10	1	Alpha	Status "Y" retransmission will follow as requested "N" request ignored

7.6 Spin Request Message (“s”)

Table 25 – Spin Request

NAME	OFFSET	LENGTH	VALUE	NOTES
Length	0	1		Binary, Length of message including this field
Type	1	1	“s”	Spin Request
Client Identifier	2	4	Integer	Optional client identifier

7.7 Spin Response Message (“c”)

Table 26 – Spin Response

NAME	OFFSET	LENGTH	VALUE	NOTES
Length	0	1		Binary, Length of message including this field
Type	1	1	“c”	Spin Response
Client Identifier	2	4	Integer	Echo back from spin request
Status	6	1	“S” “E”	Status First, server will ack your spin request with “S”server will then retransmit all information back to you over same TCP socket. Once it is done, it will send another spin response with status “E” to mark end of spin Note: spin messages are always returned to the same TCP socket where they originate

7.8 Spin Usage Example

Spin is used when you lost all order book information on your side and need to rebuild the full book.

If you detect a gap, you shouldn’t use SPIN, you should ask for retransmission instead.

Spin messages are sequenced but can arrive out of sequence.

For spin request, server will send you a snapshot of the order book at the moment when request is received.

When client system restarts, it should first join multicast feed group. Right after joining the multicast group, client should send spin request. All messages received from multicast should be queued at this moment. Client will then process all spin response first, then process all queued multicast messages. Once this is done, client should have up-to-date book information. Then client should start to process all new multicast messages in normal way.

After processed all spin response messages, during processing queued multicast messages, client should pay special attention to sequence number and use sequence number to throw away unneeded messages from multicast queue.

The following is an illustration how this portion should work.

Supposed we have a simple book

1000@10, order id is AA1, the latest sequence number for this entry is 1

900@9.99, order id is AB1, the latest sequence number for this entry is 5

500@9.98, order id is AC2, the latest sequence number for this entry is 15

A client lost its book, it started to join multicast group, then immediately requested for spin. During this period, two quote events happened.

First event, order(AA1) was partially executed for 200 shares. A multicast message of order executed was sent out with order id AA1 and sequence number 18.

Second event, order(AC2) was canceled. A multicast message of order delete was sent out with order id AC2 and sequence number 19.

If server received snapshot request before those two events happened, client first received initial snapshot of 3 messages each carrying sequence number 1, 5, 15.

Important note, snapshot response's sequence number are not sequential and can be out of sequence, you might receive those 3 messages in order of 15,5,1 as well.

Then clients started to process queued message (message 18, and message 19). For message 18, order id was AA1, client found the book entry that had sequence number 1, which was smaller than sequence number 18, client processed message 18, updated entry AA1, now the book entry had 800@10 with sequence number 18.

Next queued message was message 19, it had order id AC2, client found corresponding book entry that had sequence number 15, which was smaller than sequence number 19, client processed message 19, deleted entry AC2. Client book is now as follows, which is current.

800@10, order id is AA1, the latest sequence number for this entry is 18

900@9.99, order id is AB1, the latest sequence number for this entry is 5

Let's try a different scenario. Supposed two events happened right at moment between clients joined multicast group,

but before server received snapshot request.

So server will send its latest snapshot back to clients, it had two following messages.

800@10, order id is AA1, the latest sequence number for this entry is 18
 900@9.99, order id is AB1, the latest sequence number for this entry is 5

Client processed the snapshot first. Then processed queued multicast messages, message 18 and 19. For message 18, order id is AA1, it had sequence number 18, which was same or higher than queue multicast message 18. In this situation, client should throw away multicast message 18.

Next message 19, order id is AC2, clients found no entry in book, in this case, it should throw away multicast message 19 also.

Now client book is up to date with two entries.

800@10, order id is AA1, the latest sequence number for this entry is 18
 900@9.99, order id is AB1, the latest sequence number for this entry is 5

As you can see, once you use sequence number to filter out old messages, you will have up-to-date book in all scenarios.

7.9 Heartbeat Message (“h”)

Table 27 - Heartbeat

NAME	OFFSET	LENGTH	VALUE	NOTES
Length	0	1		Binary, Length of message including this field
Type	1	1	“h”	Heartbeat message
Client Identifier	2	4	Integer	Optional identifier

8 Field Explanation

8.1 System Event Codes

Table 28 - System Event

SYSTEM EVENT CODE	EXPLANATION
"O"	Start of Messages, this is the first message sent
"S"	Start of System Hours
"V"	Pre-open auction period begins. The Exchange is open and ready to start accepting orders
"Q"	Start of Market Hours, the Pre-open auction is concluded, and continuous trading begins
"M"	End of Market Hours, this indicates the end of continuous trading
"E"	End of System Hours, it indicates that the Exchange is closed
"C"	End of Messages, this is the last message sent

8.2 Trading Action

Table 29 – Trading Action

TRADING ACTION	EXPLANATION
"N"	Normal Trading
"V"	Suspend Trading
"H"	Halt
"D"	Delayed Open

8.3 Broken Trade Reasons

Table 30 – Broken Trade

REASON	EXPLANATION
"S"	Supervisory – The trade was manually broken by the Exchange
"A"	Amended

8.4 Message Event Flow

Table 31 – Message Event Flow

MARKET STATE	SYSTEM EVENT MESSAGE [S] SYSTEM EVENT CODE (SYSTEM OR GROUP)	
	“O” – “Start of Messages” First message sent.	S
ENQUIRY	“S” – “Start of System Hours”	G
PRE-OPEN	“V” – “Scheduled Auction Starts”	G
OPEN	“Q” – “Start Of Market Hours”	G
CLOSE	“M” – “End Of Market Hours”	G
ENDOFDAY	“E” – “End of System Hours”	G
	“C” – “End of Messages” Last message sent.	S



FUNDAMENTAL INTERACTIONS

Sales & Development Office

147 W. 26th Street. 300

New York, NY 10001

Office: (212) 725-3509 | Corporate: 212-845-9077

sales@fundamentalinteractions.com

Support Office

Hudson Street, Hoboken, NJ 07030
Support Phone: 888-851-1369
support@fundamentalinteractions.com

Development Office

Development Office
1500 District Avenue, 2nd Floor, Burlington,
MA 01803

Europe Development & Support Office

Kharkov, Ukraine
61000