LMAX FIX 4.2 API Specification

Brokers

Version 1.0

1 Preface

This document provides complete technical details on interfacing to the LMAX Trading FIX 4.2 Interface (the FIX interface). The document is intended to be used by clients as a technical reference when building systems that interact with the LMAX FIX 4.2 Broker API through FIX.

2 Introduction

2.1 Overview

2.2 Operating Hours

The LMAX opening hours are described in the following schedule. Please note that the time is expressed in EDT/EST time (America/New York time zone) and will follow America/New York DST changes schedule.

Order Books will be available only during LMAX operating hours.

	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
Close	17:00	17:00	17:00	17:00	17:00	Not open	
Open	17:05	17:05	17:05	17:05		Not open	17:05

2.2.1 FIX Session Schedule

Fix session schedule follows the same daily schedule as LMAX. At 17:02:00 New York time LMAX will disconnect all FIX sessions and reset the sequence numbers, both inbound and outbound.

TimeZone	America/New York
Connect/Reset Sequence Numbers	17:02:00
Logout/Disconnect	17:02:00

2.3 Terms

Term	Description
LMAX Gateway	Refers to the LMAX FIX Gateway system.

Client	Refers to a Broker Client that connects directly to the LMAX Exchange.
Connection	A network connection to the LMAX Gateway. Each connection belongs to exactly one client, though a client may have multiple sessions. Connections are only allowed to be logged on once.
Session	FIX Connection between LMAX Gateway and Client is comprised of three parts: logon, message exchange, and logout. These are identified by a SenderCompID assigned, at initial registration, by the LMAX.
Initiator	FIX session party that initiates the FIX Connection
Acceptor	FIX session party that accepts the FIX Connection

2.4 FIX Message Syntax

LMAX supports the standard "Tag=Value" syntax. The general format of a FIX message applies. Messages consist of a series of "Tag=Value" fields separated by a field delimiter. The delimiter is ASCII 1 (SOH) symbol. All messages begin with a standard header and are terminated with a standard trailer.

2.5 Foreign Exchange Securities Reference

For foreign exchange instruments, conventions for identifying the transactions are as follows:

- The Foreign Exchange Symbol (55) is defined in EBS (Electronic Banking System) format: CCY1/CCY2.
    - Rates are expressed as currency1 in currency2 (or currency2 per currency1) and are calculated as CCY2 divided by CCY1 (NOT CCY1 divided by CCY2)
    - e.g. GBP/USD represents a rate expressed as USD per GBP USD/JPY represents a rate expressed as JPY per USD, etc.
    - CCY1 and CCY2 are ISO currency codes

3 Session Level Protocol

3.1 Overview

This section discusses issues pertaining to the FIX Session Protocol, which is responsible for providing reliable, ordered transport of FIX Application messages. All messages sent by the Client must have one Client ID that is agreed upon in advance with LMAX.

3.2 Headers

There are two fields in the header that are used by the FIX Session Level protocol for technical routing. These are *SenderCompID* and *TargetCompID*. When a Client sends a message into the LMAX Gateway it populates the *SenderCompID* with its Client ID provided by LMAX and the *TargetCompID* with FIX-API. When the LMAX Gateway sends messages to a Client it populates the *SenderCompID* with FIX-API and

TargetCompID with the Client ID provided by LMAX for that Client. For full description of the Header please see section 5.6 - Headers and Trailers.

3.3 Establishing a Session

In alignment with the current Exchange connectivity model, all Trading Parties must connect to the LMAX Gateway. Client must reconnect if the connection is lost.

3.3.1 IP Authentication

When a Client connects to the LMAX Gateway using the FIX interface, its IP address is authenticated to make sure that it is one of the assigned IP addresses for that Client. If the IP address is not recognized then the connection is immediately dropped.

3.3.2 Logon Authentication

Upon successful connection, the Client sends in a FIX **Logon** message. This contains the assigned *SenderCompID* and the password for the Client. The LMAX Gateway authenticates the *SenderCompID* and password. Only a valid combination of *SenderCompID* and password will be allowed to proceed and establish a FIX session.

Only one FIX session per SenderCompID is allowed at a time. If a SenderCompID is already logged on, the current connection will be logged off in favour of the new logon.

3.3.3 Authorization

Messages are validated to make sure that Clients are allowed to send them. If a message fails this validation an appropriate rejection message is sent back to the sender. This is an **ExecutionReport** for **NewOrderSingle**, **OrderCancelReject** for an **OrderCancelRequest** or **BusinessLevelReject**, in all other cases.

3.3.4 Message Recovery

In normal circumstances, when a Client is not connected, the LMAX Gateway will store **ExecutionReport**(s) and **OrderCancelReject**(s), and discard all others. These stored messages are then automatically sent to the Client upon a successful logon as a response to a resend request.

3.3.5 Sequence number handling and Resetting

Session is always assigned two sequence numbers:

- Incoming sequence number is a counter for incoming message
- Outgoing sequence number is a counter for outgoing message

Both sides must maintain two values and control that they are in sync. There are two types of sequence numbers de-sync:

- Sequence number too high indicates messages loss and leads to resend procedure.
- Sequence number too low indicates some serious problem and must lead to immediate session termination and manual sequence number synchronization.

The LMAX FIX Gateway handles sequence numbers as follows:

- Sequence numbers, both inbound and outbound, will be reset to 1 each night during Exchange closed period. See section 2.2- Operating Hours for more details.
- In normal circumstances, LMAX will never reset sequence numbers intraday. After intraday logout the sequence numbers will continue on next connect.
- Messages are processed in sequence order
- Setting *ResetSeqNumFlag* to Y make session to reset sequence number each time on logon and force counter-party to do the same (standard FIX mechanism). This is not recommended since messages sent during inactivity time will be lost.
- In all cases except the **Sequence Reset - Reset** message, the FIX session should be terminated if the incoming sequence number is less than expected and the *PossDupFlag* is not set. A **Logout** message with the descriptive text should be sent to the other side before closing the session.

3.4 Session state

From the time session is created and until it is destroyed, session uses state that dictates its reaction to events.

3.4.1 FIX Session-level State Matrix

Precedence	State	Initiator	Acceptor	Description
1	Disconnected-No Connection Today	Y	Y	Currently disconnected, have not attempted to establish a connection “today”, and no *MsgSeqNum* have been consumed (next connection “today” will start at *MsgSeqNum* of 1)
2	Disconnected-Connection Today	Y	Y	Currently disconnected, have attempted to establish a connection “today” and thus *MsgSeqNum* have been consumed (next connection “today” will start at *MsgSeqNum* of (last + 1))

3	Detect Broken Network Connection	Y	Y	While connected, detect a broken network connection (e.g. TCP socket closed). Disconnect the network connection and "shutdown" configuration for this session. LMAX treats TCP disconnect as a Logout.
4	Awaiting Connection	N	Y	Session acceptor Logon awaiting network connection from counterparty
5	Initiate Connection	Y	N	Session initiator Logon establishing network connection with counterparty
6	Network Connection Established	Y	Y	Network connection established between both parties
7	Initiation **Logon** Sent	Y	N	Session initiator Logon send **Logon** message.
8	Initiation **Logon** Received	N	Y	Session acceptor Logon receive counterparty's **Logon** message.
9	Initiation **Logon** Response	N	Y	Session acceptor Logon respond to counterparty's **Logon** message with **Logon** message to handshake
10	Handle **ResendRequest**	Y	Y	Receive and respond to counterparty's **ResendRequest** sending requested application level messages and/or **SequenceReset-Gap Fill** messages for the range of *MsgSeqNum* requested
11	Receive *MsgSeqNum* Too High	Y	Y	Receive too high of *MsgSeqNum* from counterparty, queue message, and send **ResendRequest**
12	Awaiting/Processing Response to **ResendRequest**	Y	Y	Process requested *MsgSeqNum* *PossDupFlag*=Y resent messages and/or **SequenceReset-Gap Fill** messages from counterparty. Queue incoming messages with *MsgSeqNum* too high
13	No messages received in Interval	Y	Y	No inbound messages (non-garbled) received in (*HeartBeatInt* + "reasonable period of time"), send **TestRequest.** Please note that LMAX determines whether the session is alive by sending **TestRequest** and receiving the **Heartbeat** response with corresponding *TestRequestID* within 2 x *heartbeat*+2 seconds
14	Awaiting/Processing Response to **TestRequest**	Y	Y	Process inbound messages. Reset heartbeat interval-related timer when **Heartbeat** message with corresponding *TestRequestID* is received.
15	Receive **Logout** message	Y	Y	Receive **Logout** message from counterparty initiating logout/disconnect. If *MsgSeqNum* too high, send **ResendRequest**. If sent, wait a reasonable period of time for complete response to **ResendRequest**. Note that depending upon the reason for the **Logout**, the counterparty may be unable to fulfill the request. Send **Logout** message as response and wait a reasonable period of time for counterparty to disconnect the network connection. Note counterparty may send a **ResendRequest** message if **Logout** message response has *MsgSeqNum* too high and then re-initiate the **Logout** process.
16	Initiate **Logout** Process	Y	Y	Identify condition or reason to gracefully disconnect (e.g. no response after **TestRequest** messages, etc.). Send **Logout** message to counterparty. Wait a reasonable period of time for **Logout** response. During this time handle "new" inbound messages and/or **ResendRequest** if possible. Note

				that some logout/termination conditions may require immediate termination of the network connection following the initial send of the **Logout** message. Disconnect the network connection and "shutdown" configuration for this session.
17	Active/Normal Session	Y	Y	Network connection established, **Logon** message exchange successfully completed, inbound and outbound *MsgSeqNum* are in sequence as expected, and **Heartbeat,** or other messages are received within 2 x heartbeat+2 seconds. Please note that LMAX relies on **Heartbeat** response with corresponding *TestRequestID* sent within 2 x heartbeat+2 seconds from sending **TestRequest** to determine whether the session is alive.
18	Waiting for **Logon** Ack	Y	N	Session initiator waiting for session acceptor to send back **Logon** ACK.

3.4.2 FIX Logon Process State Transition Diagram

Session Initiator Action	Session Acceptor Action	Session Initiator State	Session Acceptor State
Start		Disconnected-No Connection Today Disconnected-Connection Today	Awaiting Connection
Connect		Initiate Connection (Possible) Detect Broken Network Connection	Awaiting Connection
	Accept Connection	Network Connection Established	Network Connection Established
Initiate **Logon**		Initiation **Logon** Sent	Network Connection Established
	Receive Initiation **Logon**	Initiation **Logon** Sent	Initiation **Logon** Received
	Send Initiation **Logon** Response	Initiation **Logon** Sent	Initiation **Logon** Response (Possible) Initiate **Logout** Process (Possible) Receive *MsgSeqNum* Too High
	(Possible) Send **ResendRequest**		Initiation **Logon** Response (Possible) Receive *MsgSeqNum* Too High
Receive Initiation **Logon** Response		(Possible) Active/Normal Session (Possible) Initiate **Logout** Process	Initiation **Logon** Response
(Possible) Send **ResendRequest**		(Possible) Active/Normal Session (Possible) Receive *MsgSeqNum* Too High	(Possible) Active/Normal Session (Possible) Handle **ResendRequest**
		Active/Normal Session	Active/Normal Session

3.4.3 FIX Logout Process State Transition Diagram

Logout Initiator Action	Logout Acceptor Action	Logout Initiator State	Logout Acceptor State
Start		• Active/Normal Session • No messages received in Interval • Awaiting/Processing Response to **TestRequest**	• Active/Normal Session • No messages received in Interval • Initiation **Logon** Sent • Awaiting/Processing Response to **TestRequest** • Awaiting validation of logon • Receive *MsgSeqNum* Too High • Awaiting/Processing Response to **ResendRequest** • Initiate **Logout** Process • Waiting for **Logon** Ack
Send **Logout** message		Logout Pending	
	Receive **Logout** message	Logout Pending	Logout Pending (Possible) Receive *MsgSeqNum* Too High
	Send **Logout** response	Logout Pending	Awaiting Disconnect
	(Possible) Send **ResendRequest**	Logout Pending	(Possible) Awaiting / Processing Response to **ResendRequest**
(Possible) receive **ResendRequest**		(Possible) Awaiting / Processing Response to **ResendRequest**	(Possible) Awaiting Response to **ResendRequest**
Receive **Logout** Response		Disconnected-Connection Today	Awaiting Disconnect
Disconnect		Disconnected-Connection Today	Disconnected-Connection Today

4 Application Level Protocol

4.1 Overview

This section describes the various message reference ids that are used by FIX protocol and the LMAX Gateway to allow Clients to marry up requests with responses.

4.2 Message Rejection

LMAX may reject messages for a number of reasons. Depending on the nature of the failure, different FIX message types may be used to transmit the error.

FIX Message Type used to report the error.	Type of Error
Reject (MsgType='3')	Basic message validation failure such as an unsupported tag for a message or an invalid value for an allowed tag.
BusinessReject (MsgType='j')	Conditionally required tag missing.
OrderCancelReject (MsgType='9')	To report an error encountered while processing an **OrderCancelRequest** or **OrderCancelReplaceRequest**.
ExecutionReport (MsgType='8')	To report an error encountered while attempting to place a new order with **NewOrderSingle**.
QuoteAcknowledgement (MsgType='b')	To report an error encountered while attempting to place a **MassQuote** (Mass Order)

4.3 Cancel Orders on Logout and Disconnect Functionality

4.3.1 Unexpected disconnect

In the event of unexpected disconnect, all client's working orders associated with the FIX session will be cancelled automatically. LMAX treats TCP disconnect as an implicit logout.

4.3.2 Disconnect with Logout

When the Client disconnects the session with **Logout** message, all working orders associated with the session will be cancelled automatically.

4.4 References

4.4.1 Order References

NewOrderSingle, **OrderCancelRequest**, **OrderCancelReject** and **ExecutionReport** messages use *ClOrdID* field to uniquely identify the orders.

ExecutionReport messages contain the *ClOrdID* of the **NewOrderSingle** and **OrderCancelRequest** messages to allow the Client to correlate the **ExecutionReports** to the **NewOrderSingle** and **OrderCancelRequest** messages.

OrderCancelRequest and **OrderCancelReplaceRequest** messages also use the *OrigClOrdID* to reference the original order being cancelled/amended. This ID is present in the **ExecutionReport** and **OrderCancelReject** message sent in response to an **OrderCancelRequest**.

4.4.2 Mass Quote References

MassQuote, **QuoteCancel** and **QuoteAcknowledgement** messages contain the *QuoteID* field that is used to uniquely identify the message. Each individual order within the Mass Quote will be identified by a *ClOrdID* determined by the *QuoteID* plus a sequential character [0-9, a-b] added at the end of the string. This field is populated sequentially starting from Top of Book Bid, descending, then Top of Book Ask, ascending. This field will be mapped in the corresponding ExecutionReports as *ClOrdID*. Orders in the **MassQuote** can be identified in the **ExecutionReports** based on the *ClOrdID* and *Price*. Example shown below.

Original Mass Quote

Quote ID: lmaxord1	
BID	ASK
1m @ 1.11700	1m @ 1.11706
1m @ 1.11699	1m @ 1.11707
1m @ 1.11698	1m @ 1.11708
1m @ 1.11696	1m @ 1.11710
1m @ 1.11695	1m @ 1.11711
	1m @ 1.11712

LMAX ClOrdID assignment

Quote ID: lmaxord1			
ClOrdID	BID	ClOrdID	ASK
lmaxord10	1m @ 1.11700	lmaxord15	1m @ 1.11706
lmaxord11	1m @ 1.11699	lmaxord16	1m @ 1.11707
lmaxord12	1m @ 1.11698	lmaxord17	1m @ 1.11708
lmaxord13	1m @ 1.11696	lmaxord18	1m @ 1.11710
lmaxord14	1m @ 1.11695	lmaxord19	1m @ 1.11711
		lmaxord1a	1m @ 1.11712

4.4.3 ClOrdID and QuoteID Formatting

In order to provide clients with the best performance on the exchange we have implemented the following formatting requirements on the *ClOrdID* and *QuoteID* tags.

- Value will be restricted to a character string with a minimum length of 1 character and a maximum length of 20 characters.
- Value is restricted to the ASCII characters 33-126
- Value cannot be a single character of 0
- For QuoteIDs, the client determined string must have a maximum length of 19 characters, as LMAX will add an additional character at the end of the string to represent each individual order within the MassQuote. These IDs are reported as ClOrdID in ExecutionReports reporting executions on MassQuotes.

5 Message Reference

This section provides a reference for all the messages used by the FIX interface of the LMAX Gateway. The subsection that follows describes the notation used in this part of the document.

5.1 General

The count field in a repeating block is always required when specifying a repeating block.

As per the FIX specification, if a repeating block is supplied, then the first field in the block must be present for each instance of the block.

5.2 Data Types

The FIX interface supports the following data types.

Data Type	Description
Int	Sequence of digits without commas or decimals and optional sign character (ASCII characters - and 0 - 9). The sign character utilizes one byte (i.e. positive Int is 99999 while negative int is -99999). Note that Int values may contain leading zeros (e.g. 00023 = 23).
Qty	Value capable of storing either a whole number (no decimal places) of shares (securities denominated in whole units) or a decimal value containing decimal places for non-share quantity asset classes (securities denominated in fractional units).
String	Alpha-numeric free format strings; can include any character or punctuation except the delimiter. All char fields are case sensitive (i.e. lmax != LMAX).

Boolean	char field containing one of two values: 'Y' = True/Yes 'N' = False/No
Currency	Representing a currency type using ISO 4217 Currency code (3 characters) values.
UTC Timestamp	Representing Time/date combination represented in UTC (Universal Time Coordinated, also known as GMT) only accepted in this format : YYYYMMDD-HH:MM:SS (whole seconds).
Char	Char value, can include any alphanumeric character or punctuation except the delimiter. All char fields are case sensitive (i.e. l != L).
Price	Value representing a price. Note the number of decimal places may vary.
Length	Representing the length in bytes. Value must be positive
SeqNum	Representing a message sequence number. Value must be positive
Float	Sequence of digits with optional decimal point and sign character (ASCII characters - , 0 - 9 and.); the absence of the decimal point within the string will be interpreted as the float representation of an integer value. All float fields must accommodate up to fifteen significant digits. The number of decimal places used should be a factor of business/market needs and mutual agreement between counterparties. Note that float values may contain leading zeros (e.g. 00023.23 = 23.23) and may contain or omit trailing zeros after the decimal point (e.g. 23.0 = 23.0000 = 23 = 23.). Note that fields which are derived from float may contain negative values unless explicitly specified otherwise.
NumInGroup	Value that represents the number of repeating values in a group
MultipleValueString	Field Containing one or more space delimited values.

5.3 Notation

The following notation is used in the tables that follow:

Notation	Description
*	This denotes a tag not present in FIX 4.2
➢	This denotes a nested block.
➢➢	This denotes a doubly nested block (i.e. a block within a block)

5.4 Supported Messages

Only the following FIX message types are supported by LMAX FIX Gateway

Message Type ID	Message Type Name	Information
Session level messages		
A	Logon	Used for establishing a communication between initiating and accepting parties
0	Heartbeat	Monitors the status of the communication link
1	TestRequest	Forces a heartbeat from the opposing application
2	ResendRequest	Initiates the retransmission of messages by the receiving application
3	Reject	Session level message reject
4	SequenceReset	Reset the incoming and outgoing sequence numbers
5	Logout	Logout from current session
Application messages		
D	NewOrderSingle	Used to electronically submit orders for execution
8	ExecutionReport	Communicates the state of the order
F	OrderCancelRequest	Request to cancel an order in the market.
G	OrderCancelReplaceRequest	Request to amend an order in the market.
9	OrderCancelReject	Send in response to OrderCancelRequest or OrderCancelReplaceRequest messages if they cannot be fulfilled.
i	MassQuote	Places number of orders on the market based on specified depth. Available for market makers only.
Z	QuoteCancel	Cancels and existing quote within the market
b	QuoteAcknowledgement	Confirms the acceptance or rejects the mass quote order
e	SecurityStatusRequest	Requests the status of a security
f	SecurityStatus	Returns the status of a security.
c	SecurityDefinitionRequest	Requests Securities definition

d	SecurityDefinition	Returns the Security Definition
j	BusinessMessageReject	Business Reject

5.5 Message Permissions

The following table lists the Message exchange that is supported by LMAX and Clients where Y = Supported and N= Not supported

Tag 35 =	Message Type	LMAX		Market Maker	
		Send	Receive	Send	Receive
Session Level					
A	Logon	Y	Y	Y	Y
0	Heartbeat	Y	Y	Y	Y
1	Test Request	Y	Y	Y	Y
2	ResendRequest	Y	Y	Y	Y
3	Reject	Y	Y	Y	Y
4	SequenceReset	Y	Y	Y	Y
5	Logout	Y	Y	Y	Y
Application Level					
Reference Data					
c	SecurityDefinitionRequest	N	Y	Y	N
d	SecurityDefinition	Y	N	N	Y
e	SecurityStatusRequest	N	Y	Y	N
f	SecurityStatus	Y	N	N	Y
Trade					
D	NewOrderSingle	N	Y	Y	N
F	OrderCancelRequest	N	Y	Y	N
9	OrderCancelReject	Y	N	N	Y
G	OrderCancelReplaceRequest	N	Y	Y	N
8	ExecutionReport	Y	Y	N	Y

Tag 35 =	Message Type	LMAX		Market Maker	
		Send	Receive	Send	Receive
i	MassQuote	N	Y	Y	N
b	QuoteAcknowledgement	Y	N	N	Y
Z	QuoteCancel	Y	N	Y	N
Other					
j	BusinessMessageReject	Y	Y	Y	Y

5.6 Headers & Trailers

5.6.1 Standard Header

Each session or application message is preceded by a standard header. The header identifies the message type, length, destination, sequence number, origination point and time.

Tag	Field	Description	Required	Data Type	LMAX Supported Values / Information	Used in
8	BeginString	Identifies beginning of new message and protocol version, must be first field in message.	Y	String	FIX.4.2	All supported messages
9	BodyLength	Message length, in bytes, forward to the CheckSum (10)field, , must be second field in message	Y	Int		All supported messages
35	MsgType	Defines the type of message being sent or received, must be third field in message	Y	String	See supported message types	All supported messages
49	SenderCompID	Assigned value used to identify Client sending message.	Y	String		All supported messages

Tag	Field	Description	Required	Data Type	LMAX Supported Values / Information	Used in
56	TargetCompID	Assigned value used to identify the receiving Client.	Y	String	FIX-API	All supported messages
34	MsgSeqNum	Integer message sequence number.	Y	Int		All supported messages
43	PosDupFlag	Indicates possible retransmission of this sequence number. Always required for retransmitted messages, whether prompted by the sending system or as the result of a resend request.	Y ©	Boolean	Y=POSSIBLE DUPLICATE N=ORIGINAL TRANSMISSION	All supported messages
52	SendingTime	Time of message transmission	Y	UTC Timestamp (GMT)		All supported messages
122	OrigSendingTime	Original time of message transmission. Required for message resent as a result of a Resend Request (35=2)	Y ©	UTC Timestamp (GMT)		

5.6.2 Standard Trailer

Each message, session or application, is terminated by a standard trailer. The trailer is used to segregate messages and contains the three digit character representation of the *CheckSum* (10) value.

Tag	Field	Description	Required	Data Type	LMAX Supported Values / Information	Used in
10	CheckSum	Three byte, simple checksum, always last field in message. Always defined as three characters.	Y	String		All supported messages

5.7 Session Level Messages

Only the following FIX Session message types are accepted by LMAX FIX Gateway:

- Logon (A)
- Logout(5)
- Heartbeat(0)
- Test Request(1)
- Resend Request(2)
- Reject(3)
- Sequence Reset(4)

5.7.1 Connecting to LMAX Gateway – Logon (A)

The **Logon** message must be the first message Client sends after establishing a TCP connection on the port agreed upon with LMAX. *EncryptMethod* must be 0 – None. Client must wait for a **Logon** from LMAX before sending other messages and beginning gap fill operations. Please see section *3.4.2 – Logon Process State Transition Diagram* for more details.

Client must specify a heartbeat interval in the Logon message, which the LMAX FIX Gateway will use to determine if the connection is active. When logging on, the client requests a heartbeat interval, using the *HeartBtInt*.

5.7.1.1 Client Logon

The **Logon** message authenticates the Client establishing a connection to LMAX. For authentication purposes tag *SenderCompID* will be used as the **username**, and tag *RawData* will be used as the **password**.

Tag	Field	Description	Required	Data Type	LMAX Supported Values / Information
<Standard Message Header>			Y	MsgType = A	
98	EncryptMethod	Method of encryption	Y	Int	0 =None
108	HeartBtInt	Used to determine the heartbeat interval.(Seconds)	Y	Int	Integer >= 2 AND integer <=60 Default value = 30
95	RawDataLength	Number of bytes in *RawData* field.	Y	Int	
96	RawData	Unformatted raw data. Contains the password for the given Client.	Y	String	

Tag	Field	Description	Required	Data Type	LMAX Supported Values / Information
141	ResetSeqNumFlag	Indicates both sides of a FIX session should reset sequence numbers.	N	Boolean	Y= RESET SEQUENCE NUMBERS N= DON'T RESET SEQUENCE NUMBERS
<Standard Message Trailer>			Y		

5.7.1.2 LMAX Logon

Upon receipt of a **Logon** message from the Client requesting connection and successful authentication, LMAX will issue a **Logon** message as an acknowledgment that the connection request has been accepted.

Tag	Field	Description	Required	Data Type	LMAX Supported Values / Information
<Standard Message Header>			Y	MsgType = A	
98	EncryptMethod	Method of encryption	Y	Int	0 =None
108	HeartBtInt	Used to determine the heartbeat interval.(Seconds)	Y	Int	Matched value received from Client **Logon**
141	ResetSeqNumFlag	Indicates both sides of a FIX session should reset sequence numbers.	N	Boolean	Matched value received from Client **Logon**
<Standard Message Trailer>			Y		

5.7.2 Disconnecting from the LMAX FIX Gateway- Logout (5)

At the end of the day, the client must log off and disconnect from the LMAX FIX Gateway. Please see section *3.4.3 – Logout Process State Transition Diagram* for more details.

The **Logout** message initiates or confirms the termination of a FIX session. Before actually closing the session, the logout initiator should wait for the opposite side to respond with a confirming logout message. This gives the remote end a chance to perform any Gap Fill operations that may be necessary. Disconnection without the exchange of **Logout** messages should be interpreted as an abnormal condition.

5.7.2.1 Client Logout

The client logout message just requires header and trailer message tags.

When a client wishes to logout it should perform the following steps:

- Send a **TestRequest** message. This is to make sure that the sequence numbers have not got out of sync. If they have then a **ResendRequest** is sent to obtain the messages that were lost.
- Send a **Logout** message. After sending the **Logout** message, the logout initiator should not send any messages unless requested to do so by the logout acceptor via a **ResendRequest**
- Wait for the opposite side to respond with a confirming **Logout** message. The session may be terminated if the remote side does not respond in an appropriate timeframe.
- Disconnect.

5.7.2.2 LMAX Logout

LMAX will send a **Logout** message in following scenarios:

- As a response to a **Logout** request from a client.
- The logon or authentication of a connecting Client was unsuccessful.
- Incoming sequence number is lower than expected.
- LMAX exchange is closed

The reason for the logout will be provided in *Text* tag.

Tag	Field	Description	Required	Data Type	LMAX Supported Values / Information
<Standard Message Header>			Y	MsgType = 5	
58	Text	The reason for the logout	N	String	• password is incorrect: BAD CREDENTIALS • logon has failed 7 consecutive times due to bad password: ACCOUNT LOCKED • attempt to logon outside exchange operating hours: Exchange closed • Any other reasons: DISCONNECT • MsgSeqNum too low, expecting Y but received X
<Standard Message Trailer>			Y		

5.7.3 TestRequest (1)

The **TestRequest** message forces a **Heartbeat** response from the opposing application. LMAX uses the **TestRequest** message to actively determine if the client connection is alive and the networks connecting the Client to the LMAX Exchange are functional. To determine if the session is still active LMAX will send a **TestRequest** with *TestReqID* value specified. On receipt of a **TestRequest** message, the Client should send a **Heartbeat** message containing corresponding *TestReqID*.

If the LMAX FIX Gateway receives a **Heartbeat** response with matching *TestReqID* within *2 x heartbeat+2 seconds*, then the session will be kept open and all orders will stay working in the market.

In the event of a failure to receive the **Heartbeat** message with matching *TestReqID* within the required interval, all the client's working orders on the LMAX Exchange will be cancelled and the LMAX FIX Gateway will drop the connection.

If heartbeat time specified less than 2 sec, the Cancel On Disconnect behavior is disabled by default.

Tag	Field	Description	Required	Data Type	LMAX Supported Values / Information
<Standard Message Header>			Y	MsgType = 1	
112	TestReqID**	Identifier included in **TestRequest** message to be returned in resulting **Heartbeat**	Y	String	
<Standard Message Trailer>			Y		

***The value will be returned in the resulting Heartbeat.*

5.7.4 Heartbeat (0)

The Heartbeat message matches the status of the communication link and identifies when the last of a string of messages was not received.

Tag	Field	Description	Required	Data Type	LMAX Supported Values / Information
<Standard Message Header>			Y	MsgType = 0	
112	TestReqID	Identifier included in **TestRequest** message to be returned in resulting **Heartbeat**	Y ©	String	Required when the heartbeat is the result of a **TestRequest** message. In this case, this tag must contain the *TestReqID* that was sent in the **TestRequest** message.
<Standard Message Trailer>			Y		

5.7.5 ResendRequest (2)

The **ResendRequest** is sent by the receiving application to initiate the retransmission of messages. This functionality is utilized if a sequence number gap is detected or as a part of the initialization process.

The **ResendRequest** can be used in following scenarios:

- To request a single message: *BeginSeqNo* = *EndSeqNo*
- To request a range of messages: *BeginSeqNo* = first message of the range, *EndSeqNo* = last message of the range
- To request all messages subsequent to a particular message: *BeginSeqNo*= first message of range, *EndSeqNo* = 0 (represents infinity)

The LMAX FIX Gateway will automatically send a **ResendRequest** if out of sequence message is received (sequence number too high). When the LMAX FIX Gateway processes resent messages it will ignore the *PossDupFlag* and process all messages as new irrespective of the value of this tag, therefore it is recommended that clients always send **SequenceReset** -**Gap Fill** when processing resent requests made by LMAX.

The LMAX FIX Gateway will set the *PossDupFlag* to 'Y' when resending messages to the client. The receiving application should process this message as follows: if a message with this sequence number has been previously received then ignore this message; if message with this sequence number hasn't been received then it process normally.

Tag	Field	Description	Required	Data Type	LMAX Supported Values / Information
<Standard Message Header>			Y	MsgType = 2	
7	BeginSeqNo	Beginning message sequence number	Y	Int	
16	EndSeqNo	Ending sequence number	Y	Int	0 = infinity
<Standard Message Trailer>			Y		

5.7.6 SequenceReset (4)

If the sequence numbers of messages become unsynchronized, the **SequenceReset** message is sent to resynchronize the message sequence numbers between the LMAX FIX Gateway and the Client. The **SequenceReset** message is used by the sending application to reset the incoming sequence number on the opposing side.

The **SequenceReset** message only allows increasing the sequence number. If **SequenceReset** message is received attempting to decrease the next expected sequence number the message should be rejected and treated as a serious error.

During the **SequenceReset** process, administrative messages should not be retransmitted. Instead, a special **SequenceReset-GapFill** message is generated. The administrative messages which are not to be resent are: **Logon**, **Logout**, **ResendRequest**, **Heartbeat**, **TestRequest** and **SequenceReset**.

Tag	Field	Description	Required	Data Type	LMAX Supported Values / Information
<Standard Message Header>			Y	MsgType = 4	
123	GapFillFlag	Indicates that sequence reset will replace administration or application messages that will not be resent	N	String	Y=Gap Fill message, *MsgSeqNum* field valid N=Sequence Reset, ignore *MsgSeqNum*
36	NewSeqNo	The new sequence number	Y	Int	
<Standard Message Trailer>			Y		

5.7.7 Reject (3)

SessionLevelReject message is sent when a message is received but cannot be properly processed by the session level. For example if the message received is formatted incorrectly or is missing a mandatory field.

LMAX will send the **SessionLevelReject** in the following scenarios:

- Required tag is missing
- Value supplied is incorrect data format
- Value supplied is correctly formatted but out of supported by LMAX range

Tag	Field	Description	Required	Data Type	LMAX Supported Values / Information
<Standard Message Header>			Y	MsgType = 3	
45	RefSeqNum	Reference message sequence number of rejected message	Y	Int	
371	RefTagID	The tag number of the FIX field being referenced.	N	Int	
372	RefMsgType	The *MsgType* of the FIX message being referenced.	N	Int	
373	SessionRejectReason	Code to identify reason for a session-level **Reject** message.	N	Int	0= Invalid tag number 1= Required tag missing 2= Tag not defined for this message type 5=Value is incorrect (out of range) for this tag 6=Incorrect data format for value 11=Invalid *MsgType*
58	Text	Free format text message to explain reason for rejection	N	String	Invalid tag number Required tag missing Tag not defined for this message type Value is incorrect (out of range) for this tag Incorrect data format for value Invalid *MsgType*
<Standard Message Trailer>			Y		

5.8 Application Level Messages

5.8.1 Overview

This section describes the FIX Application messages supported by LMAX platform. Below is a quick legend to the Message tables that follow:

Tag	Field	Description	Required	Data Type	LMAX Supported Values / Information	Instrument Type
Tag No	Field Name	Field Description	Y = Required Y © = Required Conditionally N = Not Required C = Conditional	Supported Data type for the field	Valid Values for Field	ALL – All instruments FX = Foreign Exchange I= CFD Indices R = CFD Rates C= CFD Commodities

5.9 Placing Order

LMAX supports 2 messages for submitting orders for any single instrument available for trading on the LMAX platform.

1. NewOrderSingle message allows the placement of a single order of specified size, price and side.
   - OrderCancelReplace message can be used to cancel an existing order and replace it with a new order.
   - OrderCancelRequest message can be used to cancel an existing order.
2. MassQuote message allows the placement of up to 12 orders (maximum 6 on each side) for a specified quantity, price and side for each order.
   - The placement of a new MassQuote message will replace the previous quote.
   - QuoteCancel can be used to remove an existing quote.
   - An “empty” MassQuote (with no prices and quantities) can be used to remove an existing quote.

5.9.0.1 Order Rejections

In the event that an OrderCancelReplace message is sent to modify an existing order, but is rejected (e.g. due to incorrect IDs, incorrect formatting), the initial order will still be active on the LMAX platform. It is advised that if an OrderCancelReplace message is rejected, for any reason, a subsequent OrderCancelRequest message is sent to remove the initial order.

In the event that a MassQuote is sent to replace a previous MassQuote, but is rejected (e.g. due to incorrect IDs, incorrect formatting, incorrect levels of depth), the initial MassQuote will still be active on the LMAX platform. It is advised that if a MassQuote is rejected, for any reason, and there is an existing MassQuote on the platform, a QuoteCancel or “empty” MassQuote is sent to remove the existing MassQuote.

5.9.0.2 Volatility Bands

In order to protect Clients from pricing errors LMAX references the price of each incoming order against their last accepted order submitted for that side. If the price difference between the previous accepted price and the new price is outside the volatility limit, LMAX will reject the order with an OUTSIDE_VOLATILITY_BAND message. LMAX will only accept a new price if it is within the volatility limit, or if LMAX Market Operations manually increase the volatility limit to allow the new price to be accepted.

Please see the example below that demonstrates this behavior.

Six orders have been submitted by a client to the market for a given instrument and side. The volatility limit for the instrument is set to 5%.

1. Order A 10@100 - accepted
2. Order B 10@101 - accepted
3. Order C 20@110 – rejected as 8.91% difference compared to previous accepted price (101)
4. Order D 20@105 – accepted as 3.96% difference compared to previous accepted price (101)
5. Order E 20@111- rejected as 5.71% difference compared to previous accepted price (105)
6. Order F 20@110 – accepted as 4.76% difference compared to previous accepted price (105)

Please note that on market opening, the first price of the trading session is compared to the LMAX closing price of the previous trading day. If other clients are pricing, the first price by a client is compared to the last accepted price from other clients.

5.9.0.3 Message Rate Threshold

In order to manage the number of messages that the LMAX Platform can process at any given point, a Message Rate Threshold is imposed on each FIX session that acts as a limit to the number of messages the FIX session can send to LMAX in one second. If the Message Rate Threshold is breached, LMAX will force the session to disconnect by sending a Logout with a “Number of messages exceeds the threshold” message. LMAX will block any subsequent Logon attempts for 15 seconds. Message Rate Thresholds are determined by LMAX. They are based on the number of instruments and levels that each FIX session will price.

5.9.1 NewOrderSingle (D)

In addition to requirements for the FIX message header, only the following fields are used by the application layer for a New Order – Single message.

Tag	Field	Description	Required	Data Type	LMAX Supported Values / Information	Instrument Type
<Standard Message Header>			Y	MsgType = D		
1	Account	Account ID of the current logged in LMAX Client	N	String		
11	ClOrdID	Unique identifier of the order as assigned by institution.	Y	String	Please see section on *ClOrdID* format requirements	
21	HandlInst	Handle Instruction	Y	Char	1=AUTOMATED EXECUTION ORDER PRIVATE NO BROKER INTERVENTION	
38	OrderQty	Order Quantity	Y	Qty	Quantity specified in Notional amounts by default. Can be set as LMAX contracts if requested.	
40	OrdType	Order Type	Y	Char	1=MARKET 2=LIMIT	
44	Price	Price per contract	Y ©	Price	Required only when entering the following Order Type:2 = Limit	
22	IDSource	Identifies class of alternative SecurityID	N	String	8 = Exchange Symbol	
48	SecurityID	LMAX Security ID	N	String	Should be the security id of the instrument downloaded from security definition.	
54	Side	Buy or Sell indicator	Y	Char	1=BUY 2=SELL	
55	Symbol	The underlying symbol for the contract	Y	String		

Tag	Field	Description	Required	Data Type	LMAX Supported Values / Information	Instrument Type
59	TimeInForce	Specifies how long the order remains in effect.	Y	Char	0=Good for Day (GFD) 3=Immediate or Cancel (IOC) 4=Fill or Kill (FOK) Limit Orders support TimeInForce values of IOC, FOK, and GFD. Market Orders support TimeInForce values of IOC, FOK.	
60	TransactTime	Time of execution/order creation	Y	UTCTimestamp	Time of order creation	
440	ClearingAccount	Supplemental account information forwarded to central counter party.	N	String		
<Standard Message Trailer>			Y			

5.9.1.1 Order State Transition

Client can cancel his/her orders that have been previously entered and still exist. In this instance LMAX will inform the Client of the new status of the order (i.e. cancelled). LMAX may also reject the cancellation request based on each request's individual status (i.e. invalid or the order has already been completed)

As the order state changes, either due to Instructions received over FIX (e.g. **OrderCancelRequest, OrderCancel/ReplaceRequest**) or matching occurring on the Order Book, the Order will transition through the Order State Diagram, and an **ExecutionReport** will be generated for each transition. **ExecutionReport** message is used to convey the current state of the order. It contains two fields which are used to communicate the current state of the order – *OrdStatus,* and the purpose of the message - *ExecType*.

The sample workflow for an order is shown in an example below; order is sent to LMAX via the LMAX Gateway and execution reports are returned informing the client of the status of the order. In the example below 2 types of order matches are used: aggressive – when the order matches retail orders on entry and passive – order is waiting to be matched on the Order Book.

An example

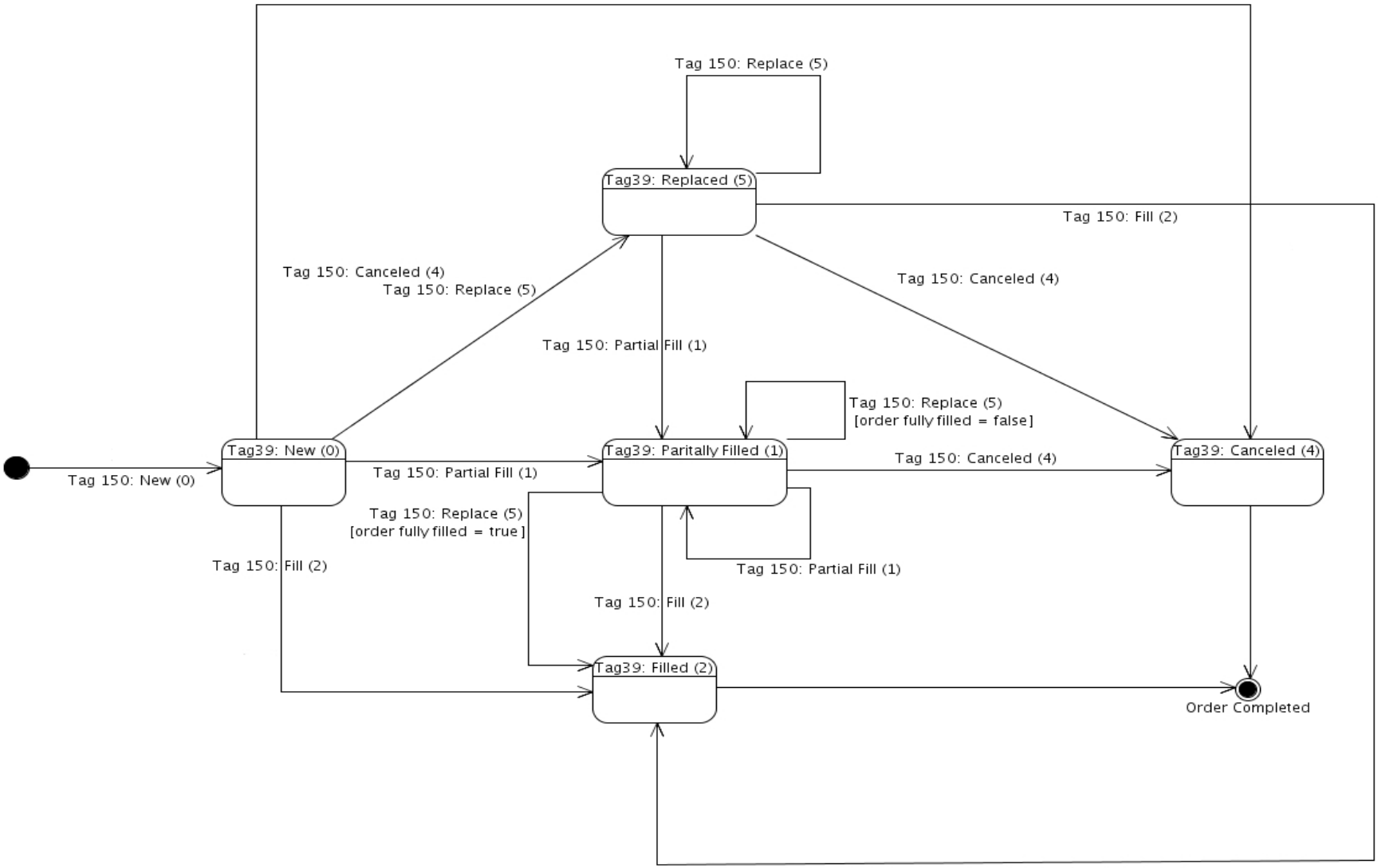
1. LMAX Client places a new Order
2. Order gets aggressively part filled against two counter orders.
3. The remainder of the Order is waiting on the Order Book. It is later being passively matched by another single counter order which causes the Order to be fully filled.

Time	Message Received	Message Sent	ExecType(150)	OrdStatus(39)	Comment
1	New Order				
2		ExecutionReport	New(0)	New(0)	Order received
2		ExecutionReport	Partial Fill(1)	Partially Filled(1)	Aggressive match against 1st counter order
2		ExecutionReport	Partial Fill(1)	Partially Filled(1)	Aggressive match against 2nd counter order
3		ExecutionReport	Fill(2)	Filled(2)	Passively matched by another aggressive counter order which fully fills the order.

If an order simultaneously exists in more than one order state, the value with highest precedence is the value that is reported in the *OrdStatus* tag.

The order statuses precedence is displayed in the table below (in highest to lowest precedence)

Precedence	OrdStatus (39)	Description
5	Filled	Order completely filled, no remaining quantity
4	Canceled	Canceled order with or without executions
3	Partially Filled	Outstanding order with executions and remaining quantity
2	Replaced	Replaced order with or without executions
1	New	Outstanding order with no executions
1	Rejected	Order has been rejected by LMAX.


Tag 150: Replace (5)
Tag39: Replaced (5)
Tag 150: Fill (2)
Tag 150: Canceled (4)
Tag 150: Replace (5)
Tag 150: Canceled (4)
Tag 150: Partial Fill (1)
Tag 150: Replace (5)
[order fully filled = false]
Tag39: New (0)
Tag39: Paritally Filled (1)
Tag39: Canceled (4)
Tag 150: New (0)
Tag 150: Partial Fill (1)
Tag 150: Canceled (4)
Tag 150: Replace (5)
[order fully filled = true]
Tag 150: Fill (2)
Tag 150: Partial Fill (1)
Tag 150: Fill (2)
Tag39: Filled (2)
Order Completed

5.9.2 OrderCancelRequest (F)

In addition to requirements for the FIX message header, the following fields are used by the application layer for an **OrderCancelRequest** message. Client can use this message to cancel existing orders placed with LMAX.

Tag	Field	Description	Required	Data Type	LMAX Supported Values / Information	Instrument Type
<Standard Message Header>			Y	MsgType = F		
1	Account	Account ID of the current logged in session	N	String		
11	ClOrdID	Unique *ClOrdID* for this cancel.	Y	String	Please see section on *ClOrdID* format requirements	
41	OrigClOrdID	The original *ClOrdID* of the order to cancel.	Y	String		
22	IDSource	Identifies class of alternative *SecurityID*.	N	String	8 = Exchange Symbol	
48	SecurityID	LMAX Security Id	N	String	Should be the security id of the instrument downloaded from security definition.	
54	Side	Buy or Sell indicator. Ignored for validation.	Y	Char	1 = Buy 2 = Sell	
55	Symbol	The underlying symbol for the contract. Ignored for validation.	Y	String		
60	TransactTime	Time this order request was initiated/ released by the trader or trading system.	Y	UTCTimestamp		
<Standard Message Trailer>			Y			

5.9.3 OrderCancelReplaceRequest (G)

In addition to requirements for the FIX message header, the following fields are used by the application layer for **OrderCancelReplaceRequest** message. The order cancel/replace request is used to change the parameters of an existing order in the market.

Tag	Field	Description	Required	Data Type	LMAX Supported Values / Information	Instrument Type
<Standard Message Header>			Y	MsgType = G		
1	Account	Account ID of the current logged in LMAX Client	N	String		
11	ClOrdID	Unique identifier of the order as assigned by institution.	Y	String	Please see section on ClOrdID format requirements	
21	HandlInst	Handle Instruction	Y	Char	1=AUTOMATED EXECUTION ORDER PRIVATE NO BROKER INTERVENTION	
22	IDSource	Identifies class of alternative *SecurityID*	N	String	8 = Exchange Symbol	
48	SecurityID	LMAX Security Id	N	String	Should be the security id of the instrument downloaded from security definition.	
38	OrderQty	Number of contracts	Y	Qty		
40	OrdType	Order Type	Y	Char	2 = Limit	
41	OrigClOrdID	ClOrdID of the previous order (NOT the initial order of the day) as assigned by the institution, used to identify the previous order.	Y	Char		
44	Price	Price per contract	Y	Price		
54	Side	Buy or Sell indicator	Y	Char	1 = Buy 2 = Sell	
55	Symbol	The underlying symbol for the contract	Y	String		
59	TimeInForce	Specifies how long the order remains in effect.	Y	Char	0 = Good for Day (GFD)	
60	TransactTime	Time of execution/order creation	Y	UTCTimestamp	Time of order creation	

Tag	Field	Description	Required	Data Type	LMAX Supported Values / Information	Instrument Type
440	ClearingAccount	Supplemental accounting information forwarded to clearing house/firm.	N	String		
<Standard Message Trailer>			Y			

5.9.4 OrderCancelReject (9)

The **OrderCancelReject** message is used when an **OrderCancelRequest** or **OrderCancel/ReplaceRequest** cannot be honoured. Message requests that change the price or quantity of the order are only granted when an outstanding quantity exists. Filled orders cannot be changed.

Reasons for rejecting a **CancelOrderRequest** are outlined below.

Reason for Cancel Reject	Order Cancel Request	Order Cancel Replace Request
Order was not found	Yes	Yes
Order does not exist because it was already filled or Cancelled	Yes	Yes
Cancel can't be routed to the target system for a technical reason	Yes	Yes
New order price exceeds maximum allowed for instrument	No	Yes
New order price exceeds minimum allowed for instrument	No	Yes
New order price is not multiple of the price increment for the Instrument (too many decimal places specified)	No	Yes
New order price is not numeric format	No	Yes
Attempting to change the side of the order	No	Yes
New quantity is not a multiple of quantity increment**	No	Yes
New quantity is not in numeric format	No	Yes
Attempting to Cancel/Replace an Order with anything other than a GFD Limit Order	No	Yes
New price of the order is outside the transient price filter (is more that a defined percentage of the last two prices on the same side of the book)	No	Yes
Specified *Security Id* or *Symbol* is Invalid	Yes	Yes
New order has duplicate *ClOrdId* to the existing working Order on the same Order Book	No	Yes

** ***Quantity increments vary across LMAX Instruments – please request the LMAX Product Files for more information.***

In addition to requirements for the FIX message header, only the following fields are used by the application layer for **OrderCancelReject** message.

Tag	Field	Description	Required	Data Type	LMAX Supported Values / Information	Instrument Type
<Standard Message Header>			Y	MsgType = 9		
1	Account	Account ID of the current logged LMAX Client	N	String		
11	ClOrdID	Unique identifier of the order as assigned by institution.	Y	String		
102	CxlRejReason		N	String	1=Unknown order 2=Broker option	
434	CxlRejResponseTo		Y	String	1=Order Cancel Request 2=Order Cancel Replace Request	
37	OrderID		Y	String	NONE	
39	OrdStatus	OrdStatus value after this cancel reject is applied.	Y	Char		
41	OrigClOrdID		Y	String	PRICE IS INVALID QUANTITY IS INVALID INVALID ORDER INSTRUCTION OUTSIDE VOLATILITY BAND INSTRUMENT DOES NOT EXIST INVALID INSTRUMENT SYMBOL INSTRUMENT IS NOT OPEN INSTRUMENT IS SUSPENDED DUPLICATE ORDER	
58	Text		N	String	Reason for Rejection	
<Standard Message Trailer>			Y			

5.9.5 Mass Quote (i)

The **MassQuote** message is used to submit multiple orders at different prices onto both sides of the market. **MassQuote** only supports quoting on a single instrument. Only one **MassQuote** per Instrument may exist at any given time in LMAX system. A subsequent **MassQuote**, if accepted, will replace the previous **MassQuote** and will cancel all the unmatched liquidity associated with the replaced **MassQuote**.

MassQuote supports maximum of 6 quote entries for each side. **MassQuote** will be rejected if number of quote entries is greater than 6 on either/both side(s).

Once the Mass Quote message is submitted to a market and after the initial Mass Quote Acknowledgement – the Quote Entries are treated as single orders. The trades generated for each of the Quote Entry in the **MassQuote** order will be reported with **ExecutionReport** Message (*Section 5.9.8.2 – Execution Report-Mass Quote Orders)*.

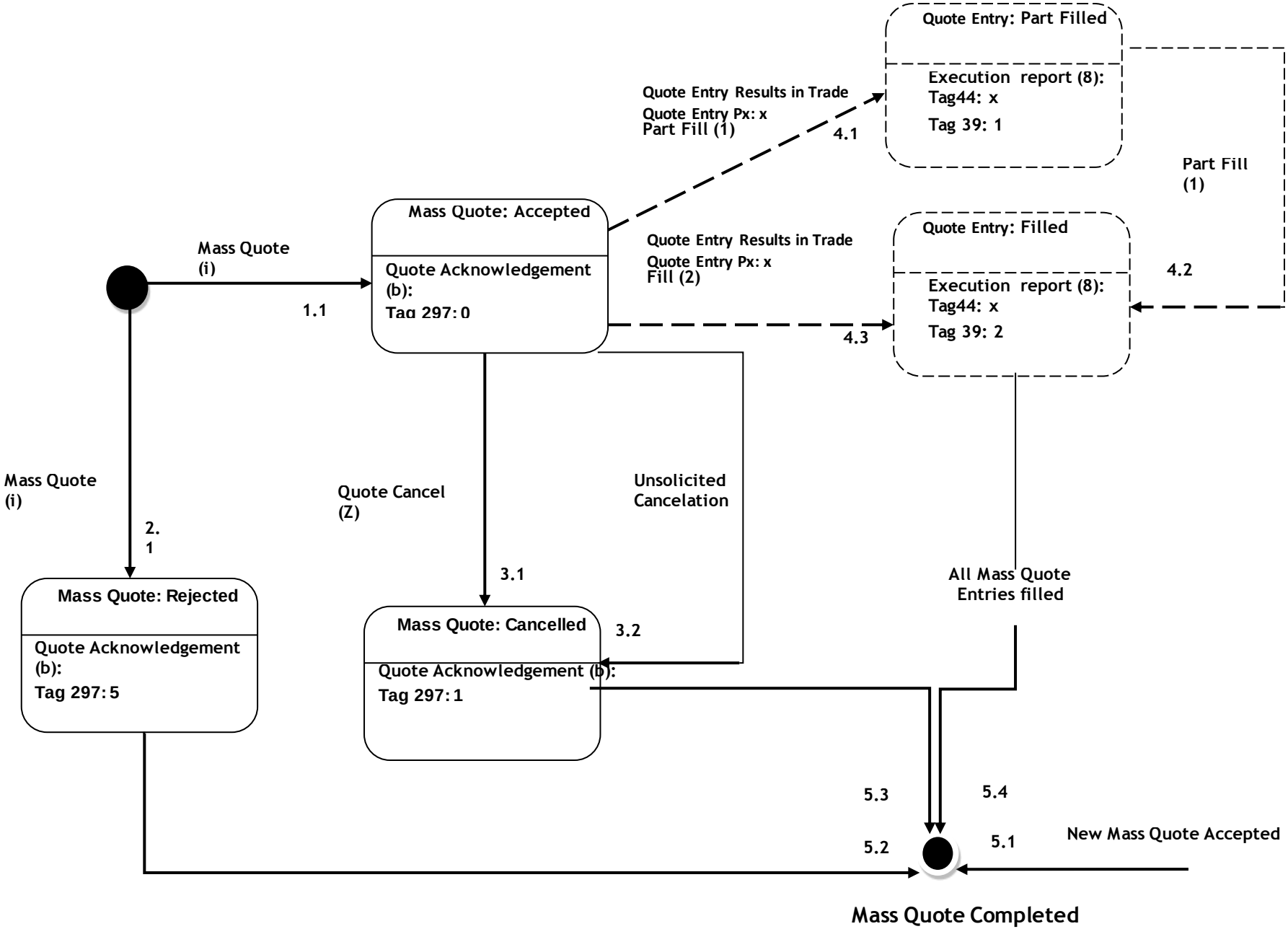
Please see section 4.4.2 (Mass Quote Reference) for more details on how the individual QuoteEntry transactions mapped to the original Mass Quote.
In addition to requirements for the FIX message header, only the following fields are used by the application layer for a **MassQuote** message.

Tag	Field	Description	Required	Data Type	LMAX Supported Values / Information	Instrument Type
<Standard Message Header>			Y	MsgType = i		
117	QuoteID	Unique identifier for this MassQuote. This will be returned on execution reports as *ClOrdID* plus an arbitrary character to represent each individual order	Y	String	Please see *section* on *ClOrdID* and *QuoteID* format requirements. The sender must guarantee that *QuoteID* is unique for given Instrument. LMAX will not verify the uniqueness of the *QuoteID*.	
296	NoQuoteSets	Number of sets of orders within this quote message	Y	Int	1	
➢302	QuoteSetID	Sequential number of the Quote set	Y	Int	We only support single quote sets. So the default value of 1 has to be used.	
➢311	UnderlyingSymbol	Underlying security's *Symbol*.	Y	String		
➢309	UnderlyingSecurityID	Underlying security's *SecurityID*.	N	String	Should be the security id of the instrument downloaded from security definition.	

Tag	Field	Description	Required	Data Type	LMAX Supported Values / Information	Instrument Type
➢305	UnderlyingIDSource	Underlying security's IDSource	N	Int	8=EXCHANGE SYMBOL Needs to be included if UnderlyingSecurityID (309) is used	
➢304	TotQuoteEntries	Total number of Quotes set across all messages, should be the sum of NoQuoteSets	Y	Int	Total number of quotes within this quote set. Max. of 6 entries supported – to represent 6 for each side of the order. 0 to indicate empty MQ.	
➢295	NoQuoteEntries	Number of quotes for this *Symbol*	Y	Int	Number of quotes within this quote set. Max. of 6 entries supported – to represent 6 for each side of the order. 0 to indicate empty MQ.	
➢➢299	QuoteEntryID	Uniquely identifies the quote as part of the Quote Set	Y ©	Int	Identifier of the Quote Entry. Max 6 for each quote. Omit for empty MQ.	
➢➢132	BidPx	Bid price, Required if *BidSize* is specified	Y ©	Price		
➢➢133	OfferPx	Offer price. Required if *OfferSize* is specified	Y ©	Price		
➢➢134	BidSize	Bid size. Required if *BidPx* is specified	Y ©	Int		
➢➢135	OfferSize	Offer size. Required if *OfferPx* is specified	Y ©	Int		
440*	ClearingAccount	Supplemental accounting information forwarded to clearing house/firm.	N	String		CCP cleared instruments
<Standard Message Trailer>			Y			

5.9.5.1 Mass Quote State Transition Diagram

As the Mass Quote state changes, either due to New Mass Quote replacing the previous one, **QuoteCancel** request**,** validations failure, the Exchange events or trades occurring on the Order Book, the Mass Quote will transition through the Mass Quote State Diagram, and a **Quote Acknowledgement** or an **ExecutionReport** will be generated for each transition.


Quote Entry: Part Filled
Execution report (8):
Tag44: x
Tag 39: 1
Quote Entry Results in Trade
Quote Entry Px: x
Part Fill (1)
4.1
Part Fill
(1)
Mass Quote: Accepted
Quote Acknowledgement
(b):
Tag 297: 0
Mass Quote
(i)
1.1
Quote Entry Results in Trade
Quote Entry Px: x
Fill (2)
Quote Entry: Filled
Execution report (8):
Tag44: x
Tag 39: 2
4.2
4.3
Mass Quote
(i)
2.
1
Quote Cancel
(Z)
3.1
Unsolicited
Cancelation
All Mass Quote
Entries filled
Mass Quote: Rejected
Quote Acknowledgement
(b):
Tag 297: 5
Mass Quote: Cancelled
3.2
Quote Acknowledgement (b):
Tag 297: 1
5.3
5.4
5.2
5.1
New Mass Quote Accepted
Mass Quote Completed

5.9.6 QuoteCancel (Z)

QuoteCancel request is used to cancel an existing **MassQuote** for a given instrument by originator of the quote. If **QuoteCancel** request was successful **QuoteAcknowledgement** with *QuoteAckStatus = 1(Cancelled)* will be sent in response.

In addition to requirements for the FIX message header, only the following fields are used by the application layer for a **QuoteCancel** message.

Tag	Field	Description	Required	Data Type	LMAX Supported Values / Information	Instrument Type
<Standard Message Header>			Y	MsgType = Z		
117	QuoteID	Unique identifier for the quote to be canceled as provided in *QuoteID* of original quote. This will be returned in **QuoteAcknowledgment** in *QuoteID* field	Y	String		
298	QuoteCancelType	Type of quote cancel	Y	Int	1	
295	NoQuoteEntries		Y	Int	1	
➢55	Symbol	Underlying Securities' Symbol	Y	Symbol		
➢48	SecurityID	Underlying Securities' ID	N	String		
<Standard Message Trailer>			Y			

5.9.7 Quote Acknowledgement (b)

This acknowledgement message is sent in response to a **MassQuote** being rejected, cancelled or accepted.

MassQuote can be cancelled only in the following scenarios:

- **QuoteCancel** request has been issued by the **MassQuote** originator
- Unsolicited cancellation of **MassQuote** by LMAX
- When new **MassQuote** has been placed, any outstanding quantity in the previous **MassQuote** for a given instrument will be cancelled.

In the last scenario only one **QuoteAcknowledgement** will be sent to notify the user that the new **MassQuote** has been accepted/rejected. In case when the new **MassQuote** has been accepted, no **QuoteAcknowledgement** will be sent to report cancelled **MassQuote**.

In the event of unsolicited cancelation of the MassQuote by LMAX **QuoteAcknowledgement** will be send to the client with *QuoteAckStatus = 1 (Cancelled).*

In the **QuoteAcknowledgement** for unsolicited MassQuote cancel *QuoteID* contains the most recently known *QuoteID* for the MassQuote that has been cancelled.

Mass Quote can be rejected in the following scenarios:

- Required or conditionally required tag is missing
- Value provided in a tag is not supported by LMAX (e.g. invalid format, not within supported bounds)
- Instrument is not available for trading (suspended, expired, closed for trading)
- Invalid prices are specified (price does not match the price increment for the instrument, outside the allowable bounds for the instrument, duplicate prices for one side, inverted prices, unsupported format)
- Quantity does not match the quantity increment for the instrument.
- Instrument cannot be identified due to the invalid values supplied in either *UnderlyingSymbol* or UnderlyingSecurityID, or both.

In addition to requirements for the FIX message header, only the following fields are used by the application layer for a Quote Acknowledgement message.

Tag	Field	Description	Required	Data Type	LMAX Supported Values / Information	Instrument Type
<Standard Message Header>			Y	MsgType = b		
297	QuoteAckStatus	The status of the quote	Y	Char	0=ACCEPTED 1=CANCELLED 5=REJECTED	
117	QuoteID	Unique identifier for the **MassQuote**, references *QuoteID* of the original **MassQuote**.	Y	String	The sender must guarantee that *QuoteID* is unique for given Instrument. LMAX will not verify the uniqueness of *QuoteID*.	

Tag	Field	Description	Required	Data Type	LMAX Supported Values / Information	Instrument Type
300	QuoteRejectReason	Reason Quote was rejected	N	Char	1=UNKNOWN SYMBOL 2=EXCHANGE SECURITY CLOSED 4=TOO LATE TO ENTER 5=UNKNOWN QUOTE 6=DUPLICATE QUOTE 8=INVALID PRICE 99 =OTHER*	
58	Text	Reason for Rejection	N	String		
296	NoQuoteSets	Number of sets of orders within this quote message	N	Int	1	
➢302	QuoteSetID	Sequential number of the Quote sets acknowledged	N	Int	1	
➢305	UnderlyingIDSource	Underlying security's IDSource	N	Int	8 = Exchange Symbol	
➢309	UnderlyingSecurityID	Underlying security's SecurityID.	N	String		
➢311	UnderlyingSymbol	Underlying security's Symbol.	N	String		
<Standard Message Trailer>			Y			

5.9.8 Execution Report (8)

ExecutionReport messages are used to confirm the successful processing of an instruction places or amends an order and also to report any fills which occur for an order, including fills for orders placed with the **MassQuote** message.

Each execution report message contains an *ExecID* which uniquely identifies the execution of an instruction on the LMAX Exchange. When the execution report is reporting a trade, then the *ExecID* is reported to the central counterparty (Clearing House) and can be used to reconcile the trade during settlement. *ExecID* is globally unique across all Order Books and over all time. The *ExecID* reporting a particular trade is also reported over the LMAX drop copy feed, if this is used.

Table below provides functions that **ExecutionReport** carries out for a Single order or MassQuote order

Function	Single Order	Mass Quote Order
Confirms the receipt or rejection of an order	Yes	No
Confirms the amendments/cancellation to an existing order	Yes	No
Provides fill information on working orders	Yes	Yes

5.9.8.1 Execution Report - Single Orders

In addition to requirements for the FIX message header, only the following fields are used by the application layer for **ExecutionReport** message.

Tag	Field	Description	Required	Data Type	LMAX Supported Values / Information	Instrument Type
<Standard Message Header>			Y	MsgType = 8		
1	Account	Account ID of the current logged in Client	Y	String		
6	AvgPx	Calculated average price of all fills on this order.	Y	Price	The value will always be zero	
11	ClOrdID	Client side order identifier	Y	String	Please see section on *ClOrdID and QuoteID Formatting* on *ClOrdID* format requirements In case of unsolicited order cancels populated with the value of the last known *ClOrderID* for the order that is modified/cancelled.	
14	CumQty	Contains the cumulated traded quantity for the order through its life.	Y	Qty		
17	ExecID	Execution ID for this fill.	Y	String	*ExecID* is sent to the central counter party for clearing the underlying asset.	
20	ExecTransType		Y	Char	0=NEW	
150	ExecType	Describes the type of execution report. Same possible values as Order Status.	Y	Char	0=NEW 1=PARTIALLY FILL 2=FILL 4=CANCELED 5=REPLACE 8=REJECTED	

21	HandlInst	Handle Instruction	N	Char	1=AUTOMATED EXECUTION ORDER PRIVATE NO BROKER INTERVENTION	
22	IDSource	Identifies class of alternative *SecurityID*	Y	Char	8=EXCHANGE SYMBOL	
31	LastPx	Price of this fill	N	Price		
32	LastShares	Quantity of contracts bough/sold on this last fill	N	Int		
151	LeavesQty	Amount of shares open for further execution	Y	Qty		
37	OrderID	Unique identifier for Order as assigned by exchange	Y	String		
103	OrderRejReason	Code to identify reason for order rejection.	N	Char	0=BROKER OPTION 1=UNKNOWN SYMBOL 2=EXCHANGE CLOSED 5=UNKNOWN ORDER 6=DUPLICATE ORDER	
38	OrderQty	Number of contracts submitted by the client	Y	Qty		
39	OrderStatus	Identifies the status of the order	Y	Char	0=NEW 1=PARTIALLY FILLED 2=FILLED 4=CANCELED 5=REPLACED 8=REJECTED	
41	OrigClOrdID	Original *ClOrdID*	N	String	Not populated in case of unsolicited order cancel.	
44	Price	Price	N	Price		
48	SecurityID	The ID of the security	N	Int		
54	Side	Side of the order	Y	Char	1=BUY 2=SELL	
55	Symbol	Symbol identifier	Y	String		
58	Text	Free format text string, can be used for rejects or information	N	String		
59	TimeInForce	Specifies how long the order remains in effect.	N	Char	0=DAY 3=IOC 4=FOK	

60	TransactTime	Time of execution/order creation	N	UTC Timestamp		
75	TradeDate	Logical Trade Date (YYYYMMDD)	Y©	Date	e.g. 20140429 – sent for Trade Executions only	
64	FutSettDate	Trade Settlement Date (YYYYMMDD)	Y©	Date	e.g. 20140501– sent for Trade Executions only	
<Standard Message Trailer>			Y			

5.9.8.2 Execution Report – Mass Quote Orders

In addition to requirements for the FIX message header, only the following fields are used by the application layer for **ExecutionReport** message.

Tag	Field	Description	Required	Data Type	LMAX Supported Values / Information	Instrument Type
<Standard Message Header>			Y	MsgType = 8		
1	Account	Account ID of the current logged in Client	Y	String		
6	AvgPx	Calculated average price of all fills on this order.	Y	Price	The value will always be zero	
11	ClOrdID	Client side order identifier for each individual order within MassQuote	Y	String	Populated with *QuoteID* values from originating **MassQuote**, plus arbitrary character defined by LMAX to represent each individual order within MassQuote (see 4.4.2)	
14	CumQty	Contains the cumulated traded quantity for the order at the given *Price* level (tag 44) throughout its life.	Y	Qty		
17	ExecID	Execution ID for this fill.	Y	String	*ExecID* is sent to the central counter party for clearing the underlying asset.	
20	ExecTransType		Y	Char	0=NEW	
150	ExecType	Describes the type of execution report. Same possible values as OrdStatus.	Y	Char	1=PARTIALLY FILLED 2=FILLED	
22	IDSource	Identifies class of alternative	N	Char	8=EXCHANGE SYMBOL	

Tag	Field	Description	Required	Data Type	LMAX Supported Values / Information	Instrument Type
		SecurityID				
31	LastPx	Price of this fill for the order at the given *Price* level (tag 44)	N	Price		
32	LastShares	Quantity of contracts bough/sold on this last fill for the order at the given *Price* level (tag 44)	N	Int		
151	LeavesQty	Amount of shares open for further execution for the order at the given *Price* level (tag 44) throughout its life.	Y	Qty		
37	OrderID	Unique identifier for Order as assigned by LMAX	Y	String		
38	OrderQty	Number of contracts submitted by the client for the order at the given *Price* level (tag 44)	N	Qty		
39	OrderStatus	Identifies the status of the order	Y	Char	1=PARTIALLY FILLED 2=FILLED	
44	Price	Price for the bid or offer order at the given Price level	Y	Price		
48	SecurityID	The LMAX ID of the security	Y	Int		
54	Side	Side of the order	Y	Char	1=BUY 2=SELL	
55	Symbol	Symbol identifier	Y	String		
59	TimeInForce	Specifies how long the order remains in effect.	N	Char	0=DAY	
60	TransactTime	Time of execution/order creation	N	UTC Timestamp		
75	TradeDate	Logical Trade Date (YYYYMMDD)	Y©	Date	e.g. 20140429 – sent for Trade Executions only	
64	FutSettDate	Trade Settlement Date (YYYYMMDD)	Y©	Date	e.g. 20140501– sent for Trade Executions only	
<Standard Message Trailer>			Y			

5.9.8.3 Execution Report – Single Order Rejection

The order will be rejected by LMAX for the following reasons:

- Order is being placed outside the exchange operating hours
- Order with duplicate *ClOrderID*
- Order with invalid *Price*, eg if price does not match the price increment for the instrument or is outside the allowable bounds for the instrument.
- *Quantity* is not specified or does not match the quantity increment for the instrument.
- Instrument is not available for trading (suspended, expired, closed for trading)
- Instrument cannot be identified due to the invalid values supplied in either *UnderlyingSymbol* or UnderlyingSecurityID, or both
- *Price* specified for the Order significantly differs from the prices of the previous two orders submitted on this side of the book on this FIX session. The price is considered significantly different if it is a certain percentage higher or lower than either of the last two prices placed on this side of the book for the instrument on this FIX session. This percentage is configured on a per Instrument basis.

5.9.8.4 Execution Report - Unsolicited Single Order Cancel

In the event of unsolicited cancelation of the order by LMAX **ExecutionReport** will be send to the client with *OrdStatus= 4 (Cancelled).*
In the **ExecutionReport** for unsolicited order cancel *ClOrdID* contains the most recently known *ClOrdID* for the order that has been cancelled. *OrigClOrderID* will be omitted. *ExecID* will be populated with the unique execution id generated by LMAX.

5.9.9 Business Message Reject (j)

The **BusinessMessageReject** message can reject an application level message which fulfills session-level rules and cannot be rejected via any other means.

In addition to requirements for the FIX message header, only the following fields are used by the application layer for **BusinessMessageReject** message.

Tag	Field	Description	Required	Data Type	LMAX Supported Values / Information	Instrument Type
<Standard Message Header>			Y	MsgType = j		
45	RefSeqNum	*MsgSeqNum* of rejected message	N	Int		
372	RefMsgType	The *MsgType* of the FIX message being referenced.	Y	String		

Tag	Field	Description	Required	Data Type	LMAX Supported Values / Information	Instrument Type
379	BusinessRejectRefID	The value of the business-level ID field on the message being referenced. Required unless the corresponding ID field was not specified.	N	String		
380	BusinessRejectReason	Code to identify reason for rejection	Y	Int	0 = Other 1 = Unknown ID 2 = Unknown Security 3 = Unsupported Message Type 4 = Application is not Available 5 = Conditionally Required Field Missing	
58	Text	Message to explain reason for rejection	N	String		
354	EncodedTextLen	Must be set if *EncodedText* field is specified and must immediately precede it.	N	Int		
355	EncodedText	Encoded (non-ASCII characters) representation of the *Text* field in the encoded format specified via the *MessageEncoding* field.	N	Data		
<Standard Message Trailer>			Y			

5.10 Reference Data Messages

5.10.1 Security Definition Request (c)

SecurityDefinitionRequest message is used for requesting a list of securities that can be traded on the exchange. This request will generate a **SecurityDefinition** message(s) in response.

LMAX does not support **SecurityDefinitionRequests** for any specific instruments or instrument types.

In addition to requirements for the FIX message header, only the following fields are used by the application layer for a **SecurityDefinitionRequest** message.

Tag	Field	Description	Required	Data Type	LMAX Supported Values / Information	Instrument Type
<Standard Message Header>			Y	MsgType = c		
207	SecurityExchange	Should be defaulted to LMAX	Y	String	LMAX	
320	SecurityRequestID	ID for users security request, to be sent back in response	Y	String		
321	SecurityRequestType	Type of request	Y	Int	0 = REQUEST SECURITY IDENTITY AND SPECIFICATIONS	
<Standard Message Trailer>			Y			

5.10.1.1 Example of SecurityDefinitionRequest

Component	Tag Name	Tag No.	Value
Standard Header	BeginString	8	FIX.4.2
Standard Header	BodyLength	9	93
Standard Header	MsgSeqNum	34	7
Standard Header	MsgType	35	c=SECURITY_DEFINITION_REQUEST
Standard Header	SenderCompID	49	lfixqauser
Standard Header	SendingTime	52	20100514-09:44:42.078
Standard Header	TargetCompID	56	FIX-API
Body	SecurityExchange	207	LMAX
Body	SecurityReqID	320	2010514104437
Body	SecurityRequestType	321	0=REQUEST_SECURITY_IDENTITY_AND_SPECIFICATIONS
Standard Trailer	CheckSum	10	180

5.10.2 Security Definition (d)

SecurityDefinition message will be generated in response to **SecurityDefinitionRequest** message for each instrument available for trading on LMAX exchange.

In addition to requirements for the FIX message header, only the following fields are used by the application layer for a Security Definition message.

Tag	Field	Description	Required	Data Type	LMAX Supported Values / Information	Instrument Type
<Standard Message Header>			Y	MsgType = d		
15	Currency	The quote currency for this security (the currency which this instrument is traded in)	N	Currency	ISO Currency code	
22	IDSource	Identifies class of alternative *SecurityID*	N	Char	8=EXCHANGE SYMBOL	
9000	PriceIncrement	Price increment is the minimum movements up or down for this security	N	Float		
9001	PriceIncrementValue	This is the Tick value	N	Float		
107	SecurityDesc	The description of the security.	N	String		
207	SecurityExchange	Exchange you are downloading instruments for	N	String	LMAX	
48	SecurityID	The LMAX ID of the security	N	String		
320	SecurityReqID	Unique ID for request, as sent by client	Y	String		

Tag	Field	Description	Required	Data Type	LMAX Supported Values / Information	Instrument Type
322	SecurityResponseID	LMAX Generated Response ID	Y	String		
167	SecurityType	The type of security.	N	String	CS=Equities FUT=Future FOR=Forex Spot CASH=Cash Spot (Indices) *	
55	Symbol	The underlying symbol	N	String		
393	TotalNumSecurities	Total number of securities.	Y	String	Total number of securities available on LMAX Exchange	
	Text	Instrument Clearing Reference	N		Value provided by a Central Clearer. Conditionally required for centrally cleared instruments.	
200	MaturityMonthYear	Month and Year of the maturity for *SecurityType =FUT*. Required if *MaturityDay* is specified. Format: YYYYMM (i.e. 199903)	Y©			FUT
205	MaturityDay	Day of month used in conjunction with *MaturityMonthYear* to specify the maturity date for *SecurityType =FUT* Required only when maturity date is available	Y©			FUT
<Standard Message Trailer>			Y			

5.10.2.1 Examples of SecurityDefinition for supported security types

Foreign Exchange

Component	Tag Name	Tag No	Value
Standard Header	BeginString	8	FIX.4.2
Standard Header	BodyLength	9	189
Standard Header	MsgSeqNum	34	5
Standard Header	MsgType	35	d=SECURITY_DEFINITION
Standard Header	SenderCompID	49	FIX-API
Standard Header	SendingTime	52	20100514-09:44:42.231
Standard Header	TargetCompID	56	lfixqauser
Body	Currency	15	JPY
Body	IDSource	22	8=EXCHANGE_SYMBOL
Body	PriceIncrement	9000	0.001
Body	PriceIncrementValue	9001	10
Body	SecurityDesc	107	AUD/JPY
Body	SecurityExchange	207	LMAX
Body	SecurityID	48	4008
Body	SecurityReqID	320	2010514104437
Body	SecurityResponseID	322	295
Body	SecurityType	167	FOR=FOREIGN_EXCHANGE_CONTRACT
Body	Symbol	55	AUD/JPY
Body	Text	58	AUD/JPY
Body	TotalNumSecurities	393	40
Standard Trailer	CheckSum	10	190

Equity

Component	Tag Name	Tag No	Value
Standard Header	BeginString	8	FIX.4.2
Standard Header	BodyLength	9	196
Standard Header	MsgSeqNum	34	7
Standard Header	MsgType	35	d=SECURITY_DEFINITION
Standard Header	SenderCompID	49	FIX-API
Standard Header	SendingTime	52	20100514-09:44:42.232
Standard Header	TargetCompID	56	lfixqauser
Body	Currency	15	GBP
Body	IDSource	22	8=EXCHANGE_SYMBOL
Body	PriceIncrement	9000	0.1
Body	PriceIncrementValue	9001	0.001
Body	SecurityDesc	107	BARCLAYS PLC

Body	SecurityExchange	207	LMAX
Body	SecurityID	48	2001
Body	SecurityReqID	320	2010514104437
Body	SecurityResponseID	322	297
Body	SecurityType	167	CS=COMMON_STOCK
Body	Symbol	55	BARC
Body	Text	58	GB0031348658
Body	TotalNumSecurities	393	40
Standard Trailer	CheckSum	10	126

Futures Contract

Component	Tag Name	Tag No	Value
Standard Header	BeginString	8	FIX.4.2
Standard Header	BodyLength	9	229
Standard Header	MsgSeqNum	34	21
Standard Header	MsgType	35	d=SECURITY_DEFINITION
Standard Header	SenderCompID	49	FIX-API
Standard Header	SendingTime	52	20100514-09:44:42.239
Standard Header	TargetCompID	56	lfixqauser
Body	Currency	15	USD
Body	IDSource	22	8=EXCHANGE_SYMBOL
Body	MaturityDay	205	30
Body	MaturityMonthYear	200	201009
Body	PriceIncrement	9000	0.01
Body	PriceIncrementValue	9001	0.1
Body	SecurityDesc	107	Copper (Aug10)
Body	SecurityExchange	207	LMAX
Body	SecurityID	48	3032
Body	SecurityReqID	320	2010514104437
Body	SecurityResponseID	322	311
Body	SecurityType	167	FUT=FUTURE
Body	Symbol	55	HGQ0
Body	Text	58	LMAHGQ001000
Body	TotalNumSecurities	393	40
Standard Trailer	CheckSum	10	093

Indices

Component	Tag Name	Tag No	Value
Standard Header	BeginString	8	FIX.4.2
Standard Header	BodyLength	9	201
Standard Header	MsgSeqNum	34	100

Standard Header	MsgType	35	d=SECURITY_DEFINITION
Standard Header	SenderCompID	49	FIX-API
Standard Header	SendingTime	52	20100514-09:44:42.281
Standard Header	TargetCompID	56	lfixqauser
Body	Currency	15	USD
Body	IDSource	22	8=EXCHANGE_SYMBOL
Body	PriceIncrement	9000	1
Body	PriceIncrementValue	9001	1
Body	SecurityDesc	107	Wall Street 30 24hr
Body	SecurityExchange	207	LMAX
Body	SecurityID	48	1008
Body	SecurityReqID	320	2010514104437
Body	SecurityResponseID	322	390
Body	SecurityType	167	CASH=CASH
Body	Symbol	55	DJI
Body	Text	58	LMADJI001000
Body	TotalNumSecurities	393	40
Standard Trailer	CheckSum	10	057

5.10.3 Security Status Request (e)

SecurityStatusRequest allows the Client to subscribe to status updates for a given security.

In addition to requirements for the FIX message header, only the following fields are used by the application layer for a Security Status Request message.

Tag	Field	Description	Required	Data Type	LMAX Supported Values / Information	Instrument Type
<Standard Message Header>			Y	MsgType = e		
22	IDSource	Identifies class of alternative *SecurityID*	N	Char	8=EXCHANGE SYMBOL	
48	SecurityID	The *SecurityID* of the instrument	N	String		
324	SecurityStatusReqID	Unique identifier for status request	Y	String		
263	SubscriptionRequestType	Indicates to the other party what type of response is expected.	Y	Char	1= SNAPSHOT PLUS UPDATES 2=DISABLE PREVIOUS	
55	Symbol	Symbol	Y	String		
<Standard Message Trailer>			Y			

5.10.3.1 Examples of Security Status request

Snapshot + Updates request for FX security

Component	Tag Name	Tag No	Value
Standard Header	BeginString	8	FIX.4.2
Standard Header	BodyLength	9	108
Standard Header	MsgSeqNum	34	96
Standard Header	MsgType	35	e=SECURITY_STATUS_REQUEST
Standard Header	SenderCompID	49	lfixqauser
Standard Header	SendingTime	52	20100514-10:14:12.359
Standard Header	TargetCompID	56	FIX-API
Body	IDSource	22	8=EXCHANGE_SYMBOL
Body	SecurityID	48	4002
Body	SecurityStatusReqID	324	201051411147
Body	SubscriptionRequestType	263	1=SNAPSHOT_PLUS_UPDATES
Body	Symbol	55	GBP/USD
Standard Trailer	CheckSum	10	193

Snapshot for Indices Security

Component	Tag Name	Tag No	Value
Standard Header	BeginString	8	FIX.4.2
Standard Header	BodyLength	9	108
Standard Header	MsgSeqNum	34	120
Standard Header	MsgType	35	e=SECURITY_STATUS_REQUEST
Standard Header	SenderCompID	49	lfixqauser
Standard Header	SendingTime	52	20100514-10:21:56.500
Standard Header	TargetCompID	56	FIX-API
Body	IDSource	22	8=EXCHANGE_SYMBOL
Body	SecurityID	48	1001
Body	SecurityStatusReqID	324	2010514112143
Body	SubscriptionRequestType	263	0=SNAPSHOT
Body	Symbol	55	FTSE
Standard Trailer	CheckSum	10	116

Disable Security Status updates

Component	Tag Name	Tag No	Value
Standard Header	BeginString	8	FIX.4.2
Standard Header	BodyLength	9	108
Standard Header	MsgSeqNum	34	207
Standard Header	MsgType	35	e=SECURITY_STATUS_REQUEST

Standard Header	SenderCompID	49	lfixqauser
Standard Header	SendingTime	52	20100514-10:50:49.156
Standard Header	TargetCompID	56	FIX-API
Body	IDSource	22	8=EXCHANGE_SYMBOL
Body	SecurityID	48	1001
Body	SecurityStatusReqID	324	2010514115037
Body	SubscriptionRequestType	263	2=DISABLE_PREVIOUS
Body	Symbol	55	FTSE
Standard Trailer	CheckSum	10	140

5.10.4 Security Status (f)

SecurityStatus message updates the Client on the current trading status of a given security.

In addition to requirements for the FIX message header, only the following fields are used by the application layer for **SecurityStatus** message.

Tag	Field	Description	Required	Data Type	LMAX Supported Values / Information	Instrument Type
<Standard Message Header>			Y	MsgType = f		
22	IDSource	Identifies class of alternative SecurityID	N	Char	8=EXCHANGE SYMBOL	
207	SecurityExchange	Exchange you are downloading instruments for.	N	String	LMAX	
48	SecurityID	The LMAX ID of the security	N	String		
324	SecurityStatusReqID	Unique identifier for status request	N	String		
326	SecurityTradingStatus	Identifies the trading status applicable to the transaction.	N	Int	2=TRADING HALT 17=READY TO TRADE 18= NOT AVAILABLE FOR TRADING 20=UNKNOWN OR INVALID	
55	Symbol	Symbol	Y	String		
<Standard Message Trailer>			Y			

5.10.4.1 Examples of Security Status messages

Update - Security is ready to trade

Component	Tag Name	Tag No	Value
Standard Header	BeginString	8	FIX.4.2
Standard Header	BodyLength	9	119
Standard Header	MsgSeqNum	34	162
Standard Header	MsgType	35	f=SECURITY_STATUS
Standard Header	SenderCompID	49	FIX-API
Standard Header	SendingTime	52	20100514-10:14:12.601
Standard Header	TargetCompID	56	lfixqauser
Body	IDSource	22	8=EXCHANGE_SYMBOL
Body	SecurityExchange	207	LMAX
Body	SecurityID	48	4002
Body	SecurityStatusReqID	324	201051411147
Body	SecurityTradingStatus	326	**17=READY_TO_TRADE**
Body	Symbol	55	GBP/USD
Standard Trailer	CheckSum	10	036

Response to unknown or invalid subscription

Component	Tag Name	Tag No	Value
Standard Header	BeginString	8	FIX.4.2
Standard Header	BodyLength	9	118
Standard Header	MsgSeqNum	34	237
Standard Header	MsgType	35	f=SECURITY_STATUS
Standard Header	SenderCompID	49	FIX-API
Standard Header	SendingTime	52	20100514-10:50:49.555
Standard Header	TargetCompID	56	lfixqauser
Body	IDSource	22	8=EXCHANGE_SYMBOL
Body	SecurityExchange	207	LMAX
Body	SecurityID	48	1001
Body	SecurityStatusReqID	324	2010514115037
Body	SecurityTradingStatus	326	**20=UNKNOWN_OR_INVALID**
Body	Symbol	55	FTSE
Standard Trailer	CheckSum	10	205

Updates - Instrument has been suspended

Component	Tag Name	Tag No	Value
Standard Header	BeginString	8	FIX.4.2
Standard Header	BodyLength	9	118
Standard Header	MsgSeqNum	34	366
Standard Header	MsgType	35	f=SECURITY_STATUS
Standard Header	SenderCompID	49	FIX-API
Standard Header	SendingTime	52	20100514-11:04:00.830
Standard Header	TargetCompID	56	lfixqauser
Body	IDSource	22	8=EXCHANGE_SYMBOL
Body	SecurityExchange	207	LMAX
Body	SecurityID	48	4002
Body	SecurityStatusReqID	324	201051412343
Body	SecurityTradingStatus	326	**2=TRADING_HALT**
Body	Symbol	55	GBP/USD
Standard Trailer	CheckSum	10	243

6 Appendix A – Message Flows

6.1 New Order Single

The sample workflow for an order is shown in an example below; order is sent to LMAX via the LMAX Gateway and an execution reports are returned informing the client of the status of the order. In the example below 2 types of order matches are used: aggressive – when the order matches retail orders on entry and passive – order is waiting to be matched on the Order Book.

An example

4. LMAX Client places a new Order
5. Order gets aggressively part filled against two counter orders.
6. The remainder of the Order is waiting on the Order Book. It is later being passively matched by another single counter order which causes the Order to be fully filled.

Time	Message Received	Message Sent	ExecType(150)	OrdStatus(39)	Comment
1	New Order				
2		ExecutionReport	New(0)	New(0)	Order received

Time	Message Received	Message Sent	ExecType(150)	OrdStatus(39)	Comment
2		ExecutionReport	Partial Fill(1)	Partially Filled(1)	Aggressive match against 1st counter order
2		ExecutionReport	Partial Fill(1)	Partially Filled(1)	Aggressive match against 2nd counter order
3		ExecutionReport	Fill(2)	Filled(2)	Passively matched by another aggressive counter order which fully fills the order.

6.2 Mass Quote

The sample workflows for a Mass Quote is shown in the examples below; Mass Quotes are sent to LMAX via the LMAX Gateway and Quote Acknowledgements/Execution Reports are returned informing the client of the status of the mass Quote.

In the examples below 2 types of Mass Quote matches are described:

- Aggressive – when the Mass Quote matches retail orders immediately;
- Passive – Mass Quote is applied to a market and waiting to be matched on the Order Book.

Example 1 – Mass Quote is received and Quote Acknowledgement is sent with status Accepted, followed by the Execution Reports

1. LMAX Client places a new Mass Quote
2. LMAX receives the Mass Quote and sends Quote Acknowledgment
3. Two levels of the Mass Quote get aggressively filled against two counter orders – top level fully filled and the second – part filled.
4. The remainder of the Mass Quote is waiting on the Order Book. The remainder of the part filled level later being passively matched by another single counter order which causes it to be to be fully filled.
5. New Mass Quote is submitted resulting in the existing Mass Quote replacement.

Time	Message Received	Message Sent	Quote Acknowledgement Status (297)	ExecType(150)	OrdStatus(39)	Comment
1	Mass Quote (i)					
2		Quote Acknowledgement(b)	Accepted(0)			Mass Quote for a given instrument is applied to a market

Time	Message Received	Message Sent	Quote Acknowledgement Status (297)	ExecType(150)	OrdStatus(39)	Comment
3		ExecutionReport		Fill(2)	Filled(2)	Mass Quote results in Trade - aggressive full match for 1st price level submitted in the Mass Quote
3		ExecutionReport		Partial Fill(1)	Partially Filled(1)	Mass Quote results in Trade - aggressive partial match against 2nd price level submitted in the Mass Quote
4		ExecutionReport		Partial Fill(1)	Filled(2)	Mass Quote results in Trade - passive match that fully fills the 2nd price level submitted in the Mass Quote
5	Mass Quote (i)					
6		Quote Acknowledgement(b)	Accepted(0)			Mass Quote replaced the previous MassQuote on a market for a given instrument

Example 2 – Mass Quote is received and Quote Acknowledgement is sent with status Rejected

1. LMAX Client places a new Mass Quote after the Instrument has been closed for trading at the end of the day.
2. LMAX receives the Mass Quote and sends Quote Acknowledgment - Rejected

Time	Message Received	Message Sent	Quote Acknowledgement Status (297)	ExecType(150)	OrdStatus(39)	Comment
1	Mass Quote (i)					
2		Quote Acknowledgement(b)	Rejected(5)			Mass quote is rejected due to Instrument being closed for trading

Example 3 –Quote Cancel is received and Quote Acknowledgement is sent with status Cancelled

1. LMAX Client places a new Mass Quote
2. LMAX receives the Mass Quote and sends Quote Acknowledgment – Accepted
3. LMAX Client sends Quote Cancel requesting to cancel the Mass Quote
4. LMAX processes the Quote Cancel request and sends Quote Acknowledgement - Cancelled

Time	Message Received	Message Sent	Quote Acknowledgement Status (297)	ExecType(150)	OrdStatus(39)	Comment
1	Mass Quote (i)					
2		Quote Acknowledgement(b)	Rejected(5)			Mass quote is rejected due to Instrument being closed for trading
3	Quote Cancel (Z)					Request to cancel Mass Quote is placed
4		Quote Acknowledgement(b)	Cancelled(1)			Mass Quote has been cancelled

Example 4 –Quote Cancel is received and Quote Acknowledgement is sent with status Rejected

1. LMAX Client places a new Mass Quote
2. LMAX receives the Mass Quote and sends Quote Acknowledgment – Accepted
3. LMAX Client sends Quote Cancel requesting to cancel the Mass Quote
4. LMAX processes the Quote Cancel request and sends Quote Acknowledgement - Rejected

Time	Message Received	Message Sent	Quote Acknowledgement Status (297)	ExecType(150)	OrdStatus(39)	Comment
1	Mass Quote (i)					
2		Quote Acknowledgement(b)	Rejected(5)			Mass quote is rejected due to Instrument being closed for trading
3	Quote Cancel (Z)					Request to cancel Mass Quote is placed
4		Quote Acknowledgement(b)	Rejected(5)			Quote Cancel Request has been rejected due to the unknown QuoteID specified in Quote Cancel message

7 Revision History

The following changes have been made to this document since its release.

Version	Release date	Comments
1.0	07/09/2015	First release