LMEsource

Client Interface Specification v4.04

THE LONDON METAL EXCHANGE

LME.COM

10 Finsbury Square, London EC2A 1AJ | Tel +44 (0)20 7113 8888
Registered in England no 2128666. Registered office as above.

Document History

Version	Date	Changes
4.00	11 Jun 2020	Initial release of document for electronic market data from the new LMEselect v10
4.01 – 4.03		LME internal updates
4.04	28 May 2021	Changes to message definitions for the following messages: • Outright Definition (301) • Strategy Definition (302) Day 2 functionality identified in grey and italic font. Changes to the following message definitions • Outright Definition (301) • Strategy Definition (302) New messages defined: • Order Cancel (335) • Order Executed (350) • Match Trade (360) Updated details for Intraday Trade Statistics (352) message. Update definition for TimeOfEvent fields. Extra detail on IOP (354) message. Additional Order update examples added in sections 7 and 8.

Contents

1 Introduction

1.1 Purpose

The LME has embarked on a program to fully rebuild its trading platform, LMEselect, and to upgrade the technology of its market data platform, LMEsource. LMEsource will be the LME's sole market data platform, disseminating incremental updates for bid, ask, trade, statistics and provide recovery and supporting services for client processing of this market data.

This document specifies the updated binary interface for the new LMEsource v4, which will disseminate electronic market data from the new LMEselect v10.

A later update to this document will include details of the non-electronic market data messages.

1.2 Delivery Phasing

This document covers all the functionality available in LMEselect 10 and LMEsource V4 however functionality will be delivered in phased releases.

Note: This specification includes references to functionality and data that will be made available in future releases to both LMEselect v10 and LMEsource V4. Functionality and/or data that will be included in a later release is specified in the following table and shown throughout the document in *dark grey italics*. The initial release of LMEselect v10 and LMEsource V4 will contain all functionality that is not specified in the table below.

Function	Reference
Option Instruments	Outright Definition (301) Strategy Definition (302)
Cancelled Trades	Order Executed (350) Match Trade (360)
Quote Requests	Quote Request (356)
Strategy Types	Strategy Type
Base Metal Options	*Base Metal Options*
TAPOs	*TAPOs*

1.3 Reading Guide

The chapters following this introduction are:

Chapter 2: System Overview

Chapter 3: Common Message Formats

Chapter 4: Electronic Message Formats

Chapter 5: Non-Electronic Message Formats

Chapter 6: Recovery

Chapter 7: Aggregated Order Book Management

Chapter 8: Full Order Book Management

Appendix A: Reference Data Values

Appendix B: Maturity Dates

Appendix C: Market Data Product Summary Table

Appendix D: Channel Matrix

Note: details of the non-electronic message formats are not included in this document. These details will be included in a future version.

1.4 Products

A range of products are provided catering for the varying needs of LMEsource clients. Each product may contain multiple distinct multicast channels, and will contain data for both futures and options. The set of products will encompass Level 1 Top-of-Book, Level 2 15 Levels Price Book and Level 3 Full book offerings.

1.4.1 Product Summary Table

The table below shows the market data products and the specific chapters in this document that are relevant to each individual product.

Section	Message Formats	LME Level 1 Top of Book	LME Level 2 15 Levels Price Book	LME Level 3 Full Book

3.1	Data Types	◆	◆	◆
3.2	Packet Structure	◆	◆	◆
3.3	Packet Header	◆	◆	◆
3.4	Control Messages	◆	◆	◆
3.5	Retransmission	◆	◆	◆
3.6	Refresh	◆	◆	◆
4.1.1	Outright Definition (301)	◆	◆	◆
4.1.2	Strategy Definition (302)	◆	◆	◆
4.2.1	Contract State (311)	◆	◆	◆
4.2.2	Instrument State (312)	◆	◆	◆
4.3.1	Level 1 Top of Book (355)	◆		
4.3.2	Level 2 Aggregate Order Book Update (353)		◆	
4.3.3	Level 3 Order Add (357)			◆
4.3.4	Level 3 Order Amend (358)			◆

4.3.5	Level 3 Order Cancel (359)			◆
4.3.6	Order Executed (350)			◆
4.3.7	Orderbook Clear (335)	◆	◆	◆
4.3.8	Quote Request (356)	◆	◆	◆
4.3.9	IOP (354)	◆	◆	◆
4.4.1	Match Trade (360)	◆	◆	
4.4.2	EOD Trade Statistics (351)	◆	◆	◆
4.4.3	Intraday Trade Statistics (352)		◆	◆
6	Recovery	◆	◆	◆
7	Level 2 Aggregate Order Book Management		◆	
8	Level 3 Full Order Book Management			◆

2 System Overview

2.1 Scope

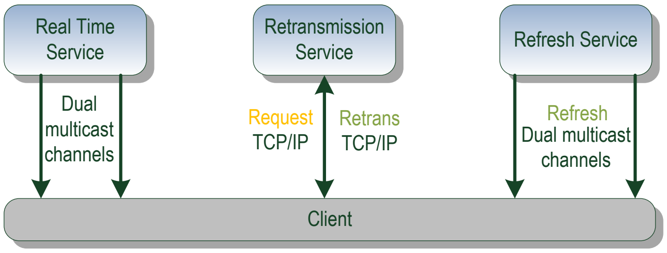


Figure 1: Access to Market Data

LMEsource provides market data represented in an efficient binary message format for all instruments traded on the LME Market. It has been designed for high throughput and low latency.

2.1.1 Multicast

Messages are published in a one-to-many fashion using the IP multicast and UDP transport protocols. Multicast is not a connection-oriented protocol. Data is sent strictly in one direction from server to clients.

2.1.2 Dual Multicast Channels

Due to the inherently unreliable nature of the UDP transport, packets may be lost or delivered out of sequence. To mitigate the risk of packet loss, the messages are duplicated and sent over two separate multicast channels (dual channels). Technically, a multicast channel corresponds to a multicast group.

Each pair of dual multicast channels has a unique identifier, which is referred to as the ChannelID.

More details regarding the configuration parameters (including the IP addresses and port numbers corresponding to the multicast channels) will be found in the LME Systems Connectivity Guide which will be provided at a later stage.

2.1.3 Recovery Mechanisms

LMEsource provides two recovery mechanisms:

- A retransmission server which provides an on-request gap-fill retransmission of lost messages. The retransmission requests and gap-fill replies are a point-to-point (TCP/IP connection).
- A refresh server provides snapshots of the market state at regular intervals throughout the business day. Snapshots are sent using multicast on separate channels for the real time messages. The time period between snapshots is 45 seconds, but this may vary across multicast channels.

2.2 Session Management

Each multicast channel maintains its own session. A session is limited to one business day. During this day the message sequence number is strictly increasing and therefore unique within the channel.

2.2.1 Start of Day

LMEsource will normally be brought up at around 00:15 – 00:30am. This start up time is not rigid and the LME has the right to adjust this time according to different trading situations.

On each channel the first message at the start of the business day is the Sequence Reset message. The Sequence Reset message carries sequence number 1. On receipt of this message, the client must clear all cached data for all instruments.

At start-up, LMEsource will disseminate reference data comprised of Outright Definition and Strategy Definition messages for all outright and strategy tradable instruments.

If a client starts listening after the start of the business day and misses the Sequence Reset message and reference data, it must use the refresh service to recover and synchronize with the real time channels.

2.2.2 Normal Transmission

Normal message transmission is expected between when the market opens for trading and when the market is closed. Heartbeats are sent at regular intervals (currently set at every 2 seconds) on each channel when there is no activity. The LME may adjust this interval.

2.2.3 End of Day

LMEsource will typically shut down at around 21:00 London time after the clearing procedure has completed. A later shutdown may occur due to special circumstances. The shutdown time is not rigid and the LME has the right to adjust this time according to different trading situations.

2.2.4 Error Recovery

2.2.4.1 System Component Failure

If a system component fails that results in a small amount of packet loss and requires a failover or restart, there will be a short interruption in multicast dissemination from either Line A or Line B. The system is deployed in an active-active configuration with Line A and Line B being generated independently and so line arbitration will allow the client to continue receiving messages – see section 6 for more information about recovery.

2.2.4.2 Disaster Recovery

In the unlikely event of a disaster recovery situation at the primary site, LMEsource will be brought up at the disaster recovery (DR) site.

During the interruption, no data will be sent, including heartbeats.

A Sequence Reset message will be sent on each channel when LMEsource is brought up. This will be followed by a snapshot of each channel. After the snapshot the market data feed will return to normal operations.

A Disaster Recovery (DR) Signal message indicating the DR status will also be sent on its dedicated channel when LMEsource is brought up – see section 3.4.3 for more information about the DR Signal message.

IP addresses and ports that have been provided for the DR site’s retransmission service should be used. See the LME Systems Connectivity Guide for more details.

2.3 Trading Sessions

Normally, trading is conducted in continuous trading session(s) every trading day. However, there are situations where there is only half day trading with fewer trading session(s), or a trading session is suspended due to a special situation. LMEsource is not affected by the number of trading sessions and will continue to provide real time data as long as the LME’s trading system is available.

2.4 Race Conditions

Due to the nature of the dissemination protocol, the real time order/trade data and reference data are disseminated via separate channels so users need to be aware that there is the potential for a race condition.

As an example suppose an Instrument State message is sent showing a change to state ‘Post Trade’, however for a very short time after this message the regular order and trade information for this instrument may continue to arrive.

3 Common Message Formats

3.1 Data Types

The information supplied in this section and its sub-sections applies to the data feed(s) marked with [◆]

Section	LME Level 1 Top of Book	LME Level 2 15 Levels Price Book	LME Level 3 Full Book
3.1	◆	◆	◆

The following table lists all the data types used by LMEsource.

Format	Description
String	ASCII characters which are left aligned and padded with spaces, unless otherwise specified.
Uint8	8 bit unsigned integer.
Uint16	Little-Endian encoded 16 bit unsigned integer.
Uint32	Little-Endian encoded 32 bit unsigned integer.
Uint64	Little-Endian encoded 64 bit unsigned integer.
Int8	Little-Endian encoded 8 bit signed integer.
Int16	Little-Endian encoded 16 bit signed integer.
Int32	Little-Endian encoded 32 bit signed integer.
Int64	Little-Endian encoded 64 bit signed integer.

3.1.1 Null Values

From time to time certain fields cannot be populated and specific values are used to represent null. The table below shows the values used to represent null for different data types.

Format	Null representation (Hex 2's complement)
Uint8	0x00
Int8	0x80
Int32	0x80000000
Int64	0x8000000000000000

3.1.2 Implied Decimal

In order to avoid decimal calculation in LMEsource, all price and percentage fields are provided as integers with implied decimal places. Implied decimal places for each field are stated as below table. Clients require to perform the actual scaling for data value.

Constant Name	Type	Implied Decimal places	Example
PERCENT	Int32	4	987654 = 98.7654
PRICE	Int64	6	123456789 = 123.456789

3.2 Packet Structure

The information supplied in this section and its sub-sections applies to the data feed(s) marked with [◆]

Section	LME Level 1 Top of Book	LME Level 2 15 Levels Price Book	LME Level 3 Full Book
3.2	◆	◆	◆

Multicast packets are structured into a common packet header followed by zero or more messages. Messages within a packet are laid out sequentially, one after another without any spaces between them.

Packet Header	Message 1	Message 2	…	Message n

The maximum length of a packet is 1500 bytes which includes the multicast headers, packet header and messages.

A packet will only ever contain complete messages. A single message will never be fragmented across packets.

3.2.1 Packet Header

All packets will begin with a common packet header. The packet header provides information including the total packet length, the number of messages within the packet, the sequence number of the first message and a send timestamp.

Offset	Field	Format	Len	Description
0	PktSize	Uint16	2	Size of the packet (including this field)
2	MsgCount	Uint8	1	Number of messages included in the packet
3	Filler	String	1	

4	SeqNum	Uint32	4	Sequence number of the first message in the packet
8	SendTime	Uint64	8	The UTC timestamp for the time this message was published by LMEsource., as the number of nanoseconds since January 1, 1970, 00:00:00 GMT. Precision is provided to the nearest microsecond.
Header length			16	

3.3 Message Structure

The information supplied in this section and its sub-sections applies to the data feed(s) marked with [◆]

Section	LME Level 1 Top of Book	LME Level 2 15 Levels Price Book	LME Level 3 Full Book
3.3	◆	◆	◆

The format of each message within a packet will vary according to the message type. However, regardless of the message type, each message will start with a two-byte message size (MsgSize) followed by a two-byte message type (MsgType). These are described in the following table.

Field	Format	Len	Description
MsgSize	Uint16	2	Message length (including this field)
MsgType	Uint16	2	Type of message. The valid values for MsgType are below:

			100 Sequence Reset **101** Logon **102** Logon Response **201** Retransmission Request **202** Retransmission Response **203** Refresh Complete **301** Outright Definition **302** Strategy Definition **311** Contract State **312** Instrument State **335** Orderbook Clear **350** Order Executed **351** EOD Trade Statistics **352** Intraday Trade Statistics **353** Aggregate Order Book Update **354** IOP **355** Top Of Book **356** Quote Request **357** Order Add **358** Order Amend **359** Order Cancel **360** Match Trade

3.4 Control Messages

The information supplied in this section and its sub-sections applies to the data feed(s) marked with [◆]

Section	LME Level 1 Top of Book	LME Level 2 15 Levels Price Book	LME Level 3 Full Book
3.4	◆	◆	◆

3.4.1 Heartbeat

Heartbeats consist of a packet header with MsgCount set to 0. They do not carry a sequence number and therefore do not increment the sequence number of the multicast channel. SeqNum is set to the sequence number of the previous message sent on the channel.

The Heartbeat message will be identical for all the services.

3.4.2 Sequence Reset (100)

The Sequence Reset message is sent on each multicast channel at start of day. It may also be sent intraday in case of a disaster recovery.

The client must ignore the sequence number of the Sequence Reset message itself, and set the next expected sequence number to NewSeqNo. The client may receive multiple sequence reset messages from all channels. Whenever the Sequence Reset message is received, clients must clear all cached data for all instruments traded in the Market and then subscribe to the refresh channels to receive the current state of the market.

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message.	**100** = Sequence Reset
4	NewSeqNo	Uint32	4	New sequence number.	Always set to 1
Total length			8		

3.4.3 Disaster Recovery Signal (105)

The Disaster Recovery (DR) Signal message is sent on a dedicated multicast channel (DR channel) whenever a site failover is triggered. In normal situations, the dedicated DR channel only carries heartbeats until the end of the business day.

When site failover begins, a DR Signal is sent with “DRStatus=1” indicating that the DR process has been activated. Clients should then clear all cached market data and prepare their own system for the site failover. When the site failover process finishes, a DR Signal will be sent with “DRStatus=2”, thereupon clients can start to rebuild the latest market image from the refresh service. The same DR Signal will be sent periodically until the end of the business day.

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message.	**105 =** DR Message
4	DRStatus	Uint32	4	Status during site failover	**1** = DR in progress **2** = DR completed
Total length			8		

3.5 Retransmission

The information supplied in this section and its sub-sections applies to the data feed(s) marked with [◆]

Section	LME Level 1 Top of Book	LME Level 2 15 Levels Price Book	LME Level 3 Full Book
3.5	◆	◆	◆

Refer to section 6.3 Retransmission service for details on the retransmission messages.

Note that when the Logon (101) or Retransmission Request (201) messages are sent to the LMEsource server, the client must also include a packet header as shown below.

Also note that the same header is used by the RTS server when sending either a Logon Response (102) or a Retransmission Response (202) messages to clients. In this case the SeqNum and SendTime fields are not relevant and can be discarded.

There is no Logoff required for the Retransmission service. The client can simply disconnect from the session.

Offset	Field	Format	Len	Values	Notes
0	PktSize	Uint16	2	32	16 bytes for this header plus 16 bytes for either the Logon (101) or Retransmission Request (201) message. When sent by the RTS, this will contain 16 bytes for this header, plus either 8 bytes for the Logon Response (102) or 16 bytes for the Retransmission Response (202)
2	MsgCount	Uint8	1	1	One message only
3	Filler	String	1		Empty Filler
4	SeqNum	Uint32	4	0	This field is not used
8	SendTime	Uint64	8	0	This field is not used
Total length			16		

After this header, the fields for either Logon (101) or Retransmission Request (201) should follow.

3.5.1 Logon (101)

The Logon message enables client authentication. This is not required for multicast channels and is only used for retransmission requests.

Normal operation: The client sends a Logon message containing username to LMEsource, which responds with a Logon Response message with the SessionStatus set to 0 (Session Active).

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message	**101 =** Logon
4	Username	String	12	Username to log on, padded with binary null characters	
Total length			16		

3.5.2 Logon Response (102)

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message	**102 =** Logon Response
4	SessionStatus	Uint8	1	Status of the session	**0 =** Session Active **5** = Invalid username or IP address **100 =** User already connected
5	Filler	String	3		
Total length			8		

3.5.3 Retransmission Request (201)

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message	**201 =** Retransmission Request
4	ChannelID	Uint16	2	Multicast Channel ID to which the retransmission relates	
6	Filler	String	2		
8	BeginSeqNum	Uint32	4	Beginning of sequence	
12	EndSeqNum	Uint32	4	Message sequence number of last message in range to be resent	
Total length			16		

3.5.4 Retransmission Response (202)

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message.	**202 =** Retransmission Response

4	ChannelID	Uint16	2	Multicast Channel ID to which the retransmission relates	
6	RetransStatus	Uint8	1	Status of the Retransmission response	See Appendix A, section 9.3
7	Filler	String	1		
8	BeginSeqNum	Uint32	4	First sequence number of the Retransmission. Only populated when RetransStatus is 0	
12	EndSeqNum	Uint32	4	Last sequence number of the Retransmission. Only populated when RetransStatus is 0	
Total length			16		

3.6 Refresh

The information supplied in this section and its sub-sections applies to the data feed(s) marked with [◆]

Section	LME Level 1 Top of Book	LME Level 2 15 Levels Price Book	LME Level 3 Full Book
3.6	◆	◆	◆

Refer to section 6.4 Refresh service for details on the Refresh Complete message.

3.6.1 Refresh Complete (203)

This message is published to mark the end of a refresh cycle, see section 6.4 for a full description of refresh.

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message	**203 =** Refresh Complete
4	LastSeqNum	Uint32	4	Sequence number with which the refresh is synchronised	Numerical
Total length			8		

4 Electronic Message Formats

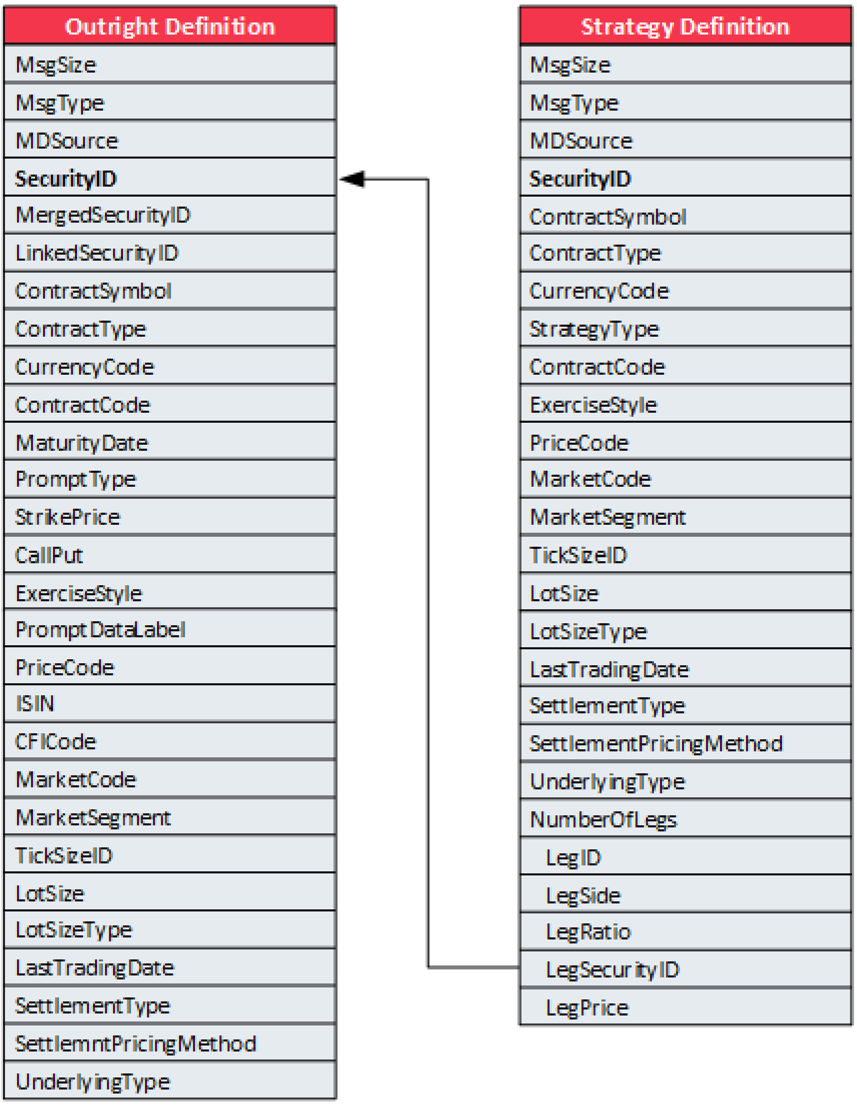
4.1 Reference Data

The information supplied in this section and its sub-sections applies to the data feed(s) marked with [◆]

Section	LME Level 1 Top of Book	LME Level 2 15 Levels Price Book	LME Level 3 Full Book
4.1	◆	◆	◆

Static reference data is organised into two messages that provide a full list of all securities/tradable instruments available on the LMEselect electronic market. The two messages are shown in the entity relationship diagram below. The bold field(s) form the primary key for each message type.

This section is only applicable to reference data for instruments traded on the LMEselect electronic market.



Outright Definition
MsgSize
MsgType
MDSource
SecurityID
MergedSecurityID
LinkedSecurityID
ContractSymbol
ContractType
CurrencyCode
ContractCode
MaturityDate
PromptType
StrikePrice
CallPut
ExerciseStyle
PromptDataLabel
PriceCode
ISIN
CFICode
MarketCode
MarketSegment
TickSizeID
LotSize
LotSizeType
LastTradingDate
SettlementType
SettlemntPricingMethod
UnderlyingType

Strategy Definition
MsgSize
MsgType
MDSource
SecurityID
ContractSymbol
ContractType
CurrencyCode
StrategyType
ContractCode
ExerciseStyle
PriceCode
MarketCode
MarketSegment
TickSizeID
LotSize
LotSizeType
LastTradingDate
SettlementType
SettlementPricingMethod
UnderlyingType
NumberOfLegs
LegID
LegSide
LegRatio
LegSecurityID
LegPrice

The SecurityID field is used to link order and trade messages to the instrument definition. Instrument State (312) messages that are only applicable to one instrument also use SecurityID to link back to the instrument definition.

4.1.1 Outright Definition (301)

Describes an individual outright tradable instrument available from LMEsource.

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	150
2	MsgType	Uint16	2	Type of message	**301 =** Outright Definition
4	MDSource	String	2	Data Source of the message	**EL =** Electronic Market
6	SecurityID	Uint64	8	The id of the security for which trades in this tradable Instrument are held	
14	MergedSecurityID	Uint64	8	The SecurityID of merged tradable instrument 0 means no merged tradable instrument	
22	LinkedSecurityID	Uint64	8	The SecurityID of linked tradable instrument. This will be populated on TaS/TaR instruments with the SecurityID of the parent instrument	

Offset	Field	Format	Len	Description	Values
				0 means no linked tradable instrument	
30	ProductCode	String	2	The product code. This identifies the metal for the instrument	e.g. ‘**AA**’, ‘**AH**’,’**CA**’
32	ContractType	Char	1	Constant contract type	**F** = Future ***O*** *= Options*
33	CurrencyCode	String	3	The currency code according to ISO 4217	e.g. ‘**USD**’
36	ContractCode	String	12	A unique code to identify the contract	e.g. ‘**PBDF**, **OCDF**, **NADT**’
48	MaturityDate	Uint32	4	The maturity date for this tradable instrument	
52	PromptType	Char	1	Identifies whether the prompt is a single or a rolling prompt	**S** = Single **R** = Rolling **O** = Options Expiry
53	StrikePrice	Int64	8	The strike price for this tradable instrument	This data field with implied decimal
61	CallPut	Char	1	For option tradable instruments,	***C*** *= Call* ***P*** *= Put*

Offset	Field	Format	Len	Description	Values
				whether this tradable instrument represents a call or a put option	Default - Space (not applicable)
62	ExerciseStyle	Int8	1	Exercise Style	***0*** *= European* ***1*** *= American* ***2*** *= Asian* NULL for Futures
63	PromptDateLabel	String	7	Prompt Date Label	See Appendix 9.1
70	PriceCode	String	2	A code to represent the 'Trade at Reference' or 'Trade at Settlement' price code	e.g. '**TA**', '**TN**'. '**TP**'
72	ISIN	String	12	ISIN Code	This field will not be populated for instruments that are not yet tradable.
85	CFICode	String	6	The instrument classification according to ISO 10962.	e.g. '**FCEPSX**', '**FCECSX**', '**OCAFPS**'
91	MarketCode	String	4	Market Code	**LME** = the base metals market **LPM** = the precious metals market.

Offset	Field	Format	Len	Description	Values
95	MarketSegment	String	12	A string identifier for the individual market segment to which the outright belongs	e.g. ‘**Base**’, ‘**Precious**’, ‘**Ferrous**’, ‘**Minor**’
107	TickSizeID	Uint16	2	Tick Size ID	
109	LotSize	Uint64	8	The lot size for this tradable instrument	e.g. ‘**20**’, ‘**25**’, ‘**5000**’
117	LotSizeType	Char	1	Lot Size Type	**S** = Standard **M** = Mini
118	LastTradingDate	Uint32	4	Last Trading Date	
122	SettlementType	Char	1	Settlement Type	**C** = Cash **P** = Physical
123	SettlementPricing Method	Char	1	Settlement Pricing Method	**D** = Daily **M** = Monthly Average
124	UnderlyingType	Char	1	Underlying Type	**C** = Commodity F = Future **O** = Other
Total length			125		

4.1.2 Strategy Definition (302)

Describes individual strategy instruments available from LMEsource.

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message.	**302** = Strategy Definition
4	MDSource	String	2	Data Source of the message	**EL** = Electronic Market
6	SecurityID	Uint64	8	The id of the security for which trades in this tradable instrument are held	
14	ProductCode	String	2	The product code. This identifies the metal for the instrument	e.g. ‘**AA**’, ‘**AH**’,’**CA**’
16	ContractType	Char	1	Constant contract type	**F** =Future ***O*** *= Options* Default = Space (i.e. not applicable)
17	CurrencyCode	String	3	The currency code according to ISO 4217	e.g. ‘**USD**’

20	StrategyType	Uint8	1	Strategy Type	See Appendix 9.2
21	ContractCode	String	12	A unique code to identify the contract. This is the contract code for the near leg of the strategy.	e.g. '**PBDF**, **OCDF**, **NADT**'
33	ExerciseStyle	Int8	1	Exercise Style	***0*** *= European* ***1*** *= American* ***2*** *= Asian*
34	PriceCode	String	2	A code to represent the 'Trade at Reference' or 'Trade at Settlement' price code	e.g. '**TA**', '**TN**'. '**TP**'
36	MarketCode	String	4	Market Code	**LME** = the base metals market **LPM** = the precious metals market.
40	MarketSegment	String	12	A string identifier for the individual market segment to which the outright belongs	e.g. '**Base**', '**Precious**', '**Ferrous**', '**Minor**'
52	TickSizeID	Uint16	2	Tick Size	

54	LotSize	Uint64	8	Lot Size	
62	LotSizeType	Char	1	Lot Size Type	**S** = Standard **M** = Mini
63	LastTradingDate	Uint32	4	Last Trading Date	
67	SettlementType	Char	1	Settlement Type	**C** = Cash **P** = Physical
68	SettlementPricingMethod	Char	1	Settlement Pricing Method	**D** = Daily **M** = Monthly Average
69	UnderlyingType	Char	1	Underlying Type	**S** = security **O** = other
70	NumberOfLegs	Uint8	1	Number of Legs of this Strategy instrument	2 to 13
	LegID	Uint8	1	An identifier for each leg in the strategy	Starts from 1, increments by 1 for each leg
	LegSide	Uint8	1	Leg direction	**1** = Bid **2** = Ask
	LegRatio	Uint32	4	Leg ratio	
	LegSecurityID	Uint64	8	Security ID of leg	Reference to the outright definition of this leg
	LegPrice	Uint64	8	Leg Price	*Underlying price for a futures leg in*

					a delta-hedge custom strategy. *Null Value otherwise.*
Total length			71 + $22n_o$		

(n_o = value of NumberOfLegs)

4.1.3 Merged Instruments

There are occasions when there are two tradable instruments for a contract which share the shame Prompt Date. This occurs when the 3M Rolling Prompt instrument coincides with a calendar dated tradable instrument. This results in two Outright Definition messages being published by LMEsource, one for each tradable instrument. In each of these Outright Definition messages, the field MergedSecurityID will contain the SecurityID for the corresponding tradable instrument.

All market data updates, quotes, trades and statistics, will be published using the SecurityID for the Rolling Prompt tradable instrument.

4.2 Status Data

The information supplied in this section and its sub-sections applies to the data feed(s) marked with [◆]

Section	LME Level 1 Top of Book	LME Level 2 15 Levels Price Book	LME Level 3 Full Book
4.2	◆	◆	◆

4.2.1 Contract State (311)

This message indicates the trading state of instruments at the contract level, e.g. Copper Futures, Aluminium Options. The ContractCode field is common across the Contract State (311), Outright Definition (301) and Strategy Definition (302) messages and is used to link the contract state to individual tradable instruments.

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message	**311** = Contract State
4	MDSource	String	2	Data Source of the message	**EL** = Electronic Market
6	TimeOfEvent	Uint64	8	The time the trading state of the contract changed. Specified as nanoseconds from midnight, January 1, 1970 UTC. Precision is provided to the nearest microsecond.	
14	ContractCode	String	13	A string that combines the contract symbol, contract type and currency	e.g. '**PBDF**, **OCDF**, **NADT**'
27	TradingState	Uint8	1	Trading State	**0** = SOD **1** = Pre-Open **2** = Open **3** = Post Trade **4** = Close **5** = EOD **6** = Technical Halt

28	StartTime	Uint64	8	Trading State Start Time	
36	EndTime	Uint64	8	Trading State End Time	
44	TradingStateCondition	Char	1	Current trading state condition	**P** = Pause **H** = Halt Blank = Active (or when pause/halt has been lifted)
45	Filler	String	3		
Total length			48		

4.2.2 Instrument State (312)

This message indicates the trading state of an individual instrument.

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message	**312** = Instrument State
4	MDSource	String	2	Data Source of the message	**EL =** Electronic Market
6	TimeOfEvent	Uint64	8	The time the trading state of the instrument changed. Specified as nanoseconds from midnight,	

				January 1, 1970 UTC. Precision is provided to the nearest microsecond.	
14	SecurityID	Uint64	8	Security ID of tradable instrument	
22	TimetableControlType	Char	1	Timetable Control Type	**A** = Automatic **M** = Manual
23	TradingState	Uint8	1	Trading State	**0** = SOD **1** = Pre-Open **2** = Open **3** = Post Trade **4** = Close **5** = EOD **6** = Technical Halt
24	StartTime	Uint64	8	Trading State Start Time	
32	EndTime	Uint64	8	Trading State End Time	
40	TradingStateCondition	Char	1	Current trading state condition	**P** = Pause **H** = Halt Blank = Active (or when pause/halt has been lifted)
41	Filler	String	3		

Total length	44

4.3 Order Book Data

With the exception of the Quote Request (356) and IOP (354) messages described in sections 4.3.8 and 4.3.9 the messages in this section are only published when the market is in the ‘Open’ state.

4.3.1 Level 1 Top Of Book (355)

The information supplied in this section and its sub-sections applies to the data feed(s) marked with [◆]

Section	LME Level 1 Top of Book	LME Level 2 15 Levels Price Book	LME Level 3 Full Book
4.3.1	◆		

The Top of Book message is generated when the top price level has been modified. There are no ‘New’, ‘Change’ or ‘Delete’ actions for the Top of Book. Whenever the price, quantity or the number of orders at the Top of Book changes, a new message is sent.

Whenever an order book is emptied as a result of market activity, a Top Of Book message with price fields set to Null and aggregate quantity / number of orders set to zero will be sent.

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message	**355** = Top of Book
4	MDSource	String	2	Data Source of the message	**EL =** Electronic Market

6	TimeOfEvent	Uint64	8	The time LMEselect updated the orderbook, triggering a Top of Book message. Specified as nanoseconds from midnight, January 1, 1970 UTC. Precision is provided to the nearest microsecond.	
14	SecurityID	Uint64	8	Security ID of tradable instrument	
22	AggregateBidQuantity	Uint64	8	Aggregated quantity available on the bid side	
30	AggregateAskQuantity	Uint64	8	Aggregated quantity available on the ask side	
38	BidPrice	Int64	8	The bid price	
46	AskPrice	Int64	8	The ask price.	
54	NumberBidExplicitOrders	Uint32	4	The total number of Bid orders in the best price level for Explicit Orders	

58	BidQtyExplicitOrders	Uint64	8	The total quantity of Bid orders in the best price level for Explicit Orders	
66	NumberAskExplicitOrders	Uint32	4	The total number of Ask orders in the best price level for Explicit Orders	
70	AskQtyExplicitOrders	Uint64	8	The total quantity of Ask orders in the best price level for Explicit Orders	
78	NumberBidImpliedOrders	Uint32	4	The total number of Bid orders in the best price level for Implied Orders	
82	BidQtyImpliedOrders	Uint64	8	The total quantity of Bid orders in the best price level for Implied Orders	
90	NumberAskImpliedOrders	Uint32	4	The total number of Ask orders in the best price level for Implied Orders	

94	AskQtyImpliedOrders	Uint64	8	The total quantity of Ask orders in the best price level for Implied Orders	
Total length			102		

4.3.2 Level 2 Aggregate Order Book Update (353)

The information supplied in this section and its sub-sections applies to the data feed(s) marked with [◆]

Section	LME Level 1 Top of Book	LME Level 2 15 Levels Price Book	LME Level 3 Full Book
4.3.2		◆	

The aggregate order book is sent whenever there is an order book change within the top 15 price levels. The ordering of the price levels in the Aggregate Order Book (353) message in the RFS snapshot will be from best to worst.

Refer to Section 7 - Aggregate Order Book Management for details on the Aggregate Order Book Update message.

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message	**353 =** Aggregate Order Book Update

4	MDSource	String	2	Data Source of the message	**EL =** Electronic Market
6	TimeOfEvent	Uint64	8	The time LMEselect updated the orderbook, triggering an Aggregate Orderbook update message. Specified as nanoseconds from midnight, January 1, 1970 UTC . Precision is provided to the nearest microsecond.	
14	SecurityID	Uint64	8	Security ID of tradable instrument	
22	NoEntries	Uint8	1	Number of book entries within the message	
23	AggregateQuantity	Uint64	8	Total quantity of orders in this price level and side	
31	Price	Int64	8	Price	
39	NumberOfExplicitOrders	Uint32	4	Number of Explicit orders	

				in this price level and side	
43	TotalQtyOfExplicitOrders	Uint64	8	Total quantity of Explicit orders in this price level and side	
51	NumberOfImpliedOrders	Uint32	4	Number of Implied orders in this price level and side	
55	TotalQtyOfImpliedOrders	Uint64	8	Total quantity of Implied orders in this price level and side	
63	Side	Int8	1	Side of the order	**1** = Bid **2** = Ask
64	PriceLevel	Uint8	1	Indicates the price level (within top 15) of the information carried in the message	**1** to **15**
65	UpdateAction	Uint8	1	Type of market data update action	**0** = New **1** = Change **2** = Delete
Total length			23 + $43n_o$		

(n_o = value of NoEntries)

4.3.3 Level 3 Order Add (357)

The information supplied in this section and its sub-sections applies to the data feed(s) marked with [◆]

Section	LME Level 1 Top of Book	LME Level 2 15 Levels Price Book	LME Level 3 Full Book
4.3.3			◆

The Order Add message is generated when an order is placed in order book. An incoming, aggressing order that matches against one or more resting orders will not be published as an Order Add message, unless it has residual volume after matching.

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message	**357** = Order Add
4	MDSource	String	2	Data Source of the message	**EL** = Electronic Market
6	TimeOfEvent	Uint64	8	The time LMEselect added this order to the orderbook. Specified as nanoseconds from midnight, January 1, 1970 UTC. Precision is provided to the nearest microsecond.	

14	T1	Uint64	8	Time when the order request reached the LMEselect gateway	NULL for Speedbump / Stop / Iceberg orders
22	T2	Uint64	8	Time when LMEselect captured the order request	NULL for Speedbump / Stop / Iceberg orders
30	T3	Unit64	8	Time when LMEselect added this order	NULL for Speedbump / Stop / Iceberg orders
38	SecurityID	Uint64	8	Security ID of tradable instrument	
46	OrderID	Uint64	8	Order ID assigned by LMEselect	
54	Side	Int8	1	Side of order	**1** = Bid **2** = Ask
55	Quantity	Uint32	4	Order Quantity	
59	Price	Int64	8	Price of order	
67	OrderBookPosition	Uint32	4	Relative order position within this side of the orderbook based upon price and time priority	
71	Filler	String	1		
Total length			72		

4.3.4 Level 3 Order Amend (358)

The information supplied in this section and its sub-sections applies to the data feed(s) marked with [◆]

Section	LME Level 1 Top of Book	LME Level 2 15 Levels Price Book	LME Level 3 Full Book
4.3.4			◆

The Order Amend message is generated when an order is amended in the orderbook.

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message	**358** = Order Amend
4	MDSource	String	2	Data Source of the message	**EL =** Electronic Market
6	TimeOfEvent	Uint64	8	The time LMEselect amended this order in the orderbook. Specified as nanoseconds from midnight, January 1, 1970 UTC. Precision is provided to the nearest microsecond.	
14	T1	Uint64	8	Time when the order request	NULL for Speedbump /

				reached the LMEselect gateway	Stop / Iceberg orders
22	T2	Uint64	8	Time when LMEselect captured the order request	NULL for Speedbump / Stop / Iceberg orders
30	T3	Unit64	8	Time when LMEselect amended this order	NULL for Speedbump / Stop / Iceberg orders
38	SecurityID	Uint64	8	Security ID of tradable instrument	
46	OrderID	Uint64	8	Order ID of amended order	
54	Side	Int8	1	Side of order	**1** = Bid **2** = Ask
55	Quantity	Uint32	4	Amended delta quantity	
59	Price	Int64	8	Price of order	
67	OrderBookPosition	Uint32	4	Relative order position within the order book / price / side	
71	Filler	String	1		
Total length			72		

4.3.5 Level 3 Order Cancel (359)

The information supplied in this section and its sub-sections applies to the data feed(s) marked with [◆]

Section	LME Level 1 Top of Book	LME Level 2 15 Levels Price Book	LME Level 3 Full Book
4.3.5			◆

The Order Cancel message is generated when an order is cancelled in the orderbook.

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message	**359** = Order Cancel
4	MDSource	String	2	Data Source of the message	**EL** = Electronic Market
6	TimeOfEvent	Uint64	8	The time LMEselect cancelled this order from the orderbook. Specified as nanoseconds from midnight, January 1, 1970 UTC. Precision is provided to the nearest microsecond.	
14	T1	Uint64	8	Time when the order request	

				reached the LMEselect gateway	
22	T2	Uint64	8	Time when LMEselect captured the order request	
30	T3	Unit64	8	Time when LMEselect cancelled this order	
38	SecurityID	Uint64	8	Security ID of tradable instrument	
46	OrderID	Uint64	8	Order ID for the cancelled order	
54	Side	Int8	1	Side of order	**1** = Bid **2** = Ask
55	Filler	String	1		
Total length			56		

4.3.6 Order Executed (350)

The information supplied in this section and its sub-sections applies to the data feed(s) marked with [◆]

Section	LME Level 1 Top of Book	LME Level 2 15 Levels Price Book	LME Level 3 Full Book
4.3.6			◆

The Order Executed message is generated when an order is executed. It will only be produced for explicit orders, not implied orders.

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message	**350** = Order Executed
4	MDSource	String	2	Data Source of the message	**EL** = Electronic Market
6	TimeOfEvent	UInt64	8	The time LMEselect created the trade that executed this order. Specified as nanoseconds from midnight, January 1, 1970 UTC. Precision is provided to the nearest microsecond.	
14	SecurityID	Uint64	8	Security ID of the trade	
22	Price	Int64	8	The price of the trade	
30	Quantity	Uint32	4	Trade volume.	
34	OrderID	Uint64	8	Order ID assigned by LMEselect	
42	MatchID	Uint64	8	An id that can be used to identify all orders that	

				matched against each other	
50	TradeCancelFlag	Uint8	1	Trade Cancel Flag	**0** = Normal trade ***1*** *= Cancelled trade*
51	TradeSide	Int8	1	The side of the trade for this order	
52	NumOfLegs	Uint8	1	Number of legs present	
	LegID	Uint64	8	Leg Security ID	
	LegSide	Uint8	1	Leg side as applicable depending on the trade side	
	LegPrice	Int64	8	Price allocated to this leg of the strategy	
	LegQuantity	Uint32	4	Quantity allocated to this leg of the strategy	
	LegMatchID	Uint64	8	Leg Match ID	
Total length			53 + 29no		

(n_o = value of NumberOfLegs)

4.3.7 Orderbook Clear (335)

The information supplied in this section and its sub-sections applies to the data feed(s) marked with [◆]

Section	LME Level 1 Top of Book	LME Level 2 15 Levels Price Book	LME Level 3 Full Book
4.3.7	◆	◆	◆

The Orderbook Clear message is generated when order book is required to be clear, for example after the transition between certain market states, or when an instrument's Trading State Condition is 'Trade Halt'.

The market state transitions that result in an Orderbook Clear message are:

1. Pre-Open to Open. Any order events entered during Pre-Open that didn't match during uncrossing are published after the Orderbook Clear message.
2. Open to Post-Trade.

Following an 'Orderbook Clear' message, if an instrument is subsequently open for trading any resting orders will be sent as Aggregate Order Book Update / Order Add messages to allow clients to rebuild the orderbook.

An Orderbook Clear message will not be published when an instrument enters or leaves 'Trade Pause'.

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message	**335** = Orderbook Clear
4	MDSource	String	2	Data Source of the message	**EL =** Electronic Market
6	TimeOfEvent	Uint64	8	The time the trading state of the	

				instrument changed, resulting in the Orderbook Clear message. Specified as nanoseconds from midnight, January 1, 1970 UTC. Precision is provided to the nearest microsecond.	
14	SecurityID	Uint64	8	Security ID of tradable instrument	
Total length			22		

4.3.8 Quote Request (356)

The information supplied in this section and its sub-sections applies to the data feed(s) marked with [◆]

Section	LME Level 1 Top of Book	LME Level 2 15 Levels Price Book	LME Level 3 Full Book
4.3.8	◆	◆	◆

The Quote Request message is generated when a request for quote is accepted by the ME.

Offset	Field	Format	Len	Description	Values
0	*MsgSize*	*Uint16*	*2*	*Size of the message*	

2	*MsgType*	*Uint16*	*2*	*Type of message*	***356** = Quote Request*
4	*MDSource*	*String*	*2*	*Data Source of the message*	***EL =** Electronic Market*
6	*TimeOfEvent*	*Uint64*	*8*	*The time LMEselect created this quote request.* *Specified as nanoseconds from midnight, January 1, 1970 UTC.* *Precision is provided to the nearest microsecond.*	
14	*SecurityID*	*Uint64*	*8*	*Security ID of tradable instrument*	
22	*QuoteRequestType*	*Uint8*	*1*	*Type of Quote Request*	***1** = Manual (a single quote request)* ***2** = Auto (a streaming quote request)*
23	*Side*	*Int8*	*1*	*Side of order*	***1** = Bid* ***2** = Ask* ***null** = Two sided quote*
24	*Quantity*	*Uint32*	*4*	*Amended delta quantity*	
28	*Filler*	*String*	*2*		
Total length			*30*		

4.3.9 IOP (354)

The information supplied in this section and its sub-sections applies to the data feed(s) marked with [◆]

Section	LME Level 1 Top of Book	LME Level 2 15 Levels Price Book	LME Level 3 Full Book
4.3.9	◆	◆	◆

The IOP is the predicted opening trade price which is calculated using the uncrossing algorithm during Pre-Open. If there are no crossed prices, no IOP is calculated or disseminated. If a tradable instrument does not have a crossed order book but does have both a bid and offer price then a mid-price (IOMP) will be calculated and published instead of the IOP.

An IOP and IOMP are mutually exclusive, i.e. when an IOP is available the IOMP is not applicable, and will be set to null, and vice versa.

At the market state transitions from Pre-Open to Open, an IOP message with IndicativeOpeningPrice, IndicativeOpeningVolume and IndicativeOpeningMidPrice set to null is published for every tradable instrument to indicate the IOP and IOMP prices are no longer valid.

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message	**354** = IOP
4	MDSource	String	2	Data Source of the message	**EL** = Electronic Market
6	TimeOfEvent	Uint64	8	The time LMEselect computed or	

				derived the IOP/IOMP. Specified as nanoseconds from midnight, January 1, 1970 UTC. Precision is provided to the nearest microsecond.	
14	SecurityID	Uint64	8	Security ID of tradable instrument	
22	IndicativeOpeningPrice	Int64	8	Indicative Opening Price	Default = NULL
30	IndicativeOpeningVolume	Uint32	4	Indicative Opening Volume	Default = 0
34	IndicativeOpeningMid Price	Int64	8	Indicative Opening Mid Price	Default = NULL
42	Filler	String	2		
Total length			44		

4.4 Trade and Price Data

4.4.1 Match Trade (360)

The information supplied in this section and its sub-sections applies to the data feed(s) marked with [◆]

Section	LME Level 1 Top of Book	LME Level 2 15 Levels Price Book	LME Level 3 Full Book

4.4.1	◆	◆	

The Match Trade message is generated each time a trade has been performed. When an incoming order matches against multiple resting orders, there will be one Match Trade message published for each matched trade.

At market open, a Match Trade message will be published for any orders entered during Pre-Open that matched

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message	**360** = Match Trade
4	MDSource	String	2	Data Source of the message	**EL =** Electronic Market
6	TimeOfEvent	Uint64	8	The time LMEselect created this trade. Specified as nanoseconds from midnight, January 1, 1970 UTC. Precision is provided to the nearest microsecond.	
14	SecurityID	Uint64	8	Security ID of the trade	
22	Price	Int64	8	The price of the trade	
30	Quantity	Uint32	4	Trade volume.	

34	MatchID	Uint64	8	An id that can be used to identify all orders that matched against each other	
42	TradeCancelFlag	Uint8	1	Trade Cancel Flag	**0** = Normal trade *__1__ = Cancelled trade*
43	SubTypeOfTrade	Int8	1	Sub type of trade	**1** = The trade resulted from an explicit order **7** = The trade resulted from an implied order
Total length			44		

4.4.2 EOD Trade Statistics (351)

The information supplied in this section and its sub-sections applies to the data feed(s) marked with [◆]

Section	**LME Level 1** **Top of Book**	**LME Level 2** **15 Levels Price Book**	**LME Level 3** **Full Book**
4.4.2	◆	◆	◆

The EOD Trade Statistics message is generated when the market moves to the Post-Trade session for instruments that have traded. LMEsource will publish the LMEselect Opening and Closing Prices, LMEselect Trading High, and LMEselect Trading Low for all contracts that have traded during the day.

Note: due to sequence of message publication the EOD Trade Statistic message may be disseminated before the Instrument State (312) message that confirms the instrument as being in the ‘Post-Trade’ or ‘Closed’ state.

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message	**351** = EOD Trade Statistics
4	MDSource	String	2	Data Source of the message	**EL** = Electronic Market
6	TimeOfEvent	Uint64	8	The time LMEselect computed or derived the designated price. Specified as nanoseconds from midnight, January 1, 1970 UTC. Precision is provided to the nearest microsecond.	
14	SecurityID	Uint64	8	The security ID of tradable instrument	
22	OpenPrice	Int64	8	Opening trade price for the current day	
30	HighPrice	Int64	8	Highest trade price for the current day	
38	LowPrice	Int64	8	Lowest trade price for the current day	

46	ClosingPrice	Int64	8	Closing (final) trade price for the current day	
Total length			54		

4.4.3 Intraday Trade Statistics (352)

The information supplied in this section and its sub-sections applies to the data feed(s) marked with [◆]

Section	LME Level 1 Top of Book	LME Level 2 15 Levels Price Book	LME Level 3 Full Book
4.4.3		◆	◆

The Intraday Trade Statistics message contains trade information for completed trades. The trade statistics information is provided on a snapshot basis.

At market open, if there has been any uncrossing activity in the instrument, three messages will be published, one each for open price, high price and low price. Each of these three messages will include any previously published OHL price.

If there was no uncrossing activity in the instrument, a single Intraday Trade Statistics message is published, with the OpenPrice, HighPrice and LowPrice fields set to null.

The OpenPrice is only populated as a result of matched trades during uncrossing, if there was no uncrossing activity in the instrument the OpenPrice will remain as null throughout the trading day.

Upon the first trade of the day, two messages will be published, one each for high price and low price.

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message	**352** = Intraday Trade Statistics

4	MDSource	String	2	Data Source of the message	**EL** = Electronic Market
6	TimeOfEvent	Uint64	8	The time LMEselect computed or derived the designated price. Specified as nanoseconds from midnight, January 1, 1970 UTC. Precision is provided to the nearest microsecond.	
14	SecurityID	Uint64	8	The Security ID of tradable instrument	
22	OpenPrice	Int64	8	Opening trade price for the current day	
30	HighPrice	Int64	8	Highest trade price for the current day	
38	LowPrice	Int64	8	Lowest trade price for the current day	
Total length			46		

5 Non-Electronic Message Formats

This section will be populated in a later version of this document.

6 Recovery

The information supplied in this section and its sub-sections applies to the data feed(s) marked with [◆]

Section	LME Level 1 Top of Book	LME Level 2 15 Levels Price Book	LME Level 3 Full Book
4	◆	◆	◆

LMEsource provides three different mechanisms for recovering missed data:

1. Line arbitration – using dual multicast channels (Line A and Line B)
2. Retransmission Server – recovery of a limited number of messages
3. Refresh Server – snapshot of current market state

These mechanisms should be used as described in the following table.

Event	Action
Packet lost on one either Line A or Line B	Try to recover data from the other line with a configurable timeout (“arbitration mechanism”).
Dropped packet(s) on both Line A and Line B	Recover dropped message(s) from the Retransmission Server.
Late start up or extended intraday outage	Wait for a refresh of the current market state and then continue with real time messages.

6.1 Gap Detection

Each packet provides the sequence number (SN) of the first message it contains. This sequence number starts at 1 and increases with each subsequent message.

The sequence numbers provided in every packet header is calculated by adding the previous sequence number and the message count, as shown in the table below:

Packet	Sequence Number	Message Count
Packet 1	1	4
Packet 2	5	2
Packet 3	7	1
Packet 4	8	3
Packet 5	11	1

If the client drops the first five packets they would request a gap fill for messages 1-11.

All messages conform to the message level sequencing. Each channel has its own sequence number. This allows recipients to detect gaps or duplicates in each message sequence number and, if appropriate, reconcile them (line arbitration) with the primary or secondary multicast groups or request retransmission of the missing / corrupted messages.

Users should use this sequence number to detect gaps in the transmission of messages.

The following diagram illustrates how the message sequence number should be used to detect gaps in the feed.

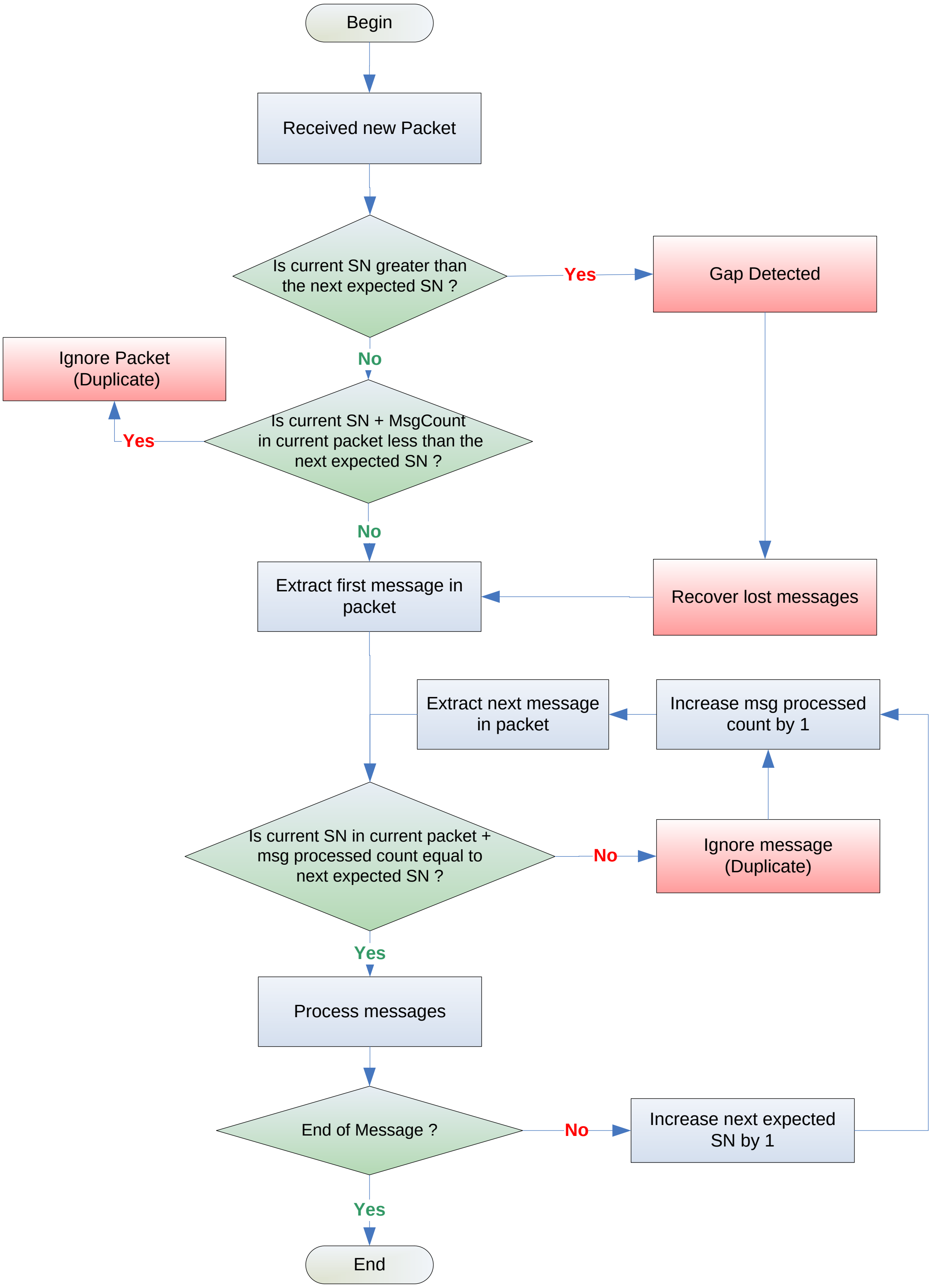

Begin
Received new Packet
Is current SN greater than the next expected SN ?
Yes
Gap Detected
No
Ignore Packet (Duplicate)
Yes
Is current SN + MsgCount in current packet less than the next expected SN ?
No
Extract first message in packet
Recover lost messages
Extract next message in packet
Increase msg processed count by 1
Is current SN in current packet + msg processed count equal to next expected SN ?
No
Ignore message (Duplicate)
Yes
Process messages
End of Message ?
No
Increase next expected SN by 1
Yes
End

Figure 2: Gap Detection using the Sequence Number (SN)

6.2 Line Arbitration

Client applications should check the sequence number (SN) and message count (MC) for every packet received. SNs are unique and increase monotonically for each service, the MC indicates the number of messages within each packet.

Line A and Line B are identical in terms of:

- SNs
- Messages that are sent
- Sequence in which messages are sent

However it is not guaranteed that the packet content between Line A and Line B will be the same. For example the third packet of the day from the Line A could contain SN 10 with MC 3, whereas the third packet of the day from Line B could contain SN 9 with MC 4. For this reason clients must arbitrate on SN (at the message level) rather than packet content. Client applications should listen to both Line A and Line B in real-time. Clients should look at packets coming from both lines and process the ones that arrive first, regardless of whether they came from Line A or Line B. It is advisable to apply the "first come – first served" rule.

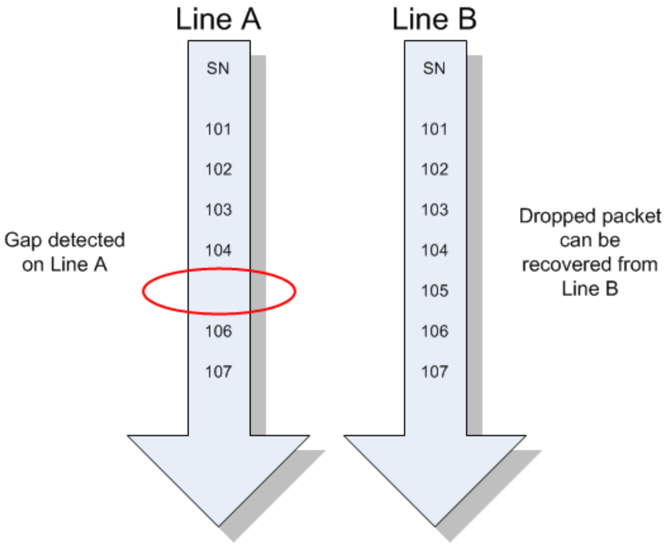


Figure 3 – Detecting Missing Packets

Additional Notes;

- The above example of a dropped packet is a simplified example assuming 1 message per packet, in reality each packet is likely to contain multiple messages

- Whilst the order of individual messages between Line A and Line B will be identical, there is no guarantee that the packets will contain exactly the same messages.

- In the example below, three packets are sent on each line, but message 'OrderUpdate3' appears in one packet from Line A but in the subsequent packet on Line B.

Primary		
Messages	**MC**	**SN**
OrderUpdate1 OrderUpdate2 OrderUpdate3	3	101
Trade1 OrderUpdate4	2	104
Trade2 Statistics1	2	106

Secondary		
SN	**MC**	**Messages**
101	2	OrderUpdate1 OrderUpdate2
103	3	OrderUpdate3 Trade1 OrderUpdate4
106	2	Trade2 Statistics1

Figure 4 – Normal Message Delivery

6.3 Retransmission Service

The retransmission service is provided via the TCP/IP protocol and is designed to allow clients to recapture a small number of missed messages already published on the real time channels.

It is not intended that clients use the retransmission service to recover data after long outages or on late start up (in these situations, clients should use the Refresh service). To that end, it aims to support the retransmission of the data covering the market activities for the last 15-30 seconds only. This figure is indicative only and may be shorter than 15 seconds if a spike happens in the market. The sequence range of messages that a client can request and the number of retransmission requests permitted per day is also limited.

The following diagram illustrates the message flow during a retransmission session:

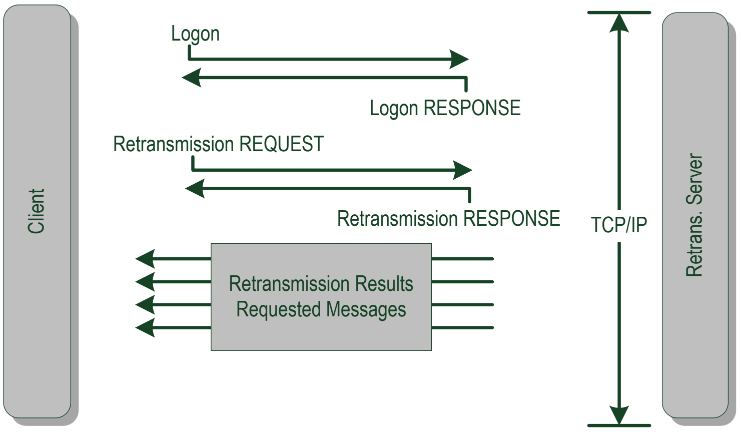


Figure 5: Retransmission Request

Logon

The client establishes a TCP/IP connection and initiates a session by sending the Logon message. Once the client is authenticated the server will respond immediately with the Logon Response message. If the client does not send a Logon message within the logon timeout interval, the server will close the connection.

Logons may be rejected for the following reasons:

- Invalid username
- User already connected

In all cases the server will close the connection after sending the Logon Response message.

Making a request

The client can make a retransmission request by sending the Retrans Request message. The server will respond with a Retrans Response message to indicate whether the request has been accepted or not.

In the case of a successful request the server will send the requested messages immediately after the Retrans Response message.

The sequence numbers will be the same as when they were first sent on the real time multicast channel. The framing of the retransmitted messages into a packet may differ from the original transmission.

Retransmission requests may be rejected for the following reasons:

- Unknown channel ID or illegal (not authorized)
- Messages not available

- Exceeds maximum sequence range
- Exceeds maximum requests in a day

In the case where the client has exceeded the maximum number of requests allowed in a day, the server will close the connection after sending the Retrans Response message.

The following diagram is a guideline of the flow of logic when making a request:

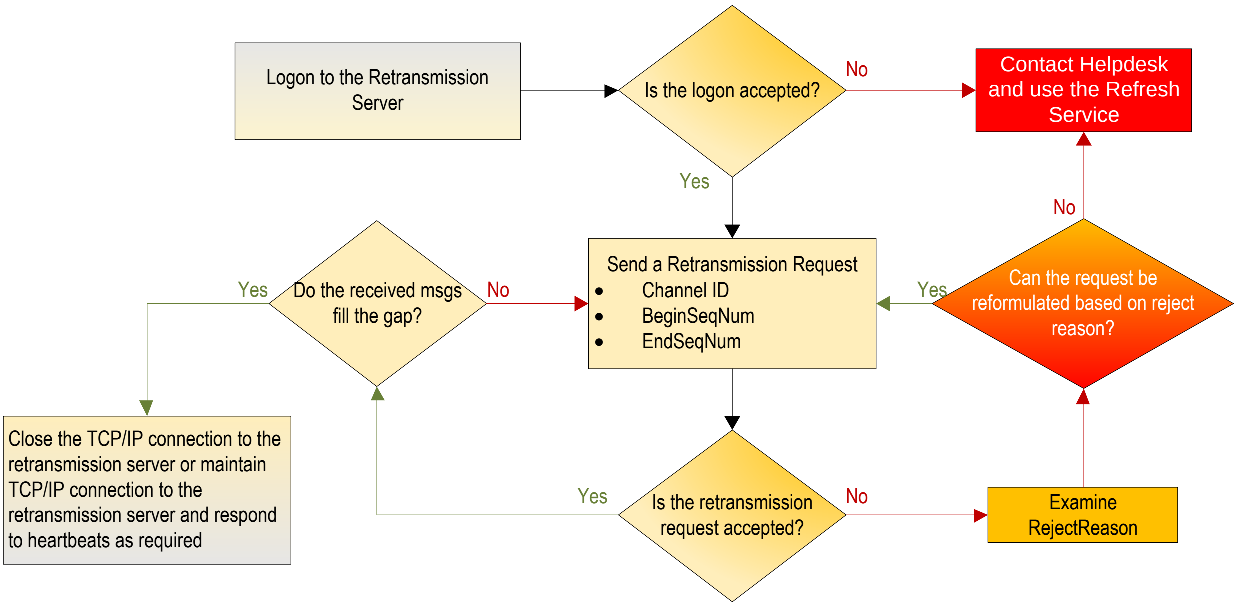

Logon to the Retransmission Server
Is the logon accepted?
No
Contact Helpdesk and use the Refresh Service
Yes
No
Send a Retransmission Request
Channel ID
BeginSeqNum
EndSeqNum
Yes
Can the request be reformulated based on reject reason?
Yes
Do the received msgs fill the gap?
No
Close the TCP/IP connection to the retransmission server or maintain TCP/IP connection to the retransmission server and respond to heartbeats as required
Yes
Is the retransmission request accepted?
No
Examine RejectReason

Figure 6: Requesting Dropped Packets

Multiple requests and concurrent sessions

Clients can send multiple requests during a session and can keep the session open during idle periods by responding to heartbeats sent by the server. Concurrent sessions however will not be supported. Each user can only have one session open at a time.

If a client makes multiple requests, the server will process them serially. Clients are unable to cancel outstanding requests.

Heartbeats

To determine the health of the user connection on the TCP/IP channel, the Retransmission Server will send regular heartbeat packets to the user. The heartbeat frequency is 30 seconds. The client application must respond with a “Heartbeat Response” packet. The time out for this heartbeat response packet is set at 5 seconds. If no response is received by the server within this timeframe, the TCP/IP session will be disconnected.

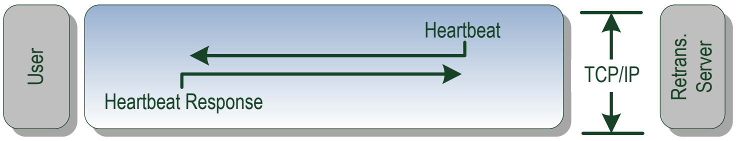


Figure 7: Retransmission Server Heartbeat Message

A "heartbeat response" packet consists in an exact copy of the incoming heartbeat packet.

Closing the session

Sessions should be terminated by gracefully closing the TCP/IP connection.

System limits

The system limits mentioned above are set as follows:

System Limit	Value
Maximum sequence range that can be requested	10,000
Maximum number of requests per day	1,000
Logon timeout (seconds)	5
Heartbeat interval (seconds)	30
Heartbeat response timeout (seconds)	5

Please note that the maximum number of requests per day limit is across all channels.

High availability

For each site, two sets of IP addresses and ports are provided for the retransmission service in order to facilitate high availability. Clients may connect to both retransmission services at the start of the day and maintain the connection during the day by responding to heartbeats.

The LME will provide a Fully Qualified Domain Name (FQDN) to access the retransmission service. Under normal operation the FQDN will resolve to the high availability retransmission service at the Active Production site. Clients may connect to the retransmission service at the start of the day and maintain the connection during the day by responding to heartbeats.

Disaster recovery

During normal conditions the retransmission service at the disaster site is not available. If clients attempt to connect, this will fail.

In the unlikely event of a disaster recovery situation, the retransmission service at the disaster site will be brought up and clients may connect via the backup IP addresses and ports.

6.4 Refresh Service

The refresh service is designed to allow clients to recover from a large-scale data loss. This can happen after a late start or during a major outage.

Synchronisation is on a per channel basis. For each real time multicast channel there exists a corresponding refresh multicast channel on which snapshots of the market state are sent at regular intervals throughout the business day. No ordering should be assumed between the various different data types unless otherwise stated – this is due to the nature of using multiple different multicast channels for refresh.

Snapshot

A snapshot of the market state is described in the table below.

Message	Snapshot Description
Instruments Definition	A full list of all Outright Definition, which includes any modifications or additions made intraday. The order is sent as: • Outright Definition (301) • Strategy Definition (302)
Instrument Status	The most recent Instrument State (312) message of declared SecurityID.
Contract Status	The most recent Contract State (311) messages of declared ContractCode.

Orders	For L1 clients: the latest level 1 Price book via Top of Book (355) messages. For L2 clients: the latest 15 levels of Price book via Aggregate Order Book (353) messages. The ordering of the price levels in the Aggregate Order Book (353) message in the RFS snapshot will be from worst to best. This is the reverse of the sequence in the real-time Aggregate Order Book (353) message. For L3 clients: the snapshot of the full orderbook via Order Add (357) messages. The OrderbookPosition field and the T1, T2 and T3 timestamp fields won’t be populated in the RFS snapshot.
Intraday Trade Statistics	The latest Intraday Trade Statistics (352) message of declared SecurityID.
EOD Trade Statistics	The latest EOD Trade Statistics (351) message of declared SecurityID.
Match Trade	A replay of all Match Trade (360) messages from start of day.
IOP	The latest IOP (354) message of declared SecurityID
Quote Request	The latest Quote Request (356) message of declared Security ID.

The ordering of refresh messages types within the multicast channels is detailed below:

Channel	Refresh Sequence
Instrument Definition	Outright Definition (301), Strategy Definition (302), EOD Trade Statistics (351)
Order (L1 clients)	Instrument State (312), Contract State (311), IOP (354), Quote Request (356), Top of Book (355)

Order (L2 clients)	Instrument State (312), Contract State (311), IOP (354), Quote Request (356), Aggregate Orderbook Update (353)
Order (L3 clients)	Instrument State (312), Contract State (311), IOP (354), Quote Request (356), Order Add (357)
Trade Statistics	Intraday Trade Statistics (352)
Trade	Match Trade (360)

Refresh complete

A Refresh Complete message is sent at the end of a snapshot indicating the sequence number with which the snapshot is synchronized.

Snapshot processing

Below is an overview of the steps to carry out in order to process a channel snapshot.

- Subscribe to the real time multicast channel and cache received messages.
- Subscribe to the corresponding refresh multicast channel and discard messages until the Refresh Complete message is received.
- Process received messages until the next Refresh Complete message is received.
- Store the LastSeqNum sequence number provided in the Refresh Complete.
- Unsubscribe to the refresh multicast channel.

- Discard the cached real time messages with sequence number less than or equal to LastSeqNum.
- Process the remaining cached real-time messages and resume normal processing.

Missed messages

The retransmission server does not support refresh channels. If a client misses messages, it must wait for the next snapshot. Similarly if a client starts listening during the middle of a snapshot, it must wait for the next snapshot.

7 Level 2 Aggregate Order Book Management

The information supplied in this section and its sub-sections applies to the data feed(s) marked with [◆]

Section	LME Level 1 Top of Book	LME Level 2 15 Levels Price Book	LME Level 3 Full Book
5		◆	

Book Identification

A book is uniquely identified by SecurityID. SecurityID is an unsigned integer representation of 8 bytes, from Outright Definition (301) messages and Strategy Definition (302) messages.

Partial Price Depth

The price level within the Aggregate Order Book Update message determines the number of price levels the order price is away from the best price for a given order book. An order with price level 1 means the order's price is the best price, a price level of 2 will be used for orders at the next best price, etc. The ordering of the price levels in the Aggregate Order Book (353) message in the RFS snapshot will be from best to worst.

LMEsource provides a view of 15 price depths of aggregate order book for the LME Markets. This view can be visualized as a number of rows in a table for each of the bid and ask sides. On each side there are a number of rows showing the aggregate quantity available at a number of price levels.

For brevity, the below examples use 5 levels of price depths to demonstrate the aggregate orderbook update mechanism. The same principles apply to 15 levels of price depths.

The table below shows the starting position of the orderbook.

Bid Side					Ask Side				
Price Level	No. Of Implied Orders	No. of Explicit Orders	Agg. Quantity	Price	Price	Agg. Quantity	No. of Explicit Orders	No. of Implied Orders	Price Level
1	1	2	700	9730	9760	500	1	2	1
2	0	1	350	9720	9770	300	2	0	2

3	0	1	150	9710	9780	100	1	0	3
4	0	1	250	9700	9790	150	1	0	4
5	-	-	-	-	-	-	-	-	5

Book Updates

Book update messages are generated by LMEsource as delta messages defined in section 4.3.2 Aggregate Order Book Update (353). Each message may contain any combination of new, changed, or deleted entries for a book. The nature of an entry is defined by its UpdateAction.

UpdateAction	Meaning	Value
New	to create/insert a new price level	0
Change	to update aggregate quantity at a price level	1
Delete	to remove a price level	2

7.1 Example 1 – Quantity Reduction and Explicit Addition

For example suppose one of the explicit ask orders at price level 9770 is reduced in quantity such that the total quantity is now 200, and at the same time a new explicit ask order is added with a price of 9850 and quantity of 300, then the following message is sent:

Offset	Field Name	Value
0	MsgSize	109
2	MsgType	353
4	MDSource	EL
6	TimeOfEvent	123456789
14	SecurityID	1234

22	NoEntries	2
23	AggregateQuantity	200
31	Price	9770
39	NumberOfExplicitOrders	2
43	TotalQtyOfExplicitOrders	200
51	NumberOfImpliedOrders	0
55	TotalQtyOfImpliedOrders	0
63	Side	2 (Ask)
64	PriceLevel	2
65	UpdateAction	1
66	AggregateQuantity	300
74	Price	9850
82	NumberOfExplicitOrders	1
86	TotalQtyOfExplicitOrders	300
94	NumberOfImpliedOrders	0
98	TotalQtyOfImpliedOrders	0
106	Side	2 (Ask)
107	PriceLevel	5
108	UpdateAction	0

The resulting orderbook should now be as follows:

Bid Side					Ask Side				
Price Level	No. Of Implied Orders	No. of Explicit Orders	Agg. Quantity	Price	Price	Agg. Quantity	No. of Explicit Orders	No. of Implied Orders	Price Level
1	1	3	700	9730	9760	500	3	2	1
2	0	1	350	9720	9770	200	2	0	2
3	0	1	150	9710	9780	100	1	0	3
4	0	1	250	9700	9790	150	1	0	4
5	-	-	-	-	9850	300	1	0	5

7.2 Example 2 – Implicit Level Adjustments

The client must adjust the price level of entries below deleted or inserted entries. Potential level adjustments must be carried out after each single entry in the Aggregate Order Book Update message.

For example, if a bid order with price 9740 and quantity 50 is added to the order book above, it will cause the following message to be sent:

Offset	Field Name	Value
0	MsgSize	66
2	MsgType	353
4	MDSource	EL
6	TimeOfEvent	123456789
14	SecurityID	1234
22	NoEntries	1

23	AggregateQuantity	50
31	Price	9740
39	NumberOfExplicitOrders	1
43	TotalQtyOfExplicitOrders	50
51	NumberOfImpliedOrders	0
55	TotalQtyOfImpliedOrders	0
63	Side	1 (Bid)
64	PriceLevel	1
65	UpdateAction	0

After processing this message, the client's book should look as follows:

Bid Side					Ask Side				
Price Level	No. Of Implied Orders	No. of Explicit Orders	Agg. Quantity	Price	Price	Agg. Quantity	No. of Explicit Orders	No. of Implied Orders	Price Level
1	0	1	50	9740	9760	500	3	2	1
2	1	3	700	9730	9770	200	2	0	2
3	0	1	350	9720	9780	100	1	0	3
4	0	1	150	9710	9790	150	1	0	4
5	0	1	250	9700	9850	300	1	0	5

The price levels for the existing Bid orders must all be incremented even though there will not be Aggregate Order Book Update messages sent for these increments. The implied bid order with a price of 9730 at (new) price level 2 remains in the orderbook, even though it is no longer at the best price.

7.3 Example 3 – Implicit Deletions

If a new book entry causes the bottom entry of a book to be shifted out of the book (i.e. more than 5 price levels away from the best price), the client must delete the excess entry. If the book shrinks again, LMEsource resends the entries that have temporarily fallen out.

For example, if a bid order with price 9750 and quantity 250 is added to the book above, and the bid quantity at price 9710 is reduced from 150 to 110, it will cause the following message to be sent:

Offset	Field Name	Value
0	MsgSize	109
2	MsgType	353
4	MDSource	EL
6	TimeOfEvent	123456789
14	SecurityID	1234
22	NoEntries	2
23	AggregateQuantity	250
31	Price	9750
39	NumberOfExplicitOrders	1
43	TotalQtyOfExplicitOrders	250
51	NumberOfImpliedOrders	0
55	TotalQtyOfImpliedOrders	0
63	Side	1 (Bid)
64	PriceLevel	1
65	UpdateAction	0
66	AggregateQuantity	110

74	Price	9710
82	NumberOfExplicitOrders	1
86	TotalQtyOfExplicitOrders	110
94	NumberOfImpliedOrders	0
98	TotalQtyOfImpliedOrders	0
106	Side	1 (Bid)
107	PriceLevel	5
108	UpdateAction	1

After processing this message, the client's book should look as follows:

Bid Side					Ask Side				
Price Level	No. Of Implied Orders	No. of Explicit Orders	Agg. Quantity	Price	Price	Agg. Quantity	No. of Explicit Orders	No. of Implied Orders	Price Level
1	0	1	250	9750	9760	500	3	2	1
2	0	1	50	9740	9770	200	2	0	2
3	1	3	700	9730	9780	100	1	0	3
4	0	1	350	9720	9790	150	1	0	4
5	0	1	110	9710	9850	300	1	0	5

Price 9750 and quantity 250 is added according to the message. Price 9700 and quantity 250 must be deleted by the client.

Price 9710 quantity must be reduced to 110. The AOB update messages uses the price level 5 to reflect the new price level of the price 9710 after the addition of the new price level a with a price of 9750.

7.4 Example 4 – Explicit Additions

If orders are removed so that there are now less than 5 levels visible then the server will also automatically send the additional level(s) that are now revealed.

For example, if the bid order with price 9750 and quantity 250 is now removed from the book above each price level will shift up one position, and this reveals a 5th level which needs to be disseminated. This will cause the following message to be sent:

Offset	Field Name	Value
0	MsgSize	109
2	MsgType	353
4	MDSource	EL
6	TimeOfEvent	123456789
14	SecurityID	1234
22	NoEntries	2
23	AggregateQuantity	250
31	Price	9750
39	NumberOfExplicitOrders	1
43	TotalQtyOfExplicitOrders	250
51	NumberOfImpliedOrders	0
55	TotalQtyOfImpliedOrders	0
63	Side	1 (Bid)
64	PriceLevel	1

65	UpdateAction	2
66	AggregateQuantity	250
74	Price	9700
82	NumberOfExplicitOrders	1
86	TotalQtyOfExplicitOrders	250
94	NumberOfImpliedOrders	0
98	TotalQtyOfImpliedOrders	0
106	Side	1 (Bid)
107	PriceLevel	5
108	UpdateAction	0

The resulting order book should now be:

Bid Side					Ask Side				
Price Level	No. Of Implied Orders	No. of Explicit Orders	Agg. Quantity	Price	Price	Agg. Quantity	No. of Explicit Orders	No. of Implied Orders	Price Level
1	0	1	50	9740	9760	500	3	2	1
2	1	3	700	9730	9770	200	2	0	2
3	0	1	350	9720	9780	100	1	0	3
4	0	1	110	9710	9790	150	1	0	4
5	0	1	250	9700	9850	300	1	0	5

7.5 Example 5 – Additional Order at an Existing Price Level

If a new order is entered into the book at an existing price level, the number of orders and the quantity at that level is incremented.

For example, if an ask order with price 9780 and quantity 200 is added to the book above the following message will be sent:

Offset	Field Name	Value
0	MsgSize	66
2	MsgType	353
4	MDSource	EL
6	TimeOfEvent	123456789
14	SecurityID	1234
22	NoEntries	21
23	AggregateQuantity	300
31	Price	9780
39	NumberOfExplicitOrders	2
43	TotalQtyOfExplicitOrders	300
51	NumberOfImpliedOrders	0
55	TotalQtyOfImpliedOrders	0
63	Side	2 (Ask)
64	PriceLevel	3
65	UpdateAction	1

The resulting orderbook should now be as follows:

Bid Side					Ask Side				
Price Level	No. Of Implied Orders	No. of Explicit Orders	Agg. Quantity	Price	Price	Agg. Quantity	No. of Explicit Orders	No. of Implied Orders	Price Level
1	0	1	50	9740	9760	500	3	2	1
2	1	3	700	9730	9770	200	2	0	2
3	0	1	350	9720	9780	300	2	0	3
4	0	1	110	9710	9790	150	1	0	4
5	0	1	250	9700	9850	300	1	0	5

7.6 Example 6 – An Existing Order has its Quantity Amended

Amendments to order quantity are published as price level update, irrespective of whether the quantity is increased or decreased, or how many orders there are at the price level.

For example, if one of the ask orders with price 9770 is revised such that the aggregate quantity at the price level is 150, the following message will be sent:

Offset	Field Name	Value
0	MsgSize	109
2	MsgType	353
4	MDSource	EL
6	TimeOfEvent	123456789
14	SecurityID	1234
22	NoEntries	2
23	AggregateQuantity	150

31	Price	9770
39	NumberOfExplicitOrders	2
43	TotalQtyOfExplicitOrders	0
51	NumberOfImpliedOrders	0
55	TotalQtyOfImpliedOrders	0
63	Side	2 (Ask)
64	PriceLevel	2
65	UpdateAction	1

The resulting order book should now be:

Bid Side					Ask Side				
Price Level	No. Of Implied Orders	No. of Explicit Orders	Agg. Quantity	Price	Price	Agg. Quantity	No. of Explicit Orders	No. of Implied Orders	Price Level
1	0	1	50	9740	9760	500	3	2	1
2	1	3	700	9730	9770	150	2	0	2
3	0	1	350	9720	9780	300	2	0	3
4	0	1	110	9710	9790	150	1	0	4
5	0	1	250	9700	9850	300	1	0	5

8 Level 3 Full Order Book Management

The information supplied in this section and its sub-sections applies to the data feed(s) marked with [◆]

Section	LME Level 1 Top of Book	LME Level 2 15 Levels Price Book	LME Level 3 Full Book
6			◆

Book Identification

A book is uniquely identified by SecurityID. SecurityID is an unsigned integer representation of 8 bytes from Outright Definition (301) messages and Strategy Definition (302) messages.

Book Updates

The information needed to build an order book view from the message flow is contained within the following messages:

- Order Add (357)
- Order Amend (358)
- Order Cancel (359)
- Order Executed (350)

Orders are ranked by orderbook position based upon price and time priority, with 1 denoting the highest ranked order. When an order is cancelled or fully filled, all existing orders below it should shift their position up one step to fill the “void”.

The Order Add (357) message signals that a new order is placed in the order book. If there are already orders in that position, recipients should check the price of existing orders (with the same orderbook position). If the price of the existing order(s) is worse than the price of new order, all existing orders below the new order should shift down one position. If the price of the existing order(s) is better than the price of new order, it signifies an error (something has gone wrong with the order book).

The Order Amend (358) message signals that the order has been modified. The current rank may or may not be lost in the process. The OrderBookPosition field will show the new rank within the book. The order must be removed from its previous position and inserted at the new OrderBookPosition if the new position is not the same as previous one. Removal of an order causes existing orders below it to shift their position up one level. An order inserted at an existing position should be handled as per an Order Add (357).

The Order Cancel (359) message tells the recipient to remove the order referenced. If the order cancelled or fully filled is the last order at that position, the deleted order causes all existing orders below it to shift their position up one step to fill the "void".

The Order Execute (350) messages tells the recipient to deduct the traded order from the order book. If the order is fully filled, the associated order (OrderID) should be removed from the order lists. If it was the only remaining order at that order position, all existing orders below it should shift their position up one step to fill the "void". If the order is partially executed, the executed quantity should be deducted from the associated order in the order lists.

In certain failure scenarios LMEsource may send an 'Orderbook Clear' message at which point clients should clear all orders in the book for the specified instrument.

Following an 'Orderbook Clear' message any existing orders for the instrument will be resent as Order Add (357) messages to rebuild the current image.

The table below shows the starting position of the orderbook for the following examples:

Bid Side				Ask Side			
Orderbook Position	OrderID	Quantity	Price	Price	Quantity	OrderID	Orderbook Position
1	0003	500	9730	9760	500	1004	1
2	0004	200	9730	9770	100	1001	2
3	0002	350	9720	9770	200	1002	3
4	0001	150	9710	9780	100	1003	4
5	0005	250	9700	9790	150	1005	5

8.1 Example 1 – Addition of a New Order

A new Bid order with price 9720 is received. The following message is published by LMEsource.

Offset	Field Name	Value
0	MsgSize	72
2	MsgType	357

4	MDSource	EL
6	TimeOfEvent	123456789
14	T1	123456789
22	T2	123456789
30	T3	123456789
38	SecurityID	1234
46	OrderID	0006
54	Side	1 (Bid)
55	Quantity	75
59	Price	9720
67	OrderBookPosition	4
71	Filler	-

The new Bid order 0006 is inserted into the orderbook at position 4. The existing order 0002 at the same price of 9720 remains in its higher position of 3 due to its time priority. Orders 0001 and 0005 must have their orderbook positions adjusted down one step by the client.

Bid Side				Ask Side			
Orderbook Position	OrderID	Quantity	Price	Price	Quantity	OrderID	Orderbook Position
1	0003	500	9730	9760	500	1004	1
2	0004	200	9730	9770	100	1001	2
3	0002	350	9720	9770	200	1002	3
4	0006	75	9720	9780	100	1003	4

5	0001	150	9710	9790	150	1005	5
6	0005	250	9700	-	-	-	-

8.2 Example 2 – A Resting Order has its Quantity Decreased

The Bid order with OrderID 0002 and price of 9720 has its quantity reduced from 350 to 300. The following message is published by LMEsource.

Offset	Field Name	Value
0	MsgSize	72
2	MsgType	358
4	MDSource	EL
6	TimeOfEvent	123456789
14	T1	123456789
22	T2	123456789
30	T3	123456789
38	SecurityID	1234
46	OrderID	0002
54	Side	1 (Bid)
55	Quantity	300
59	Price	9720
67	OrderBookPosition	3
71	Filler	-

The quantity of Bid order 0002 is reduced from 350 to 300. It remains at orderbook position 3.

Bid Side				Ask Side			
Orderbook Position	OrderID	Quantity	Price	Price	Quantity	OrderID	Orderbook Position
1	0003	500	9730	9760	500	1004	1
2	0004	200	9730	9770	100	1001	2
3	0002	300	9720	9770	200	1002	3
4	0006	75	9720	9780	100	1003	4
5	0001	150	9710	9790	150	1005	5
6	0005	250	9700	-	-	-	-

8.3 Example 3 – Cancellation of an Order

The Ask order with OrderID 1001 and price of 9770 is pulled by the originating trader. The following message is published by LMEsource.

Offset	Field Name	Value
0	MsgSize	72
2	MsgType	359
4	MDSource	EL
6	TimeOfEvent	123456789
14	T1	123456789
22	T2	123456789
30	T3	123456789

38	SecurityID	1234
46	OrderID	1001
54	Side	2 (Ask)
55	Filler	-

The client must remove OrderID 1001 from their orderbook, and adjust the orderbook position for the remaining Ask Orders up by one step.

Bid Side				Ask Side			
Orderbook Position	OrderID	Quantity	Price	Price	Quantity	OrderID	Orderbook Position
1	0003	500	9730	9760	500	1004	1
2	0004	200	9730	9770	200	1002	2
3	0002	300	9720	9780	100	1003	3
4	0006	75	9720	9790	150	1005	4
5	0001	150	9710	-	-	-	-
6	0005	250	9700	-	-	-	-

8.4 Example 4 – An Aggressing Order is Entered that Executes Against a Resting Order

An aggressing Bid Order with an OrderID of 0007, a price of 9760 and quantity 100 is entered by a trader. This order fully executes against the resting ask order with the OrderID of 1004.

The aggressing order is not published. An Order Executed message for the resting Ask Order with OrderID 1004 is published. No Order Amend for OrderID 1004 is published.

Offset	Field Name	Value
0	MsgSize	53
2	MsgType	350
4	MDSource	EL
6	TimeOfEvent	123456789
14	SecurityID	1234
22	Price	9760
30	Quantity	100
34	OrderID	1004
42	MatchID	9988
50	TradeCancelFlag	0
51	TradeSide	2
52	NumOfLegs	0

The client must update the quantity for OrderID 1004 in their orderbook. The orderbook should now look like:

Bid Side				Ask Side			
Orderbook Position	OrderID	Quantity	Price	Price	Quantity	OrderID	Orderbook Position
1	0003	500	9730	9760	400	1004	1
2	0004	200	9730	9770	200	1002	2
3	0002	300	9720	9780	100	1003	3
4	0006	75	9720	9790	150	1005	4

5	0001	150	9710	-	-	-	-
6	0005	250	9700	-	-	-	-

8.5 Example 5 – An Aggressing Order is Entered that Executes against Multiple Resting Orders at Different Price Levels

An aggressing Ask Order with an OrderID of 1006, a price of 9720 and quantity 800 is entered by a trader. This order fully executes against the resting bid orders with OrderIDs of 0003, 0004 and 0002. Orders 0003 and 0004 are fully filled, and Order 0002 partially filled.

The aggressing order is not published. Three Order Executed message for the resting Bid Orders are published. No Order Amend for OrderID 0002 is published.

The following messages are published:

Offset	Field Name	Value
0	MsgSize	53
2	MsgType	350
4	MDSource	EL
6	TimeOfEvent	123456789
14	SecurityID	1234
22	Price	9730
30	Quantity	500
34	OrderID	0003
42	MatchID	7766
50	TradeCancelFlag	0
51	TradeSide	1
52	NumOfLegs	0

Offset	Field Name	Value
0	MsgSize	53
2	MsgType	350
4	MDSource	EL
6	TimeOfEvent	123456789
14	SecurityID	1234
22	Price	9730
30	Quantity	200
34	OrderID	0004
42	MatchID	7767
50	TradeCancelFlag	0
51	TradeSide	1
52	NumOfLegs	0

Offset	Field Name	Value
0	MsgSize	53
2	MsgType	350
4	MDSource	EL
6	TimeOfEvent	123456789
14	SecurityID	1234
22	Price	9720

30	Quantity	100
34	OrderID	0002
42	MatchID	7768
50	TradeCancelFlag	0
51	TradeSide	1
52	NumOfLegs	0

The client must remove Orders 0003 and 0004, and update the quantity for OrderID 0002 in their orderbook. The remaining Bid orders have their Orderbook position updated. The orderbook should now look like:

Bid Side				Ask Side			
Orderbook Position	OrderID	Quantity	Price	Price	Quantity	OrderID	Orderbook Position
1	0002	200	9720	9760	400	1004	1
2	0006	75	9720	9770	200	1002	2
3	0001	150	9710	9780	100	1003	3
4	0005	250	9700	9790	150	1005	4
-	-	-	-	-	-	-	-

9 Appendix A – Reference Data Values

9.1 Prompt Date Label

The prompt date label can be one of the following, depending upon the contract and type of prompt. Monthly 2BD (second business day) prompts do not have any labels.

Prompt Type	Possible Values
Daily	Sequential contiguous labels: D1, D2, D3...D90
Weekly	Sequential contiguous labels: W1, W2, W3...W26
Monthly and quarterly prompts	A three character month code with a two digit year code. E.g. JUN20, JUL20, AUG20...AUG30
Semi-annual prompts	Sequential contiguous labels: S1, S2, S3…S20. S is always June or December, with S1 being the nearest of these.
Annual prompts	Sequential contiguous labels: A1, A2, A3…A10. A is always December, with D1 being the nearest December prompt.
Specific Prompts	DEC1 - the December monthly in the next calendar year according to the current trading day DEC2 - the December monthly in the second next calendar year according to the current trading day DEC3 - the December monthly in the third next calendar year according to the current trading day 15M - the 3rd Wednesday (monthly) prompt that falls in the month 15 months from the current month 27M - the 3rd Wednesday (monthly) prompt that falls in the month 27 months from the current month 63M - the 3rd Wednesday (monthly) prompt that falls in the month 63 months from the current month 127M - the 3rd Wednesday (monthly) prompt that falls in the month 123 months from the current month
Rolling Prompts	TOM, CASH, 3M

9.1.1 Prompt Date Label Precedence

It is possible for a tradable instrument to have more than one prompt date label, however the Outright Definition (301) message will only include one prompt date label. The precedence for which prompt date label is published is:

1. A rolling prompt date label has the highest precedence
2. A specific label takes precedence over duration labels (e.g. monthly, weekly, daily).
3. Annuals duration labels take precedence over semi-annual / quarterly / monthly / weekly / daily duration labels.
4. Semi-annual duration labels take precedence over quarterly / monthly / weekly / daily duration labels.
5. Quarterly labels take precedence over monthly / weekly / daily duration labels.
6. Monthly duration labels take precedence over weekly / daily duration labels.
7. Weekly duration labels take precedence over daily duration labels.

9.2 Strategy Type

Strategy Type	Value
Outright	0
Carry	1
Custom (Futures)	*2*
Three Month Average	*3*
Six Month Average	*4*
Twelve Month Average	*5*
Carry Average	*6*
Call Spread	*7*
Put Spread	*8*
Custom (Delta Hedge)	*9*
Custom (Options)	*10*

9.3 Retransmission Status

Value	Description
0	Request Accepted
1	Unknown/Unauthorized channel ID
2	Messages not available
100	Exceeds maximum sequence range
101	Exceeds maximum requests in a day

10 Appendix B – Maturity Dates

10.1 Futures

An LME future is defined by a symbol and a maturity date, the *prompt date*. There are three different types of prompt dates for futures:

- Rolling prompt dates – these prompt dates are relative to the current trading day. When trades in these contracts are sent to clearing, the date is "frozen" into a calendar date. The principle rolling prompts are:
  1) 3M (Three months) – this prompt date represents the settlement business day three months from today.
  2) CASH – this prompt date represents the settlement business day after tomorrow.
  3) TOM (Tomorrow) – this prompt date represents tomorrow.
  4) Near Month 1 – this is the first monthly prompt date after the CASH date
  5) Near Month 2 – this is the second monthly prompt date after the CASH date
  6) Near Month 3 - this is the third monthly prompt date after the CASH date. This date must fall before the 3M date, and will not always be present.

Rolling prompt codes are present in the PromptType and PromptDateLabel fields in the Outright Definition (301) message. Not all LME Metal Futures have rolling prompt dates.

The 3M date is not applicable to LME Precious Futures.

- Single prompt dates - these prompt dates are calendar dates, written in the format YYYYMMDD, where YYYY is the year, MM is the month (01-12) and DD is the day (01-31). The LME uses the concept of "Monthly", "Weekly", "Daily" contracts, but all these contract types represent a single prompt date, and there is no difference between them in LMEsource. (For "Monthly" contracts, the prompt date is either the 3rd Wednesday in the month or the last Business Day in the month. For "Weekly" contracts, the prompt date is the Wednesday in each week.)

Single prompt dates are present in the MaturityDate field in the Outright Definition (301) message.

LMEprecious Future "Quarterly" prompt dates are considered "Monthly" contracts in LMEsource.

To know what prompt dates that are available, it is necessary to have access to an LME trading calendar. A quick and incomplete summary of the trading calendar is:

For a Base Metal Future, 3M and CASH are always present, and TOM is usually present. TOM is not present if that date is a US national holiday. There is one prompt date per settlement business day between the TOM and the 3M contract, thereafter a prompt date every Wednesday for 3 months, and then prompt dates on the 3rd Wednesday of each month for a number of months, depending on the underlying commodity. Average contracts are present that span the months after the 3M contract.

For LMEprecious Futures, CASH is always present, and TOM is usually present. TOM is not present if that date is a US national holiday. There is an additional daily prompt on the settlement business day after the CASH date. Thereafter there is a monthly prompt date on the 3rd Wednesday of each month for 24/25 months, followed by 12 quarterly prompts on the 3rd Wednesday of the third month (Mar, Jun, Sep, Dec) in each quarter.

For the index, there is one prompt date every 2nd Wednesday in the month for 12 months. There are no rolling prompts and no average prompts for the index.

For LMEminis, there exist one prompt date every 3rd Wednesday in the month for 12 months. There are no rolling prompts for LMEminis.

For LME Monthly Cash Settled Futures (CSFs), there is one prompt date on the last working day of each month. The CSF prompt date does not need to fall on a settlement business day.

For LME Monthly Average Futures (MAFs), there is one prompt date on the last working day of each month. The MAF prompt date does not need to fall on a settlement business day.

10.2 Base Metal Options

A base metal option is defined by a symbol, an option type, a strike price and an expiration date.

Expiration dates are calendar dates, written in the format YYYYMMDD, where YYYY is the year, mm is the month (01-12) and DD is the day (01-31). For metals, there is one expiration date per month: the first Wednesday in the month. For the index, the expiration date is the second Wednesday in the month. For both metals and the index, the expiration date is rolled forward one day if the expiration date is a non-business day. The expiration date for Metal Options does not need to fall on a settlement business day.

10.3 TAPOs

A TAPO is defined by a symbol (only metals), an option type, a strike price and an expiration date. For TAPOS, the only allowed expiration date is the single expiration date in format YYYYMMDD. There is one expiration date per month on the last trading day of the month.

11 Appendix C – Channel Matrix

The table below shows the mapping between multicast channel IDs and the LMEsource messages. please refer to “LME Systems Connectivity Guide” document for the Channel IPs and UDP port number for both test and/or production environment.

Category	LMEsource Message	Base Contracts		Growth Contracts		Precious contracts	
		Realtime Channel	Refresh Channel	Realtime Channel	Refresh Channel	Realtime Channel	Refresh Channel
Futures	Outright Definition (301) Strategy Definition (301)	100	600	250	750	150	650
	EOD Trade Statistics (351)						
Options	*Outright Definition (301)* *Strategy Definition (301)*	*101*	*601*	-	-	-	-
	EOD Trade Statistics (351)						
Futures Level 1	Top of Book (355) Orderbook Clear (335)	104	604*	254	754	154	654
	Match Trade (360)						
	Quote Request (356)						
	IOP (354)						
	Contract State (311) Instrument State (312)						
Futures Level 2	Aggregate Order Book (353) Orderbook Clear (335)	105	605*	255	755	155	655
	Match Trade (360)						
	Quote Request (356)						
	IOP (354)						
	Contract State (311) Instrument State (312)						
	Order Add (357) Order Amend (358)	106	606*	256	756	156	656

Futures Level 3	Order Cancel (359) Orderbook Clear (335) Order Executed (350) *Quote Request (356)* IOP (354) Contract State (311) Instrument State (312)						
Options Level 1	*Top of Book (355)* *Orderbook Clear (335)* *Match Trade (360)* *Quote Request (356)* *IOP (354)* *Contract State (311)* *Instrument State (312)*	*107*	*607**	-	-	-	-
Options Level 2	*Aggregate Order Book (353)* *Orderbook Clear (335)* *Match Trade (360)* *Quote Request (356)* *IOP (354)* *Contract State (311)* *Instrument State (312)*	*108*	*608**	-	-	-	-
Options Level 3	*Order Add (357)* *Order Amend (358)* *Order Cancel (359)* *Orderbook Clear (335)* *Order Executed (350)* *Quote Request (356)* *IOP (354)*	*109*	*609**	-	-	-	-

	Contract State (311) *Instrument State (312)*						
Futures	Intraday Trade Statistics (352)	110**	610**	260**	760**	160**	660**
Options	*Intraday Trade Statistics (352)*	*111***	*611***	-	-	-	-
Futures Trade History	Match Trade (350)	-	614	-	764	-	664
Options Trade History	*Match Trade (350)*	-	*615*	-	-	-	-
All	DR Signal (105)	81	-	81	-	81	-

* Match Trade and Order Execute messages are not published in these RFS channels. A replay of all Match Trade (360) messages from SOD is published in the Trade history channels.

** Intraday Trade Statistics are only available to Level 2 and Level 3 participants.