LMEsmart FIX API Member Guide

Version 2.30

Please respond to:
posttradeoperations@lme.com
Post Trade Operations, +44 20 7113 8201

Change History

Revision	Sections	Notes
0.4 23rd Mar 16	All	Updated following detailed LME review
0.5 5th Apr 16	All	Additional comments from LME
0.6 5th Apr 16	All	Extended trade categories Added commodity derivatives indicator
0.7 7th Apr 16	All	Updated to new template
0.8 12th Apr 16	Appendix D	Updated examples
0.9 14th Apr 16	5.5 12.8 Appendix D	“Closing” system state added Clearing status “pending” re-instated Added extra examples
2.0 15th Apr 16	-	Incremented to 2.0 for release
2.1 26th Apr 16	4.3 6.4.1 6.4.2 7.1 7.2 Appendix A Appendix D	Removed references to averages Added SecurityType(167) into instrument block Updated size of Symbol (55) Added Market (1301) Added notes on user accounts Added Examples for parties block
2.2 10th May 16	7, 8	Added client strategy identifier Modified Trade Time to be milliseconds precision
2.3 19th Jul 16	6.4.2 7.1 7.2 8.3 12.3 Appendix A Appendix D	Corrected message types for new trades (should be E) Clarified values for Side (54) Added tags missing from dictionary for non-trade messages (371, 372, 373, 379, 380) AccountType(581) for Gross OSA is 7 Corrected PartyIDSource (447) description Corrected TradeTime (5179) format LegLastPx (637) not required for Options

Revision	Sections	Notes
		Modified cancellation fields required – swap ClOrdId (11) and OrigClOrdId (41) Modify *Market* (1301) descriptions and examples Added *Symbol* (55) and *SecurityType* (167) to download Securities request CommodityDerivativeIndicator (20023) is only a required field for MiFID II
2.4 2nd Sep 16	6.4.2 7.1 7.2 7.3 7.4 7.5 8.1 8.3 12.3 Appendix A Appendix B Appendix D	Added PartyRole (452) = 60 to support IBs. Added AccountType (581) = 9 (Unallocated House) Added AccountType (581) = 90 (Unallocated Client) Removed section 12.3 as auto-generation of client halves is back in scope Pending state changed to Paired Extra error codes added and redundant ones removed Added details on providing Give-Up with unknown counterparty Added details of auto-generation of counterpart for exchange trades Premium and underlying futures price no longer mandatory TradeTime can be provided in TERs along with session code if populated by LMEsmart Prices with trailing zeroes have them truncated Removed reference to MaturityAveragePrompt (10001) MiFID fields not mandatory in parties block Account(1) is conditionally required Trade Mass Status Request can have Symbol(55) and ContractType(167) Trade Mass Status Request can return zero records PartyIdSource(447) mandatory Trade Mass Status Request includes trades in all states Clarified conditional fields in Execution Reports sent back to members

Revision	Sections	Notes
		Removed MRCPS CFI Code Examples in this document are for simple exchange trades (house to house)
2.5 2nd Dec 16	5.3.5 6.3.1 6.4.1 6.4.2 6.6 7.4.1 8 8.3 Appendix A Appendix B	Updated OrigClOrdId to 30 chars Added error codes 1164, 1180, 1181, 1276 Added error codes for clearing rejections Clarified comp ids Clarified that format of ClOrdId is just a suggested format, but not enforced by system MMMYY format accepted for MaturityDate(541) Clarified when business reject (35=j) and session reject (35=3) messages are used Market (1301) in in trade mass status request responses ComplexTradeComponentId (20020) is 35 chars Premium and Underlying price are required for options and TAPO Updated what can be returned in 102 and 103 (i.e. just “99” – other error) Note on strategies being returned as individual reports in response to trade mass status request Party 3 and 24 are conditional
2.6 25th Jan 17	5.3.1 6.4.1 7.3 7.5.2 8.3 Appendix A Appendix B	Updated what can be returned in 102 and 103 (i.e. just “99” – other error) 5440 (Clearing Status) can have value of W for cancelled “UNKNOWN” CFICode value Error code 1045 added for invalid unique product id Added details of validation on logon Client crosses can be cancelled in all states
2.7 10th Feb 17	6.4.1, 7.4.1, 7.5.1, 9.3, Appendix A	Security Definition only contains partial instrument block Indent on ComplexTradeComponentId (20020) Trading Capacity (10051) conditional in responses

<table>
<tr><th>Revision</th><th>Sections</th><th>Notes</th></tr>
<tr><td></td><td></td><td>Clearing Status (5440) = 0 is not seen on the interface

ClOrdId (11) can be 35 characters for responses from LMEsmart (due to _AUTO trades)

Change CFICode description

Examples of market are LME and LPM

Clarified that both 637 and 5474 are received after price substitution</td></tr>
<tr><td>2.8
10th Mar 17</td><td>6.4.2, 7.5, 9.4,
Appendix A
Appendix C</td><td>New template

5477 (Public Ref) is only first 5 chars in alleged trades

PartyRole 304 can be received in Execution Reports from LMEsmart

Use of conditional FIX Tag 11 (ClOrdId) as first in repeating block is not FIX conformant

Member Definition responses contain “na” for empty fields

AlgoId changed to AlgoID</td></tr>
<tr><td>2.9
24th Apr 17</td><td>6.4.2,
7.4.1
8.3
Appendix A</td><td>New Matching rules as of 20th May 2017

Description of Party Role 24 should be Client Code

Account(1) and AccountType(581) are provided on cancellation execution reports

Clarified that TradeTime (5179) is in London Time.

Max PartyId(448) size is 128 characters</td></tr>
<tr><td>2.10
4th Aug 17</td><td>7.4.1
7.5.1
Appendix A
Appendix B</td><td>New fix tags for MiFID II compliance that can be provided in new trade messages and received in execution reports:

<table>
<tr><th>Fix Tag</th><th>Name</th></tr>
<tr><td>20030</td><td>Traded Price</td></tr>
<tr><td>20031</td><td>Traded Premium</td></tr>
<tr><td>10052</td><td>Country Of Branch Of Client</td></tr>
<tr><td>820</td><td>Trade Link Id</td></tr>
<tr><td>20032</td><td>Cancellation Flag</td></tr>
<tr><td>20033</td><td>Cancel Link Id</td></tr>
</table>

<table>
<tr><th>PartySrc (447)</th><th>PartyRole (452)</th><th>Name</th></tr>
<tr><td>P</td><td>3</td><td>Client Short Code</td></tr>
<tr><td>P</td><td>122</td><td>Decision Maker Short Code</td></tr>
</table>
</td></tr>
</table>

Revision	Sections	Notes

Other new fix tags that can be received in execution reports (but do not need to be provided in new trade messages):

Fix Tag	Name
10054	Strategy Link Id
1903	Regulatory Trade Id (TVTIC)

PartySrc (447)	PartyRole (452)	Name
D	6	Correspondent Broker

The following Fix tags have been amended:

Fix Tag	Name	Change
5179	TradeTime	Now a UTC Timestamp rather than localtime

PartyRole (452)	Name	Change
300	Investment Decision User Id	Now an integer (representing the short code)
301	Execution Decision User Id	Now an integer (representing the short code)

The following trade categories (sent in Tag 5681 – ExchangeTradeType) have been added:

Name	Code
OTC Take-Off	10
Exception - Reportable	8
Exception – Non-Reportable	TBC

The following trade categories (sent in Tag 5681 – ExchangeTradeType) have been removed:

Name	Code
Reversal	6
Correction	7

To replace the Reversal and Correction trade category a new pair of FIX tags are provided (as listed above):

- 20032 – Cancellation Flag
- 20033 – Cancel Link Id

Instead of sending a Reversal or Correction trade category, a member should send a new trade with the same trade category as the original trade. This new trade should have

<table>
<tr><th>Revision</th><th>Sections</th><th>Notes</th></tr>
<tr><td></td><td></td><td>the Cancellation Flag set and the Cancel Link Id set to the Matching Ref Num (5935) of the original trade.

There are the following additional error codes:

<table>
<tr><th>Error Code</th><th>Error Text</th></tr>
<tr><td>1046</td><td>Traded Price Invalid</td></tr>
<tr><td>1047</td><td>Traded Premium Invalid</td></tr>
<tr><td>1048</td><td>Trade Link Id invalid</td></tr>
<tr><td>1049</td><td>Cancellation Flag invalid</td></tr>
<tr><td>1050</td><td>Cancel Link Id invalid</td></tr>
<tr><td>1051</td><td>Client Short Code invalid</td></tr>
<tr><td>1052</td><td>Decision Maker Short Code invalid</td></tr>
<tr><td>1301</td><td>Cancellation Flag provided without Cancel Link Id</td></tr>
</table>

The AlgoID party role (role 304) has been removed and will no longer be returned on execution reports for trades executed on LMEselect.</td></tr>
<tr><td>2.11
14th August 2017</td><td>6.4.2
7.4.1
8.3
Appendix A</td><td>Correspondent Broker will be provided in a party with a PartyRole(452)=26 (and not 6 as previously stated)

Added explanation that Correspondent Broker only comes from trades from the LMEselect venue

Clarified which roles are mandatory in the parties block, these being:

1 = Executing Firm
11 = Order Origination Trader
17 = Contra Firm
36 = Entering Trader
300 = Investment Decision User ID
301 = Execution Decision User ID
302 = Investment Decision Country Code
303 = Execution Decision Country Code

The numeric code for the “Exception – Non-Reportable” trade category is 16 (as provided in FIX tag 5681 – ExchangeTradeType)

Corrected description of ExchangeTradeType(5681) as provided in section 8.3 on Trade Execution Report responses from Trade History requests (the list of codes allowed for this tag should now match that given in the New Trade List section)</td></tr>
<tr><td>2.12
18th August 2017</td><td>Appendix D</td><td>Updated extra message examples to cover use of new MiFID related fields.</td></tr>
</table>

<table>
<tr><th>Revision</th><th>Sections</th><th>Notes</th></tr>
<tr><td></td><td>Appendix B</td><td>There are the following additional error codes:
<table><tr><th>Error Code</th><th>Error Text</th></tr><tr><td>1053</td><td>Country of branch of client is invalid</td></tr></table></td></tr>
<tr><td>2.13
13th October 2017</td><td>6.4.2
7.4.1
Appendix A</td><td>Corrected description for CancellationFlag (20032) – this referered to the CancelLinkId (20033) being the Matching Slip Id, but this should be the Matching Reference Number

Corrected description of tag CancellationFlag (20032) – the C code allowed in this tag means “Correction” and not “Cancellation”

Corrected description of TradeTime(5179) format. This should be a UTCTime type rather than UTCTimestamp, e.g. “08:35:22.123”

Corrected description for TVTIC (1903) – this field is not mandatory.

The following parties are now optional:

• 300 = Investment Decision User ID
• 302 = Investment Decision Country Code
• 303 = Execution Decision Country Code</td></tr>
<tr><td>2.14
19th October 2017</td><td>6.4.2
7.4.1
Appendix A</td><td>CommodityDerivativeIndicator (20023) is a mandatory field (was previously listed as conditional)

ComplexTradeComponentID (20020) is an optional field (was previously listed as conditional)

Trading Capacity (10051) is a mandatory field (was previously listed as conditional)

Updated description of TimeBracket(943) – K3 and higher and D3 and higher codes are no longer valid

One new error code has been added

<table><tr><th>Error Code</th><th>Error Text</th></tr><tr><td>1302</td><td>A party role was provided that is not allowed on inbound messages</td></tr></table></td></tr>
<tr><td>2.15
1st November 2017</td><td></td><td>Two extra parties added to the parties block. These can only be included on reports received with an origin of LMEselect, they cannot be sent in to LMEsmart by the member:

<table><tr><th>PartySrc (447)</th><th>PartyRole (452)</th><th>Name</th></tr><tr><td>N</td><td>66</td><td>Market Maker Identifier</td></tr><tr><td>N</td><td>35</td><td>Liquidity Provider Identifier</td></tr></table></td></tr>
</table>

Revision	Sections	Notes
2.16 27th November 2017	7.5.1 8.3	Changed example value for SelectTradeNumber (5940) due to change in format of content of this tag: Previous: TA-ZS-20150114-000509 New: TAZS20150114n000509
2.17 7th December 2017	7.5.1 8.3	Changed example value for SelectTradeNumber (5940) due to change in format of content of this tag: Previous: TAZS20150114n000509 New: TAZS20150114X0000509
2.18 29th May 2018	7.5.1 7.5.2 8.1 8.3 Appendix A	New value for OrdStatus (39): 9 = Pending Acceptance New value for ExecType (150): 9 = Pending Acceptance These new values will be used when an RIB submits a trade to a clearing member for acceptance before being presented for matching.
2.19 12th June 2018	7.2 7.5.1 7.5.2 8.1 8.3 Appendix A	New value for OrdStatus (39): 7 = Pending Validation This value will be used when a trade half fails price band validation and requires LME manual validation before being accepted for matching. The corresponding value for ExecType(150) when this occurs will be: 9 = Pending Acceptance Added new tag SelectImpliedTradeNumber (5941). This tag will be populated when a carry on LMEselect has been matched with two outrights via an implied. For the owner of the carry order, this tag will contain the LMEselect Trade ID of the carry fill. (tag 5940 currently holds the LMEselect Trade ID of the underlying outright trade). Added two new CFI codes to tag (461): OPEFPS and OCEFPS included for European precious options. Added new tag AggressorIndicator (1057) This tag will only be populated for trades that have originated from LMEselect. New TAS price codes TA, TB, TC through to TZ have been added to the Abbreviated Price Code tag (5474).

Revision	Sections	Notes
2.20 11th July 2018	5.3.1 7.1 7.4.1 Appendix A Appendix B	The LMEsmart FIX API supports ASCII character range 32 – 122. Any characters used outside of this range may result in undesired formatting being returned Added/amended error codes: 1220 - A price code has been used for an invalid contract and venue combination 1221 - Price Code used with an invalid Prompt Date or a Prompt Date Code 1222 - A TAS Price code is present but not in all legs Clarified potential sequence numbers in logon responses.
2.21 31st October 2018	Change History, 7.5.1, 8.3	Tag 5442 (Matching Slip Id) is Conditional – it is not provided for certain OrdStatus values (7, 8 and 9) Update to the change history for version 2.19 to clarify that when an Execution Report has the OrdStatus (39) value of: 7 = Pending Validation Then the ExecType(150) will be: 9 = Pending Acceptance These values for OrdStatus (39) and ExecType (150) will be used when a trade half fails price band validation and requires LME manual validation before being accepted for matching.
2.22 3rd March 2020	7.4.1, 7.5.1, 8.3, Appendix A (Fix Tags), Appendix B (Error Codes)	New rejection error codes related to Pre-Trade Transparency (PTT) Auctions. In general, these will only be received if sending a directed bid/offer on an auction and it is rejected. The exception is Error Code 1072 which will be received if a trade is below PTT LIS Thresholds but outside the allowed auction times. New value for OrdStatus(39) of “P” to indicate a trade has triggered a PTT Auction. New value for ExecType(150) of “P” to indicate an execution report has been sent because the trade has triggered a PTT auction PublicReference(5477) should be set to the Auction Id when sending directed bids/offers on a PTT Auction In the case where a directed bid/offer is filled at the end of an auction and creates a new trade half, two tags are used for reference back to the original directed bid/offer: • PrivateReference (5476) - set to a combination of the auction id, original bid/offer client order id and fill number • ClOrdId(11) - set to the ClOrdId of the original directed bid/offer plus the text “_AUCN” (where N is the fill number)

Revision	Sections	Notes
2.23 16th June 2020	-	LME internal review only, not officially published
2.24 20th August 2020	-	LME internal review only, not officially published
2.25 11th November 2020	7.5.1 7.5.1.1 Appendix A	Details on use of "_SYS" as appended text in ClOrdId (11) for GUI or system initiated cancellations. There are three different cancellation cases which affect the value in ClOrdId(11) when the OrdStatus(39)=4 (Cancelled): • GUI cancellation, single half – text "_SYS" is appended to a system created cancellation request id • GUI cancellation, multiple halves – text "_SYS_N" is appended to a system created cancellation request id where N is a number • System cancellation (at end of PTT Auction) – text "_SYS" is appended to the original client order id of the half being cancelled. Note that initiating trade halves will be cancelled at the end of the PTT Auction if the original matching trade half was cancelled, regardless of whether the initiating trade half is now paired with a directed bid/offer or not (if it has been paired the original half is cancelled and a new one created).
2.26 29th January 2021	3, 7.5.1, 11, Appendix A	Fix trade acceptance for RIB trades New section added describing this functionality (section 11) New FIX Messages (only seen if member is enabled for FIX trade acceptance for RIB trades): RIB Trade Module Request (rb1) RIB Trade Module Response (rb2) New FIX Tags (only seen if member is enabled for FIX trade acceptance for RIB trades): Trade Module Id (20038) Acceptance Status (20039) RIB Acceptance Source (20040) New error code (only seen if member is enabled for FIX trade acceptance for RIB trades): 1287 (IB trade rejected by GCM) Updated Appendix description of Fix Tag 5447 (Request Id) as this can also be used in a RIB Trade Module Request as well as a Download Request Clarified that tag 10051 (Trading Capacity) is not provided on alleged trades

Revision	Sections	Notes
2.27 3rd March 2021	8.3	FIX Tags below are also included in Execution Reports sent in response to a Trade Mass Status request if they have a value (also only seen if member is enabled for FIX trade acceptance for RIB trades): Trade Module Id (20038) RIB Acceptance Source (20040)
2.28 8th July 2022	6.4.2	Clarified used of PartyRole=60 (Introducing Broker) in relation to RIB trades. This party should only be populated if it is the RIB sending in the execution report.
2.29 18th July 2023	7.5.1, 8.3, Appendix A	Updated all SelectOrderNumber (5939) examples in section 7.5.1, 8.3 & Appendix A with: 23138010000000020 Updated all SelectTradeNumber (5940) examples in section 7.5.1, 8.3 & Appendix A with: 6023138010000000009 Updated SelectImpliedTradeNumber (5941) description in section 7.5.1 & 8.3 to: “If TradeVenue is Select and the trade executed as the result of an implied carry. Contains a reference number indicating that the order was executed in Select as part of an implied scenario (for both outright and carries).”
2.30 13th September 2024	5.3.1 5.3.2 7.5.1, 8.3	Updated description of HeartBtInt in 5.3.1 to state recommended value should be 60 seconds or more. This is also stated in the section describing heartbeat intervals (5.3.2). Updated SelectImpliedTradeNumber (5941) description in section 7.5.1 & 8.3 to: “If TradeVenue is Select and the trade executed as the result of an implied carry. Contains a reference number indicating that the order was executed in Select as part of an implied scenario (for both outright and carries).”

Associated Documents

Ref	Title	Source	Version	Date
[1]	FIX Spec: http://www.fixtradingcommunity.org/pg/structure/tech-specs/fix-version/44	N/A	4.4	N/A

Contents

1 Introduction

This document provides a guide for members to using the FIX API of the LME's matching service (LMEsmart). LMEsmart provides a post-trade registration and matching service for the three LME trading venues – LMEselect, Ring and Inter-office (Telephone).

This guide covers how members can submit trades executed in the Ring or over the Telephone via a FIX API and receive updates on the status of these trades as they pass through the matching system to clearing. Details on how to request reports is also covered.

To aid members in moving from the existing matching system (LMEsmart) to the new system (LMEsmart), a summary of the key differences between the two are provided.

This document is divided into the following sections:

- Terminology Conventions
- System Usage Overview
- Connectivity
- FIX Session Layer
- Business Layer Overview
- Sending Trades
- Cancelling Trades
- Requesting Trade History
- Requesting Reference Data Reports
- News
- Accepting or Rejecting RIB Trades
- Recovery
- Changes from LMEsmart
- Support

There are also four appendices:

- FIX Tags
- Error Codes
- Deviations from FIX Protocol
- Message Examples

2 Guide to Reading This Document

The following sections should be read in order to assist with understanding the terminology and conventions in use throughout the remainder of the document.

2.1 Terminology

Throughout this guide the following terminology is in use:

Term	Description
LMEsmart	The matching process at LME available to LME members to access via a FIX session. LMEsmart acts as a server in FIX sessions i.e. it accepts client connections but does not initiate them.
Matching client/ LMEsmart client	An application used by Members for Trade capture which also establishes FIX sessions to LMEsmart in order to contribute trade halves to it.
FIX Connection	Is comprised of three parts: logon, message exchange and logout
FIX Session	Is comprised of one or more FIX connections that span the duration of a working day
Member	A firm or organization that is an LME member
User	In LMEsmart reference data, a User is associated with a member organization. A User is allowed to see all Trades entered by or alleged to a Member with which they are associated.
Client	A firm or organization that is a customer of a Member and for whom that Member acts.

2.2 Conventions

In order to fully describe the FIX messages supported by LMEsmart, the layouts have been placed within tables to assist with readability. The following guidelines apply:

- An arrow (→) indicates that a field is a sub-component and appears within the parent block.
- Within a repeating block, tags must be in the order specified.
- Where the mandatory requirement for a field to be present is indicated as **No (conditional)**, there are rules that dictate whether a value is required based on a values in another field.

3 System Usage Overview

The main purpose of the LMEsmart system is for members to be able to register trades for matching and clearing. The FIX API provides one mechanism for members to register such trades.
In order to use the FIX API the member system must first establish a FIX session. The FIX API uses standard FIX session establishment and maintenance techniques which are detailed in section 5.
Once a FIX session is established then trades can be sent. The following steps will occur:

- Send the trade – the member system should send a New Trade message for a trade half. See section 7.4.1 for details.
- Receive confirmation of trade – the LMEsmart system will respond with a Trade Execution Report message with an UNMATCHED state for each trade half. If there is an error with the trade (e.g. it fails validation) then a Trade Execution Report messages with a REJECTED state will be sent.
- Receive updates on trade state – as the trade half progresses through matching and clearing the LMEsmart system will send out Trade Execution Report messages with status updates. The progression of states for a successfully matched and cleared trade is:
    - UNMATCHED
    - MATCHED
    - SENT TO CLEARING
    - CLEARED

The LMEsmart system also provides a number of other features accessible via the FIX API:

- Requesting Trade History – a filter is applied such that the matching client may request a trade history of an individual trade or a specified subset of trades according to Symbol, Contract Type or trade date – see section 8.1
- Requesting Reference Data Reports – the matching client may request lists of valid instrument details or a list of current LME members and relevant contact details – see section 9.1
- News – From time to time LME will inform the matching client of significant market events via a simple unsolicited News message – see section 10
- Accepting/Rejecting RIB Trades – if opted into this feature, the matching client may accept or reject RIB entered trades – see section 11.

4 Connectivity

LMEsmart FIX server interfaces will support Transport Level Security (TLS). This will provide additional security on top of the measures already provided by LME network infrastructure.

4.1 COMPIDs

A CompID is a unique identifier for each machine connected over FIX. The CompID of each client must be registered with LME before FIX communications can begin. A single client may have multiple connections to the server (i.e. multiple FIX sessions, each with its own CompID).

The CompID of the server will be an LME allocated identifier. The messages sent to the server should contain the CompID assigned to the member firm in the field SenderCompID (49) and ”FGW” in the field TargetCompID (56).

Please contact the LME for instructions on how to register a member CompID.

4.2 Production IP Addresses and Ports

The IP address of each client must be registered with LME before FIX communications can begin. The IP addresses and ports of the production servers and test servers can be found in the LMEsmart Member Connectivity Guide.

LME will assign each registered client to a primary IP address and port and a secondary IP address and port. Clients should use the assigned primary IP address and port whenever it is available. The secondary IP address and port should only be used in the unlikely event the primary is unavailable.

4.3 User Accounts

Each process or user that requires the ability to connect to the Member FIX API will need a user account created that specifies the username, password and FAX key to be used in login. Please contact the LME for instructions on how to get a new user account for the FIX API.

User accounts can only be used by one process/user; if an attempt is made to login with an already in use set of credentials then this login will be rejected. If login is attempted unsuccessfully more than three times in a row then the user account will be locked and the LME should be contacted to unlock this.

4.4 Failover and Recovery

LMEsmart has been designed with fault tolerance and disaster recovery technology that ensures that matching may continue in the unlikely event of a process or site outage.

If the client is unexpectedly disconnected from the server, it should wait at least 10 seconds and attempt to re-connect to primary site. The client should only attempt to connect to the secondary IP address and port if so requested by LME.

5 FIX Session Layer

5.1 Session Management – Overview

There follows a brief overview of the means by which FIX sessions between LMEsmart and Members' client applications are established and maintained. For more details please see the individual per session level message descriptions that follow.

A FIX Connection is comprised of three parts: Logon, message exchange, and Logout.

A FIX Session is comprised of one or more FIX connections, meaning that a FIX Session spans multiple logins.

At market pre-open, LMEsmart will start 'listening' for matching client connections. Between market close and market pre-open times LMEsmart service will not be available for client connections. N.B. although matching clients may connect to LMEsmart once market pre-open time is reached, they may not submit new trades until full market open time is reached.

An LMEsmart client session is established when a Member's matching client establishes a connection and sends a Logon message to the LMEsmart server. The Logon request is validated and, if it is accepted by LMEsmart, then a Logon response is returned to the matching client.

When necessary, Heartbeat and Test messages may be exchanged between LMEsmart and matching client in order to confirm that the session is still active.

Standard FIX 4.4 rules apply to sequence number synchronization at a session level. If the FIX session is lost for some reason then, on re-establishment of the session, standard FIX4.4 message recovery conventions apply.

Either party to a FIX session may send a Reject message when they encounter an incoming message that does not conform to FIX session rules or does not contain a mandatory field.

At market close, LMEsmart sends a Logout message to all connected matching clients. It is expected that the matching client will respond with a corresponding Logout message and then, following a short pause, the matching client will be disconnected.

5.2 Standard Header and Trailer

All messages exchanged between the LMEsmart server and a matching client will contain a standard header block:

Tag	Name	Req'd	Description
8	*BeginString*	Y	Always 'FIX4.4' – must be 1st field in the message
9	*BodyLength*	Y	Message length in bytes - must be 2nd field in the message
35	*MsgType*	Y	Defines message type and must be 3rd field in the message
49	*SenderCompID*	Y	Identifies the source of the message
56	*TargetCompID*	Y	Identifies the intended recipient of the message
34	*MsgSeqNum*	Y	Incrementing sequence number used to identify gaps in transmission / reception
43	*PossDupFlag*	N	Indicates possible retransmission of message with this sequence number. See section 5.4
52	*SendingTime*	Y	Time of message transmission
97	*PossResend*	N	Indicates that the message may contain information that has been sent already with a different sequence number
122	*OrigSendingTime*	N	Required for messages sent as a result of a resend request
369	*LastMsgSeqNum*	N	The last *MsgSeqNum* (tag 34) value received by the FIX interface component

Messages from LMEsmart will identify "FGW" as the sending company in the *SenderCompID* (49) field. Matching clients should use the same value in – 'FGW' – in the TargetCompID (56) field in messages sent to LMEsmart. Each matching client will have their own identifier to include in the *SenderCompID* field of messages sent to LMEsmart. This matching client identity is then automatically used in the *TargetCompID* field of messages sent from LMEsmart to the matching client.

All messages exchanged between the LMEsmart server and a matching client will contain a standard trailer:

Tag	Name	Req'd	Description
10	*CheckSum*	Y	A three byte simple checksum see FIX specification for a detailed description. This field MUST always be the final field in the message.

5.3 Session Level Message Details

5.3.1 Logon

Having established a connection to the LMEsmart server, the first message that the matching client sends to LMEsmart must be a ***Logon*** message. If any other message is sent then the LMEsmart server immediately closes the connection.

A ***Logon*** message sent from a matching client to the LMEsmart server should be formatted as follows:

Tag	Name	Req'd	Description
	Standard Header	Y	MsgType = A , Component block – see 5.2 above
98	*EncryptMethod*	Y	Always set to 0 – indicates the message is not encrypted -
108	*HeartBtInt*	Y	Heartbeat Interval required. In seconds e.g. 60. Used by both sides of the connection. Note, the LME recommends the value should be 60 seconds or more
95	*RawDataLength*	Y	Used for password encryption – see below
96	*RawData*	Y	Used for password encryption – see below
141	*ResetSeqNumFlag*	N	Indicates that both sides of the FIX session should reset sequence numbers – see 5.4.4
789	*NextExpectedMsgSeqNum*	N	Required when re-connecting intra-day or after failover see 5.4 below
553	*Username*	Y	Contains user name assigned by the LME
554	*Password*	Y	Contains the password assigned by the LME – must be encrypted
	Standard Trailer	Y	See 5.2 above

On receipt of a ***Logon*** message, The LMEsmart server validates the contents:

- Does the *Username* value correspond to a known user account?
- Does the decrypted password correspond to the password associated with the *Username* provided?

If these tests succeed then LMEsmart responds with a ***Logon*** message that informs the matching client that they have successfully logged on to the LMEsmart system. The MsgSeqNum(34) in the ***Logon*** response message from LMEsmart can have a sequence greater than 1 (even if this is the first logon of the day by the matching client). If the MsgSeqNum(34) in the Logon response is greater than the matching client was expecting then standard FIX session recovery mechanisms should be

used to request any missed sequences (i.e. by sending a ***ResendRequest***). The format of the server generated ***Logon*** response message is as follows:

Tag	Name	Req'd	Description
	Standard Header	Y	MsgType = A , Component block – see 5.2 above.
98	*EncryptMethod*	Y	Always set to 0 – indicates the message is not encrypted (the password however maybe)
108	*HeartBtInt*	Y	Heartbeat Interval required. In seconds e.g. 60. Reflects the heartbeat interval included in the matching client's *Logon* message
	Standard Trailer	Y	See 5.2 above

In the case that the matching client's logon attempt fails validation for some reason then the LMEsmart server responds with a ***Logout*** message formatted as follows:

Tag	Name	Req'd	Description
	Standard Header	Y	MsgType = 5 ,Component block – see 5.2 above
58	*Text*	N	A string that describes the reason for the Logon failure
	Standard Trailer	Y	See 5.2 above

N.B. if the Username provided in the ***Logon*** message is the same as that provided by an existing matching client FIX session then the logon attempt will fail and the client will receive a ***Logout*** message containing an appropriate error message. In this case the Member should contact the LMEsmart service desk and request that they terminate the existing client session so that they can successfully connect and logon to LMEsmart.

5.3.1.1 Password Encryption

The following entities make up the encrypted password:

- A hash pattern on the source password
- The 64 byte ascii fax key – this is provided by LMEsmart for the LMEsmart FIX API users
- The “HmacSHA1” encryption algorithm
- An 8 byte increasing number generated by the matching client. Which is generated by the matching client as follows:
    - Is an integer number.
    - Has a value of the current (GMT) system time in milliseconds*
    - Is higher than the previous midnight’s (GMT) time in milliseconds*
    - Is lower than the following midnight’s (GMT) time in milliseconds*
    - Has a higher value than the number used in the previous logon request.

      **Milliseconds since 1/1/70 (unix epoch time)*

The matching client puts the number generated in the *RawData* (96) field prefixed by “m:”, e.g. “m:1065641126118” The *RawDataLen* (95) field is be set to the total length of *RawData* (including the prefix).

The LMEsmart FIX server component uses the *RawData* (96) and *RawDataLength* (95) fields to decrypt the *Password* field provided such that it can be compared to the reference data for the given company and user.

On receipt of a ***Logon*** message, the LMEsmart server validates the contents using the following rules:

- Logon is in format as given above
- Username and password are correct
- That the user does not have an existing active session
- That the *SendingTime* (52) is within a configurable range of the current UTC time.
- That the *RawData (96)* number is within a configurable range of the current milliseconds since 1/1/70.

5.3.2 Heartbeat and Test Messages

Heartbeat messages are used to monitor the status of the communication link and identify incoming sequence number gaps. When sending the ***Logon*** message matching clients send a required heartbeat interval using the *HeartBtInt* (108) tag. The LMEsmart server simply echoes the heartbeat interval value provided by the matching client when returning the ***Logon*** response message to the matching client. Thereafter both LMEsmart and the matching client should assume that the suggested heartbeat interval is in use. The LME recommends the heartbeat interval should be 60 seconds or more.

Heartbeats must then be sent in both directions at the specified interval:

- The LMEsmart server sends heartbeat messages at the required interval and receives heartbeats from matching clients, unless other messages are sent.
- A matching client sends heartbeat messages at the required interval and receives heartbeats from the server, unless other messages are sent.

If an application has not received any message from the opposite application for a period of time equal to 1.5 heartbeat intervals then a ***Test Request*** message should be sent. The opposite application should respond immediately with a ***Heartbeat*** message with *TestReqId* field set to the same value as that contained in the ***Test Request*** message that prompted it. If the opposite application does not respond within 1.2 heartbeat intervals then the connection is closed.

Heartbeat messages have the following format:

Tag	Name	Req'd	Description
	Standard Header	Y	MsgType = 0, Component block – see 5.2 above
112	*TestReqID*	N	Required when the heartbeat is a result of a *Test Request* message. Number reset to zero at start of day and incremented each time a test message is sent
	Standard Trailer	Y	See 5.2 above

Test Request messages have the following format:

Tag	Name	Req'd	Description
	Standard Header	Y	MsgType = 1, Component block – see 5.2 above
112	*TestReqID*	Y	The value will be returned in the answering *Heartbeat* message
	Standard Trailer	Y	See 5.2 above

5.3.3 Resend Request

If either LMEsmart or a matching client detects a gap in the sequence numbers included in the messages that they have received then they can send a ***Resend Request*** in order to tell the opposite application to send/resend the messages corresponding to the missing sequence numbers. See section 5.4 below for further discussion on message recovery. The format of the ***Resend Request*** message is as follows:

Tag	Name	Req'd	Description
	Standard Header	Y	MsgType = 2, Component block – see 5.2 above
7	*BeginSeqNo*	Y	Message sequence number of the first message in the range to be resent
16	*EndSeqNo*	Y	Message sequence number of the last message in the range to be resent. If the request is for all messages subsequent to a given message then this field should have its value set to zero
	Standard Trailer	Y	See 5.2 above

5.3.4 Sequence Reset

In order to reset the connection to a known position with regards to sequence numbers, either party to a FIX session may send a ***Sequence Reset*** message. See section 5.4 below for further discussion on message recovery. ***Sequence Reset*** messages have the following format:

Tag	Name	Req'd	Description
	Standard Header	Y	MsgType = 4, Component block – see 5.2 above
123	*GapFillFlag*	N	Y = Gap Fill message
36	*NewSeqNo*	Y	The new sequence number to be expected by the opposite application
	Standard Trailer	Y	See 5.2 above

5.3.5 Reject

Session level ***Reject*** messages can be used when a breach of FIX session level rules is detected. Session level *Reject* messages should not be confused with business level *Reject* messages which deal with business context dependent data omissions or unsupported features and are discussed later on this guide – see 6.6 below

LMEsmart should not send any session level reject messages, but the format of these is given here for completeness as the client application may wish to use them.

A ***Reject*** message has the following format:

Tag	Name	Req'd	Description
	Standard Header	Y	MsgType = 3, Component block – see 5.2 above
45	*RefSeqNum*	Y	*MsgSeqNum* of rejected message
371	*RefTagID*	N	The tag number of the FIX field being referenced
372	*RefMsgType*	N	The *MsgType* of the FIX message being referenced
373	*SessionRejectReason*	N	Code to identify the reason for the session level message rejection – see list below
	Standard Trailer	Y	See 5.2 above

A list of possible codes to be included in session level *Reject* messages (in *SessionRejectReason)* is given below:

Code	Description
0	Invalid FIX tag number
1	Required FIX tag missing
2	FIX tag not defined for this *MsgType*
3	Undefined FIX tag
4	FIX tag specified without a value
5	Value is out-of-range for this FIX tag
6	Incorrect data format for value
7	Decryption problem
8	Signature problem
9	*CompID* problem
10	*SendingTime* accuracy problem
11	Invalid *MsgType*
12	XML Validation error
13	FIX tag appears more than once
14	FIX tag specified out of the correct order
15	Repeating group fields out of order
16	Incorrect *NumInGroup* count for repeating group
17	Non ‘data’ value includes field delimiter (SOH)
98	Service not available at this time
99	Other

5.3.6 Logout

A ***Logout*** message initiates and confirms the termination of a FIX session. The LMEsmart server and matching client should always terminate a FIX session gracefully by using the ***Logout*** message.
The initiator of the session termination should send a ***Logout*** message and then wait to receive an answering ***Logout*** message from the opposite application prior to closing the socket connection. This approach ensures that the opposite application has received and processed all messages up to and including the ***Logout***.

A standard Logout message has the following format:

Tag	Name	Req'd	Description
	Standard Header	Y	MsgType = 5, Component block – see 5.2 above
58	*Text*	N	Additional description - optional
	Standard Trailer	Y	See 5.2 above

As discussed above (see 5.3.1), the Logout message can also be used to refuse a Logon attempt by the matching client and in this case an error text field is included in the ***Logout*** response

5.4 Session Level Message Recovery

The LMEsmart server conforms to the FIX4.4 standard for sequence number handling and recovery. Sequence numbers are used to detect messages that have been missed due to a break in communications. In order to support recovery, the LMEsmart server and matching clients must save last receive and sent sequence numbers and cache the corresponding FIX format messages. There are a number of ways in which message recovery can work and the following sections summarise them:

5.4.1 Gap Fill Processing

When processing a backlog of FIX messages during resend processing, it is important that administrative messages are not resent. For example, it is not necessary to resend ***Heartbeat*** or ***Test Request*** messages that may have been included in the original message stream. In this case a ***Sequence Reset*** message (see 5.3.4) is sent instead to indicate that the message which would have been sent with the next sequence number was an administrative message.

The ***Sequence Reset*** with Gap Fill Flag set to “Y” indicates that the opposite application should ignore the fact that there is a gap in sequence numbers and be reassured that all business level messages will be included in the replay.

5.4.2 Recovery using Resend Request

If the LMEsmart server or matching client detects a gap in the incoming message sequence numbers then they may send a ***Resend Request*** to the opposite application (see 5.3.3 above for message details).

It is strongly recommended that the ***Resend Request*** should contain an *EndSeqNo* field set to zero. In other words the component should throw away or ignore any messages with sequence number greater than the first missing sequence number – this avoids any danger of duplication. For example if the application receives messages with sequence numbers 3,4,5,7 then it should discard the message with sequence number 7 and then ask the opposite application for a resend with *BeginSeqNo* =6 and *EndSeqNo* = 0.

In all cases resent messages include a *PossDupFlag* field (tag 43) in the message header to indicate that the message has been sent as the result of a *ResendRequest*.

5.4.2.1 Example 1 – Using A ResendRequest

In this case the receiver (IN) was expecting a message with sequence number 10 but receives a message with sequence number 12. The receiver sends (OUT) a ResendRequest message. The sender resends the missing messages marked as possible duplicates and then resumes sending messages 'normally' from the point where the resend processing is complete. The message type – 'Biz' just indicates any business layer message e.g. Trade Capture report etc.

Direction	MsgType	MsgSeqNum	BeginSeqNo	EndSeqNo	NewSeqNo	PossDup	GapFill
IN	Biz	12	-	-	-	N	-
OUT	Resend Req	155	10	0	-	N	-
IN	Biz	10	-	-	-	Y	-
IN	Biz	11	-	-	-	Y	-
IN	Biz	12	-	-	-	Y	-
IN	Biz	13	-	-	-	N	-

5.4.2.2 Example 2 – Using ResendRequest With GapFill

As per Example 1 above but this time a ***Sequence Reset*** message in Gap Fill mode is sent to the recovering application. The reset is required to step over Administration messages e.g. Heartbeat. In this case there are two administration messages to ignore and they had sequence numbers 11 and 12.

Direction	MsgType	MsgSeqNum	BeginSeqNo	EndSeqNo	NewSeqNo	PossDup	GapFill
IN	Biz	14	-	-	-	N	-
OUT	Resend Req	198	10	0	-	N	-
IN	Biz	10	-	-	-	Y	-
IN	Seq Reset	11	-	-	13	Y	Y
IN	Biz	13	-	-	-	Y	-
IN	Biz	14	-	-	-	Y	-
IN	Biz	15	-	-	-	N	-

5.4.3 Recovery using NextExpectedSeqNum

Logon messages can also act as resend requests since this message may contain a *NextExpectedMsgSeqNum* field (tag 789). This field is included following a FIX session reconnection that takes place intraday. In this case the opposite application can calculate the range of messages that it should resend in order to ensure that the reconnected session is up to date.

If the intraday reconnection occurs due to some failover scenario then the ***Logon*** message may have the *ResetSeqNumFlag* field set to 'Y' to indicate that no recovery is to be attempted by either party to the connection.

5.4.3.1 Example – Use of NextExpectedSeqNum at Logon

In this case, either party to a FIX session can decide to automatically resend messages to the opposite application if it detects a sequence number gap. In the example below LMEsmart detects that the *NextExpectedSeqNum* value contained in the matching client's Logon message (150) is less than the value it would expect it to be (154) and so, having accepted the Logon attempt by returning a Logon message of its own to the client, the LMEsmart proceeds to send the messages that the matching client has missed. Note the GapFill required to 'step over' the LMEsmart's original Logon response (154)

Direction	MsgType	NextExpecSeqNo	MsgSeqNum	NewSeqNo	PossDup	GapFill
IN	Logon	150	20	-	N	-
OUT	Logon	21	154	-	N	-
OUT	Biz	-	150	-	Y	-
OUT	Biz	-	151	-	Y	-
OUT	Biz	-	152	-	Y	-
OUT	Biz	-	153	-	Y	-
OUT	Reset Seq	-	154	155	Y	Y
OUT	Biz	-	155	-	N	-

The processing is identical whichever end of the connection – matching client or LMEsmart – detects the opposite application's sequence number discrepancy.

5.4.4 Reset Sequence Numbers

This case should be a very rare exception and the most likely reason for sequence numbers to be reset is that during the interchange of Logon messages one or other party to the FIX session detects that the sequence number of an incoming Logon message is **LESS** than the next sequence number it is expecting. This indicates that the opposite application has lost context - for example it may have 'forgotten' some messages that it has already sent. The most likely cause for this eventuality to arise is that there has been a database failure or a failover to a backup system.

At this point the party that notices the discrepancy should send a Logout message containing meaningful descriptive text to the opposite application. Having received a Logout response, the offending party should realise (probably via operator intervention) that only a sequence number reset will correct the situation.

5.4.4.1 Example Reset Sequence Numbers Following Logon

In the example below LMEsmart is expecting the matching client to send a Logon message whose MsgSeqNum is 25 – but the message it sees has MsgSeqNum = 20. This implies that the matching client has 'lost' its context. The only solution is to reset sequence numbers.

Direction	MsgType	MsgSeqNum	ResetSeqNum	Text
IN	Logon	20	-	
OUT	Logout	144	-	Reset Sequence Numbers
IN	Logon	1	Y	
OUT	Logon	1	N	
IN	Biz	2	-	

It is quite likely that this scenario would involve human operator intervention – a configuration flag should be set to indicate that one or other party should request sequence numbers to be reset at the next Logon attempt.

5.4.4.2 Example Reset Sequence Numbers Mid-Session

A sequence reset can also be achieved mid -session without a Logon exchange. The following example illustrates this behaviour:

Direction	MsgType	MsgSeqNum	NewSeqNo	PossDup
IN	Biz	194	-	N
OUT	Reset Seq	8	10001	N
IN	Biz	10001	-	N

This type of reset is only attempted if one party has encountered some unrecoverable error but is happy to proceed without any kind of manual intervention. There is a danger in this approach in that important messages may have been missed by one party or the other and therefore it is unlikely that this behaviour will be experienced in the LMEsmart system.

5.5 Start and End of Day

On business days, the LMEsmart server will accept connections between the start of day (00:45) and the end of day (21:55). However the services available to a connected client depend on the current market state:

- Market Pre-open (00:45 to 01:00) – a client may connect and logon to LMEsmart but they may not submit Trades at this point. Download requests will be processed.
- Market Open (01:00 to 20:00) – A connected and logged on client may now submit Trades.
- Market Pre-close (20:00 to 21:45) – During this period, a connected and logged on client may not submit Trades to LMEsmart but Download requests will be processed.
- Market Closing (21:45 – 21:55) - During this period, a connected and logged on client may not submit Trades to LMEsmart but Download requests will be processed. Any outstanding trades will be abandoned and a Trade Execution Report sent for each one with the abandoned status.
- Market Close (21:55 to 00:45) – At the start of this period all connected clients are logged out of the LMEsmart system and their connections terminated. Clients will not be able to logon to LMEsmart.

When Market Pre-open time is reached, matching clients should, as soon as possible, establish a connection and Logon to the LMEsmart server. This ensures that the client won't miss any Trade Updates that may start to arrive as soon as Market Open time is reached.

At Market Close time, the LMEsmart clients will be logged off and disconnected. Prior to the start of the following working day, the matching client should reset sequence numbers such that both it and theLMEsmart server will start the day at sequence number one.

The market pre-open, open, pre-close, closing and close times may change in future and therefore these times should be configurable at the LMEsmart client.

6 Business Layer Overview

This section discusses some elements that are common to business layer messages exchanged between LMEsmart and matching clients. The sections following this one discuss the content of the business layer messages in much more detail.

6.1 User Access and Permissions

Having established a FIX session with LMEsmart at the start of the day, the matching client is subsequently automatically sent all Trade updates and News messages that the user of the session is entitled to. Users are created and their permissions assigned using the LMEsmart Administration application.
If, having initially established an LMEsmart session, the matching client is disconnected for some reason then, on reconnection, the user will receive all the messages they have missed since the connection was lost. Conventional FIX session message recovery processing is used to achieve this objective - see section 11 below.

6.1.1 Late Connection

If, for some reason, the matching client establishes a session after the business day has begun then the user of the session does not receive all the Trade Update and News messages that they have missed by connecting late. The matching client is just sent all the outstanding Alleged Trades that are associated with their company. However, the matching client can request the latest status of all the current day's Trades involving their member by sending a ***Trade Mass Status Request*** to LMEsmart – see section 8.1

6.2 Trade Representation

Business layer messages that include Trade details use common FIX message blocks and message fields to contain specific Trade data. The instrument being traded is described by an Instrument block (see 6.4.1 below). If an instrument consists of multiple Legs, i.e. a Carry, then the instrument block contains details of all legs involved.

Trade specific data e.g. price and size are conveyed in FIX fields which have been identified as being suitable for this purpose. In the case of a multi-leg Trade, these fields are included in a per leg block within the overall Trade half message. Regardless of how many legs a Trade may involve, Trade half data is contained within a single business layer message.

Multiple related trade halves can be contained in a single New Trade or Execution Report message. In this case, the trade half data is enclosed in a repeating block containing an entry for each of the related trade halves. This feature has been introduced to support 'Strategy' trades which are entered, matched and cleared as a single unit. The client can use this feature to capture the details of a more complex investment plan and ensure that the relationship between the component trade halves is preserved in LMEsmart.

6.3 Trade Identifiers

There are a number of different Trade and Trade half identifiers used in the LMEsmart FIX interface. The following sections enumerate these identifiers and explain their purpose and scope.

6.3.1 Matching Client Sourced Identifiers

Tag	Name	Description
11	*ClOrdID*	Matching client's own reference for a Trade half message. Used in New Trade and Cancel Trade messages. This ID should be unique over time so the following structure is recommended: 'AAA-YYYYMMDD-nnnnnnnn' Where: AAA = Client firm's mnemonic YYYYMMDD = current date nnnnnnnn = an incrementing number reset at the start of each day (Note this is just a suggested format to ensure uniqueness, but this format is not enforced by the system, the system merely checks ClOrdId is unqiue for the member on the current day)
41	*OrigClOrdID*	The value of *ClOrdID* (11) included in the original message – used in Trade Cancel requests
20020	*ComplexTradeComponentID*	Client's unique strategy identifier. Value should be unique and it should conform to MiFID Complex Trade Component ID requirements

6.3.2 LMEsmart Sourced Identifiers

Tag	Name	Description
17	*ExecID*	LMEsmart will populate this with the Trade Half Update Id. This value uniquely identifies a particular Trade Update message. The ExecID is a string of up to 21 digits
37	*OrderID*	LMEsmart will populate this with the Trade Half Id This is a unique internal identifier (across days) for a Trade half. All legs in the trade half belong to the same Trade Half Id (since they are treated as one trade half with one state). The Trade Half Id is an integer up to 20 digits.

Tag	Name	Description
5935	*MatchingRefNo*	LMEsmart populates this with the Matching Reference Number - A unique trade reference number for a matched trade across days (with the two halves that make up the trade having this same number). This is assigned when the trade halves are matched. It is a 16 character reference, made up of two parts: YYYYMMDD – system date nnnnnnnn = eight digit matching sequence number for the day – it is the same as the *TrdMatchId* (880) value (see below) but zero filled to 8 digits e.g. '00000543'
880	*TrdMatchId*	LMEsmart will populate this with the Matching Sequence No. This is an eight digit sequence number used by the matching system to identify a matched trade on a given day (with the two halves that make up the trade having this same number). This number resets to 1 at the start of each day and increments from there. This number is used as a component of the *MatchingRefNo* (tag 5935)
5442	*MatchingSlipID*	Identifier for a leg in a trade half. For example a trade half with two legs would have two slip ids associated with it. This is unique across all slips for all trades on a given day. Slips are correlated across the same leg on the two sides of a trade, Slips are 8 digits, the first 7 digits of which are a sequence number, starting at 1000001, which resets at the start of each day. The 8th digit is 1 for the buy side of the leg and 2 for the sell side of the leg. For example: 10000071 and 10000072 are slip ids for buy and sell legs of a given trade half.
20021	*StrategyID*	LMEsmart specified identifier of the Trade Strategy. Only required if multiple trade halves submitted i.e. NoTrades (897) is > 1. A number containing up to 14 digits
20022	*StrategyMatchID*	Unique Strategy match ID for the day assigned by LMEsmart. The same id is used for both Strategy halves. Required if *ExecType* = 2 (Matched) and for all post match reports. A number containing up to 14 digits

6.4 Common Message Blocks

The following blocks of data are included in all business layer messages that convey Trade half data. The detailed business layer message descriptions that are included in the following sections refer back to these common elements.

6.4.1 Instrument Block

Tag	Name		Req'd	Description
55	*Symbol*		Y	Metal or other code combined with currency e.g. 'CAD'
167	*SecurityType*		Y	Indicates the contract type. Current supported values are: F – Future T – Traded Option A - TAPO
461	*CFICode*		Y	Type of security. Can have value of "UNKNOWN" for trades in a mass status request that were rejected.
10010	*NoOfInstrumentLegs*		Y	The number of instrument legs contained within the message. 1 to n for request and trade level messages (to support carry trades). Always = 1 for Outright and Option Trades
->	20004	*InstrumentLegNo*	Y	Indicates which leg of a carry is being described. Always = 1 for Outright and Option Trades
->	10004	*PromptType*	Y	Defines prompt type that is to be used. Valid values: S = Single Prompt Date R = Rolling Prompt Date
->	10000	*MaturityRollingPrompt*	Cond	Required if *PromptType* = 'R'. Valid values: TOM = Tomorrow C = Cash 3M = 3 Months
->	541	*Maturity Date*	Cond	Required if *PromptType* = 'S'. This represents the Prompt Date for Futures or the Expiration date for Options and TAPOs Format : YYYYMMDD (E.g. '20161221') MMMYY (input only) (E.g. 'DEC16') When sent back by LMEsmart this will be populated with the absolute date in YYYYMMDD format for rolling prompts (i.e. *PromptType*= 'R') or if a Maturity Date in format of MMMYY was used on input.
->	202	*StrikePrice*	Cond	Supplied if *CFICode* (461) defines an option or TAPO contract

Tag	Name		Req'd	Description
->	5678	*Volatility*	Cond	Required for Options or TAPO.
->	20008	*UniqueProductId*	N	ESMA Unique product identifier

6.4.2 Parties Block

The parties block provides a structured mechanism for conveying information concerning the member firms, individual dealers and other related individuals and organisations that are responsible for a Trade half.

Tag	Name		Req'd	Description
453	*NoPartyIds*		Y	Number of parties specified
->	447	*PartyIDSource*	Y	This field is required if the PartyID (448) is specified. Identifies the class or source of the PartyID (448) value. Set to: D = LME Clearing Member E = PartyID is ISO country code G = Party ID is a MIC code I = 3 char broker code N = Party ID is a Client code P = Short code
->	448	*PartyID*	Y	ID of the firm or trader. Required if NoPartyIDs (453) > 0
->	452	*PartyRole*	Y	Identifies the type or role of PartyID (448) e.g. Executing Broker: 1 = Executing Firm[a] 3 (if 447=N) = Client ID[b] 3 (if 447=P) = Client Short Code[b][c] 4 = Clearing Firm 7 = Entering Firm 11 = Order Origination Trader[a] 17 = Contra Firm[a] 24 = Client Code[b] 26 = Correspondent Broker[d] 35 = Liquidity Provider Identifier [d] 36 = Entering Trader[a] 60 = Introducing Broker (IB)[e] 66 = Market Maker Identifier [d] 122 = Decision Maker Short Code[b][c] 300 = Investment Decision User ID[c] 301 = Execution Decision User ID[a][c] 302 = Investment Decision Country Code 303 = Execution Decision Country Code [a] mandatory role [b] Conditional role - required on both sides of a client trade if T4 booking model or on client side of a client trade in T2 booking model [c] Short Code – 8 byte integer [d] Only present in execution reports from LMEsmart if from the LMEselect venue.

				[e] The introducing broker party should only be populated if the Trade Execution report is being sent to LMEsmart by the registered intermediating broker (RIB) themselves. It should not be populated in Trade Execution Reports sent in by members (even if this is a trade brokered by a RIB).

Some examples of typical combinations of PartyIDSource, PartyID and PartyRole in the context of LMEsmart are given below.

Description	PartyIDSource(447)	PartyRole(452)	PartyID(448) (Example Value)
Executing Member	D	1	ABC
Counterparty Member	D	17	XYZ
Trader Id	N	11	JSMITH22
Entering Trader Id	N	36	JSMITH22
Client Id (LEI)	N	3	515800XX4X5XXX1XXX91
Client Code (DPRS)	N	24	DEF001

6.5 Overview of Business Layer Messages

Where possible, LMEsmart employs message types included in the FIX 4.4 definition, but where no appropriate message type exists in this standard, custom message types have been created - **cm2, cm3** and **cm4**. The usage of these custom message types is included in the message definitions found in the following sections.

6.5.1 Matching Client Initiated Messages

The following table lists the Business Layer messages that matching clients may send to LMEsmart:

Name	FIX MsgType	Description
New Trades List	E	Used to allow a matching client to submit a new Trade Half to LMEsmart. May also be used to submit a set of related Trade Halves to LMEsmart Response from LMEsmart is expected in an Execution Report (MsgType = 8).
Trade Cancel Request	F	Used by the matching client to cancel a live trade in LMEsmart Response from LMEsmart is expected in an Execution Report (MsgType = 8) or in a Trade Cancel Reject (MsgType = 9)
Trader Mass Status Request	AF	Allows the matching client to request status of a filtered set of live or historic orders. Response from LMEsmart is a Download Request Response (MsgType cm3) followed by a set of Execution

Name	FIX MsgType	Description
		Reports (MsgType = 8)
Download Request	cm2	Request for download of reference data Response from LMEsmart is a Download Request Response (MsgType cm3) followed by a set of either Security Definition (MsgType = d) or Member Definition (MsgType = cm4) messages

6.5.2 LMEsmart Initiated Messages

The following table lists the Business Layer messages that LMEsmart may send to matching clients:

Name	FIX MsgType	Description
Execution Report	8	Used to inform matching clients of status changes to the Trades that they have submitted via LMFI.
Trade Cancel Reject	9	Used to inform the matching client that a previous Cancel request from them has been rejected for some reason.
Download Request Response	cm3	Sent in response to a matching client initiated Trade or Reference Data download request (cm2). Indicates whether given request has been accepted or not. Does not include requested data – assuming that the request has been accepted then the data follows in further messages.
News	B	Allows the matching client to receive unsolicited service desk announcements.
Security Definition	d	Used to send security reference date to the matching client in response to a download request (cm2).
Member Definition	cm4	Used to send member reference data to the matching client in response to a download request (cm2).
Business Message Reject	j	Indicates to the matching client that LMEsmart could not process a matching client message. The reason for the rejection is specified.

6.6 Business Message Reject

A business layer reject message is sent when a valid FIX 4.4 message has been received, but from an application's point of view it is an inappropriate message. Examples include:

- Unsupported message type
- Compulsory field not included
- Conditionally required field not included
- Invalid value for a specific tag. This would be returned if the data type, length or precision of the value was incorrect.
- A received message contains a given FIX tag multiple times (outside a repeating block where this would be allowed)

The format of a business reject message is:

Tag	Name	Req'd	Description
	Standard Header	Y	MsgType = j, Component block – see 5.2 above
45	*RefMsgSeqNum*	N	*SeqNum* of the message being rejected
372	*RefMsgType*	Y	The *MsgType* of the rejected message
379	*BusinessRejectRefID*	Cond	Specifies the business level 'ID' of the message being rejected and is required unless no such identifier was supplied in the message. E.G. *ClOrdID* or *OrigClOrdID*
380	*BusinessRejectReason*	Y	Specifies the reason that the message was rejected based on the following values: 0 – Other 1 – Unknown ID 2 – Unknown security 3 – Unsupported message 4 – Application not available 5 – Conditionally required field not provided 6 – Not authorised
58	*Text*	N	Additional information about the message in the form of free format text.
	Standard Trailer	Y	See 5.2 above

7 Sending Trades

7.1 Sending trades to LMEsmart

Once connected and logged on to LMEsmart, client applications can contribute Trade halves on behalf of the firm with which the application is associated. The format required is given in section 7.4.1. The LMEsmart FIX API supports ASCII character range 32 – 122. Any characters used outside of this range may result in undesired formatting being returned

In certain cases LMEsmart will automatically generate trade halves in response to a new trade from a client application. There are two cases where this occurs:

- If the client application provides a half trade with a category of Give-Up Executor and a UNA counterparty. LMEsmart will generate the other half for the UNA counterparty so this trade can be matched and cleared. When the counterparty is know the client application should send a cancel for the original trade and a new trade with the known counterparty.
- If the client application provides a trade half with a client account type in a T4 market. LMEsmart will automatically generate the matching counterpart side of this trade.

7.2 Receiving trade updates from LMEsmart

LMEsmart will send trade status updates to the client application in the form of Trade Execution Reports. These reports inform the matching client of the progress of their contributed Trade halves through the LME's matching and clearing systems. The format of Trade Execution Reports is given in section 7.5.1.

Trade half progress is indicated by changes to the trade status value returned in the Trade Execution Reports in the OrdStatus(39) tag. The following statuses may be observed:

- Unmatched
- Matched
- Cancelled
- Rejected
- Sent To Clearing
- Cleared
- Paired
- Abandoned
- Pending Acceptance
- Pending Validation

The client application can receive Trade Execution Reports that do not contain a change of status, but just provide additional information:

- Price Conversion i.e. price code converted to absolute price value following the publication of reference price data

LMEsmart will also send Trade Execution Reports to the client application in two other cases:

- If a trade half has been entered by another party where the client application's member is the counterparty, i.e. an alleged trade. This provides the opportunity for the client application to input the counterparty half trade into the matching system.
- If a trade half has been entered into LMEsmart via an alternative source but the client application's member is the member or counterparty in the trade half, e.g. trade halves that are entered into LME's electronic trading platform (LMEselect).

7.3 Cancelling Trades

In general the client application can cancel Trade halves provided that they have not already been matched.

There are two exceptions to this when trades in other states can be cancelled:

- Give-Up Executor trades with a UNA counterparty – these can be cancelled within the same business day as they were registered even if they are cleared.
- Client crosses - in general these can be cancelled within the same business day as they were registered in any state. If the trade is in the "sent to clearing" state then it can only be cancelled if it is before the clearing house opens for clearing. If it is later than this then the client must wait for an update from the clearing house before cancelling (i.e. either pending or cleared).

The format for cancellation messages is given in section 7.4.2.

Trade half cancellation attempts are responded to by LMEsmart – it will return either an Execution Report with status set to cancelled or, if the cancellation is not permitted, a Cancel Reject message containing the reason why the request failed.

7.4 Client Initiated Messages

7.4.1 New Trades List

The matching client contributes new Trade halves by sending a ***New Trades List*** message. This message is capable of delivering a list of related trade halves at once – such a collection of trade halves is also known as a 'Strategy Trade'. It is anticipated that in the vast majority of cases, only one trade half will be delivered in each **New Trades List** message. The key field that determines how many trades are present in the message is *NoTrades* (897).

The LMEsmart FIX API supports ASCII character range 32 – 122. Any characters used outside of this range may result in undesired formatting being returned

The **New Trades List** message is formatted as follows:

Tag	Name	Req'd	Description
	Standard Header	Y	MsgType = E, Component block – see 5.2 above
	Parties	Y	Standard set of fields see 6.4.2

Tag	Name	Req'd	Description
1	*Account*	Cond	Account mnemonic. Required on both sides of a client trade if T4 booking model or on client side of a client trade in T2 booking model.
581	*AccountType*	Y	Indicates the type of account associated with the original order. One of: • 1 = ISA • 2 = House • 3 = Omnibus • 7 = Gross OSA • 9 = Unallocated House • 90 = Unallocated Client
5681	*ExchangeTradeType*	Y	Exchange defined type of trade. Valid values: • 0 – NORMAL • 2 – GIVE-UP EXECUTOR • 3 – TRANSFER • 8 – EXCEPTION - REPORTABLE • 9 – OTC BRING ON • 10 – OTC TAKE OFF • 14 – GIVE-UP CLEARER • 15 – FINANCING • 16 – EXCEPTION NON-REPORTABLE
7931	*VenueID*	Y	Venue of execution • 0 – Inter-office • 1 – Ring • 2 – Select • 3 – Basis Ring
1301	*MarketID*	Y	Market of execution. • 'LME' = LME Base Metals • 'LPM' = LME Precious Metals
943	*TimeBracket*	Cond	LME session Time e.g. R1 Either TimeBracket or TradeTime must be supplied
5179	*TradeTime*	Cond	This is the time at which the contract was agreed between the counterparties. Format - "hh:mm:ss.sss". Either TradeTime or TimeBracket must be present Note that this time is in UTC.
75	*TradeDate*	Y	Trade Date – format YYYYMMDD Defaults to current business date

Tag	Name		Req'd	Description
20025	*StrategyClOrdId*		Cond	Client's unique strategy identifier - only required if multiple trade halves submitted i.e. NoTrades (897) is > 1. Value should be unique Should be prefixed with firm Mnemonic and business date to guarantee uniqueness
20023	*CommodityDerivativeIndicator*		Y	Nature of associated risk. Values: • 0 – Trade is risk reducing • 1 – Trade is NOT risk reducing
897	*NoTrades*		Y	The number of single trade half descriptions to follow. Usually 1 but may be any value up to 64
->	11	*ClOrdId*	Y	Client specified unique identifier of the Trade half. Should be prefixed with firm Mnemonic and business date to guarantee uniqueness
->		*Instrument*	Y	Standard description fields See 6.4.1
->	20020	*ComplexTradeComponentID*	N	Client's unique strategy identifier. Value should be unique and conform to MiFID Complex Trade Component ID requirements
->	10051	*TradingCapacity*	Y	Values allowed: • 'DEAL' – Dealing on own account • 'MTCH' – Matched Principal • 'AOTC' – Any Other Capacity
->	10052	*CountryOfBranchOfClient*	Cond	Required for client trades
->	820	*TradeLinkId*	N	To satisfy Transaction Reporting obligations, members will be required to populate this trade reference across multiple trade halves

Tag	Name			Req'd	Description
->	20032		*CancellationFlag*	N	Possible values are: • C = Correction • R = Reversal This can be used to correct or reverse a previous trade. The member should send a new trade with the same trade category as the original trade. The new trade should have the Cancellation Flag set and the Cancel Link Id set to the MatchingRefNo (5935) of the original trade that is being reversed/cancelled. Note that using this flag is NOT the same as sending an explicit cancel message as described in section 7.4.2.
->	20033		*CancelLinkId*	Cond	Required if the Cancellation Flag (20032) is set. This should be the MatchingRefNo (5935) of the original trade that is being reversed/cancelled
->	31		*LastPx*	Cond	Option premium. Required for Options or TAPO
->	20031		*TradedPremium*	N	Option premium before any commission or other costs
->	810		*UnderlyingPx*	Cond	Underlying Futures price. Required for Options or TAPO
->	423		*PriceType*	Y	Possible values: • 0 = Current • 2 = Historic
->	54		*Side*	Y	Indicates the side of the trade. Values: • 1 = Side 1 (by convention the buy side) • 2 = Side 2 (by convention the sell side)
->	555		*NoLegs*	Y	Number of trade legs attached to the trade half (1 – 124). Set to '1' for outright trades and options. Greater than '1' for carry trades
->	->	20005	*LegInstrument*	Y	Corresponds to InstrumentLegNo. in Instrument block
->	->	624	*LegSide*	Y	Represents whether the Trade in this leg was a Buy or a Sell. Values: • 1 = Buy • 2 = Sell
->	->	10003	*LegLastQty*	Y	Trade Quantity of this leg

Tag	Name			Req'd	Description
->	->	5474	*AbbreviatedPrice*	Cond	Price code. Either *AbbreviatedPrice* or *LegLastPx* must be supplied unless *CFICode* (461) defines an option or TAPO contract. See Appendix A for details of permitted codes
->	->	637	*LegLastPx*	Cond	Actual price for this trade leg Either *AbbreviatedPrice* or *LegLastPx* must be supplied unless the *CFICode* (461) defines an option or TAPO contract.
->	->	20030	*TradedPrice*	N	Traded price for this leg before any commission or other costs
60	*TransactTime*			Y	The time this message was generated specified as a UTC timestamp, e.g. YYYYMMDD-HH:mm:ss.SSS based on a 24 hour clock.
5476	*PrivateReference*			N	Free form text up to 80 chars
5477	*PublicReference*			Cond	Free form text up to 12 chars. The first 5 chars are used for matching purposes and are NOT case sensitive Required if *ExchangeTradeType* is one of: • GIVE-UP EXECUTOR • GIVE-UP CLEARER Required if this execution report is a directed bid/offer on a PTT Auction. In this case the PublicReference should be set to the Auction Id (which is in the format YYMMDDNNNNNN, e.g. 200225000001).
	Standard Trailer			Y	See 5.2 above

7.4.2 Cancel Trade

The matching client can cancel a previously entered Trade half or Trade strategy by sending a **Cancel Trade** message to LMEsmart. This message can be used to cancel an individual Trade Half identified by *OrigClOrdID*, or to cancel a related set of Trade Halves (Strategy Trade) identified by *ComplexTradeComponentID*. In the latter case, all Trade Halves contained in a Strategy Trade will be cancelled.

A **Cancel Trade** message has the following format:

Tag	Name	Req'd	Description
	Standard Header	Y	MsgType = F, Component block – see 5.2 above
11	*ClOrdId*	Y	Matching client specified unique identifier of the cancel request
41	*OrigClOrdID*	Cond	*ClOrdId* (11) of the trade being cancelled
20025	*StrategyClOrdId*	Cond	Client's unique strategy identifier of the strategy to be cancelled - one of *OrigClOrdID* or *StrategyClOrdId* must be supplied
60	*TransactTime*	Y	The time this message was generated specified as a UTC timestamp, e.g. YYYYMMDD-HH:mm:ss.SSS based on a 24 hour clock.
	Standard Trailer	Y	See 5.2 above

7.5 Server Initiated Messages

7.5.1 Execution Report

The LMEsmart server informs the matching client of Trade half status changes by sending it Trade Execution Reports. The member only receives Trade Execution Reports for Trade halves that it is qualified to see i.e. that it is a party to. These report messages have the following format:

Tag	Name	Req'd	Description
	Standard Header	Y	MsgType = 8, Component block – see 5.2 above
17	*ExecID*	Y	Server specified identifier for this message. Will be zero if message is in response to a Trade Mass Status Request (*MsgType* = AF). LMEsmart will populate this with the Trade Half Update Id See 6.3.2 for more details.
11	*ClOrdID*	Cond	This tag is only included at this level if *ExecType* (150) = Cancelled (4). When at this level, this tag contains the cancellation request id. This is either: • The ClOrdId from the original Trade Cancel request (MsgType = F) if the cancellation was originated via FIX • A system generated cancellation request id if the cancellation was originated via the GUI or the system (at the end of a PTT Auction). In the second case, this tag will contain _SYS on the end. A number can also be included on the end of the _SYS if a GUI triggered cancellation resulted in more than one half being cancelled.

Tag	Name	Req'd	Description
150	*ExecType*	Y	The reason the execution report was generated. Values: • 0 = Accepted • 2 = Matched • 4 = Cancelled • 8 = Rejected • 9 = Pending Acceptance • I = Status Response • R = Price Conversion • S = Clear Update • T = Abandoned • P = PTT Auction Triggered
39	*OrdStatus*	Y	Values: • 0 = Unmatched • 2 = Matched • 4 = Cancelled • 7 = Pending Validation • 8 = Rejected • 9 = Pending Acceptance • T = Abandoned • V = Sent To Clearing • W = Cleared Trade • X = Paired • P = PTT Auction N.B. In the LMEsmart data dictionary (Reference [5]) this field is known as Trade State
5440	*ClearingStatus*	Cond	Indicates the clearing status of Trades that have been matched and sent to clearing. Values: • 2 = Cleared • X = Paired • Y = Rejected • W = Cancelled
5507	*TrdMatchTime*	Cond	Required if *ExecType* = 2 (Matched) and for all post match reports (*OrdStatus* = V,W) LMEsmart timestamp for the match. Format – 'YYYYMMDD-HH:MM:SS.sss'
103	*OrdRejReason*	Cond	A code specifying the reason for the Reject Required if ExecType (150) = 8 (Rejected) Values: • 1 = Unknown security • 2 = Exchange closed

Tag	Name	Req'd	Description
			• 5 = Unknown order • 6 = Duplicate order (*ClOrdId*) • 99 = Other
58	*Text*	Cond	In the case of a Reject Report (if *ExecType(150)* is Rejected) then this field contains LMEsmart specific text – including further error status codes – to better describe the reason for the Rejection. Appendix B for a list of possible error messages
	Parties	Y	Standard set of fields see 6.4.2 N.B. When an alleged trade half report is sent to a member then the information concerning the counterparty in the Parties block will be confined to the member mnemonic. I.e. no trader information, investment decision responsibility or client data will be included
1	*Account*	Cond	Account mnemonic This field is not provided on Execution reports generated for a Trade alleged to a member
581	*AccountType*	Cond	Indicates the type of account associated with the original order. The following values are applicable: • 1 – ISA • 2 – House • 3 – Omnibus • 7 – Gross OSA • 9 – Unallocated House • 90 – Unallocated Client This field is not provided on Execution reports generated for Trade alleged to a member
5681	*ExchangeTradeType*	Y	Corresponds to the value in the originating New Trade message
7931	*VenueID*	Y	Corresponds to the value in the originating New Trade message
1301	*MarketID*	Y	Corresponds to the value in the originating New Trade message
943	*TimeBracket*	Cond	Corresponds to the value in the originating New Trade message

Tag	Name	Req'd	Description
5179	*TradeTime*	Cond	Corresponds to the value in the originating New Trade message or if a TimeBracket(943) value was provided then the trade time as substituted by the matching system for that time bracket. E.g. if a TimeBracket(943) of R1 was provided for contract CAD then the system would substitute the TradeTime as 12:04:00. This value is in UTC
75	*TradeDate*	Y	Corresponds to the value in the originating New Trade message
60	*TransactTime*	Y	The time this message was generated
5476	*PrivateReference*	Cond	Matches the value in the original New Trade request. This field is also used to provide a reference to the original directed bid/offer when this is filled and a new trade created as part of post-auction processing on PTT Auctions. In this case the system will set the PrivateReference to: AuctionId-ClientOrderIdOfBidOffer-FillNumber E.g. 20200225000001-ABC2002250000001234-1 This field is not provided on Execution reports generated for Trade alleged to a member
5477	*PublicReference*	Cond	Must match the value in the order. Where this is included on a trade alleged to a member only the first 5 characters will be included
20025	*StrategyClOrdId*	Cond	Corresponds to the value in the originating New Trade message. For use in cancellation responses please see 0 below

Tag	Name		Req'd	Description
20023	*CommodityDerivativeIndicator*		Y	Nature of associated risk. Values: • 0 – Trade is risk reducing • 1 – Trade is NOT risk reducing
20021	*StrategyID*		Cond	LMEsmart specified identifier of the Trade Strategy. Only required if multiple trade halves submitted i.e. NoTrades (897) is > 1. See 6.3 for more details
20022	*StrategyMatchId*		Cond	Unique Strategy match ID for the day assigned by LMEsmart. The same id is used for both Strategy halves. Required if *ExecType* = 2 (Matched) and for all post match reports. . See 6.3 for more details
897	*NoTrades*		Y	The number of single trade half descriptions to follow. Usually 1 but may be any value up to 64
->	11	*ClOrdId*	Cond	In general this is the Client specified identifier of the Trade. The exception to this is when the system has auto-generated a Trade. In this case additional characters will be appended to the ClOrdId. There are three different auto-generation cases: • Client Trades – text "_AUTO" is appended • UNA Replacement – text "_REPL" is appended • Post auction filling – text "_AUCN" is appended (where N is a number indicating the fill number) Must be present at this level for all trade update reports which are not a response to a cancellation request. Note that the use of this non-mandatory tag as the first in a repeating block is not fix conformant.
->	41	*OrigClOrdId*	Cond	Original client specified identifier of the Trade Only required if the Execution report is sent in response to a cancellation request i.e.

Tag	Name		Req'd	Description
				ExecType (150) is set to 4 (Cancellled)
->	37	*OrderID*	Y	LMEsmart specified identifier of the Trade. LMEsmart will populate this with the Trade Half Id. See 6.3 for more details.
->		*Instrument*	Y	Standard description fields See 6.4.1 Corresponds to the values in the originating New Trade message.
->	20020	*ComplexTrade ComponentID*	N	Corresponds to the values in the originating New Trade message.
->	31	*LastPx*	Cond	Option premium.
->	20031	*TradedPremium*	N	Option premium before any commission or other costs
->	810	*UnderlyingPx*	Cond	Underlying Futures price This field is not provided on Execution reports generated for Trade alleged to a member
->	423	*PriceType*	Y	Corresponds to the value in the originating New Trade message
->	10051	*TradingCapacity*	Y	Corresponds to the value in the originating New Trade message This field is not provided on Execution reports generated for Trade alleged to a member
->	10052	*CountryOfBranch OfClient*	Cond	Corresponds to the value in the originating New Trade message
->	820	*TradeLinkId*	N	Corresponds to the value in the originating New Trade message
->	20032	*CancellationFlag*	N	Corresponds to the value in the originating New Trade message
->	20033	*CancelLinkId*	N	Corresponds to the value in the originating New Trade message

Tag	Name		Req'd	Description
->	20038	*TradeModuleId*	Cond	Only set for trade halves entered by a RIB and where the Member has FIX RIB Trade Acceptance enabled. This is the id of the Trade Module that this trade half belong to (where a trade module represents all the trade halves on one side of a trade). E.g. 1-20180420-00000001-2
->	20040	*RIB Acceptance Source*	Cond	Only set for trade halves entered by a RIB and where the Member has FIX RIB Trade Acceptance enabled where the OrdStatus is anything other than Pending Acceptance. This indicates the source of the acceptance or rejection of the RIB trade: M = Member FIX (API) W = Member Web (GUI)
->	54	*Side*	Y	Corresponds to the value in the originating New Trade message
->	880	*TrdMatchId*	Cond	Unique match ID for the day assigned by LMEsmart. The same id is used for both halves. Required if *ExecType* = 2 (Matched) and for all post match reports (*OrdStatus* = V,W). See 6.3 for more details
->	5934	*ClearingRefNo*	Cond	Clearing reference number – required for trades updated by Clearing system. Unique clearing number assigned to the Trade by Clearing (same for both Trade halves of a Matched Trade).
->	5935	*MatchingRefNo*	Cond	Unique match number assigned to the matched Trade by LMEsmart. This is the same for both halves. It consists of *TrdMatchID* (880) above but with date pre-pended to guarantee uniqueness across business days. See 6.3 for more details

Tag		Name		Req'd	Description
->		1903	*RegulatoryTradeId*	N	The Trading Venue Transaction Identification Code (TVTIC). This will be set by LMEsmart as either: If TradeVenue is Select: SelectTradeNumber If TradeVenue is not Select: Matching Reference Number
->		5939	*SelectOrderNumber*	Cond	If TradeVenue is Select e.g. '23138010000000020' This field is not provided on Execution reports generated for Trade alleged to a member
->		5940	*SelectTradeNumber*	Cond	If TradeVenue is Select e.g. '6023138010000000009'
->		5941	*SelectImpliedTrade Number*	Cond	If TradeVenue is Select and the trade was executed as the result of an implied carry. Contains a reference number indicating that the order was executed in Select as part of an implied scenario (for both outright and carries)
->		10054	*StrategyLinked OrderId*	Cond	If TradeVenue is Select
->		1057	*AggressorIndicator*	Cond	Represents whether the trade was an aggressor in the electronic order book. Is only populated if the trade originated from the LMEselect system. Values: • Y • N
->		555	*NoLegs*	Y	Number of trade legs attached to the trade half (1 – 124). Set to '1' for Outright and Option trade halves. Greater than '1' for carry trade halves
->	->	20005	*LegInstrument*	Y	Corresponds to InstrumentLegNo. in Instrument block
->	->	624	*LegSide*	Y	Represents whether the Trade in this leg was a Buy or a Sell. Values: • 1 = Buy • 2 = Sell

Tag		Name		Req'd	Description
->	->	10003	*LegLastQty*	Y	Trade Quantity of this leg
->	->	5474	*AbbreviatedPrice*	Cond	Price code Either *AbbreviatedPrice* or *LegLastPx* must be supplied unless *CFICode* (461) defines an option or TAPO contract. After price substitution both the AbbreviatedPrice and LegLastPx will be provided. See Appendix A for details of permitted codes
->	->	637	*LegLastPx*	Cond	Actual price for this trade leg Either *AbbreviatedPrice* or *LegLastPx* must be supplied unless *CFICode* (461) defines an option or TAPO contract. After price substitution both the AbbreviatedPrice and LegLastPx will be provided.
->	->	20030	*TradedPrice*	N	Corresponds to the value in the originating New Trade message If no value was originally provided then this will be set to the same value as LegLastPx by LMEsmart.
->	->	5442	*MatchingSlipID*	Cond	The slip id to identify this leg of the Trade half as allocated by LMEsmart. See 6.3 for more details Not provided for OrdStatus: • 7 = Pending Validation • 8 = Rejected • 9 = Pending Acceptance
			Standard Trailer	Y	See 5.2 above

7.5.1.1 Fields Included In Trade Cancellation Requests and Responses

Cancellation responses are formatted slightly differently to other trade updates returned to the client in Execution Reports. In a cancellation response *ClOrdId* (11) is used to identify the cancellation request id rather than the trade affected. The cancellation request id can be either:

- The original cancellation request id sent by the client if the cancellation was done via FIX
- A system generated cancellation request id if the cancellation was done via the GUI or initiated by the system at the end of a PTT Auction. System generated cancellation request ids can be identified by the presence of _SYS on the end.

In a regular trade update ClOrdId (11) is used to identify the trade that has been updated.

Also, in a cancellation response, *OrigClOrdId* (41) is used to identify the client trade that has been cancelled and corresponds to the value included on the same tag in the original cancellation request.

In the case of a strategy cancellation response then ClOrdId (11) corresponds to the value included in the original cancel request and identifies which cancellation request is being responded to. Also, in this case, *StrategyClOrdId* (20025) corresponds to the value included in the original cancellation request and *OrigClOrdId* (41) contains the client order ids of the original trade halves included in the now cancelled strategy.

7.5.2 Trade Cancel Reject

Tag	Name	Req'd	Description
	Standard Header	Y	MsgType = 9, Component block – see 5.2 above
11	*ClOrdID*	Y	ClOrdID (11) that was submitted with the order cancel request that is being rejected
41	*OrigClOrdID*	Y	The *OrigClOrdID* included in the Cancel request that is being rejected
20025	*StrategyClOrdId*	Cond	Client's unique strategy identifier of the strategy to be cancelled - one of *OrigClOrdID* or *StrategyClOrdId* must be supplied depending on which was supplied in the original cancel request
37	*OrderID*	Y	LMEsmart specified identifier of the Trade for which the cancel was submitted. Set to 'NONE' if the OrigClOrdID was unknown
39	*OrdStatus*	Y	Values: 0 = Unmatched 2 = Matched 4 = Cancelled 7 = Pending Validation 8 = Rejected 9 = Pending Acceptance T = Abandoned

Tag	Name	Req'd	Description
			V = Sent To Clearing W = Cleared Trade X = Paired
102	*CxlRejReason*	Y	A code specifying the reason for the Rejection of the original Cancel request. Values: 99 = Some other reason
58	*Text*	N	An LMEsmart code specifying the reason for the Reject (in the case that *CxlRejReason* = 99) see Appendix B for error text messages that may appear here
	Standard Trailer	Y	See 5.2 above

8 Requesting Trade History

An LMEsmart client can request Trade history reports via the LMEsmart server. For example this request can be used to recover trade data that the matching client lacks because it was not connected from the beginning of the current business day and therefore missed some earlier trade updates. Also, the matching client may want to retrieve trade data from a previous business day. The message type used in this case is ***a Trade Mass Status Request.***

On receiving a ***Trade Mass Status Request***, LMEsmart replies with a ***Download Request Response*** message that indicates whether the request has been accepted or not. In the case that the request has been accepted, the response also includes a count of the trade halves that met the request criteria. Note that a request may be accepted but return zero trade halves (e.g. if there were no trades for that symbol or the symbol was unknown).

After the Download Request Response, LMEsmart will send each of the individual ***Execution Report*** messages that met the request criteria. Trades in all states are sent. Note that any trades that were originally part of a strategy will be returned as individual Execution Reports.

There is a limit to the number of ***Trade Mass Status Request***s that an individual user may make in a single working day. This is to prevent the matching system from being used in an inappropriate fashion i.e. to prevent users from 'polling' for updates rather than receiving the update stream as intended.

8.1 Trade Mass Status Request

This request message has the following format:

Tag	Name	Req'd	Description
	Standard Header	Y	MsgType = AF, Component block – see 5.2 above
584	*MassStatusReqID*	Y	Unique ID of mass status request as assigned by the requestor
585	*MassStatusReqType*	Y	Values: 1 = History (up to 5 business days) 2 = Current (Today)
11	*ClOrdID*	Cond	ID of a specific Trade to be returned. The user cannot supply both *ClOrdID and ComplexTradeComponentID*
20025	*StrategyClOrdId*	Cond	Client's unique strategy identifier of the strategy whose trade details are to be returned
55	*Symbol*	N	Optionally request results for a given Symbol e.g. 'CAD'
167	*SecurityType*	N	Optionally request results for a given contract type. Values: F- Future T – Traded Option A - TAPO
39	*OrdStatus*	N	Values: 0 = Unmatched 2 = Matched 4 = Cancelled 7 = Pending Validation 8 = Rejected 9 = Pending Acceptance T = Abandoned V = Sent To Clearing W = Cleared Trade X = Paired
5507	*TrdMatchTime*	Cond	Time at which trade was matched. Format 'YYYYMMDD-HH:MM:SS.sss' The query will only use the date – 'YYYYMMDD' as the filtering criteria. Only valid for 'History' queries. Required for 'History' queries if *TradeDate* (75) is not supplied
75	*TrdDate*	Cond	Trade date of trades to be returned in 'YYYYMMDD' format. Only valid for 'History' queries and required if *TrdMatchTime* (5507) is not provided

Tag	Name	Req'd	Description
	Standard Trailer	Y	See 5.2 above

8.2 Download Request Response

LMEsmart responds to a ***Trade Mass Status Request*** from a matching client by sending it a ***Download Request Response*** message. This message contains the outcome of the request i.e. either success or failure. In the case that the request has been successfully processed then the ***Download Request Response*** message will contain the number of results to follow – each result will be delivered in an ***Execution Report*** message.

The ***Download Request Response*** message returned by LMEsmart is formatted as follows:

Tag	Name	Req'd	Description
	Standard Header	Y	MsgType = cm3, Component block – see 5.2 above
5449	*ReqResponseTo*	Y	Type of request being responded to: 1 = Trade Mass Status Request
584	*MassStatusReqID*	Cond	Matching client specified *MassStatusReqID* of the Trade Mass Status Request that is being acknowledged (*ReqResponseTo* = 1*)*
5469	*ReqResponseStatus*	Y	Status of the matching client's request. Values: 1 = Accepted. Results to follow 2 = Rejected
5448	*NumMsg*	Y	Number of results messages to follow this response. Set to '0' in the case that the request has been rejected (*ReqResponseStatus* = 2)
6396	*RejectCode*	N	Reason that matching client's request has been rejected – (*ReqResponseStatus* = 2)
58	*Text*	N	Explanatory text in the case that the matching client's request has been rejected
60	*TransactTime*	Y	The time that this Request Response message was created
	Standard Trailer	Y	See 5.2 above

8.3 Execution Report

The result set of a successful ***Trade Mass Status Request*** is sent to the matching client in a series of ***Execution Reports*** – one per Trade half. These ***Execution Report*** messages are very similar to those containing Trade status updates (see 7.5.1 above). The key differences to note in this case are:

- The ***Execution Report*** message includes a *MassStatusReqID* field which reflects the *MassStatusReqID* included in the original request
- The *ExecType* field in the ***Execution Report*** message can only be 'I' indicating that this ***Execution Report*** was created as a result of a Mass Status Request.

The ***Execution Report*** messages returning the result set from a successful ***Trade Mass Status Request*** are formatted as follows:

Tag	Name	Req'd	Description
	Standard Header	Y	MsgType = 8, Component block – see 5.2 above
584	*MassStatusReqID*	Cond	Unique ID of mass status request as assigned by the requestor. Only required if ExecType (4) = I – Mass status response and in this case contains the value included in the original request
150	*ExecType*	Y	The reason the execution report was generated. Values: • 0 = Accepted • 2 = Matched • 4 = Cancelled • 8 = Rejected • 9 = Pending Acceptance • I = Status Response • R = Price Conversion • S = Clear Update • T = Abandoned • P = PTT Auction Triggered

Tag	Name	Req'd	Description
39	*OrdStatus*	Y	Values: • 0 = Unmatched • 2 = Matched • 4 = Cancelled • 7 = Pending Validation • 8 = Rejected • 9 = Pending Acceptance • T = Abandoned • V = Sent To Clearing • W = Cleared Trade • X = Paired • P = PTT Auction N.B. In the LMEsmart data dictionary (Reference [5]) this field is known as Trade State
5440	*ClearingStatus*	Cond	Indicates the clearing status of Trades that have been matched and sent to clearing. Values: • 2 = Cleared • X = Paired • Y = Rejected • W = Cancelled
5507	*TrdMatchTime*	Cond	Required if *ExecType* = 2 (Matched) and for all post match reports (*OrdStatus* = V,W) LMEsmart timestamp for the match. Format – 'YYYYMMDD-HH:MM:SS.sss'
103	*OrdRejReason*	Cond	A code specifying the reason for the Reject Required if ExecType (150) = 8 (Rejected) Values: • 1 = Unknown security • 2 = Exchange closed • 5 = Unknown order • 6 = Duplicate order (*ClOrdId*) • 99 = Other
58	*Text*	Cond	In the case of a Reject Report (if *ExecType(150)* is Rejected) then this field contains LMEsmart specific text – including further error status codes – to better describe the reason for the Rejection. See Appendix B for a list of possible error messages
	Parties	Y	Standard set of fields see 6.4.2 N.B. When an alleged trade half report is sent

Tag	Name	Req'd	Description
			to a member then the information concerning the counterparty in the Parties block should be confined to the member mnemonic. I.e. no trader information, investment decision responsibility or client data should be included
1	*Account*	Cond	Account mnemonic This field is not provided on Execution reports generated for Trade alleged to a member
581	*AccountType*	Cond	Indicates the type of account associated with the original order. The following values are applicable: • 1 – ISA • 2 – House • 3 – Omnibus • 7 – Gross OSA • 9 – Unallocated House • 90 – Unallocated Client This field is not provided on Execution reports generated for Trade alleged to a member
5681	*ExchangeTradeType*	Y	Exchange defined type of trade. Valid values: • 0 – NORMAL • 2 – GIVE-UP EXECUTOR • 3 – TRANSFER • 8 – EXCEPTION - REPORTABLE • 9 – OTC BRING ON • 10 – OTC TAKE OFF • 14 – GIVE-UP CLEARER • 15 – FINANCING • 16 – EXCEPTION NON-REPORTABLE
7931	*VenueID*	Y	Venue of execution • 0 – Inter-office • 1 – Ring • 2 – Select • 3 – Basis Ring
1301	*MarketID*	Y	Market of execution: • 'LME' = LME Base Metals

Tag	Name	Req'd	Description
			• 'LPM' = LME Precious Metals
943	*TimeBracket*	Cond	LME session Time e.g. R1
5179	*TradeTime*	Cond	This is the time at which the contract was agreed between the counterparties. Format - "hh:mm:ss.sss".
75	*TradeDate*	Y	Trade Date – format YYYYMMDD
60	*TransactTime*	Y	The time this message was generated
5476	*PrivateReference*	Cond	Free form text up to 80 chars
5477	*PublicReference*	Cond	Free form text up to 12 chars. The first 5 chars are used for matching purposes and are NOT case sensitive. Required if *ExchangeTradeType* is one of: • GIVE-UP EXECUTOR • GIVE-UP CLEARER Required if this execution report is a directed bid/offer on a PTT Auction. In this case the PublicReference should be set to the Auction Id (which is in the format YYMMDDNNNNNN, e.g. 200225000001).
20025	*StrategyClOrdId*	Cond	Client's unique strategy identifier - only required if multiple trade halves submitted i.e. NoTrades (897) is > 1. Value should be unique
20023	*CommodityDerivativeIndicator*	Y	Nature of associated risk. Values: • 0 – Trade is risk reducing • 1 – Trade is NOT risk reducing
20021	*StrategyID*	Cond	LMEsmart specified identifier of the Trade Strategy. Only required if multiple trade halves submitted i.e. NoTrades (897) is > 1. See 6.3 for more details
20022	*StrategyMatchId*	Cond	Unique Strategy match ID for the day assigned by LMEsmart. The same id is used for both Strategy halves. Required if *ExecType* = 2 (Matched) and for all post

Tag	Name		Req'd	Description
				match reports. . See 6.3 for more details
897	*NoTrades*		Y	The number of single trade half descriptions to follow. Usually 1 but may be any value up to 64
->	11	*ClOrdId*	Y	Client specified identifier of the Trade
->	37	*OrderID*	Y	LMEsmart specified identifier of the Trade. LMEsmart will populate this with the Trade Half Id. See See 6.3 for more details
->		*Instrument*	Y	Standard description fields See 6.4.1 Corresponds to the values in the originating New Trade message.
->	20020	*ComplexTradeComponentID*	N	Client's unique strategy. See See 6.3 for more details
->	31	*LastPx*	Cond	Option premium.
->	810	*UnderlyingPx*	Cond	Underlying Futures price This field is not provided on Execution reports generated for Trade alleged to a member
->	423	*PriceType*	Y	Possible values: • 0 = Current • 2 = Historic
->	10051	*TradingCapacity*	Y	Values allowed: • 'DEAL' – Dealing on own account • 'MTCH' – Matched Principal • 'AOTC' – Any Other Capacity This field is not provided on Execution reports generated for Trade alleged to a member
->	20038	*TradeModuleId*	Cond	Only set for trade halves entered by a RIB and where the Member has FIX RIB Trade Acceptance enabled. This is the id of the Trade Module that this trade half belong to (where a trade module represents all the trade halves on one side of

Tag	Name		Req'd	Description
				a trade). E.g. 1-20180420-00000001-2
->	20040	*RIB Acceptance Source*	Cond	Only set for trade halves entered by a RIB and where the Member has FIX RIB Trade Acceptance enabled where the OrdStatus is anything other than Pending Acceptance. This indicates the source of the acceptance or rejection of the RIB trade: M = Member FIX (API) W = Member Web (GUI)
->	54	*Side*	Y	Indicates Buy or Sell. Values: • 1 = Buy • 2 = Sell
->	880	*TrdMatchId*	Cond	Unique match ID for the day assigned by LMEsmart. The same id is used for both halves. Required if *ExecType* = 2 (Matched) and for all post match reports (*OrdStatus* = V,W). See 6.3 for more details
->	5934	*ClearingRefNo*	Cond	Clearing reference number – required for trades updated by Clearing system. Unique clearing number assigned to the Trade by Clearing (same for both Trade halves of a Matched Trade).
->	5935	*MatchingRefNo*	Cond	Unique match number assigned to the matched Trade by LMEsmart. This is the same for both halves. It consists of *TrdMatchID* (880) above but with date pre-pended to guarantee uniqueness across business days. See See 6.3 for more details
->	5939	*SelectOrderNumber*	Cond	If TradeVenue is Select e.g. '23138010000000020' This field is not provided on Execution reports generated for Trade alleged to a member
->	5940	*SelectTradeNumber*	Cond	If TradeVenue is Select e.g. '6023138010000000009'

Tag		Name		Req'd	Description
->		5941	*SelectImpliedTrade Number*	Cond	If TradeVenue is Select and the trade was executed as the result of an implied carry. Contains a reference number indicating that the order was executed in Select as part of an implied scenario (for both outright and carries)
->		1057	*AggressorIndicator*	Cond	Represents whether the trade was an aggressor in the electronic order book. Is only populated if the trade originated from the LMEselect system. Values: • Y • N
->		555	*NoLegs*	Y	Number of trade legs attached to the trade half (1 – 124). Set to '1' for Outright and Option trade halves. Greater than '1' for carry trade halves
->	->	20005	*LegInstrument*	Y	Corresponds to InstrumentLegNo. in Instrument block
->	->	624	*LegSide*	Y	Represents whether the Trade in this leg was a Buy or a Sell. Values: • 1 = Buy • 2 = Sell
->	->	10003	*LegLastQty*	Y	Trade Quantity of this leg
->	->	5474	*AbbreviatedPrice*	Cond	Price code. Either *AbbreviatedPrice* or *LegLastPx* must be supplied unless *CFICode* (461) defines an option or TAPO contract. See Appendix A for details of permitted codes
->	->	637	*LegLastPx*	Cond	Actual price for this trade leg Either *AbbreviatedPrice* or *LegLastPx* must be supplied unless *CFICode* (461) defines an option or TAPO contract.
->	->	5442	*MatchingSlipID*	Cond	The slip id to identify this leg of the Trade half as allocated by LMEsmart. See 6.3 for more details. Not provided for OrdStatus: • 7 = Pending Validation • 8 = Rejected • 9 = Pending Acceptance
			Standard Trailer	Y	See 5.2 above

9 Requesting Reference Data Reports

LMEsmart clients can request a set of either Contract or Member reference data. In this case ***Download Request*** messages are sent to LMEsmart by the matching client.

As for Trade reports above, the first response that LMEsmart sends to the matching client is delivered in a ***Download Request Response*** message which indicates whether the request has been successful or not. In the case that the request has been successful then, depending on the type of request, the results are sent in a set of one or more ***Security Definition or Member Definition*** messages. A request may be accepted but return zero results (e.g. if the symbol was unknown).

9.1 Download Request

The Download Request message has the following format:

Tag	Name	Req'd	Description
	Standard Header	Y	MsgType = cm2, Component block – see 5.2 above
5447	*RequestID*	Y	Unique ID of download request as assigned by the requestor
5446	*RequestType*	Y	Values: 1 = Contracts 2 = Members
55	*Symbol*	N	Optionally request results for a given Symbol e.g. 'CAD'
167	*SecurityType*	N	Optionally request results for a given contract type. Values: F- Future T – Traded Option A - TAPO
60	*TransactTime*	Y	The time the download request was created
	Standard Trailer	Y	See 5.2 above

9.2 Download Request Response

The immediate response returned by LMEsmart in response to a ***Download Request*** comes in the form of a ***Download Request Response*** message. This message contains the result of the request i.e. whether the request has succeeded or failed. In the case that the request was successful then the Download Request Response message contains a count of results to follow – depending on the type of request, each result to be delivered in a ***Security Definition*** or ***Member Definition*** message. The format of the Download Request Response message is as follows:

Tag	Name	Req'd	Description
	Standard Header	Y	MsgType = cm3, Component block – see 5.2 above
5449	*ReqResponseTo*	Y	Type of request being responded to: 2 = Download Request
5447	*RequestID*	Cond	Matching client specified RequestID of the Download request being responded to (*ReqResponseTo* = 2)
5469	*ReqResponseStatus*	Y	Status of the matching client request. Values: 1 = Accepted. Results to follow 2 = Rejected
5448	*NumMsg*	Y	Number of results messages to follow this response. Set to '0' in the case that the request has been rejected (*ReqResponseStatus* = 2)
6396	*RejectCode*	N	Reason that the matching client's request has been rejected – (*ReqResponseStatus* = 2)
58	*Text*	N	Explanatory text in the case that the matching client's request has been rejected
60	*TransactTime*	Y	The time that this Request Response message was created
	Standard Trailer	Y	See 5.2 above

9.3 Security Definition

In the case that a ***Download Request*** for a set of security definitions has been successful then LMEsmart will return the results to the matching client in a set of one or more ***Security Definition*** messages. Each message formatted as follows:

Tag	Name	Req'd	Description
	Standard Header	Y	MsgType = d, Component block – see 5.2 above
5447	*RequestID*	Cond	Matching client specified RequestID of the Download request being responded to
55	*Symbol*	Y	Metal or other code combined with currency e.g. 'CAD'
167	*SecurityType*	Y	Indicates the contract type. Current supported values are: F – Future T – Traded Option A - TAPO
461	*CFICode*	Y	Type of security
541	*Maturity Date*	Cond	This represents the Prompt Date for Futures or the Expiration date for Options and TAPOs Format : YYYYMMDD (E.g. '20161221')
653	*SecDefStatus*	Y	Status of the security described by the Instrument block. Values: 0 = Active 1 = Inactive
	Standard Trailer	Y	See 5.2 above

9.4 Member Definition

In the case that a ***Download Request*** for a set of member definitions has been successful then LMEsmart will return the results to the matching client in a set of one or more **Member *Definition*** messages. Each message formatted as follows:

Tag	Name	Req'd	Description
	Standard Header	Y	MsgType = cm4, Component block – see 5.2 above
5447	*RequestID*	Cond	Matching client specified RequestID of the Download request being responded to
5322	*FirmID*	Y	Mnemonic
5364	*MemberName*	Y	Firm Name
5365	*MemberAddress*	Y	Address, if this is not available then will be set to "na"
9675	*ContactPhoneNumber*	Y	Member's contact phone number, if this is not available then will be set to "na"
9677	*ContactEmailAddress*	Y	Member's contact email address, if this is not available then will be set to "na"
5366	*ContactFax*	Y	Member's FAX number, if this is not available then will be set to "na"
5367	*AltPhone1*	Y	Contact phone number of firm's compliance department, if this is not available then will be set to "na"
5368	*AltPhone2*	Y	Contact phone number of the firm's Matching department, if this is not available then will be set to "na"
	Standard Trailer	Y	See 5.2 above

10 News

Market announcements, for example price substitution complete, are sent to the matching client in a ***News*** message. This message is formatted as follows:

Tag	Name	Req'd	Description
	Standard Header	Y	MsgType = B, Component block – see 5.2 above
148	*Headline*	Y	Specifies headline text or Announcement type
58	*Text*	Y	Announcement body text
42	*OrigTime*	Y	Time of message origination - UTC
	Standard Trailer	Y	See 5.2 above

11 Accepting or Rejecting RIB Trades

11.1 Enabling FIX Acceptance for RIB Trades

An LMEsmart client can use the FIX API to accept or reject trades as entered on behalf of the GCM by a Registered Intermediating Broker (RIB).

In order to use this facility, the GCM must first notify the LME that they wish to use the FIX Trade Acceptance mechanism. Only when this has been enabled on behalf of the GCM by the LME will the FIX tags and messages described below be available.

11.2 Identifying RIB Trades

The LMEsmart client will be able to identify Trade Execution Reports that form part of RIB entered trades that are in the Pending Acceptance state by the presence of the Trade Module Id (20038) tag on a Trade Execution Report. The Trade Module Id identifies the group of Trade Execution Reports that make up one logical side of a RIB trade and as such there can be multiple Trade Execution Reports with the same Trade Module Id.

For example, in the case of a RIB entered trade between two clients of different GCMs there would be three Trade Execution Reports with the same Trade Module Id sent to one of the GCMs as outlined in the table below - the first two represent the two halves of the necessary client trade and the third the GCMs half of the exchange trade:

Member	Counterparty	Buy/Sell	Account Type
ABC	XXX	Buy	Client
ABC	XXX	Sell	House
ABC	XXX	Buy	House

ABC – sample member code
XXX – anonymized member code

11.3 Accepting or Rejecting RIB Trades

LMEsmart clients that are enabled for FIX Trade Acceptance will be able to accept or reject Trade Modules by using the **RIB Trade Module Request** message.

Only one **RIB Trade Module Request** needs to be sent for a single trade module rather than one per Trade Execution Report (trade half) in a trade module, e.g. in the example above only one **RIB Trade Module Request** message is required and not three separate ones.

LMEsmart will respond with a **RIB Trade Module Response** indicating whether the request to accept or reject was successful. It will then send **Trade Execution Reports** as normal for the further state updates for each of the trade halves in the module, i.e. UNMATCHED, MATCHED, SENT TO

CLEARING and CLEARED. All subsequent updates will also include the TradeModuleId (20038) and RIB Acceptance Source (20040) tag for reference.

The format of these message is discussed in the next sections.

11.4 RIB Trade Module Request

The **RIB Trade Module Request** has the following format:

Tag	Name	Req'd	Description
	Standard Header	Y	MsgType = rb1, Component block – see 5.2 above
5447	RequestID	Y	Unique ID of request as assigned by the member, e.g. ABC01202006220000006
20038	TradeModuleId	Y	The id of the Trade Module as sent by LMEsmart in tag 20038 in TERs for this trade module. E.g. 1-20180420-00000001-2
20039	*AcceptanceStatus*	Y	Whether the trade module should be accepted or rejected: • 1 = Accept • 2 = Reject
	Standard Trailer	Y	See 5.2 above

11.5 RIB Trade Module Response

The **RIB Trade Module Response** has the following format:

Tag	Name	Req'd	Description
	Standard Header	Y	MsgType = rb2, Component block – see 5.2 above
5447	*RequestID*	Y	The same id as the member sent in the original request
20038	*TradeModuleId*	Y	The id of the Trade Module the member requested acceptance/rejection of. E.g. 1-20180420-00000001-2

Tag	Name	Req'd	Description
20039	*AcceptanceStatus*	Y	Whether the member requested acceptance or rejection: • 1 = Accept • 2 = Reject
5469	*ReqResponseStatus*	Y	The status of the request itself, i.e. whether LMEsmart successfully processed the request for acceptance/rejection or whether it could not process the request. Values: • 1 = Request Accepted • 2 = Request Rejected A request would be rejected if the trade halves in the Trade Module were not in the Pending Acceptance state. This would be because either they have already been accepted or because the module id is incorrect.
	Standard Trailer	Y	See 5.2 above

12 Recovery

There are a number of possible failure scenarios and the following sections describe the Recovery implications of each of them. The scenarios to be considered are:

1. Matching client disconnects during the day and, later, reconnects
2. LMEsmart has to be restarted intra-day
3. LME Site Failover

12.1 Matched Client Connection Restarted Intraday

This situation is covered by standard FIX session recovery procedure (see section 5.4 above). LMEsmart considers the matching client session to be still active even though the connection has been temporarily lost. All messages that would have been destined for the matching client are buffered up ready to be sent to the matching client when it reconnects.

Once the matching client reconnects to LMEsmart, it sends a Logon message which contains the sequence number that the matched client next expects to receive. The LMEsmart server then automatically resends the messages that the matching client has missed while it was disconnected. Eventually the missing messages have been sent and LMEsmart sends the new messages that it has buffered up during the recovery activity. Finally, the matching client catches up with the message backlog and LMEsmart sends new messages to it immediately - as would be the case if the disconnection had never occurred.

If the matching client application does not implement the next expected sequence number at Logon logic then the missing messages may still be recovered by using a Resend request. This request should include the next sequence number that the matching expected to receive and the end sequence number value should be set to zero i.e. send me everything subsequent to the start sequence number. LMEsmart will then apply the same logic as described for resends following Logon and the matching client will be brought up to date.

12.2 LMEsmart Restarted Intraday

In the highly unlikely scenario that LMEsmart has to be restarted during the working day then, if necessary, the matching clients will have their connections re-routed to an active LMEsmart instance. However there may be a short delay in this occurring and there is obviously potential for matching client messages to be lost.

This situation is the reverse of that described in the previous section. In this case it is possible that the LMEsmart server discovers during the Login interchange that it has missed one or more messages sent by the matching client. LMEsmart server will send the next expected sequence number value in its Logon response and the matching client may automatically resend the missing messages.

If the matching client application has not implemented the next expected sequence number at Logon logic then LMEsmart will, at the first opportunity, send a Resend request to the matching client and hence receive the missing messages.

It is also possible that during the events leading up to the restart of LMEsmart, the matching client also lost messages intended for it. In this case recovery is exactly as per matching client restart described in the previous section. In the worst case scenario, boith LMEsmart and the matching client may need to recover missing messages simultaneously but this is perfectly ok and is covered by the recovery rules described above except that the recovery occurs simultaneously.

12.3 Site Failover

Even less likely than LMEsmart having to be restarted, site failover recovery is achieved in exactly the same way. This is possible because LMEsmart context is mirrored between the primary and failover LME sites.

13 Changes from LMEsmart 1.X

The FIX API matching interface provided by LMEsmart 2.X will be modified from that provided in version 1.X. The majority of these modifications are required to bring the FIX API matching interface into line with the format supported by LMEmercury and LMEselect. In this way, a consistent approach is adopted across all LME platforms. These modifications are summarised below:

13.1 Representation of Multiple Legs in Carry Trades

LMEsmart currently sends a Trade Execution Report per leg and an additional 'parent' Trade Execution Report. The modified approach is to no longer send multiple Trade Execution Reports messages per leg of a carry trade half. Instead, multiple leg details will be included in a repeating block in the Trade Execution Report. The number of legs in the block will be given by the NoLegs(555) tag.

As a consequence of this change, the following fields currently used by LMEsmart are no longer required:

- SeriesNumber (5296)
- LegReport (5444)
- ParentID (9580)

13.2 Leg Fields

The usage of FIX tags in each leg will also be standardised with LMEselect and LMEmercury.

Original FIX Field	New FIX Field
-	LegInstrument (20005) (links to the instrument – see below)
Side (54)	LegSide (624) (note this actually refers to whether it is a buy or sell so not strictly a "side")
Abbreviated Price (5474)	Abbreviated Price (5474) (but this is now only used to contain a price code and is never used for an actual price)
Price (44) (futures)	LegLastPx (637)
LegQty (687)	LegLastQty (10003)

Note that the previous version of LMEsmart also used the Price(44) field to contain a premium value for options contracts. Any premium should now be provided in the LastPx (31) tag (which is part of the common fields of a half trade and not in the leg block).

13.3 Instrument Fields

To conform to other LME FIX interfaces (LMEselect and LMEmercury), instrument details will be contained in a common Instrument block (see section 6.4.1). The following table summarises this change:

Original FIX Field	New FIX Field
Symbol (55)	*Symbol* (55)
PutOrCall (201)	*CFICode* (461)
StrikePrice(202)	*StrikePrice* (202)
Volatility (5678)	*Volatility* (5678)
CarryIndicator (5479)	*NoOfInstrumentLegs* (10010)
*PromptDate (*5475*)* *ExplicitPromptDate (5936)*	*PromptType (*10004) *MaturityRollingPrompt* (10000) *MaturityDate* (541)

13.4 Parties Block

The modified version of LMEsmart represents Party and Counterparty detail within a Parties block (see section 6.4.2). As a consequence, the following fields currently used by LMEsmart in Trade Capture reports, are no longer required:

- *FirmID* (5322)
- *ContraBroker* (375)
- *ClientID* (109)
- *SelectClientID* (5937)
- *SelectTraderID* (5938)

13.5 Other Redundant Fields

Due to the changes in Trade lifecycle required in the new version of LMEsmart and due to general data dictionary rationalisation, the following fields used by the current release of LMEsmart are also no longer required:

- ClearingMatchID (5441)
- ClearingSlipID (5443)
- CrossIndicator (5478)
- AccountCode (20011)
- 4WayAgreement (5298)
- LateIndicator (7928)
- ContraAccount (6004)

13.6 Trade Status Codes

In order that LMEsmart can conform to MiFID requirements, the following Trade Status values for OrdStatus (39) will no longer be valid:

- P = Pending Branch Acceptance (PAB)
- U = Not to Clearing (NC)
- 6 = Pending Cancel (PCX)
- Q = Pending Cat4 Acceptance (PAC)
- Y = Clear Error

13.7 Clearing Status Codes

The following Clearing Status Codes no longer apply:

- W = Deleted (DE)

14 Support

Email: posttradeoperations@lme.com
Telephone: +44 (0) 20 7113 8201

Appendix A – FIX Tags

The Data Types used in the FIX tags descriptions are summarized below. For all tags the LMEsmart FIX API supports ASCII character range 32 – 122. Any characters used outside of this range may result in undesired formatting being returned

Data Type	Description	Example
Price	Floating point value representing a price quantity. May be up to 30 characters. If there are any trailing zeroes then this will be truncated in responses from LMEsmart to the member, e.g. 2000.20 would be sent back as 2000.2.	'1050.65'
Integer	A string representing a whole number. May be used for enumerated types but also for trade size for example	'1' or '17500'
Char	A single character usually used for an enumerated type	'A'
Date	Date only in 'YYYMMDD' format	'20160316'
LocalTimestamp	London local time – 'HHMMSS.sss'	'084412.437'
UTCTime	Formatted as 'HH:MM:SS.sss'	e.g. '08:23:12.013'
UTCTimestamp	Formatted as 'YYYYMMDD-HH:MM:SS.sss'	e.g. '20160316-08:23:12.013'
String(n)	String of length up to n. Where necessary, actual format details will be contained in the field description	'ZZZNN2016031600000056'

Tag	FieldName	Data Type	Description
1	*Account*	String (60)	Clearing Account Code - Mnemonic

Tag	FieldName	Data Type	Description
11	*ClOrdID*	String (30)	Format: Firm Mnemonic (3), Business Date (DDMMYY), User defined String (21) (Note this is just a suggested format to ensure uniqueness, but this format is not enforced by the system, the system merely checks ClOrdId is unqiue for the member on the current day) Execution reports from the LME may contain up to 35 characters due to additional characters are appended to the original ClOrdId. There are two cases where this applies: • Auto-generation of trade halves • System or GUI initiated cancellations There are three different auto-generation cases: • Client Trades – text "_AUTO" is appended • UNA Replacement – text "_REPL" is appended • Post auction filling – text "_AUCN" is appended (where N is a number indicating the fill number) There are three different cancellation cases: • GUI cancellation, single half – text "_SYS" is appended to a system created cancellation request id • GUI cancellation, multiple halves – text "_SYS_N" is appended to a system created cancellation request id where N is a number • System cancellation (at end of PTT Auction) – text "_SYS" is appended to the original client order id of the half being cancelled. Note that initiating trade halves will be cancelled at the end of the PTT Auction if the original matching trade half was cancelled, regardless of whether the initiating trade half is now paired with a directed bid/offer or not (if it has been paired the original half is cancelled and a new one created).
17	*ExecID*	String (16)	Unique identifier of a Trade Update message as assigned by LMEsmart A large integer which is16 digits long and zero filled. E.g. '0000000012345932'
31	*LastPx*	Price	Option Premium – Options and TAPOs
37	*OrderID*	String (14)	Unique identifier of a Trade Half as assigned by LMEsmart An integer which is 14 digits long and zero filled. E.g. '00000000837451'

Tag	FieldName	Data Type	Description
39	*OrdStatus*	Char	This is the trade state as recorded by the LMEsmart system. Values: 0 = Unmatched 2 = Matched 4 = Cancelled 7 = Pending Validation 8 = Rejected 9 = Pending Acceptance T = Abandoned V = Sent To Clearing W = Cleared Trade X = Paired P = PTT Auction
41	*OrigClOrdID*	String (30)	The value of *ClOrdID* (11) included in the original message – used in Trade Cancel requests. Format is as for *ClOrdID* (11)
42	*OrigTime*	UTCTimestamp	Time of message origination (News)
54	*Side*	Integer	Indicates Side. Values: 1 = Side 1 (by convention the buy side) 2 = Side 2 (by convention the sell side)
55	*Symbol*	String (12)	E.g. 'CAD' i.e. includes currency identifier with 2 char commodity code
58	*Text*	String (100)	Information related to the message in which this field is found. For example error text in a Logon rejection message or the reason for a New Trade being rejected in an Execution Report. Or the body of a news message
60	*TransactTime*	UTCTimestamp	The time that a message was generated
75	*TradeDate*	Date	'YYYYMMDD' – Local Market Date
102	CxlRejReason	Integer	A code specifying the reason for the Rejection of the original Cancel request. Values: 99 = Some other reason
103	OrdRejReason	Integer	A code specifying the reason for the Order rejection Required if ExecType (150) = 8 (Rejected) Values: 99 = Other
122	*OrigSendingTime*	UTCTimestamp	Required for messages sent as a result of a resend request
148	*Headline*	String (80)	Headline text in a news message which contains a brief summary of the news item being reported

Tag	FieldName	Data Type	Description
150	*ExecType*	Char	The reason the execution report was generated. Values: 0 = Accepted 2 = Matched 4 = Cancelled 8 = Rejected 9 = Pending Acceptance I = Status Response R = Price Conversion S = Clear Update T = Abandoned P = PTT Auction Triggered
167	*SecurityType*	String(12)	Indicates the contract type. Current supported values are: F – Future T – Traded Option A - TAPO
202	*StrikePrice*	Price	Required if instrument is an option or TAPO contract
371	*RefTagID*	Integer	The tag number of the FIX field being referenced
372	*RefMsgType*	String	The *MsgType* of the FIX message being referenced
373	*SessionRejectReason*	String	Code to identify the reason for the session level message rejection – see list below
379	*BusinessRejectRefID*	String	Specifies the business level 'ID' of the message being rejected and is required unless no such identifier was supplied in the message. E.G. *ClOrdID* or *OrigClOrdID*
380	*BusinessRejectReason*	String	Specifies the reason that the message was rejected based on the following values: 0 – Other 1 – Unknown ID 2 – Unknown security 3 – Unsupported message 4 – Application not available 5 – Conditionally required field not provided 6 – Not authorised
423	*PriceType*	Integer	Possible values: 0 = Current 2 = Historic

Tag	FieldName	Data Type	Description
447	*PartyIDSource*	Char	Values: D = LME Clearing Member E = PartyID is ISO country code G = Party ID is a MIC code I = 3 char broker code N = Party ID is a Client code
448	*PartyID*	String (128)	ID of Firm or Trader
452	*PartyRole*	Integer	Identifies the type or role of PartyID (448) e.g. Executing Broker: 1 = Executing Firm[a] 3 (if 447=N) = Client ID[b] 3 (if 447=P) = Client Short Code[b][c] 4 = Clearing Firm 7 = Entering Firm 11 = Order Origination Trader[a] 17 = Contra Firm[a] 24 = Client Code[b] 26 = Correspondent Broker[d] 35 = Liquidity Provider Identifier [d] 36 = Entering Trader[a] 60 = Introducing Broker (IB) 66 = Market Maker Identifier [d] 122 = Decision Maker Short Code[b][c] 300 = Investment Decision User ID[c] 301 = Execution Decision User ID[a][c] 302 = Investment Decision Country Code 303 = Execution Decision Country Code [a] mandatory role [b] Conditional role - required on both sides of a client trade if T4 booking model or on client side of a client trade in T2 booking model [c] Short Code – 8 byte integer [d] Only present in execution reports from LMEsmart if from the LMEselect venue.
453	*NoPartyIDs*	Integer	Number of parties listed in Parties block
461	*CFICode*	String (8)	Possible values: ‘FCEPS’ – Physically settled futures contracts ‘FFICS’ – Index Futures ‘OPAFPS’ – Metal future put option ‘OCAFPS’ – Metal future call option ‘OPEFPS’ – Precious future put option (European) ‘OCEFPS’ – Precious future call option (European) ‘OPXTCS’ – Metal TAPO Put ‘OCXTCS’ – Metal TAPO call ‘OPEICS’ – Index put option ‘OCEICS’ – Index call option ‘FCECS’ – Cash settled futures contracts (e.g. LMEmini, MAF and Steel scrap/rebar

Tag	FieldName	Data Type	Description
541	*MaturityDate*	Date	Required if *PromptType* = 'S'. Will be populated by LMEsmart with the absolute date for rolling prompts (i.e. *PromptType*= 'R') Represents: Prompt date for Futures Expiration date for Options and TAPOs Format : 'YYYYMMDD'
555	*NoLegs*	Integer	Number of trade legs attached to the trade half (1 – 124). Set to '1' for outright trades and options. Greater than '1' for carry trades
581	*AccountType*	Integer	One of: 1 = ISA 2 = House 3 = Omnibus 7 = Gross OSA 9 = Unallocated House 90 = Unallocated Client
584	*MassStatusReqID*	String (20)	Unique ID of mass status request as assigned by the requestor. In order to guarantee uniqueness the string should be prefaced with a source identifier and the current date. This ID should be zero filled or otherwise filled to the full 20 chars. E.g. – 'ZZZNNYYYYMMDD0000006' Where 'ZZZ' is member code NN' member instance identifier e.g. '01' 'YYYYMMDD' is the current date
585	*MassStatusReq Type*	Integer	Values: 1 = History (up to 5 business days) 2 = Current (Today)
624	*LegSide*	Integer	Valid values: 1 = Buy 2 = Sell
637	*LegLastPx*	Price	Actual price for a trade leg
653	*SecDefStatus*	Integer	Status of the security described by the Instrument block. Values: 0 = Active 1 = Inactive
820	*TradeLinkID*	String(35)	To satisfy Transaction Reporting obligations, members will be required to populate a trade reference across multiple trade halves. This is a freeform reference.

Tag	FieldName	Data Type	Description
880	*TrdMatchId*	String (8)	The Matching sequence number. Same for two halves of a matched trade and unique within the day. See 6.3 for more details
897	*NoTrades*	Integer	The number of single trade half descriptions to follow. Usually 1 but may be any value up to 64
943	*TimeBracket*	String (2)	Values: R1 – Ring One R2 – Ring Two R3 – Ring Three R4 – Ring Four K1 – Kerb One K2 – Kerb Two C1 – Basis Ring One C2 – Basis Ring Two C3 – Basis Ring Three C4 – Basis Ring Four D1 – Basis Kerb One D2 – Basis Kerb Two
1057	*AggressorIndicator*	String (1)	Represents whether the trade was an aggressor in the electronic order book. Is only populated if the trade originated from the LMEselect system. Values: • Y • N
1301	*MarketId*	String (3)	Indicates the market the trade is for. Values: • 'LME' = LME Base Metals • 'LPM' = LME Precious Metals
1903	*RegulatoryTradeId*	String(21)	Trading Venue Transaction Identification Code (TVTIC) If Origin is Select then this is the Select Trade Id (5940) If Origin is not Select then this is the Matching Reference Number (5935)
5179	*TradeTime*	UTCTime	Time at which trade was agreed between counterparties
5322	*FirmID*	String (3)	Member firm's three character mnemonic Used in Member Definition message
5364	*MemberName*	String (30)	Used in Member Definition message

Tag	FieldName	Data Type	Description
5365	*MemberAddress*	String (100)	Used in Member Definition message
5366	*ContactFax*	String (30)	Used in Member Definition message
5367	*AltPhone1*	String (30)	Used in Member Definition message
5368	*AltPhone2*	String (30)	Used in Member Definition message
5440	*ClearingStatus*	Char	Indicates the clearing status of Trades that have been matched and sent to clearing. Values: 2 = Cleared Y = Rejected X = Paired W = Cancelled
5442	*MatchingSlipID*	String (8)	The slip id to identify this leg of the Trade half as allocated by LMEsmart. See 6.3 for more details
5446	*RequestType*	Integer	Values: 1 = Contracts 2 = Members
5447	*RequestID*	String (20)	Matching client specified Request identifier of either: Download request RIB Trade Module Request In order to guarantee uniqueness the string should be prefaced with a source identifier and the current date. This ID should be zero filled or otherwise filled to the full 20 chars. E.g. – 'ZZZNNYYYYMMDD0000006' Where 'ZZZ' is member code NN' member instance identifier e.g. '01' 'YYYYMMDD' is the current date
5448	*NumMsg*	Integer	Number of results messages to follow this response. Set to '0' in the case that the request has been rejected (*ReqResponseStatus* = 2)
5449	*ReqResponseTo*	Integer	Type of request being responded to: 1 = Trade Mass Status Request 2 = Download Request

Tag	FieldName	Data Type	Description
5469	*ReqResponse Status*	Integer	Status of the client request (which can be a trade mass status request, download request or a request to accept/reject a RIB trade). Values: 1 = Accepted 2 = Rejected
5474	*AbbreviatedPrice*	String (12)	Contract Price. Price of the leg can be expressed as: (a) A price code expressions (e.g. S + 10 which means settlement price plus ten). Valid price codes are S, C, MC, M3 & B (basis) as well as TAS codes TA, TB, TC … TZ. This is only valid for futures. (b) A differential (e.g. -10 which means ten units lower than the price of the first leg). A valid differential is a number prefixed with either (+) or (-).
5476	*PrivateReference*	String (80)	Free form text up to 80 chars If the execution report is a fill of a directed bid/offer on a PTT Auctions then this is: AuctionId-ClientOrderIdOfBidOffer-FillNumber E.g. 20200225000001-ABC20022500000001234-1
5477	*PublicReference*	String (12)	First 5 chars are used for trade matching and are not case sensitive If the execution report is a directed bid/offer on a PTT Auction then this is the Auction Id (which is in the format YYMMDDNNNNNN, e.g. 200225000001).
5507	*TrdMatchTime*	UTCTimestamp	LMEsmart timestamp for the match. Does not change once it has been created at the time of the match
5678	*Volatility*	Price	Required for Options or TAPO

Tag	FieldName	Data Type	Description
5681	*ExchangeTrade Type*	Integer	Exchange defined type of trade. Valid values: 0 – NORMAL 2 – GIVE-UP EXECUTOR 3 – TRANSFER 8 – EXCEPTION - REPORTABLE 9 – OTC BRING ON 10 – OTC TAKE OFF 14 – GIVE-UP CLEARER 15 – FINANCING 16 – EXCEPTION NON-REPORTABLE
5934	*ClearingRefNo*	String (16)	Clearing reference number – required for trades updated by Clearing system. Unique clearing number assigned to the Trade by Clearing (same for both Trade halves of a Matched Trade).
5935	*MatchingRefNo*	String (16)	Same as *TrdMatchId* (880) but preceded by date to guarantee uniqueness i.e. format – 'YYYYMMDDnnnnnnnn'. See 6.3 for more details
5939	*SelectOrder Number*	String (22)	e.g. '23138010000000020'
6396	*RejectCode*	String (3)	See Appendix B
7931	*VenueID*	Integer	Venue of execution 0 – Inter-office 1 – Ring 2 – Select 3 – Basis Ring
9675	*ContactPhone Number*	String (20)	Member's contact phone number
9677	*ContactEmail Address*	String (50)	Member's contact email address
10000	*MaturityRolling Prompt*	String (10)	Required if *PromptType* = 'R'. Valid values: TOM = Tomorrow C = Cash 3M = 3 Months
10003	*LegLastQty*	Integer	Trade Quantity of this leg
10004	*PromptType*	Char	Values: S = Single Prompt Date R = Rolling Prompt Date
10010	*NoOfInstrument Legs*	Integer	Used in Instrument block to specify the number of leg details included. Introduces a repeating group of fields

Tag	FieldName	Data Type	Description
10051	*TradingCapacity*	String (4)	Values allowed: 'DEAL' – Dealing on own account 'MTCH' – Matched Principal 'AOTC' – Any Other Capacity
10052	*CountryOfBranchOf Client*	String(2)	Two letter ISO country code, e.g. GB
10054	*StrategyLinked OrderId*	String(50)	For LMEselect trades only
5940	*SelectTrade Number*	String (21)	e.g. '6023138010000000009'
5941	SelectImpliedTrade Number	String (21)	e.g. '6023138010000000009'
20004	*InstrumentLegNo*	Integer	Indicates which leg of a carry. Used in instrument block
20005	*LegInstrument*	Integer	Indicates which leg of a carry. Used in Trade capture/ Execution reports – to avoid replication?(of 20004)
20008	*UniqueProductID*	String	ESMA Unique Product Identifier
20020	*ComplexTrade ComponentID*	String (35)	Client's unique strategy identifier - only required if multiple trade halves submitted i.e. NoTrades (897) is > 1. See 6.3 for more details
20021	*StrategyID*	String (14)	LMEsmart specified identifier of the Trade Strategy. Only required if multiple trade halves submitted i.e. NoTrades (897) is > 1. See 6.3 for more details
20022	*StrategyMatchId*	String (14)	Unique Strategy match ID for the day assigned by LMEsmart. The same id is used for both Strategy halves. Required if *ExecType* = 2 (Matched) and for all post match reports. See 6.3 for more details
20023	*Commodity DerivativeIndicator*	Char	Nature of associated risk. Values: 0 – Trade is risk reducing 1 – Trade is NOT risk reducing
20025	*StrategyClOrdId*	String (14)	Client's unique strategy identifier - only required if multiple trade halves submitted i.e. NoTrades (897) is > 1. Value should be unique Should be prefixed with firm Mnemonic and business date to guarantee uniqueness
20030	*TradedPrice*	Price	Actual price for a trade leg before any commission or other costs

Tag	FieldName	Data Type	Description
20031	*TradedPremium*	Price	Option Premium for Options and TAPOs before any commission or other costs
20032	*CancellationFlag*	Char	Either: C – Correction R – Reversal If this is provided then CancelLinkId must also be provided in each leg.
20033	*CancelLinkId*	String (8)	Should only be provided if the CancellationFlag is set to C or R. The CancelLinkId is the matching reference number (5935) from the original Trade
20038	*TradeModuleId*	String (20)	Identifier for a RIB entered trade module. Multiple TERs can have the same id. E.g. 1-20180420-00000001-2
20039	*Acceptance Status*	Integer	Used in conjunction with accepting and rejected RIB trade modules. 1 = Accept 2 = Reject
20040	*RIB Acceptance Source*	Char(1)	This indicates the source of the acceptance or rejection of the RIB trade: M = Member FIX (API) W = Member Web (GUI)

Appendix B – Error Codes

If a submitted trade or trades is rejected then LMEsmart will return a generic “99” (other error) error code in *OrdRejReason (103).* In the case of a rejected cancel trade request the generic “99” (other error) error code will be returned in *CxlRejReason (102)*.
The *Text* (58) tag will contain a more detailed error message which includes an LMEsmart error code. The possible LMEsmart error codes that can be encountered in *Text (58)* in these circumstances are listed below.

Note that where the error text is “XXX invalid” (e.g. Product Invalid), this can include cases where the item itself is valid in its own right, but is invalid in the context it has been used in (if there is not a specific error code for that context). E.g. a Product Code of “CA” is a valid Product Code but might not be valid when used in combination with a specific Contract Type and Currency.

Error Code	Error Text
Invalid Fields	
1001	Product invalid
1002	Currency invalid
1003	Contract Type invalid
1004	Prompt Date invalid.
1005	Member invalid
1006	Clearing Member invalid
1007	Counterparty invalid
1008	User invalid
1010	Session code invalid
1011	Trade Category invalid
1012	Price Type invalid
1013	Venue invalid
1014	Market invalid
1015	BuySell invalid
1016	Volume invalid
1017	Price invalid
1018	Trade Date invalid
1019	Trade Time invalid
1020	Transaction Time invalid
1021	CashPhysical invalid
1022	CFICode invalid
1023	Trading Capacity invalid
1024	Account Type invalid
1025	Client Code invalid
1026	Public Reference invalid

Error Code	Error Text
1027	Private Reference invalid
1028	Complex Trade Component Id invalid
1029	CallPut invalid
1030	Strike Price invalid
1031	Premium invalid
1032	Volatility invalid
1034	Client Order Id invalid
1035	Underlying Futures Price invalid
1036	Account Code invalid
1037	Commodity Derivative Indicator invalid
1038	Investment Decision within firm invalid
1039	Country of investment decision invalid
1040	Execution Decision within firm invalid
1041	Country of execution decision invalid
1042	Currency not valid for Product
1043	Client Id invalid
1045	Unique Product Id invalid
1046	Traded Price Invalid
1047	Traded Premium Invalid
1048	Trade Link Id invalid
1049	Cancellation Flag invalid
1050	Cancel Link Id invalid
1051	Client Short Code invalid
1052	Decision Maker Short Code invalid
Mandatory Fields	
1101	Mandatory Field is missing: Trade Date

Error Code	Error Text
1102	Mandatory Field is missing: Member
1103	Mandatory Field is missing: Counterparty
1104	Mandatory Field is missing: Contract Code
1105	Mandatory Field is missing: Contract Type
1106	Mandatory Field is missing: Trade Category
1107	Mandatory Field is missing: Price Type
1108	Mandatory Field is missing: Venue
1109	Mandatory Field is missing: Market
1110	Mandatory Field is missing: BuySell
1112	Mandatory Field is missing: Prompt Date
1113	Mandatory Field is missing: Volume
1114	Mandatory Field is missing: Price or Price Code
1115	Mandatory Field is missing: Time or Session Code
1116	Mandatory Field is missing: Client Order Id
1117	Mandatory Field is missing: Premium
1118	Mandatory Field is missing: Strike Price
1119	Mandatory Field is missing: CallPut
1120	Mandatory Field is missing: Volatility
1121	Mandatory Field is missing: Underlying futures price
1122	Mandatory Field is missing: Option exercise type
1123	Mandatory Field is missing: Public Reference
1124	Mandatory Field is missing: Cash/Physical
1125	Mandatory Field is missing: Settlement Date
1126	Mandatory Field is missing: Trading Capacity
1127	Mandatory Field is missing: Commodity Derivative Indicator
1128	Mandatory Field is missing: Investment Decision within firm

Error Code	Error Text
1129	Mandatory Field is missing: Country of investment decision
1130	Mandatory Field is missing: Execution Decision within firm
1131	Mandatory Field is missing: Country of execution decision
1132	Mandatory Field is missing: Account Code
1133	Mandatory Field is missing: Client Code
1134	Mandatory Field is missing: Client Id
1135	Mandatory Field is missing: Account Type
1136	Mandatory Field is missing: Select Trader Id
1137	Mandatory Field is missing: Select Trade Id
1138	Mandatory Field is missing: Select Order Id
1139	Mandatory Field is missing: Select Client Id
1140	Mandatory Field is missing: Green Trade Flag
1141	Mandatory Field is missing: Aggressor
1142	Mandatory Field is missing: Clearing Member
1143	Mandatory Field is missing: Entering Trader
1144	Client account fields missing for client trade
1145	Client account fields present for exchange trade
Permissions	
1151	Member does not have permission on market
1152	Member does not have permission on currency
1153	Member is disabled
1154	Counterparty is disabled
1155	User is disabled
1156	Contract is disabled
1157	Cat3 member cannot enter a client cross
1163	Market is disabled

Error Code	Error Text
1164	Combination of contract and contract type is not available for the given venue and market
LMEprecious	
1180	Volatility is prohibited from entry on LMEprecious contracts
1181	Underlying Futures Price is prohibited from entry on LMEprecious contracts
LMEmercury	
1190	Rejected by clearing - invalid party information
1191	Rejected by clearing – unknown instrument
1192	Rejected by clearing – unauthorized report trades
1193	Rejected by clearing - invalid trade type
1194	Rejected by clearing - other
Price Codes	
1201	More than one price code used
1202	Price code used in leg other than first of a carry
1203	Leg contains price code other than a differential.
1204	Price code C is present but not in all legs
1205	Price code V is present but not in all legs
1206	Price Code V is used with an invalid Prompt Date or a Prompt Date Code.
1207	Price Code YS is used with an invalid Prompt Date or a Prompt Date Code.
1208	Price Code S is used with an invalid Prompt Date or a Prompt Date Code
1211	Price Code entered with a Trade Date not equal to current day.
1212	Price Code of B entered on a trade that is not a carry
1213	Price Code YS used with no absolute price available
1214	Price Code V used with no absolute price available

Error Code	Error Text
1215	Price invalid for contract
1217	Price Code invalid for market
1219	Both price and price code are used
1220	A price code has been used for an invalid contract and venue combination
1221	Price Code used with an invalid Prompt Date or a Prompt Date Code
1222	A TAS Price code is present but not in all legs
Dates/Times	
1230	Trade date is in the future
1231	Trade date is a holiday
1232	Trade time is in the future
1233	Trading deadline has passed
1234	Matching deadline has passed
1236	Invalid combination of venue and session code is used
1237	A trade is entered with both a Session Code and a Trade Time.
1238	Prompt Date Short Codes are used with a Trade Date not equal to current day
1239	A prompt date is duplicated in at least two legs of a single Carry.
1240	A trade time has been entered for a venue that uses session codes
General	
1251	System is not in a state that allows trade submission
1252	Order reference already exists.
1253	The volume of a single trade is greater than the Max Quantity
1254	A futures outright or Carry contains a Price equal to zero.
1255	Incorrect Tick Size is entered for the relevant contract
1256	Total buy volume does not equal total sell for all legs of a carry.
1257	The number of legs in a carry exceeds 124

Error Code	Error Text
1258	Invalid contract type for carry
1259	Strike Price entered does not conform to Strike Price gradations.
1260	A client trade cannot be entered with a trade category of Give Up Executor
1261	A client cross cannot be entered as a ring trade
1262	Ring trades can only be entered by Category 1 members
1263	Basis ring trades need to involve at least one Cat 1 member
1264	An exchange trade cannot have the category of OTC Take Off
1265	Price Banding threshold broken
1266	The volume of a single trade is less than the Min Quantity
1269	Trade half cancellation request for unknown trade half or trade half in incorrect state
1270	Trade half rejection request for unknown trade half or trade half in incorrect state
1271	Trade half received with member that is not the same as logged in member
1272	Strategy cancellation request for unknown strategy
1273	Invalid combination of price type, venue and trade category
1274	Number of trades in a strategy exceeds the max allowed
1275	A currency is used that is not appropriate for the venue
1276	A T4 exchange trade cannot have a client account type
1277	Trade date is not valid for trade category and price type
1278	Trade half rejected by counterparty
1279	Market is not in a state that allows trade submission
1280	Member UNA used with invalid trade category
1281	Counterparty UNA used with invalid trade category
1282	Account type U used with invalid trade category
1283	Member UNA not allowed for trades that are not auto-generated

Error Code	Error Text
1284	Invalid Account Reference Data
1285	Invalid combination of PartyRole and PartyIdSource in parties block
1287	IB trade rejected by GCM
1301	Cancellation Flag provided without Cancel Link Id
FIX Specific	
1290	User has exceeded the number of trade mass status requests allowed in a day
1291	Failed to parse incoming FIX message
1292	System is not currently open for download or status requests
1293	Unsupported message type submitted
1294	Invalid query request received
System	
1300	System Error
1301	Cancellation Flag provided without Cancel Link Id
1302	A party role was provided that is not allowed on inbound messages
PTT Auction	
1054	Auction id is invalid or for a closed auction
1071	Auction Id must be provided if counterparty is AUC
1072	Trade is below PTT LIS Thresholds but outside auction times
1073	PTT Auction entry must match details of the auction
1075	Buy Sell distribution in auction entry is not compatible with auction
1076	Carry volume in auction entry is not compatible with auction

Appendix C – Deviations from FIX Protocol

Changed Message Definitions

The following message structures do not adhere to the FIX 4.4. protocol standards:

Type	Name	Change To Definition
E	New Trades List	FIX title is 'New Order List' May contain multiple trade halves and not just multiple trade half legs. NoTrades (897) used to indicate the number of trade halves included
F	Cancel Trade	FIX title is 'Order Cancel Request' Includes optional non-standard field-*ComplexTradeComponentID* – to allow multiple trade halves that were entered together to be cancelled together
8	Trade Execution Report	May include multiple related trade halves in a single report. Includes use of a non-mandatory tag (ClOrdId, 11) as the first tag in a repeating block. The repeating block is the trades block signified by the NoTrades (897) tag.
9	Trade Cancel Reject	FIX title is 'Order Cancel Request' Includes optional non-standard field-*ComplexTradeComponentID* – to allow a Cancel request for multiple related trade halves to be rejected
AF	Trade Mass Status Request	FIX title is 'Order Mass Status Request' Includes optional non-standard field-*ComplexTradeComponentID* – to allow a request to retrieve a set of related trade halves.
cm2	Download Request	Not a standard FIX message type
cm3	Download Request Response	Not a standard FIX message type
cm4	Member Definition	Not a standard FIX message type

Changed Component Blocks

The following component blocks do not adhere to the FIX 4.4. protocol standards:

Name	Change To Definition
Instrument	Includes LME specific fields e.g. *PromptType* (10004)

User Defined Tags & Changed Tag Definitions

Fields with a Tag ID in the range 10000 to 20000 are specified as user defined fields. According to the official FIX standard this range is still reserved and the range 20000 and above should be utilised for user defined fields. The range 10000 to 20000 has been used by LMEsmart in order to conform to other LME interfaces that also utilise this range. See Appendix A for definitions of these user defined fields.

Appendix D - Message Examples

The following examples do not show the FIX session protocol fields but just shows a selected set of the business level fields.

Example 1 - trade half submitted (to matching)

Member ABC buys 20 lots of CA $4935.45 for 17/06/2016 from Counterparty XYZ

Tag Id			Name	Example	Comments
Standard Header					MsgType (35) = E
453			*NoPartyIDs*	8	There are 8 parties to the trade
->	447		*PartyIDSource*	D	D = Clearing Member
->	448		*PartyID*	ABC	Executing firm code
->	452		*PartyRole*	1	1 = Executing Firm
->	447		*PartyIDSource*	D	D = Clearing Member
->	448		PartyID	XYZ	Contra firm code
->	452		*PartyRole*	17	17 = Contra Firm
->	447		*PartyIDSource*	N	N = Client code
->	448		*PartyID*	JBLOGGS	Order origination trader name
->	452		*PartyRole*	11	11 = Order Origination Trader
->	447		*PartyIDSource*	N	N = Client code
->	448		*PartyID*	RSMITH	Entering trader name
->	452		*PartyRole*	36	36 = Entering Trader
->	447		*PartyIDSource*	P	P = Short Code
->	448		*PartyID*	78963258	Short Code
->	452		*PartyRole*	300	300 = Investment Decision ID
->	447		*PartyIDSource*	E	E = ISO Country Code
->	448		*PartyID*	GB	ISO Country Code
->	452		*PartyRole*	302	302 = Investment Decision Country Code
->	447		*PartyIDSource*	P	P = Short Code
->	448		*PartyID*	78963259	Short Code
->	452		*PartyRole*	301	301 = Execution Decision ID
->	447		*PartyIDSource*	E	E = ISO Country Code
->	448		*PartyID*	GB	ISO Country Code
->	452		*PartyRole*	303	303 = Execution Decision Country Code
581			*Account Type*	2	2 = House
5681			*ExchangeTrade Type*	0	0 = Normal
7931			*Venue ID*	0	0 = Inter-office
1301			*MarketID*	LME	LME = LME Base Metals
5179			*TradeTime*	10:52:19.123	
75			*TradeDate*	20160412	
20023			*CommodityDerivativeIndicator*	1	1 – Trade is not risk reducing
897			*NoTrades*	1	
->	11		*ClientOrderID*	ABC2016031 7000000123	
->	55		*Symbol*	CAD	
->	167		*SecurityType*	F	F = Future
->	461		*CFI Code*	FCEPS	
->	10010		*NoOfInstrumentLegs*	1	
	->	20004	*InstrumentLegNo*	1	
	->	10004	*PromptType*	S	S = Single Prompt Date
	->	541	*MaturityDate*	20160617	

->	10051		*TradingCapacity*	DEAL	DEAL – Dealing on own account
->	423		*PriceType*	0	0 = Current
->	54		*Side*	1	1 = buy in first leg
->	555		*NoLegs*	1	
	->	20005	*LegInstrument*	1	
	->	624	*LegSide*	1	1 = Buy
	->	10003	*LegLastQty*	20	
	->	637	*LegLastPx*	4935.45	
60			*TransactTime*	20160412-15:12:22.456	
Standard Trailer					

Example 2 – trade half accepted (from matching)

Trade Half Execution Report sent to member ABC acknowledging acceptance of trade half submission

Tag Id		Name	Example	Comments
Standard Header				***MsgType (35) = 8***
17		*ExecId*	00000000 2345932	Message Id generated by matching
150		*ExecType*	0	0 = Accepted
39		*OrdStatus*	0	0 = Unmatched
453		*NoPartyIDs*	8	There are 8 parties to the trade
->	447	*PartyIDSource*	D	D = Clearing Member
->	448	*PartyID*	ABC	Executing firm code
->	452	*PartyRole*	1	1 = Executing Firm
->	447	*PartyIDSource*	D	D = Clearing Member
->	448	*PartyID*	XYZ	Contra firm code
->	452	*PartyRole*	17	17 = Contra Firm
->	447	*PartyIDSource*	N	N = Client code
->	448	*PartyID*	JBLOGGS	Order origination trader name
->	452	*PartyRole*	11	11 = Order Origination Trader
->	447	*PartyIDSource*	N	N = Client code
->	448	*PartyID*	RSMITH	Entering trader name
->	452	*PartyRole*	36	36 = Entering Trader
->	447	*PartyIDSource*	P	P = Short Code
->	448	*PartyID*	78963258	Short Code
->	452	*PartyRole*	300	300 = Investment Decision ID
->	447	*PartyIDSource*	E	E = ISO Country Code
->	448	*PartyID*	GB	ISO Country Code
->	452	*PartyRole*	302	302 = Investment Decision Country Code
->	447	*PartyIDSource*	P	P = Short Code
->	448	*PartyID*	78963259	Short Code
->	452	*PartyRole*	301	301 = Execution Decision ID
->	447	*PartyIDSource*	E	E = ISO Country Code
->	448	*PartyID*	GB	ISO Country Code
->	452	*PartyRole*	303	303 = Execution Decision Country Code
581		*Account Type*	2	2 = House
5681		*ExchangeTrade Type*	0	0 = Normal
7931		*Venue ID*	0	0 = Inter-office
1301		*MarketID*	LME	L = LME Base Metals
5179		*TradeTime*	10:52:19.123	
75		*TradeDate*	20160412	
20023		*CommodityDerivativeIndicator*	1	1 – Trade is not risk reducing
897		*NoTrades*	1	
->	11	*ClientOrderID*	ABC2016031 7000000123	
->	37	*OrderID*	00000000 00002341	Trade Half Id generated by matching

->	55		*Symbol*	CAD	
->	167		*SecurityType*	F	F = Future
->	461		*CFI Code*	FCEPS	
->	10010		*NoOfInstrumentLegs*	1	
	->	20004	*InstrumentLegNo*	1	
	->	10004	*PromptType*	S	S = Single Prompt Date
	->	541	*MaturityDate*	20160617	
->	423		*PriceType*	0	0 = Current
->	10051		*TradingCapacity*	DEAL	DEAL – Dealing on own account
->	54		*Side*	1	1 = buy in first leg
->	555		*NoLegs*	1	
	->	20005	*LegInstrument*	1	
	->	624	*LegSide*	1	1 = Buy
	->	10003	*LegLastQty*	20	
	->	637	*LegLastPx*	4935.45	
60			*TransactTime*	20160412-15:12:32.256	
Standard Trailer					

Example 3 – trade half alleged to counterparty (from matching)

Trade Half Execution Report sent to member XYZ notifying them that a trade half has been alleged to them (by ABC). This contains the same details as the original side from ABC (see example 2), but omits any client and account related information if present.

Tag Id			Name	Example	Comments
Standard Header					MsgType (35) = 8
17			*ExecId*	00000000 2345933	Message Id generated by matching
150			*ExecType*	0	0 = Accepted
39			*OrdStatus*	0	0 = Unmatched
453			*NoPartyIDs*	2	There are 2 parties in the message
->	447		*PartyIDSource*	D	D = Clearing Member
->	448		*PartyID*	ABC	Entering firm code
->	452		*PartyRole*	1	1 = Executing Firm
->	447		*PartyIDSource*	D	D = Clearing Member
->	448		*PartyID*	XYZ	Contra firm code
->	452		*PartyRole*	17	17 = Contra Firm
5681			*ExchangeTrade Type*	0	0 = Normal
7931			*Venue ID*	0	0 = Inter-office
1301			*MarketID*	LME	LME = LME Base Metals
897			*NoTrades*	1	
->	11		*ClientOrderID*	ABC2016031 7000000123	
->	37		*OrderID*	00000000 00002341	Trade Half Id generated by matching for the origiinal side of the trade
->	55		*Symbol*	CAD	
->	167		*SecurityType*	F	F = Future
->	461		*CFI Code*	FCEPS	
->	10010		*NoOfInstrumentLegs*	1	
	->	20004	*InstrumentLegNo*	1	
	->	10004	*PromptType*	S	S = Single Prompt Date
	->	541	*MaturityDate*	20160617	
->	423		*PriceType*	0	0 = Current
->	54		*Side*	1	1 = buy in first leg
->	555		*NoLegs*	1	
	->	20005	*LegInstrument*	1	
	->	624	*LegSide*	1	1 = Buy
	->	10003	*LegLastQty*	20	
	->	637	*LegLastPx*	4935.45	
60			*TransactTime*	20160412-15:12:32.956	
Standard Trailer					

Example 4 – trade matched (from matching)

Trade Half Execution Report sent to member ABC notifying them that trade has been matched. This contains all the original detail of the trade half.

Tag Id		Name	Example	Comments
Standard Header				**MsgType (35) = 8**
17		*ExecId*	00000000 2345934	Message Id generated by matching
150		*ExecType*	2	2 = Matched
39		*OrdStatus*	2	2 = Matched
5507		*TrdMatchTime*	20160412-15:15:22.123	
453		*NoPartyIDs*	8	There are 8 parties to the trade
->	447	*PartyIDSource*	D	D = Clearing Member
->	448	*PartyID*	ABC	Executing firm code
->	452	*PartyRole*	1	1 = Executing Firm
->	447	*PartyIDSource*	D	D = Clearing Member
->	448	PartyID	XYZ	Contra firm code
->	452	*PartyRole*	17	17 = Contra Firm
->	447	*PartyIDSource*	N	N = Client code
->	448	*PartyID*	JBLOGGS	Order origination trader name
->	452	*PartyRole*	11	11 = Order Origination Trader
->	447	*PartyIDSource*	N	N = Client code
->	448	*PartyID*	RSMITH	Entering trader name
->	452	*PartyRole*	36	36 = Entering Trader
->	447	*PartyIDSource*	P	P = Short Code
->	448	*PartyID*	78963258	Short Code
->	452	*PartyRole*	300	300 = Investment Decision ID
->	447	*PartyIDSource*	E	E = ISO Country Code
->	448	*PartyID*	GB	ISO Country Code
->	452	*PartyRole*	302	302 = Investment Decision Country Code
->	447	*PartyIDSource*	P	P = Short Code
->	448	*PartyID*	78963259	Short Code
->	452	*PartyRole*	301	301 = Execution Decision ID
->	447	*PartyIDSource*	E	E = ISO Country Code
->	448	*PartyID*	GB	ISO Country Code
->	452	*PartyRole*	303	303 = Execution Decision Country Code
581		*Account Type*	2	2 = House
5681		*ExchangeTrade Type*	0	0 = Normal
7931		*Venue ID*	0	0 = Inter-office
1301		*MarketID*	LME	LME = LME Base Metals
5179		*TradeTime*	10:52:19.123	
75		*TradeDate*	20160412	
20023		*CommodityDerivativeIndicator*	1	1 – Trade is not risk reducing
897		*NoTrades*	1	
->	11	*ClientOrderID*	ABC2016031 7000000123	
->	37	*OrderID*	00000000 00002341	Trade Half Id generated by matching

->	55		*Symbol*	CAD	
->	167		*SecurityType*	F	F = Future
->	461		*CFI Code*	FCEPS	
->	10010		*NoOfInstrumentLegs*	1	
	->	20004	*InstrumentLegNo*	1	
	->	10004	*PromptType*	S	S = Single Prompt Date
	->	541	*MaturityDate*	20160617	
->	423		*PriceType*	0	0 = Current
->	10051		*TradingCapacity*	DEAL	DEAL – Dealing on own account
->	54		*Side*	1	1 = buy in first leg
->	880		*TrdMatchId*	10001001	Match sequence num for trade generated by matching (unique today)
->	5935		*MatchingRefNo*	2016041210001001	Match ref number for trade generated by matching (unique across all days)
->	1903		*RegulatoryTradeId*	2016041210001001	TVTIC
->	555		*NoLegs*	1	
	->	20005	*LegInstrument*	1	
	->	624	*LegSide*	1	1 = Buy
	->	10003	*LegLastQty*	20	
	->	637	*LegLastPx*	4935.45	
	->	5442	*MatchingSlipId*	00005431	Slip Id (leg id) generated by matching
60			*TransactTime*	20160412-15:12:33.234	
Standard Trailer					

Example 5 – trade cleared (from matching)

Trade Half Execution Report sent to member ABC notifying them that the trade has been cleared. This contains all the original detail of the trade half.

Tag Id		Name	Example	Comments
Standard Header				**MsgType (35) = 8**
17		*ExecId*	00000000 2345935	Message Id from matching
150		*ExecType*	S	S = Clear Update
39		*OrdStatus*	W	W = Cleared Trade
5440		*ClearingStatus*	2	2 = Cleared
453		*NoPartyIDs*	8	There are 8 parties to the trade
->	447	*PartyIDSource*	D	D = Clearing Member
->	448	*PartyID*	ABC	Executing firm code
->	452	*PartyRole*	1	1 = Executing Firm
->	447	*PartyIDSource*	D	D = Clearing Member
->	448	PartyID	XYZ	Contra firm code
->	452	*PartyRole*	17	17 = Contra Firm
->	447	*PartyIDSource*	N	N = Client code
->	448	*PartyID*	JBLOGGS	Order origination trader name
->	452	*PartyRole*	11	11 = Order Origination Trader
->	447	*PartyIDSource*	N	N = Client code
->	448	*PartyID*	RSMITH	Entering trader name
->	452	*PartyRole*	36	36 = Entering Trader
->	447	*PartyIDSource*	P	P = Short Code
->	448	*PartyID*	78963258	Short Code
->	452	*PartyRole*	300	300 = Investment Decision ID
->	447	*PartyIDSource*	E	E = ISO Country Code
->	448	*PartyID*	GB	ISO Country Code
->	452	*PartyRole*	302	302 = Investment Decision Country Code
->	447	*PartyIDSource*	P	P = Short Code
->	448	*PartyID*	78963259	Short Code
->	452	*PartyRole*	301	301 = Execution Decision ID
->	447	*PartyIDSource*	E	E = ISO Country Code
->	448	*PartyID*	GB	ISO Country Code
->	452	*PartyRole*	303	303 = Execution Decision Country Code
581		*Account Type*	2	2 = House
5681		*ExchangeTrade Type*	0	0 = Normal
7931		*Venue ID*	0	0 = Inter-office
1301		*MarketID*	LME	LME = LME Base Metals
5179		*TradeTime*	10:52:19.123	
75		*TradeDate*	20160412	
20023		*CommodityDerivativeIndicator*	1	1 – Trade is not risk reducing
897		*NoTrades*	1	
->	11	*ClientOrderID*	ABC2016031 7000000123	
->	37	*OrderID*	00000000 00002341	Trade Half Id generated by matching
->	55	*Symbol*	CAD	
->	167	*SecurityType*	F	F = Future

->	461		*CFI Code*	FCEPS	
->	10010		*NoOfInstrumentLegs*	1	
	->	20004	*InstrumentLegNo*	1	
	->	10004	*PromptType*	S	S = Single Prompt Date
	->	541	*MaturityDate*	20160617	
->	423		*PriceType*	0	0 = Current
->	10051		*TradingCapacity*	DEAL	DEAL – Dealing on own account
->	54		*Side*	1	1 = buy in first leg
->	880		*TrdMatchId*	10001001	Match sequence num for trade generated by matching (unique today)
->	5935		*MatchingRefNo*	2016041210001001	Match ref number for trade generated by matching (unique across all days)
->	5934		*ClearingRefNo*	0000000 12345678	Clearing ref num assigned by clearer
->	1903		*RegulatoryTradeId*	2016041210001001	TVTIC
->	555		*NoLegs*	1	
	->	20005	*LegInstrument*	1	
	->	624	*LegSide*	1	1 = Buy
	->	10003	*LegLastQty*	20	
	->	637	*LegLastPx*	4935.45	
	->	5442	*MatchingSlipId*	00005431	Slip Id (leg id) generated by matching
60			*TransactTime*	20160412-15:12:45.678	
Standard Trailer					

Example 6 – trade rejected (from matching)

Trade Half Execution Report sent to member ABC notifying them that a trade has been rejected. This contains all the original detail of the trade half along with the reason for rejection.

Tag Id		Name	Example	Comments
Standard Header				MsgType (35) = 8
17		*ExecId*	00000000 2345936	Message Id generated by matching
150		*ExecType*	8	8 = Rejected
39		*OrdStatus*	8	8 = Rejected
103		*OrdRejReason*	1153	1153 is a specific error code – see text in tag 58
58		*Text*	Member is disabled	
453		*NoPartyIDs*	8	There are 8 parties to the trade
->	447	*PartyIDSource*	D	D = Clearing Member
->	448	*PartyID*	ABC	Executing firm code
->	452	*PartyRole*	1	1 = Executing Firm
->	447	*PartyIDSource*	D	D = Clearing Member
->	448	PartyID	XYZ	Contra firm code
->	452	*PartyRole*	17	17 = Contra Firm
->	447	*PartyIDSource*	N	N = Client code
->	448	*PartyID*	JBLOGGS	Order origination trader name
->	452	*PartyRole*	11	11 = Order Origination Trader
->	447	*PartyIDSource*	N	N = Client code
->	448	*PartyID*	RSMITH	Entering trader name
->	452	*PartyRole*	36	36 = Entering Trader
->	447	*PartyIDSource*	P	P = Short Code
->	448	*PartyID*	78963258	Short Code
->	452	*PartyRole*	300	300 = Investment Decision ID
->	447	*PartyIDSource*	E	E = ISO Country Code
->	448	*PartyID*	GB	ISO Country Code
->	452	*PartyRole*	302	302 = Investment Decision Country Code
->	447	*PartyIDSource*	P	P = Short Code
->	448	*PartyID*	78963259	Short Code
->	452	*PartyRole*	301	301 = Execution Decision ID
->	447	*PartyIDSource*	E	E = ISO Country Code
->	448	*PartyID*	GB	ISO Country Code
->	452	*PartyRole*	303	303 = Execution Decision Country Code
581		*Account Type*	2	2 = House
5681		*ExchangeTrade Type*	0	0 = Normal
7931		*Venue ID*	0	0 = Inter-office
1301		*MarketID*	LME	LME = LME Base Metals
5179		*TradeTime*	10:52:19.123	
75		*TradeDate*	20160412	
20023		*CommodityDerivativeIndicator*	1	1 – Trade is not risk reducing
897		*NoTrades*	1	
->	11	*ClientOrderID*	ABC2016031	

				7000000123	
->	37		*OrderID*	00000000 00002341	Trade Half Id generated by matching
->	55		*Symbol*	CAD	
->	167		*SecurityType*	F	F = Future
->	461		*CFI Code*	FCEPS	
->	10010		*NoOfInstrumentLegs*	1	
	->	20004	*InstrumentLegNo*	1	
	->	10004	*PromptType*	S	S = Single Prompt Date
	->	541	*MaturityDate*	20160617	
->	423		*PriceType*	0	0 = Current
->	10051		*TradingCapacity*	DEAL	DEAL – Dealing on own account
->	54		*Side*	1	1 = buy in first leg
->	555		*NoLegs*	1	
	->	20005	*LegInstrument*	1	
	->	624	*LegSide*	1	1 = Buy
	->	10003	*LegLastQty*	20	
	->	637	*LegLastPx*	4935.45	
60			*TransactTime*	20160412-15:12:32.256	
Standard Trailer					

Example 7 – carry trade half submitted (to matching)

Tag Id		Name	Example	Comments
Standard Header				MsgType (35) = E
453		*NoPartyIDs*	8	There are 8 parties to the trade
->	447	*PartyIDSource*	D	D = Clearing Member
->	448	*PartyID*	ABC	Executing firm code
->	452	*PartyRole*	1	1 = Executing Firm
->	447	*PartyIDSource*	D	D = Clearing Member
->	448	PartyID	XYZ	Contra firm code
->	452	*PartyRole*	17	17 = Contra Firm
->	447	*PartyIDSource*	N	N = Client code
->	448	*PartyID*	JBLOGGS	Order origination trader name
->	452	*PartyRole*	11	11 = Order Origination Trader
->	447	*PartyIDSource*	N	N = Client code
->	448	*PartyID*	RSMITH	Entering trader name
->	452	*PartyRole*	36	36 = Entering Trader
->	447	*PartyIDSource*	P	P = Short Code
->	448	*PartyID*	78963258	Short Code
->	452	*PartyRole*	300	300 = Investment Decision ID
->	447	*PartyIDSource*	E	E = ISO Country Code
->	448	*PartyID*	GB	ISO Country Code
->	452	*PartyRole*	302	302 = Investment Decision Country Code
->	447	*PartyIDSource*	P	P = Short Code
->	448	*PartyID*	78963259	Short Code
->	452	*PartyRole*	301	301 = Execution Decision ID
->	447	*PartyIDSource*	E	E = ISO Country Code
->	448	*PartyID*	GB	ISO Country Code
->	452	*PartyRole*	303	303 = Execution Decision Country Code
581		*Account Type*	2	2 = House
5681		*ExchangeTrade Type*	0	0 = Normal
7931		*Venue ID*	0	0 = Inter-office
1301		*MarketID*	LME	LME = LME Base Metals
5179		*TradeTime*	10:52:19.123	
75		*TradeDate*	20160412	
20023		*CommodityDerivativeIndicator*	1	1 – Trade is not risk reducing
897		*NoTrades*	1	
->	11	*ClientOrderID*	ABC20160317000000124	
->	55	*Symbol*	CAD	
->	167	*SecurityType*	F	F = Future
->	461	*CFI Code*	FCEPS	
->	10010	*NoOfInstrumentLegs*	2	2 instruments – one for each leg

	->	20004	*InstrumentLegNo*	1	
	->	10004	*PromptType*	R	R = Rolling Prompt Date
	->	10000	*MaturityRollingPrompt*	C	C = Cash
	->	20004	*InstrumentLegNo*	2	
	->	10004	*PromptType*	R	R = Rolling Prompt Date
	->	10000	*MaturityRollingPrompt*	3M	3M = 3 Months
->	423		*PriceType*	0	0 = Current
->	10051		*TradingCapacity*	DEAL	DEAL – Dealing on own account
->	54		*Side*	1	1 = buy in first leg
->	555		*NoLegs*	2	2 legs
	->	20005	*LegInstrument*	1	
	->	624	*LegSide*	1	1 = Buy
	->	10003	*LegLastQty*	20	
	->	637	*LegLastPx*	4935.45	
	->	20005	*LegInstrument*	2	
	->	624	*LegSide*	2	1 = Sell
	->	10003	*LegLastQty*	20	
	->	637	*LegLastPx*	4700.55	
60			*TransactTime*	20160412-15:12:22.456	
Standard Trailer					

Example 8 – strategy trade submitted (to matching)

Strategy with two trades in it – one for SND and one for CAD

Tag Id			Name	Example	Comments
Standard Header					MsgType (35) = E
453			*NoPartyIDs*	4	There are 4 parties to the trade
->	447		*PartyIDSource*	D	D = Clearing Member
->	448		*PartyID*	ABC	Executing firm code
->	452		*PartyRole*	1	1 = Executing Firm
->	447		*PartyIDSource*	D	D = Clearing Member
->	448		PartyID	XYZ	Contra firm code
->	452		*PartyRole*	17	17 = Contra Firm
->	447		*PartyIDSource*	N	N = Client code
->	448		*PartyID*	JBLOGGS	Order origination trader name
->	452		*PartyRole*	11	11 = Order Origination Trader
->	447		*PartyIDSource*	N	N = Client code
->	448		*PartyID*	RSMITH	Entering trader name
->	452		*PartyRole*	36	36 = Entering Trader
->	447		*PartyIDSource*	P	P = Short Code
->	448		*PartyID*	78963258	Short Code
->	452		*PartyRole*	300	300 = Investment Decision ID
->	447		*PartyIDSource*	E	E = ISO Country Code
->	448		*PartyID*	GB	ISO Country Code
->	452		*PartyRole*	302	302 = Investment Decision Country Code
->	447		*PartyIDSource*	P	P = Short Code
->	448		*PartyID*	78963259	Short Code
->	452		*PartyRole*	301	301 = Execution Decision ID
->	447		*PartyIDSource*	E	E = ISO Country Code
->	448		*PartyID*	GB	ISO Country Code
->	452		*PartyRole*	303	303 = Execution Decision Country Code
581			*Account Type*	2	2 = House
5681			*ExchangeTrade Type*	0	0 = Normal
7931			*Venue ID*	0	0 = Inter-office
1301			*MarketID*	LME	LME = LME Base Metals
5179			*TradeTime*	10:52:19.123	
75			*TradeDate*	20160412	
20023			*CommodityDerivativeIndicator*	1	1 – Trade is not risk reducing
20020			*ComplexTrade ComponentId*	ABCSTRAT001	
897			*NoTrades*	2	2 trades in this startegy
->	11		*ClientOrderID*	ABC2016031 7000000133	
->	55		*Symbol*	SND	
->	167		*SecurityType*	F	F = Future
->	461		*CFI Code*	FCEPS	
->	10010		*NoOfInstrumentLegs*	1	
	->	20004	*InstrumentLegNo*	1	
	->	10004	*PromptType*	S	S = Single Prompt Date
	->	541	*MaturityDate*	20160617	
->	423		*PriceType*	0	0 = Current
->	10051		*TradingCapacity*	DEAL	DEAL – Dealing on own account
->	54		*Side*	1	1 = buy in first leg
->	555		*NoLegs*	1	
	->	20005	*LegInstrument*	1	
	->	624	*LegSide*	1	1 = Buy
	->	10003	*LegLastQty*	20	
	->	637	*LegLastPx*	4935.45	
->	11		*ClientOrderID*	ABC2016031 7000000133	
->	55		*Symbol*	CAD	
->	167		*SecurityType*	F	F = Future
->	461		*CFI Code*	FCEPS	

->	10010		*NoOfInstrumentLegs*	1	
	->	20004	*InstrumentLegNo*	1	
	->	10004	*PromptType*	S	S = Single Prompt Date
	->	541	*MaturityDate*	20160617	
->	423		*PriceType*	0	0 = Current
->	10051		*TradingCapacity*	DEAL	DEAL – Dealing on own account
->	54		*Side*	1	1 = buy in first leg
->	555		*NoLegs*	1	
	->	20005	*LegInstrument*	1	
	->	624	*LegSide*	1	1 = Buy
	->	10003	*LegLastQty*	20	
	->	637	*LegLastPx*	4935.45	
60			*TransactTime*	20160412-15:12:22.456	
Standard Trailer					

Example 9 – Trade Mass Status Request submitted (to matching)

Request to retrieve trades for the current day that have been rejected.

Tag Id	Name	Example	Comments
Standard Header			MsgType (35) = AF
584	*MassStatusReqID*	198762	Client supplied request id
585	*MassStatusReqType*	2	2 = current day
39	*OrdStatus*	8	8 = Rejected
Standard Trailer			

Example 10 – Trade Mass Status Request acknowledged (from matching)

Tag Id	Name	Example	Comments
Standard Header			MsgType (35) = cm3
5449	*ReqResponseTo*	1	1 = Mass Status Request response
584	*MassStatusReqID*	198762	Client supplied request id
5469	*ReqResponseStatus*	1	1 = success
5448	*NumMsg*	3	No of results to follow
Standard Trailer			

Example 11 – Trade Mass Status Request Response (from matching)

Matching responds with the execution report in response to trade mass status request (there would be three of these sent in response to the request in example 9).

Tag Id			Name	Example	Comments
Standard Header					MsgType (35) = 8
584			*MassStatusReqID*	198762	Client supplied request id
150			*ExecType*	I	I = Response to a request
39			*OrdStatus*	8	8 = Rejected
103			*OrdRejReason*	1153	1153 is a specific error code
58			*Text*	Member is disabled	
453			*NoPartyIDs*	4	There are 4 parties to the trade
->	447		*PartyIDSource*	D	D = Clearing Member
->	448		*PartyID*	ABC	Executing firm code
->	452		*PartyRole*	1	1 = Executing Firm
->	447		*PartyIDSource*	D	D = Clearing Member
->	448		PartyID	XYZ	Contra firm code
->	452		*PartyRole*	17	17 = Contra Firm
->	447		*PartyIDSource*	N	N = Client code
->	448		*PartyID*	JBLOGGS	Order origination trader name
->	452		*PartyRole*	11	11 = Order Origination Trader
->	447		*PartyIDSource*	N	N = Client code
->	448		*PartyID*	RSMITH	Entering trader name
->	452		*PartyRole*	36	36 = Entering Trader
->	447		*PartyIDSource*	P	P = Short Code
->	448		*PartyID*	78963258	Short Code
->	452		*PartyRole*	300	300 = Investment Decision ID
->	447		*PartyIDSource*	E	E = ISO Country Code
->	448		*PartyID*	GB	ISO Country Code
->	452		*PartyRole*	302	302 = Investment Decision Country Code
->	447		*PartyIDSource*	P	P = Short Code
->	448		*PartyID*	78963259	Short Code
->	452		*PartyRole*	301	301 = Execution Decision ID
->	447		*PartyIDSource*	E	E = ISO Country Code
->	448		*PartyID*	GB	ISO Country Code
->	452		*PartyRole*	303	303 = Execution Decision Country Code
581			*Account Type*	2	2 = House
5681			*ExchangeTrade Type*	0	0 = Normal
7931			*Venue ID*	0	0 = Inter-office
1301			*MarketID*	LME	LME = LME Base Metals
5179			*TradeTime*	10:52:19.123	
75			*TradeDate*	20160412	
20023			*CommodityDerivativeIndicator*	1	1 – Trade is not risk reducing
897			*NoTrades*	1	
->	11		*ClientOrderID*	ABC2016031 7000000123	
->	37		*OrderID*	00000000 00002341	Trade Half Id generated by matching
->	55		*Symbol*	CA	
->	167		*SecurityType*	F	F = Future
->	461		*CFI Code*	FCEPS	
->	10010		*NoOfInstrumentLegs*	1	
	->	20004	*InstrumentLegNo*	1	
	->	10004	*PromptType*	S	S = Single Prompt Date

	->	541	*MaturityDate*	20160617	
->	423		*PriceType*	0	0 = Current
->	10051		*TradingCapacity*	DEAL	DEAL – Dealing on own account
->	54		*Side*	1	1 = buy in first leg
->	555		*NoLegs*	1	
	->	20005	*LegInstrument*	1	
	->	624	*LegSide*	1	1 = Buy
	->	10003	*LegLastQty*	20	
	->	637	*LegLastPx*	4935.45	
60			*TransactTime*	20160412-15:12:32.256	
Standard Trailer					

Example 12 - Download Request – Members (to matching)

Tag Id	Name	Example	Comments
Standard Header			MsgType (35) = cm2
5447	*RequestID*	198799	Client supplied request id
5446	*RequestType*	2	2 = Download Members request
60	*TransactTime*	20160412-14:16:23.122	Time of request
Standard Trailer			

Example 13 - Download Request Acknowledgement (from matching)

Tag Id	Name	Example	Comments
Standard Header			MsgType (35) = cm3
5449	*ReqResponseTo*	2	2 = Download Request response
5447	*RequestID*	198799	Client supplied request id
5469	*ReqResponseStatus*	1	1 = success
5448	*NumMsg*	21	No of results to follow
Standard Trailer			

Example 14 - Member Definition (from matching)

Sent in response to the download request in example 12.

Tag Id	Name	Example	Comments
Standard Header			MsgType (35) = cm4
5447	*RequestID*	198799	Client supplied request id
5322	*FirmID*	ABC	Member ID
5364	*MemberName*	ABC Holdings	Member Name
5365	*MemberAddress*	Some Address	
9675	*ContactPhoneNo*	Some phone number	
Standard Trailer			