Order Entry Gateway
Binary Specification

Please respond to:

newtradingplatform@lme.com

Table of Contents

1 Session Management 9
1.1 Authentication 9
1.1.1 Comp ID 9
1.1.2 Password Encryption 9
1.1.3 Password 9
1.1.4 Change Password 9
1.2 Establishing a Binary Session 10
1.3 Message Sequence Numbers 10
1.4 Heartbeat and Test Request 11
1.5 Terminating a Binary Session 11
1.6 Re-establishing a Binary Session 11
1.7 Sequence Reset 12
1.8 Fault Tolerance 12
1.9 Checksum Validation 12
2 Recovery 13
2.1 General Message Recovery 13
2.2 Resend Request 13
2.3 Logon Message Processing – Next Expected Message Sequence 14
2.4 Possible Duplicates 14
2.5 Possible Resends 14
2.6 Gap Fills 14
2.7 Transmission of Missed Messages 15
3 Service Description 16
3.1 Security Identification 16
3.2 Security Creation 16
3.2.1 Strategies 16
3.3 Order Submission 18
3.4 Order Types 19
3.5 Order Validity Conditions 20
3.6 Order Types and Permitted Order Validity Conditions 21
3.7 Order Identification 21
3.8 Order Expiry 22

3.9 Order Restatement 22
3.10 Order Amendment 22
3.11 Order Cancellation 23
3.12 Mass Cancellation 24
3.13 Cancel on Disconnect 25
3.14 Mass Quote 25
3.15 Request for Quote (RFQ) 27
3.16 Speed Bumps 27
3.17 Message Throttling 36
3.18 Security Definition Throttle 37
3.19 Merged Order Books 37
3.20 Self Execution Prevention (SEP) 39
3.21 Market Maker Protection (MMP) 40
3.22 Inflight Order Processing 41
3.23 Trade Reporting 42
3.24 Client ID Usage 43
4 Message Definitions 44
4.1 Inbound Messages 44
4.2 Outbound Messages 44
4.3 Data Types 45
4.4 Message Composition 47
4.4.1 Fields Presence Map 47
4.4.2 Repeating Blocks and Nested Repeating Blocks 48
4.5 Required Fields 49
4.6 Message Header 49
4.7 Message Trailer 50
4.8 Administrative Messages 51
4.8.1 Logon (5) 51
4.8.2 Heartbeat (0) 52
4.8.3 Test Request (1) 52
4.8.4 Resend Request (2) 52
4.8.5 Sequence Reset (4) 56
4.8.6 Logout (6) 56

4.8.7 Reject (3) 57
4.9 Other Messages 57
4.9.1 Business Message Reject (7) 57
4.9.2 News (40) 58
4.10 Application Messages 60
4.10.1 Security Definition Request (10) 60
4.10.2 Security Definition (11) 62
4.10.3 New Order Single (12) 65
4.10.4 Amend Order (13) 72
4.10.5 Order Amend Rejected (14) 80
4.10.6 Cancel Order (15) 81
4.10.7 Order Cancel Rejected (16) 81
4.10.8 Execution Report (8) 82
4.10.9 Mass Cancel Request (17) 107
4.10.10 Mass Cancel Report (18) 109
4.10.11 Mass Quote (22) 112
4.10.12 Mass Quote Ack (23) 116
4.10.13 Quote Request (20) 123
4.10.14 Quote Request Ack (21) 123
4.10.15 MMP Reset Request (30) 126
4.10.16 MMP Reset Ack (31) 126

Document History

Version	Date	Change Description
1.0	27/07/2020	Initial draft
1.1	28/01/2022	Internal review

Preface

This document describes the binary interface protocol of the LME Order Entry Gateway.

The terminology used, message format, message flow and event models described throughout this document are similar to that of FIX 5.0 SP2 protocol specifications (https://www.fixtrading.org) where applicable, with some specific changes for performance and adaptability reasons.

Note message flow examples in this document are illustrations and do not contain all the mandatory fields. The presence of (…) denotes that fields have been omitted.

Bit position is shown as BP in the message definitions.

Delivery Phasing

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

Functionality that will be included in a later release is specified in the following table and shown throughout the document in *dark grey italics*. The initial release will contain all functionality that is **not** specified in the table.

Function	Reference
Futures strategies: • **3 Month Average** • **6 Month Average** • **12 Month Average** • **Carry Average**	3.2.1.1 Exchange Defined Strategy Types 4.10.1 Security Definition Request (10)
Options strategies: • **Call spread** • **Put spread**	3.2.1.1 Exchange Defined Strategy Types 4.10.1 Security Definition Request (10)
Custom strategies	3.2.1. Strategies 3.2.1.2 Custom Strategies 4.10.1 Security Definition Request (10) 4.10.2 Security Definition (11) (Example Message Flows) 4.10.8 Execution Report (8) (Example Message Flow)
Option contracts	3.2 Security Creation 3.2.1.1 Exchange Defined Strategy Types 3.2.1.2 Custom Strategies 4.10.1 Security Definition Request (10) 4.10.2 Security Definition (11) (Example Message Flows)

Function	Reference
Order types: • **Market to Limit** • **Stop Loss** • **Post Only** • **One Cancels Other**	3.4 Order Types 3.6 Order Types and Permitted Order Validity Conditions 4.10.3 New Order Single (12) 4.10.4 Amend Order (13) 4.10.8 Execution Report (8) 4.10.8.1 Execution Report Matrix
Order validity condition: • **Fill or Kill (FOK)**	3.5 Order Validity Conditions 3.6 Order Types and Permitted Order Validity Conditions 4.10.3 New Order Single (12) 4.10.4 Amend Order (13) 4.10.8 Execution Report (8)
Mass Quote	2.7 Transmission of Missed Messages 3.7 Order Identification 3.3 Order Submission 3.10 Order Amendment 3.11 Order Cancellation 3.12 Mass Cancellation 3.14 Mass Quote 3.17 Message Throttling 3.24 Client ID Usage 4.1 Inbound Messages 4.2 Outbound Messages 4.4.2 Repeating Blocks and Nested Repeating Blocks 4.10.7 Order Cancel Rejected (16) 4.10.8 Execution Report (8) 4.10.8.1 Execution Report Matrix 4.10.9 Mass Cancel Request (17) 4.10.10 Mass Cancel Report (18) 4.10.11 Mass Quote (22) 4.10.12 Mass Quote Ack (23)

Function	Reference
Request for Quote	2.7 Transmission of Missed Messages 3.15 Request for Quote (RFQ) 3.17 Message Throttling 4.1 Inbound Messages 4.2 Outbound Messages 4.10.13 Quote Request (20) 4.10.14 Quote Request Ack (21)
Speed Bumps	3.16 Speed Bumps 4.10.8 Execution Report (8) 4.10.8.1 Execution Report Matrix
Self Execution Prevention	3.20 Self Execution Prevention (SEP) 4.10.3 New Order Single (12) 4.10.4 Amend Order (13) 4.10.8 Execution Report (8) 4.10.8.1 Execution Report Matrix
Market Maker Protection	2.7 Transmission of Missed Messages 3.7 Order Identification 3.17 Message Throttling 3.21 Market Maker Protection (MMP) 4.1 Inbound Messages 4.2 Outbound Messages 4.9.2 News (40) 4.10.15 MMP Reset Request (30) 4.10.16 MMP Reset Ack (31)

1 Session Management

1.1 Authentication

1.1.1 Comp ID

A participant user should use the Comp ID (a unique session identifier) provided by the Exchange for each session in order to connect to the gateway. A single participant may have multiple connections to the gateway, i.e. multiple binary order entry sessions, each with its own Comp ID.

The messages sent to the gateway should contain the Comp ID assigned to the client in the field Comp ID in the header section of a message.

1.1.2 Password Encryption

The binary protocol requires Password and New Password to be encrypted when they are sent in the Logon (5) message from the client to the gateway.

To encrypt the password, the client is expected to use a 2048-bit RSA (https://en.wikipedia.org/wiki/RSA_(cryptosystem)) public key circulated ~~(through a different medium)~~ by the Exchange on https://www.lme.com/en/Trading/Initiatives/New-trading-platform/Access. The binary output of the RSA encryption must be represented in Big Endian PKCS #1 with padding scheme OAEP (https://en.wikipedia.org/wiki/PKCS_1) and then converted to an alphanumeric value by means of standard base-64 encoding (http://en.wikipedia.org/wiki/Base64) when communicating with the gateway.

1.1.3 Password

The client should specify their password in the Password field of the Logon (5) message. This password must be in encrypted form. For security reasons, the client is expected to prefix the login time, in UTC format (YYYYMMDDHHMMSS), to the password before encryption. The client must ensure that login time is in accurate UTC.

The gateway will extract the login time prefix from the decrypted password string and validate that it is within the acceptable tolerance of the actual current time. A logon request from the client that fails this validation is rejected by the gateway.

The gateway validates the password, any validation failure will result in logon attempt being unsuccessful.

Repeated failures in password validation will result in the client account being locked. The participant is expected to contact the Exchange to unlock the client account.

1.1.4 Change Password

Each new Comp ID will be assigned a password by the Exchange. The client is expected to change this password upon initial logon.

Each new Comp ID will be assigned a password on registration. The client is expected to change the password upon first logon whenever a password is (re)issued by the Exchange.

Password change request can be made together with Logon (5) request. The client should specify the encrypted new password in the New Password field and the current encrypted password in the Password field.

The new password must comply with Exchange's password policy. The status of the new password (i.e. whether it is accepted or rejected) will be specified in the Session Status response from the gateway. The new password, if accepted, will be effective for subsequent logins. If the new password provided fails validation, the gateway will reject the logon attempt.

1.1.4.1 Password Policy

The Exchange requires the password to contain:

- Minimum of 8 characters
- At least one number
- Combination of uppercase and lowercase characters.

1.2 Establishing a Binary Session

The client must wait for a successful Logon (5) response from the gateway before sending additional messages. If any message is received from the client before the exchange of logon messages, the TCP/IP connection with the client will be disconnected.

If a logon attempt fails, the gateway will send a Logout (6) and terminate the session; the Session Status of the Logout (6) message will indicate the reason for the logout.

If a client has already logged on, and if the gateway receives another connection attempt followed by a Logon (5) message with the same Comp ID, the gateway will terminate both connections without sending a Logout (6) or Reject (3) message.

If a session level failure occurs due to a message sent by the client which contains a sequence number that is less than what is expected and the PossDup is not set to 1 (Yes), then the gateway will send a Logout (6) and terminate the binary connection. In this scenario, the inbound sequence number will not be incremented but the outbound sequence number will be incremented.

If the gateway does not respond to the session initiation (client initiated Logon message), it is recommended that the client wait for a duration of 60 seconds prior to terminating the connection. The client is expected to retry session initiation after an elapsed time duration of 60 seconds.

If a client is disconnected abruptly or via a Logout message from the gateway, it is recommended that the client wait for a duration of 10 seconds prior to reconnecting to the gateway.

1.3 Message Sequence Numbers

The client and the gateway will each maintain a separate and independent set of incoming and outgoing message sequence numbers. Sequence numbers should be initialized to one (1) at the start of the day and be incremented throughout the session. Either side of a binary session will track the:

- Next Expected MsgSeqNum (starting at 1)
- Next To Be Sent MsgSeqNum (starting at 1) to the contra-party.

Monitoring sequence numbers will enable either parties to identify and react to the missed messages and gracefully synchronize applications when reconnecting a binary session.

Any message sent by either side of a binary session will increment the sequence number unless explicitly specified for a given message type.

If any message sent by one side of a binary session contains a sequence number that is LESS than the Next Expected MsgSeqNum then the other side of this session is expected to send a Logout message and terminate the binary connection immediately, unless the PossDup indicator is set to 1 (Yes)

A binary session will not be continued to the next trading day. Both sides are expected to initialize (reset to 1) the sequence numbers at the start of each day. At the start of each trading day if the client starts with a sequence number greater than 1 then the gateway will terminate the session immediately without any further exchange of messages.

1.4 Heartbeat and Test Request

The client and the gateway will use the Heartbeat (0) message to monitor the communication line during periods of inactivity and to verify that the interfaces at each end are available.

The gateway will send a Heartbeat anytime it has not transmitted a message for the duration of the heartbeat interval. The client is expected to employ the same logic.

If the gateway detects inactivity for a period longer than 3 heartbeat intervals, it will send a Test Request message to force a Heartbeat from the client. If a response to the Test Request is not received within a reasonable transmission time (recommended being an elapsed time equivalent to 3 heartbeat intervals), the gateway will send a Logout (6) and break the TCP/IP connection with the client. The client is expected to employ similar logic if inactivity is detected on the part of the gateway.

1.5 Terminating a Binary Session

Session termination can be initiated by either the gateway or the client by sending a Logout (6) message. Upon receiving the logout request, the contra party will respond with a Logout (6) message signifying a logout reply. Upon receiving the logout reply, the receiving party will terminate the connection.

If the contra-party does not reply with either a Resend Request (2) or a Logout (6) reply, the logout initiator should wait for 60 seconds prior to terminating the connection.

The client is expected to terminate each binary session at the end of each trading day before the gateway service is shut down. Any open binary connection will be terminated by the gateway by sending a Logout (6) when the service is shut down. Under exceptional circumstances, the gateway may initiate the termination of a connection during the trading day by sending the Logout (6) message.

If, during the exchange of logout messages, the client or the gateway detects a sequence gap, it should send a Resend Request (2).

1.6 Re-establishing a Binary Session

If a binary connection is terminated during the trading day, it may be re-established via an exchange of Logon messages.

Once the binary session is re-established, the message sequence numbers will continue from the last message successfully transmitted prior to the termination as described in 2.7 Transmission of Missed Messages.

1.7 Sequence Reset

Gap-fill mode can be used by one side when skipping session level messages which can be ignored by the other side.

During a binary session the gateway or the client may use the Sequence Reset (4) message in Gap Fill mode if either side wishes to increase the expected incoming sequence number of the other party.

It will not be possible to reset the client sequence number to 1 using the Logon message. Should a reset be required the participant should contact the Exchange.

The client is required to support a manual request by Exchange to initialize sequence numbers prior to the next login attempt.

1.8 Fault Tolerance

After a failure on the client side or on the gateway side, the client is expected to be able to continue with the same session.

If the sequence number is reset to one (1) by the gateway, all previous messages from the gateway will not be available for the client side.

The client and the gateway are expected to negotiate on the Next Expected MsgSeqNum and Next To Be Received Sequence number by contacting the Exchange prior to initiating the new session and consequently manually setting the sequence number for both ends after having a direct communication with the participant.

1.9 Checksum Validation

The gateway performs a checksum validation on all incoming messages into the input services. Incoming messages that fail the checksum validation will be rejected and the connection will be dropped by the gateway without sending a logout.

Conversely, the gateway stamps an identically calculated checksum field on all outgoing messages from the input interfaces. In case of a checksum validation failure, the client is expected to drop the connection and take any appropriate action before reconnecting. Messages that fail the checksum validation should not be processed.

This checksum is a CRC32C value with the polynomial 0x1EDC6F41, presented as a 32-bit unsigned integer (http://en.wikipedia.org/wiki/Cyclic_redundancy_check#CRC-32C).

2 Recovery

2.1 General Message Recovery

Message gaps may occur which are detected via the tracking of incoming sequence numbers. Recovery will be initiated if a gap is identified when an incoming message sequence number is found to be greater than Next Expected MsgSeqNum during Logon or the Sequence Number at other times.

The Resend Request (2) will indicate the Start Sequence and End Sequence of the message gap identified and when replying to a Resend Request (2), the messages are expected to be sent strictly honouring the sequence.

If messages are received outside of the Start Sequence and End Sequence, then the recovering party is expected to queue those messages until the gap is recovered.

During the message recovery process, the recovering party will increment the Next Expected MsgSeqNum accordingly based on the messages received. If messages applicable to the message gap are received out of sequence then the recovering party will drop these messages.

The party requesting the Resend Request (2) can specify “0” in the End Sequence to indicate that they expect the sender to send ALL messages starting from the Start Sequence. In this scenario, if the recovering party receives messages with a sequence greater than the Start Sequence, out of sequence, the message will be ignored.

Administrative messages such as Sequence Reset (4), Heartbeat (0) and Test Request (1) which can be considered irrelevant for a retransmission could be skipped using the Sequence Reset (4) message in gap-fill mode. Note that the gateway expects the client to skip Sequence Reset (4) messages when replying to a Resend Request (2) at all times.

When resending messages, the gateway would use either PossDup or PossResend indicator to indicate whether the messages were retransmitted earlier. If PossDup is set, it indicates that the same message with the given sequence number with the same business content may have been transmitted earlier. In the case where PossResend is set, it indicates that the same business content may have been transmitted previously but under the different message sequence number. In this case business contents needs to be processed to identify the resend. For example, in execution reports the Exec ID may be used for this purpose.

2.2 Resend Request

The client may use the Resend Request (2) message to recover any lost messages. This message may be used in one of three modes:

1. To request a single message. The Start Sequence and End Sequence should be the same.
2. To request a specific range of messages. The Start Sequence should be the first message of the range and the End Sequence should be the last of the range.
3. To request all messages after a particular message. The Start Sequence should be the sequence number immediately after that of the last processed message and the End Sequence should be zero (0).

2.3 Logon Message Processing – Next Expected Message Sequence

The session initiator should supply the Next Expected MsgSeqNum the value next expected from the session acceptor in Sequence Number. The session acceptor should validate the logon request including that Next Expected MsgSeqNum does not represent a gap. It then constructs its logon response with Next Expected MsgSeqNum containing the value next expected from the session initiator in Sequence Number having incremented the number above the logon request if that was the sequence expected.

The session initiator must wait until the logon response is received in order to submit application messages. Once the logon response is received, the initiator must validate that Next Expected MsgSeqNum does not represent a gap.

In case of gap detection from either party (lower than the next to be assigned sequence) recover all messages from the last message delivered prior to the logon through the specified Next Expected MsgSeqNum sending them in order, then gap fill over the sequence number used in logon and proceed sending newly queued messages with a sequence number one higher than the original logon.

Neither side should generate a Resend Request (2) based on Sequence Number of the incoming Logon message but should expect any gaps to be filled automatically by following the Next Expected Sequence processing described above. Whilst the gateway is resending messages to the client, the gateway does not allow another Resend Request (2) from the client. If a new Resend Request (2) is received during this time, the gateway will terminate the session immediately without sending the Logout (6) message.

Note that indicating the Next Expected MsgSeqNum in Logon (5) is mandatory.

2.4 Possible Duplicates

The gateway handles possible duplicates in the same way as the FIX protocol. The client and the gateway use the PossDup field to indicate that a message may have been previously transmitted with the same Sequence Number.

2.5 Possible Resends

The gateway does not handle possible resends for the client-initiated messages (e.g., New Order, Mass Quote, etc.) and all the messages will be processed without considering the value in the PossResend field. Any message with duplicate Client Order ID will be rejected based on the Client Order ID uniqueness check and those messages that conform to the uniqueness check will be processed as normal messages.

The gateway may use the PossResend field to indicate that an application message may have already been sent under a different sequence number. The client should validate the contents (e.g., Exec ID) of such a message against those of messages already received during the current trading day to determine whether the new message should be ignored or processed.

2.6 Gap Fills

The following messages are expected to be skipped using gap-fills when being retransmitted:

1. Logon

2. Logout
3. Heartbeat
4. Test Request
5. Resend Request
6. Sequence Reset

All other messages are expected to be replayed within a retransmission.

2.7 Transmission of Missed Messages

Following messages will be sent to the client when it reconnects if such messages were generated during a period when this client was disconnected from the gateway:

- Execution Report (includes order reject)
- Order Amend Rejected
- Order Cancel Rejected
- Mass Cancel Report
- Business Message Reject
- Reject
- *Quote Request ~~Rejected~~Ack*
- *Mass Quote Ack*
- Security Definition
- *MMP Reset Ack*
- News.

In the unlikely event the disconnection was due to an outage of the gateway, Business Message Reject and Reject messages may not be retransmitted, and the other messages which will be retransmitted to the client will include a PossResend set to 1 (Yes).

3 Service Description

3.1 Security Identification

Each Tradable Instrument will be identified using the Security ID field which can be a maximum of 19 digits.

3.2 Security Creation

A Security Definition Request (10) can be submitted to create a new tradable instrument:

Instrument Request Type	Binary Fields
Options strike	*Security Type = 2* *Security Sub Type = 0* *Maturity Date* *Strike Price* *Put or Call*
Strategy	Security Type = 3 Security Sub Type = 1 to 10 Leg Security ID Leg Ratio Leg Side *Leg Price*

3.2.1 Strategies

A trader can submit Security Definition Request (10) for an Exchange defined strategy type or a *custom strategy. A Delta Hedge strategy can be submitted as a custom strategy.*

~~Strategy legs must be specified in order of expiry. *For a Delta Hedge custom strategy, the option legs must precede those of the future(s).* Security Definition Requests (10) containing strategy legs that are not in correct order will not be permitted and the Security Definition (11) returned will contain the Security Response Type = '5' Reject security proposal and the Security Reject Reason (1607) = '12' Invalid instrument structure~~.Validation on the strategy requested can return the following Security Reject Reason values:

Validation	Security Reject Reason
Valid strike price	102 = Invalid strike price
Valid Leg Security ID	103 = LegSecurityID (602) does not exist

Validation	Security Reject Reason
Valid Security Sub Type	105 = Invalid SecuritySubType (762)
Valid number of legs for the strategy	12 = Invalid instrument structure
Valid leg ratio for the strategy	12 = Invalid instrument structure
Legs are in the same contract	12 = Invalid instrument structure
Option strikes conform to the strategy	*12 = Invalid instrument structure*
Legs are in expiry order	12 = Invalid instrument structure
Strategy contains the same Leg Security ID	12 = Invalid instrument structure

The accepted strategy definition is the buy side of the strategy. A Strategy Definition Request (10) will also be accepted which is the inverse (sell side) of an already created strategy or an Exchange defined strategy that has not been created. The Security Definition (11) returned will contain the SecurityResponseType = '2' Accept security proposal with revisions as indicated in the message and the Security ID of the existing strategy.

3.2.1.1 Exchange Defined Strategy Types

The following defined strategy types are supported:

Futures Strategies

Security Sub Type	Strategy Name	Definition (from buy perspective)
1	Carry	Buy near leg, sell far leg
3	*Average 3M*	*Buying 3 consecutive (monthly) legs*
4	*Average 6M*	*Buying 6 consecutive (monthly) legs*
5	*Average 12M*	*Buying 12 consecutive (monthly) legs*
6	*Carry Average*	*Buy an outright (e.g. 3M), sell a Future Average (e.g. first quarter 2023).*

An Average strategy is only permitted in monthly prompts and only the front leg needs to be specified as the remaining legs will be consecutive.

A Carry Average is the only permitted nested strategy type.

Options Strategies

Security Sub Type	Strategy Name	Definition (from buy perspective)
7	*Call Spread*	*Buy a (call) strike, sell a (call) higher strike within the same option expiry*
8	*Put Spread*	*Buy a (put) strike, sell a (put) lower strike within the same option expiry*

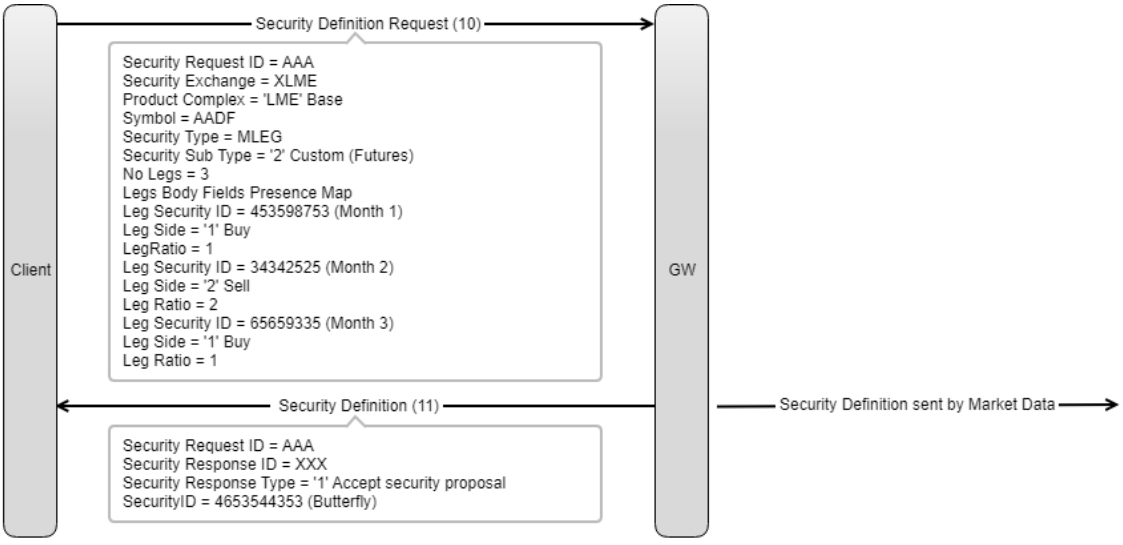
3.2.1.2 Custom Strategies

A non-Exchange defined strategy can be submitted in a Security Definition Request as a custom strategy using either:

- *Security Sub Type = '2' Custom (Futures)*
- *Security Sub Type = '9' Custom (Delta Hedge)*
- *Security Sub Type = '9' Custom (Options).*

A custom strategy may consist of up to five legs in a Futures contract or premium quoted Option. Each leg in the strategy must be in the same contract except for a delta hedge custom strategy in premium-based options where the last 1 to 2 legs belong to the underlying futures contract. Note, an Exchange defined strategy cannot be used within a custom strategy.

~~*The custom strategy is defined from a buy side perspective by specifying the Security ID of each of the legs, the leg ratio and leg side.*~~ *For example, a Futures Butterfly is defined as buy Month 1, sell Month 2 twice and buy Month 3.*



3.3 Order Submission

It is possible to submit orders for outright futures, options series or strategies using any of the order types specified in 3.4.1 Order Types. An individual order can be submitted using New Order Single (12) whereas *multiple orders can be submitted using Mass Quote (22)*.

~~3.4 Order Management~~

~~3.5~~3.4 Order Types

The following order types are supported:

Order Type	Binary Field
Limit An order submitted with a price and volume that will trade at the limit price or better for as much of its stated volume as is available in the order book.	Order Type = 2 Order Price
Market to Limit *An order submitted with a volume specified but no price. The order is executed at the best available price(s) up / down to their assigned limit price. Any order volume which is not fully executed rests in the order book as a limit order at the assigned limit price.*	*Order Type = 10*
Stop Loss *An order that is submitted but not visible in the order book until it is triggered by the last traded price and/or best bid/offer. Once triggered the order is entered into the order book as a Market to Limit order.*	*Order Type = 3* *Trigger Price* *Trigger Price Type*
Stop Limit An order that is submitted but not visible in the order book until it is triggered by the last traded price and/or best bid/offer. Once triggered the order is entered into the order book as a Limit order with at the specified price.	Order Type = 4 Trigger Price Trigger Price Type Order Price
Iceberg An order submitted with a visible order quantity and a total order quantity. The visible order quantity must be fully executed before it can be replenished with the next visible order quantity.	Order Type = 11 Order Price Display Quantity Order Quantity
Post Only *The order must rest in the order book before it can trade. If the order can be executed on entry into the order book it is rejected. If an amendment to the order can result in execution it also is rejected and the original order remains.*	*Order Type = 12* *Order Price*

Order Type	Binary Field
One Cancels Other (OCO) – Market to Limit *A single order which is a combination of a Limit and a Stop. On submission the Limit price and a Stop trigger price is specified.* *A partial trade at the Limit price will reduce the quantity available in the OCO. If the order is traded out at the Limit price the Stop component will be cancelled. Similarly if the Stop is triggered then the Limit component is cancelled.* *Note: No Execution Report will be generated for the cancelled component.* *If the Stop component is triggered the order becomes a Market to Limit order.*	*Order Type = 13* *Order Price* *Trigger Price* *Trigger Price Type*
One Cancels Other (OCO) – Limit *A single order which is a combination of a Limit and a Stop. On submission the Limit price and a Stop trigger price is specified.* *A partial trade at the Limit price will reduce the quantity available in the OCO. If the order is traded out at the Limit price the Stop component will be cancelled. Similarly if the Stop is triggered then the Limit component is cancelled.* *Note: No Execution Report will be generated for the cancelled component.* *If the Stop component is triggered it becomes a Limit order at a new price.*	*Order Type = 14* *Order Price* *Trigger Price* *Trigger Price Type* *Trigger New Price*

~~3.6~~3.5 Order Validity Conditions

Validity Condition	Binary Field
Day An order that will expire at the end of the day.	Time In Force = 0
Good Till Cancelled (GTC) An order that is valid until it is either cancelled or matched.	Time In Force = 1
Immediate or Cancel (IOC) An order that is executed at the stated price or better for as much order volume that is available. Any order volume that cannot be traded is cancelled.	Time In Force = 3

Validity Condition	Binary Field
Fill or Kill (FOK) *An order that is only executed if there is sufficient volume available, at the stated price or better, for them to execute fully. Otherwise the entire order is cancelled.*	*Time In Force = 4*
Good Till Date (GTD) The order is valid until the end of the trading date specified.	Time In Force = 6 Expire Date

Note: A GTC or GTD order cannot be entered into the TOM prompt.

A GTD will be rejected if the expiry date entered is the current trading date. A GTD in a single prompt will be rejected if the date entered exceeds the last trading date.

~~3.7~~3.6 Order Types and Permitted Order Validity Conditions

Order Type	Day	GTC	IOC	FOK	GTD
Limit	✔	✔	✔	✔	✔
Market to Limit	✔	✔	✔	✔	✔
Stop Loss	✔	✔			✔
Stop Limit	✔	✔			✔
Iceberg	✔	✔			✔
Post Only	✔	✔			✔
OCO Market to Limit	✔	✔			✔
OCO Limit	✔	✔			✔

~~3.8~~3.7 Order Identification

The client must specify a Client Order ID when submitting a New Order Single (12), ~~Order~~ Amend Order (13), ~~Order~~ Cancel Order (15) or Mass Cancel Request (17). As with the FIX protocol, the client should ~~comply with and~~ ensure that the each Client Order ID for this Comp ID is unique for the duration of the trading day and has not been used already for any of the currently persisted orders belonging to this Comp ID.

~~Note~~In addition, ~~the~~a Quote ID supplied on a Mass Quote (22) and MMP Reset Request ID supplied in a MMP Reset Request (30) must also be unique and must not have been specified as a Client Order ID.

When an order is accepted, the system assigns an Order ID that is unique for all orders and quotes.

When modifying or cancelling an order, the Original Client Order ID is used to identify the original order.

~~3.9~~3.8 Order Expiry

No Execution Report will be sent for orders with a Time in Force = '0' Day when they expire at the end of the trading day.

At the end of the day, the order originator will receive an Execution Report with Exec Type = 'C' and Order Status = '12' Expired for Time in Force = '1' Good Till Cancelled and Time in Force = '6' Good Till Date in the following cases:

- ~~Expiring orders submitted with Time in Force = '6' Good Till Date (GTD).~~ Expire Date has passed
- Last trading date for the tradable instrument has passed
- Any other Exchange specific configuration for order expiry of persisted orders
- Strategy contains a leg that has expired or meets any of the conditions above.
- ~~GTC orders if the tradable instrument expires at the end of the trading day~~

A GTD order with an expiry date between the current and next business day will be cancelled on the new business day prior to member connectivity and therefore not receive an Execution Report.

~~3.10~~3.9 Order Restatement

GTC/GTD orders that have not hit their expiry condition are persisted when the respective tradable instrument enters the Close state. The order originator is notified by Execution Report with Exec Type = '3' and Order Status = 3 = Done for Day.

On initial logon on the next trading day, Execution Reports are sent for persisted orders that have been returned with Exec Type = 'D' Restated, Order Status = '0' New or '1' Partially Filled and Exec Restatement Reason = '1' GT renewal / restatement.

~~3.11~~3.10 Order Amendment

An order can be amended by using Amend Order (13) and specifying the Original Client Order ID. The client can optionally specify the Order ID in the Amend Order (13). If Order ID is specified the system will validate whether the Order ID is associated with the correct order as identified using the Original Client Order ID. The Amend Order (13) will be rejected if the specified Order ID is invalid based on this validation.

The following order attributes can be modified if they have been specified on the original order:

- Order Price
- Trigger Price
- *Trigger New Price*
- Order Quantity
- Display Quantity

- Expiry Date
- Order Capacity
- Order Restrictions
- Execution Decision Within Firm
- Client Branch Country ~~Code~~
- Text.

The following optional order attributes can be modified. If the attribute is not present in the Amend Order (13), this indicates that the default value will be used:

- Direct Electronic Access
- Aggregated Order
- Pending Allocation Order
- Liquidity Provision Order
- Risk Reduction Order.

The following optional order attributes can be modified. If the attribute is not present in the Amend Order (13) it indicates that the value have been removed or not applicable:

- Text
- Investment Decision Within Firm
- Investment Decision Country ~~Code~~
- Execution Decision Country ~~Code~~.

The client cannot amend an order that is fully filled or cancelled or expired.

Amend Order (13) cannot be used to amend an order submitted using Mass Quote.

The Trigger Price or *Trigger New Price* cannot be amended if the Stop order has been triggered.

If the client sends an Amend Order (13) for an order for which an Amend Order (13) or a Cancel Order (15) is already being processed the incoming Amend Order (13) is rejected.

~~3.12~~3.11 Order Cancellation

An individual order can be cancelled using Cancel Order (15) by specifying the Original Client Order ID.

The client can optionally specify the Order ID in the Cancel Order (15). If the Order ID is specified the system will validate whether the Order ID is associated with the correct order as identified using the Original Client Order ID. The Cancel Order (15) will be rejected if the specified Order ID is invalid based on this validation.

A successful cancellation will return an Execution Report (8). If the cancellation request is rejected, an Order Cancel Rejected (16) is sent containing the reason for rejection.

The client may not cancel an order that is fully filled or cancelled or expired.

Cancel Order (15) cannot be used to cancel an order submitted using Mass Quote.

If the client sends a Cancel Order (15) for an order for which a Cancel Order (15) is already being processed the incoming Cancel Order (15) is rejected.

If the client sends a cancel request for an order for which an amendment is being processed the incoming cancel request will be processed (i.e. accepted or rejected) once the outcome of the amendment is known.

~~3.13~~3.12 Mass Cancellation

Multiple orders/quotes can be cancelled using Mass Cancel Request (17) by specifying which orders/*quotes* are to be cancelled:

Cancellation Type	Binary Field
All orders/*quotes* for a Comp ID	Mass Cancel Request Type = 7 Mass Cancel Scope
All orders/*quotes* for a specific tradable instrument	Mass Cancel Request Type= 1 Mass Cancel Scope Security ID
All orders/*quotes* for a specific contract	Mass Cancel Request Type= 3 Mass Cancel Scope Contract Code
All orders/*quotes* for a specific contract and side of the market	Mass Cancel Request Type= 3 Mass Cancel Scope Contract Code Side
All quotes for a specific Quote ID	*Mass Cancel Request Type = 101* *Mass Cancel Scope = 2* *Quote ID*
All orders/*quotes* for a specific end client	Mass Cancel Request Type= 7 Mass Cancel Scope Broker Client ID

If the Mass Cancel Request is accepted, Execution Reports will be sent for each order cancellation and will include the Client Order ID provided on the Mass Cancel Request (17). The Mass Cancel Report (18) will reflect the action taken and indicate the Total Affected Orders.

If the Mass Cancel Request is rejected, the Mass Cancel Response = '0' Cancel Request Rejected and will include the Mass Cancel Reject Reason.

~~3.14~~3.13 Cancel on Disconnect

The gateway will not automatically cancel a user's non-persisted orders and quotes in the event of a Logout. A user should explicitly cancel such orders and quotes prior to Logout using a Mass Cancel Request (17).

On order submission, a user can specify whether non-persisted orders should be cancelled on system disconnection (due to, for example, a network issue or in the event of inactivity such as too many missed heartbeats) using Cancel on Disconnect.

On detection of a loss of connectivity, the system will determine whether a user's non-persisted orders are to be cancelled based on Cancel on Disconnect attribute for an order. Orders from a Mass Quote are by default classified as non-persisted orders and are therefore automatically cancelled.

This feature does not guarantee that all live orders will be successfully cancelled as executions that occur very near to the time of disconnect may not be reported to the client. It also depends on the tradable instrument trading state when the abrupt disconnection is identified by the Exchange system.

~~3.15~~3.14 Mass Quote

Multiple orders can be submitted by permissioned trading users in multiple tradable instruments in the same contract using a Mass Quote (22).

Orders in the form of quotes are submitted as a quote pair (bid and offer) in a quote entry with a Quote Entry ID. Up to 20 quote entries can be submitted in a Mass Quote.

Each quote entry is related to a Quote Set which identifies the tradable instrument. All the quote sets must belong to the same contract. Within each quote set, up to three prices levels can be specified using Quote Price Level.

Quote entries within a Mass Quote can be ~~both amended~~replaced or cancelled using a Mass Quote. ~~Amendments and cancellations can be made simultaneously.~~

~~Amendments are made by sending r~~Replacement quotes ~~to~~can overwrite or cancel both sides of a quote entry or a single side of a quote pair leaving the other side unchanged. Note, an existing ~~order~~quote entry is identified using the Security ID, Quote Price Level and Side. If there is no such ~~order~~quote entry, the ~~amend~~ instruction will result in a new ~~order~~quote.

Cancellation is indicated by a zero value for Bid Size for bid side or Offer Size for offer side and a null value for Bid Price or Offer Price. ~~Cancellation can be for both sides of a quote pair or a single side of a quote pair leaving the other side unchanged. One side of a quote entry can be amended and the other side can be cancelled.~~

Unchanged is indicated by a ~~-1~~ value of -1 for Bid Size for bid side or Offer Size for offer side and a null value for Bid price or Offer Price. The quote side that has not changed will retain its current price time priority.

During mass quote processing, resting quotes that will be updated are removed to prevent execution with replacement quotes. An Execution Report will be sent for each quote side cancelled. An Execution Report will be sent to report quotes that have been replaced and replacement quotes that have been rejected. Order Cancel Reject (16) will be returned for cancellations that have been rejected.

The Quote Price Level that is assigned to a quote pair will remain unchanged by either cancellation or replacement. For example, if three price level have been specified in a Quote Set, the cancellation of quote pair at a quote price level will not reorder the quote price level of the other two quote entries in the Quote Set.

A single sided quote can be submitted with a dummy quote to make up the quote pair. The dummy quote is indicated by a value of -1 for Bid Size/Offer Size and a null value for Bid Price/Offer Price as specified in 4.3 Data Types.

For each quote entry side in the Mass Quote, an Execution Report (8) is returned to indicate whether the quote entry has been accepted or rejected. The Execution Report (8) can be mapped back to the quote in the Mass Quote message through the:

1. *Quote ID returned as Client Order ID*
2. *Quote an amendment Entry ID returned as Secondary Client Order ID.*

If ~~an amendment~~ a cancellation to a quote side fails validation ~~Order Amend Rejected (14) will be returned. Similarly for a cancellation validation failure~~ Order Cancel Rejected (16) will be returned.

Crossed quotes submitted in the same quote pair will be rejected but will be executed if supplied in different quote pairs.

If a Mass Quote message is rejected, the gateway will return a Mass Quote Ack (23) containing the Quote Reject Reason. For example, if quotes for different contracts have been specified.

An order resulting from a quote is always treated as a Limit order which expires at the end of the current day. In the event of a system related connection loss, orders from Mass Quotes will be automatically cancelled, see 3.11 Cancel on Disconnect.

An order resulting from a quote is assigned the order attributes as defined in the Mass Quote message. The following attributes are not included in the Mass Quote message and the resulting order is assigned the following default values:

Attribute	Assigned Value
Cancel on Disconnect	*N = No*
Aggregated Order	*N = No*
Pending Allocation Order	*N = No*
Liquidity Provision Order	*Y = Yes*
Risk Reduction Order	*N = No*
Text	*Absent*

For an existing quote that is being amended, only the following attributes will be amended: ~~All other order attributes are non-amendable and will retain their original value.~~

- *Bid Size*
- *Bid Price*

- *Offer Size*
- *Offer Price.*

All other order attributes that are ~~non-~~not amendable ~~and~~ will retain their original value.

~~3.16~~3.15 Request for Quote (RFQ)

~~An RFQ~~A Quote Request (20) indicates a trading interest in a specific instrument which is published to market participants by the Market Data service.

The Quote Request (20) will include the Quote Request Type which specifies whether a single quote or streaming quotes are requested. It can optionally specify the side and the quantity for which a price is required.

A Quote Request Ack (21) will be returned by the gateway in response to a Quote Request.

Trading participants can respond to an RFQ using standard order and quote functionality.

~~3.17~~3.16 Speed Bumps

Exchange contracts may be configured with speed bumps. A speed bump will only be applicable to New Order Single (12) and Amend Order (13).

Passive orders, cancellations using Cancel Order (15) or Mass Cancel Request (17)~~,~~ and Mass Quote (22) ~~and Post Only orders~~ will be exempt.

The status of an order in a speed bump will be reported in Exec Type Reason in the Execution Report (8):

101 = Order accepted but speed bump applied

102 = Order added after speed bump

103 = Order cancelled whilst in speed bump delay

104 = Original order is in speed bump enforced delay

105 = Order updated after speed bump delay

106 = Amend is in speed bump delay

107 = Order amended after speed bump delay

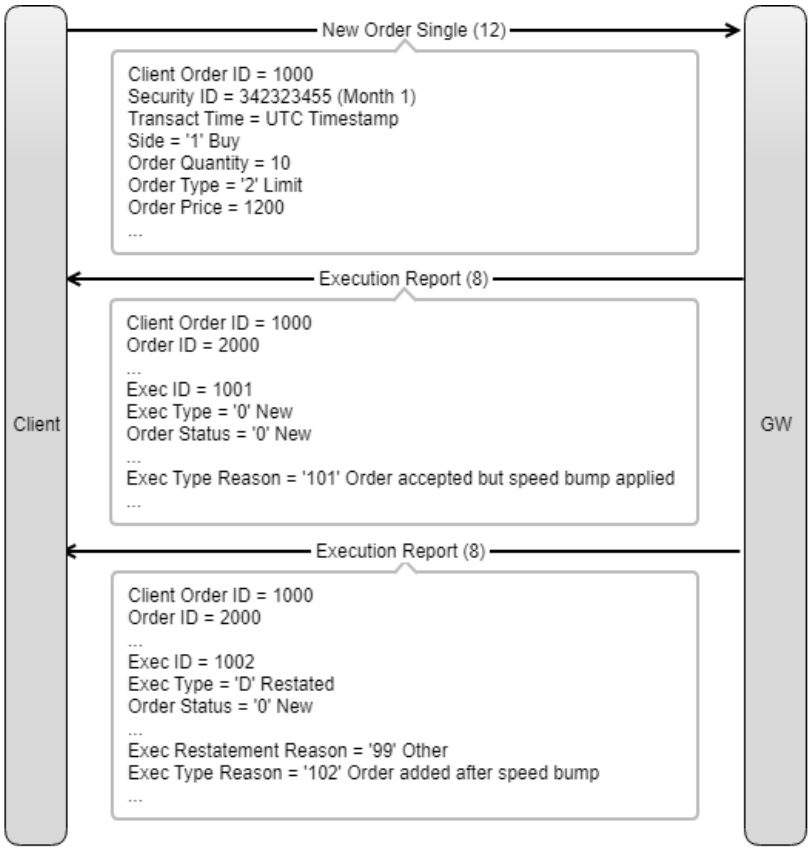
108 = Order rejected after speed bump delay

109 = Unsolicited cancel while in speed bump

Order submission is speed bumped

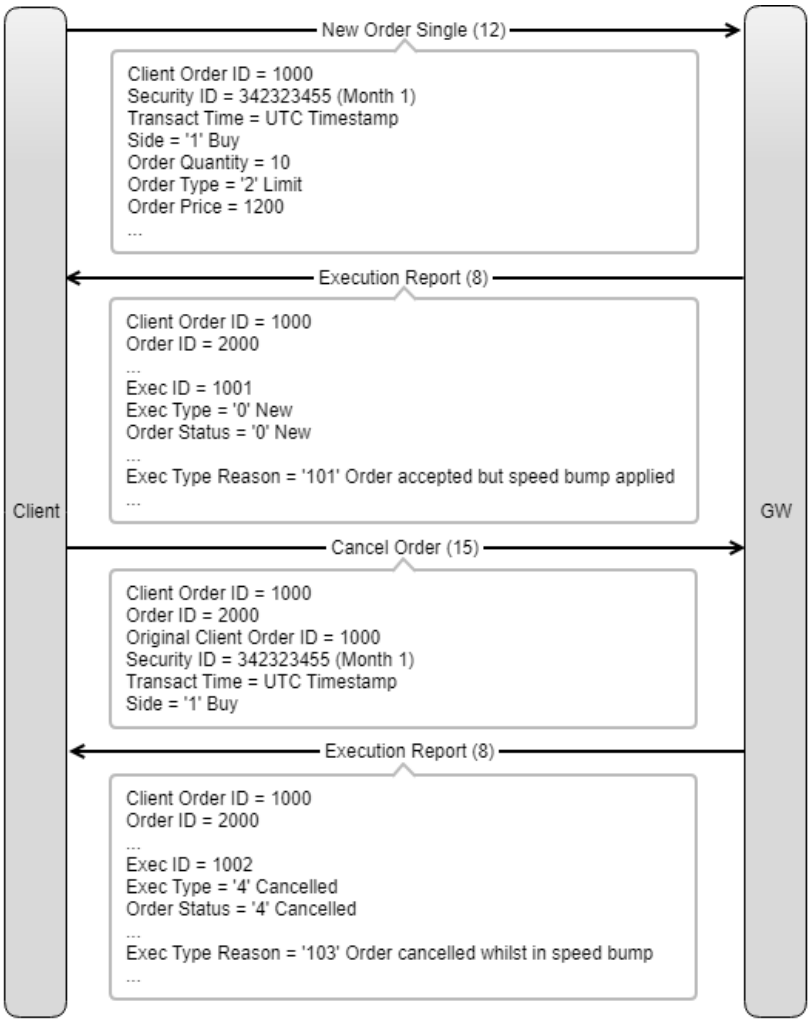
If an order is submitted but is subject to a speed bump, the order is held and not added to the order book until the order has been released from the speed bump. The Execution Report sent in acknowledgement includes an Exec Type Reason = '101' Order accepted but speed bump applied.

The Execution Report sent once the order has cleared the speed bump and is added to the order book includes Exec Type = 'D' Restated and Exec Type Reason = '102' Order added after speed bump.



Order cancellation for a speed bumped order

An order cancellation submitted while an order is in the speed bump will be processed without any delay as the Cancel Order (15) is not subject to speed bump conditions. The Execution Report sent in response to the cancellation includes Exec Type Reason = '103' Order cancelled whilst in speed bump delay.

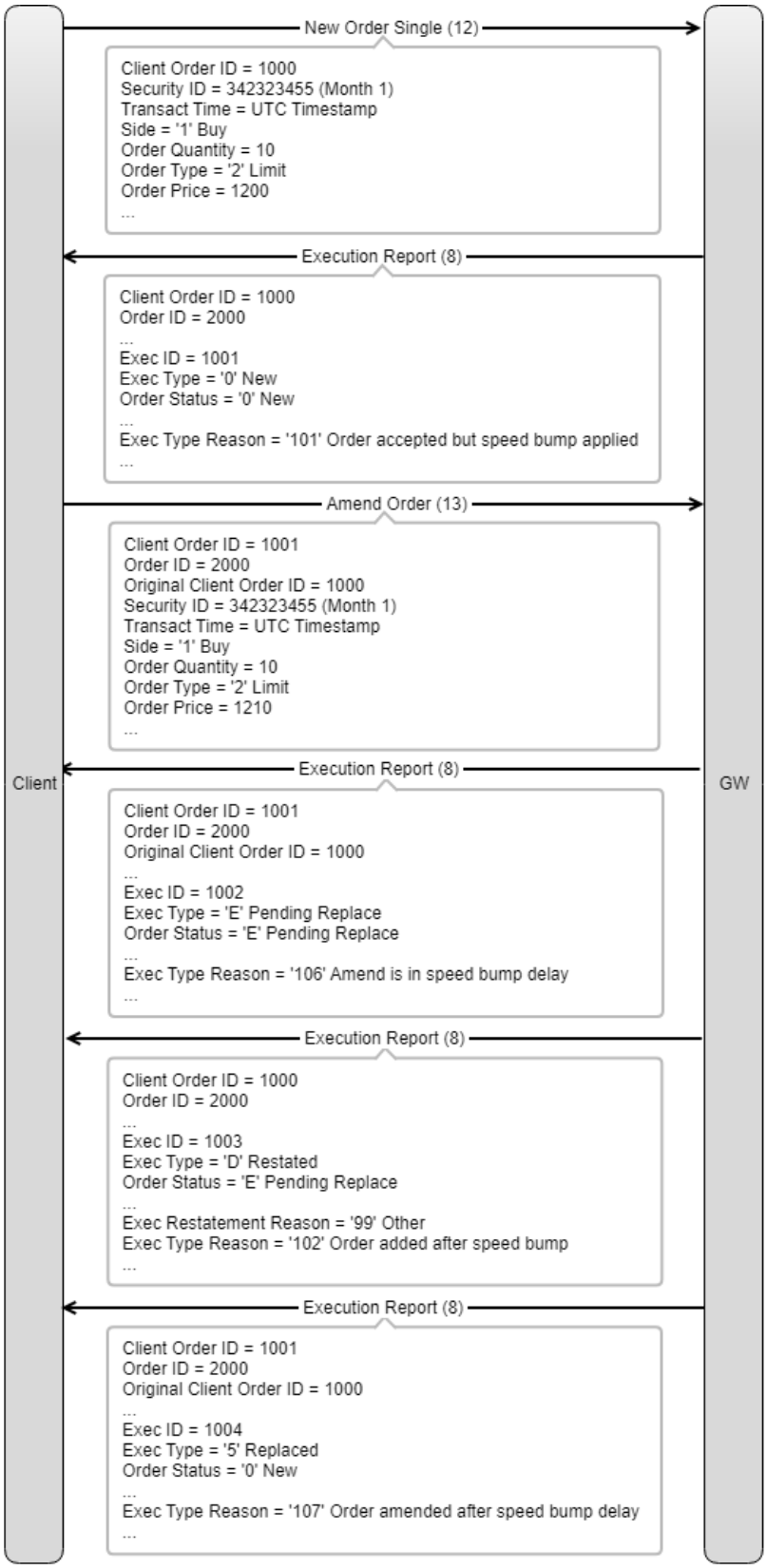


Executable order amendment for a speed bumped order will be speed bumped

An order is submitted which is subject to a speed bump. An Amend Order (13) is submitted while the order submission is in the speed bump queue. The amended order is executable and therefore speed bumped. The Execution Report for the order revision includes Exec Type Reason = '106' Amend is in speed bump delay. The Amend Order will not be processed until the original order has cleared the speed bump.

The Execution Report sent when the original order submission is released from the speed bump and added to the order book includes Exec Type = 'D' Restated, Order Status = 'E' Pending Replace and Exec Type Reason = '102' Order added after speed bump.

Another Execution Report is sent when the order revision clears the speed bump and replaces the original order. The Execution Report includes Exec Type = '5' Replaced and Exec Type Reason (2431) = '107' Order amended after speed bump delay.

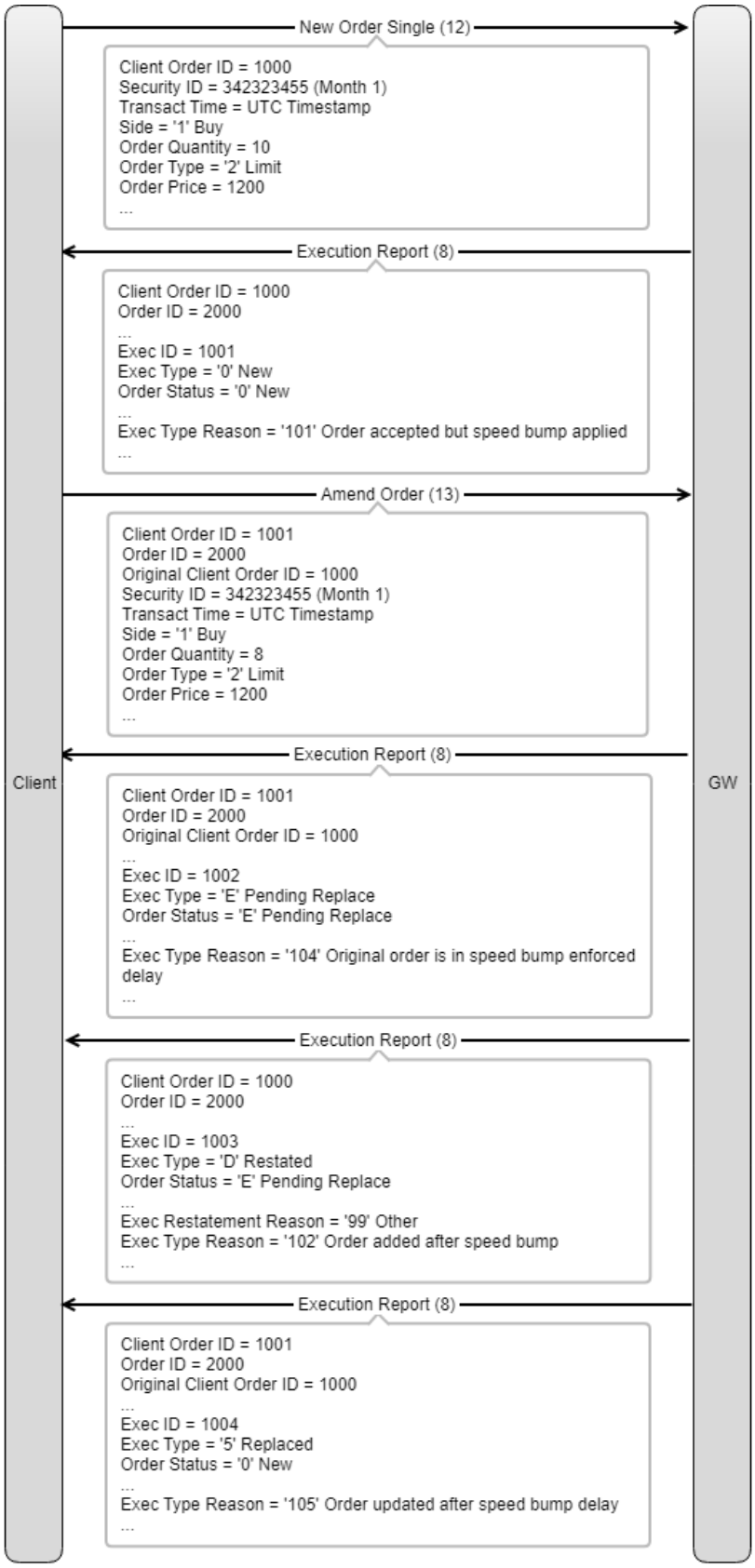

Client
GW
New Order Single (12)
Client Order ID = 1000
Security ID = 342323455 (Month 1)
Transact Time = UTC Timestamp
Side = '1' Buy
Order Quantity = 10
Order Type = '2' Limit
Order Price = 1200
...
Execution Report (8)
Client Order ID = 1000
Order ID = 2000
...
Exec ID = 1001
Exec Type = '0' New
Order Status = '0' New
...
Exec Type Reason = '101' Order accepted but speed bump applied
...
Amend Order (13)
Client Order ID = 1001
Order ID = 2000
Original Client Order ID = 1000
Security ID = 342323455 (Month 1)
Transact Time = UTC Timestamp
Side = '1' Buy
Order Quantity = 10
Order Type = '2' Limit
Order Price = 1210
...
Execution Report (8)
Client Order ID = 1001
Order ID = 2000
Original Client Order ID = 1000
...
Exec ID = 1002
Exec Type = 'E' Pending Replace
Order Status = 'E' Pending Replace
...
Exec Type Reason = '106' Amend is in speed bump delay
...
Execution Report (8)
Client Order ID = 1000
Order ID = 2000
...
Exec ID = 1003
Exec Type = 'D' Restated
Order Status = 'E' Pending Replace
...
Exec Restatement Reason = '99' Other
Exec Type Reason = '102' Order added after speed bump
...
Execution Report (8)
Client Order ID = 1001
Order ID = 2000
Original Client Order ID = 1000
...
Exec ID = 1004
Exec Type = '5' Replaced
Order Status = '0' New
...
Exec Type Reason = '107' Order amended after speed bump delay
...

Non-executable order amendment for a speed bumped order will not be speed bumped

An order is submitted which is subject to a speed bump. An Amend Order (13) is submitted while the order submission is in the speed bump queue. The amended order will rest in the order book and is therefore not subject to speed bump conditions. The Amend Order will not be processed until the original order has cleared the speed bump therefore the Execution Report for the revision includes Exec Type = 'E' Pending Replace and Exec Type Reason = '104' Original order is in speed bump enforced delay.

The Execution Report sent once the order submission has cleared the speed bump and is added to the order book includes Exec Type = 'D' Restated and Exec Type Reason = '102' Order added after speed bump.

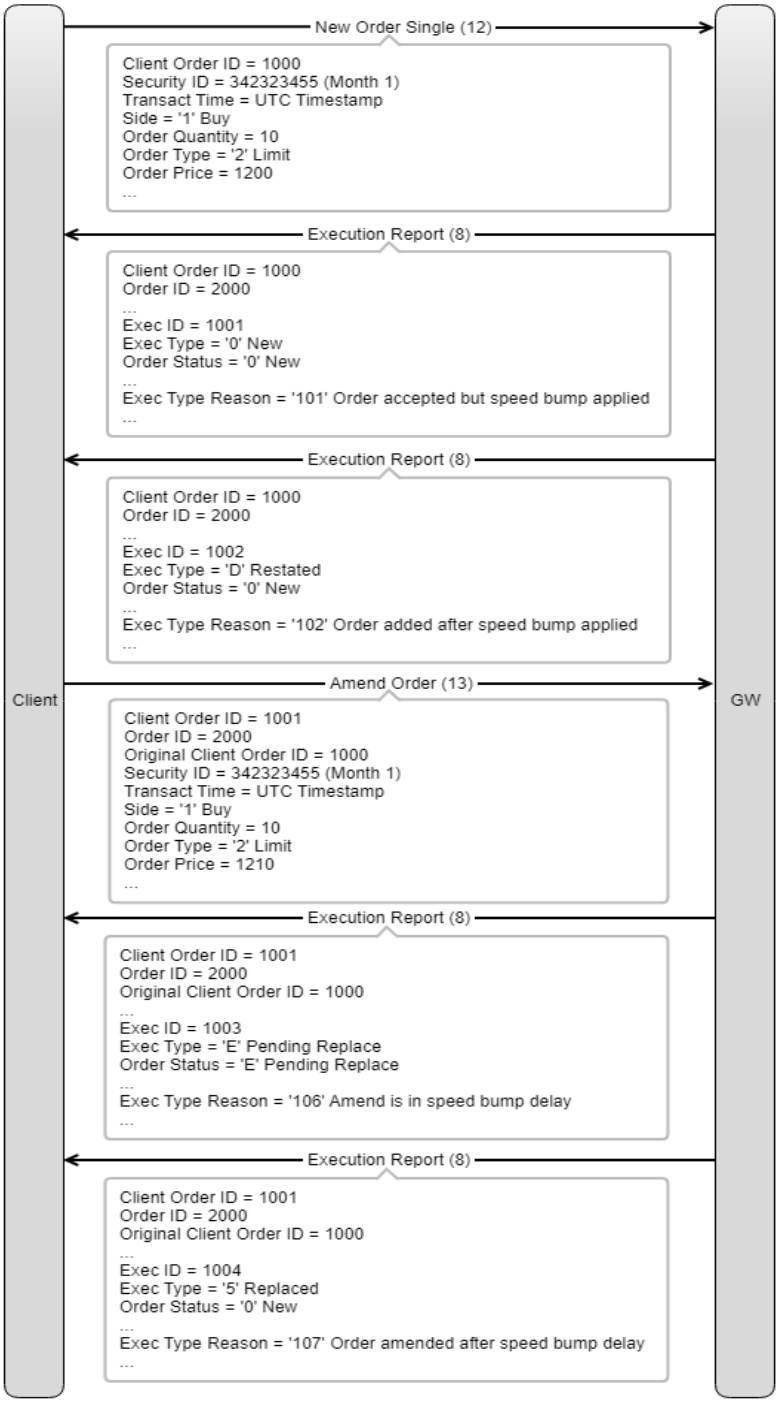
When the order is replaced the Execution Report includes Exec Type = '5' Replaced and Exec Type Reason = '105' Order updated after speed bump delay.


Client
GW
New Order Single (12)
Client Order ID = 1000
Security ID = 342323455 (Month 1)
Transact Time = UTC Timestamp
Side = '1' Buy
Order Quantity = 10
Order Type = '2' Limit
Order Price = 1200
...
Execution Report (8)
Client Order ID = 1000
Order ID = 2000
...
Exec ID = 1001
Exec Type = '0' New
Order Status = '0' New
...
Exec Type Reason = '101' Order accepted but speed bump applied
...
Amend Order (13)
Client Order ID = 1001
Order ID = 2000
Original Client Order ID = 1000
Security ID = 342323455 (Month 1)
Transact Time = UTC Timestamp
Side = '1' Buy
Order Quantity = 8
Order Type = '2' Limit
Order Price = 1200
...
Execution Report (8)
Client Order ID = 1001
Order ID = 2000
Original Client Order ID = 1000
...
Exec ID = 1002
Exec Type = 'E' Pending Replace
Order Status = 'E' Pending Replace
...
Exec Type Reason = '104' Original order is in speed bump enforced delay
...
Execution Report (8)
Client Order ID = 1000
Order ID = 2000
...
Exec ID = 1003
Exec Type = 'D' Restated
Order Status = 'E' Pending Replace
...
Exec Restatement Reason = '99' Other
Exec Type Reason = '102' Order added after speed bump
...
Execution Report (8)
Client Order ID = 1001
Order ID = 2000
Original Client Order ID = 1000
...
Exec ID = 1004
Exec Type = '5' Replaced
Order Status = '0' New
...
Exec Type Reason = '105' Order updated after speed bump delay
...

Executable order amendment for a resting order will be speed bumped

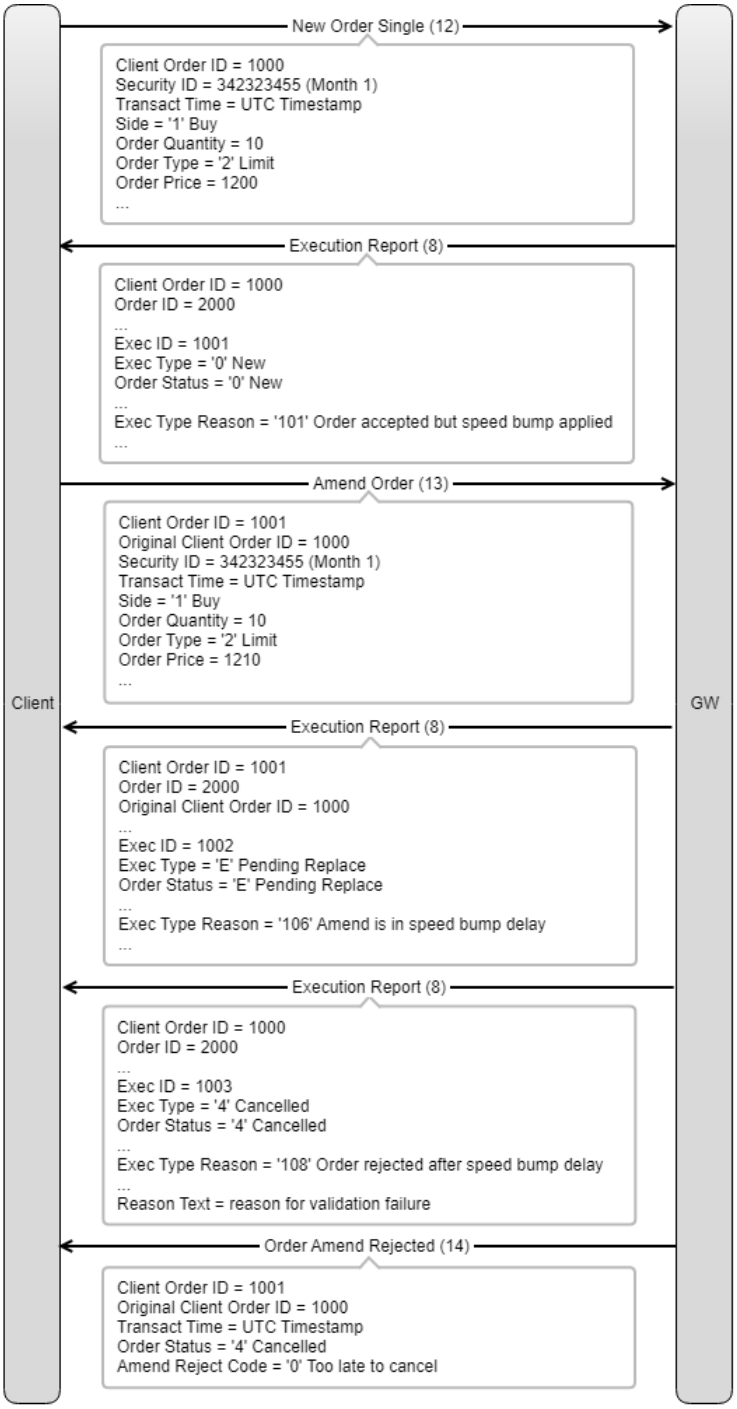
An order amendment is submitted for a resting order that was previously speed bumped. The Amend Order is speed bumped as the amended order will not provide liquidity. The Execution Report for the amendment includes Exec Type= 'E' Pending Replace with and Exec Type Reason = '106' Amend is in speed bump delay.

When the order is replaced the Execution Report includes Exec Type = '5' Replaced and Exec Type Reason = '107' Order amended after speed bump delay.



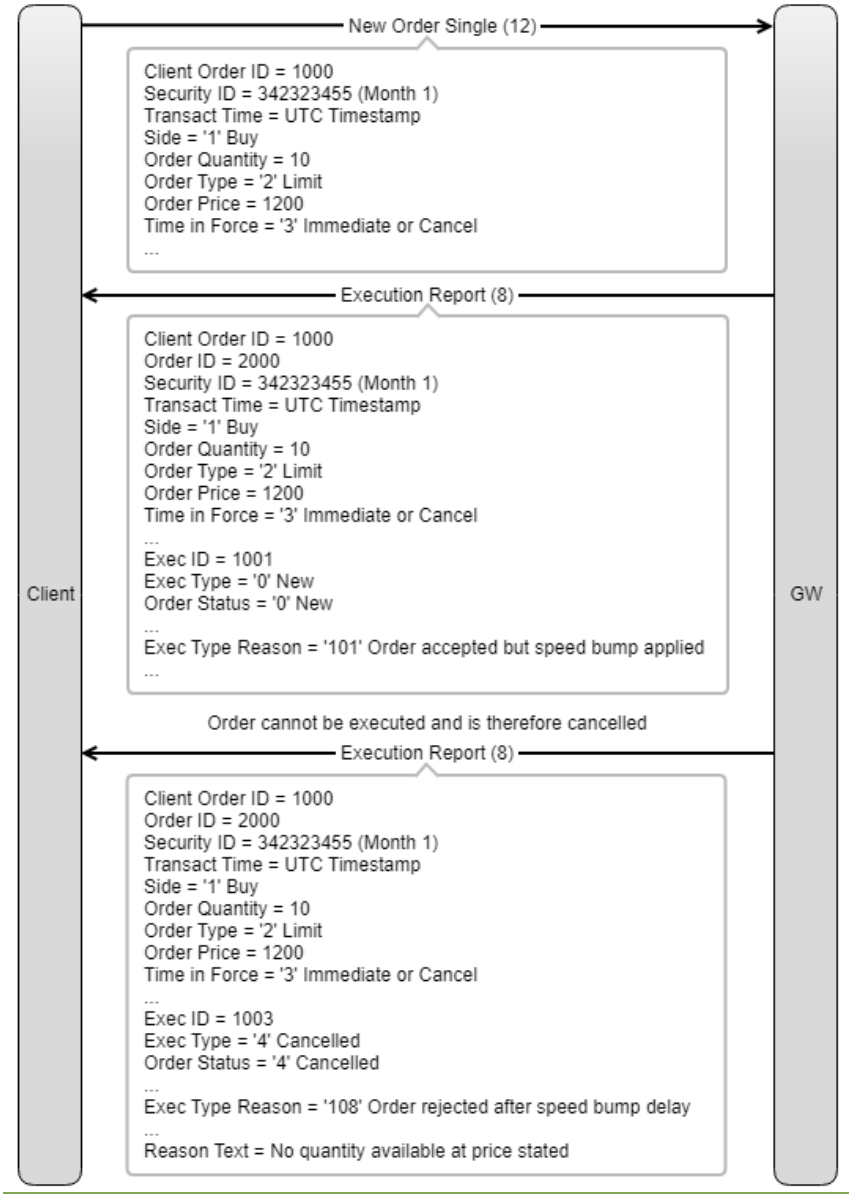
Speed bumped order is cancelled due to validation failure (inflight speed bumped ~~revision~~amendment is also cancelled)

An order is submitted which is subject to a speed bump. An Amend Order (13) is accepted which is also subject to speed bump conditions. The original order submission fails business validation on clearing the speed bump and is cancelled. The Execution Report includes Exec Type = '4' Cancelled and Exec Type Reason = '108' Order rejected after speed bump delay with the reason for the business validation failure in Reason Text. An Order Amend Rejected (14) is sent for the order amend.



Immediate or Cancel speed bumped order fails validation and is cancelled

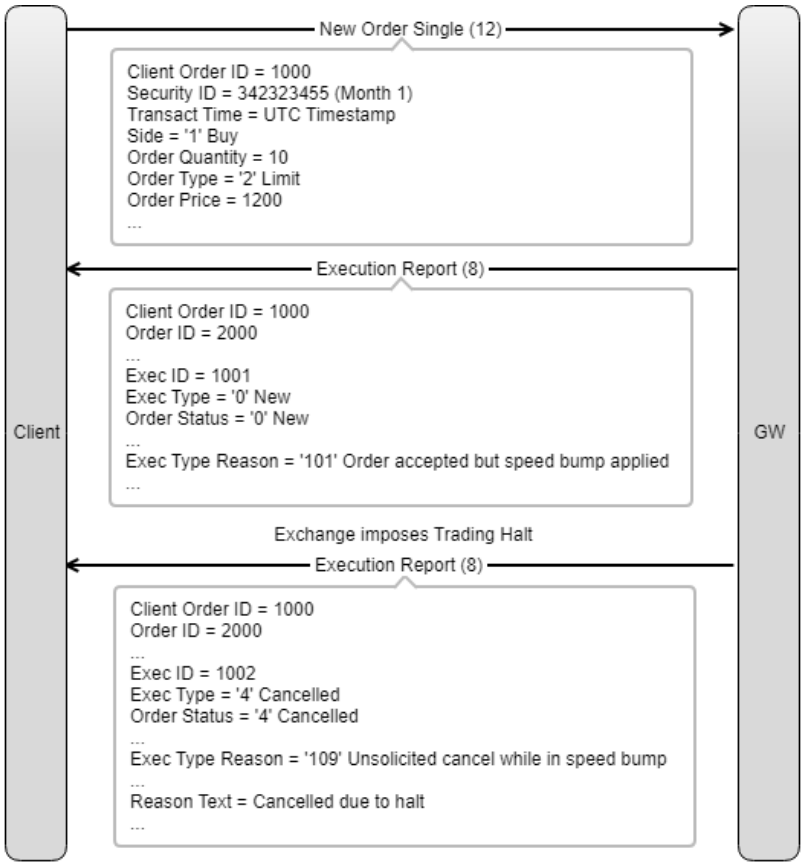
An Immediate or Cancel order is submitted which is subject to a speed bump. The order fails validation on clearing the speed bump as it cannot be executed and is therefore cancelled. The Execution Report includes Exec Type = '4' Cancelled and Exec Type Reason = '108' Order rejected after speed bump delay with the reason in Reason Text = No quantity available at price stated.



Note: In the absence of a speed bump an IOC will be rejected if no quantity is available at the price stated.

Unsolicited order cancellation while in speed bump

An order is submitted which is speed bumped. While the order is in the speed bump, the Exchange invokes a Trading Halt and all orders are pulled. The Execution Report sent for the order in the speed bump includes Exec Type) = '4' Cancelled with Exec Type Reason = '109' Unsolicited cancel while in speed bump and Reason Text = ~~Trading Halt~~Cancelled due to halt.



~~3.18~~3.17 Message Throttling

The Exchange imposes a message throttle which limits the maximum number of messages that can be submitted per second by a Comp ID using the following messages:

- New Order Single (12)
- Amend Order (13)
- *Quote Request (20)*
- Security Definition Request (10)
- *Mass Quote (22)*
- *MMP Reset Request (30)*.

Security Definition Requests are included in the message throttle but also have their own throttle limits.

Each Mass Quote message is counted as a single message irrespective of the number of quote pairs present in the message.

Messages submitted in excess of the throttle limit in any given whole second will result in those messages being rejected by the gateway and will be notified by a Business Message Reject (7).

Note, Cancel Order (15) and Mass Cancel (17) messages are exempt from throttling.

~~3.19~~3.18 Security Definition Throttle

The number of Security Definition Request (10) messages that can be submitted by a Comp ID is set at per day rate and also included in the per second message throttle. A user breaching the daily limit will have further Security Definition Request (10) submissions rejected by the gateway.

3.19 Merged Order Books

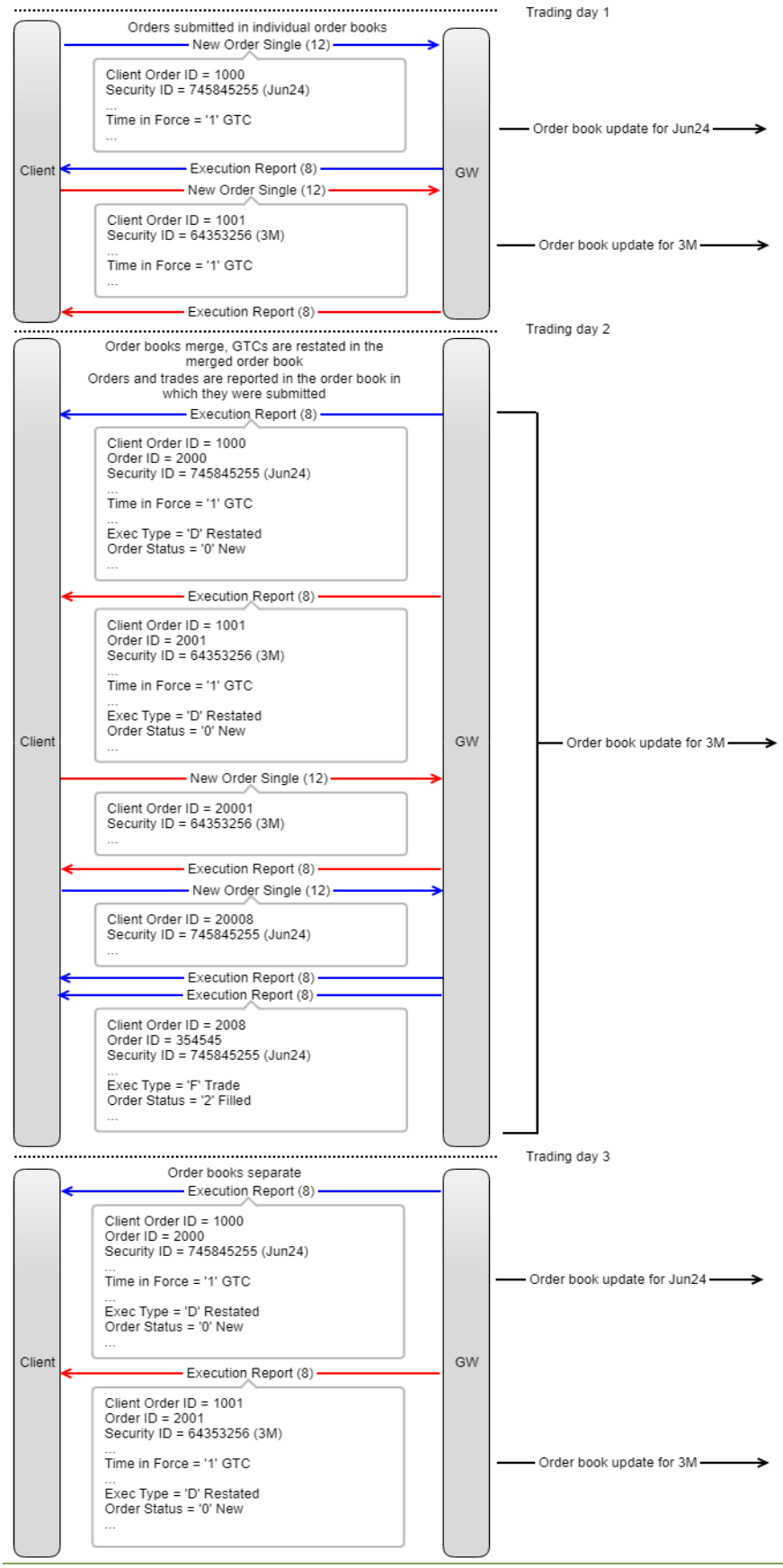
The LME prompt date structure for futures is such that two different prompts can share the same actual date on specific trading dates, for example, on the 3rd Wednesday of a month a 3M rolling prompt date will have the same prompt date as the monthly prompt date. On the trading date on which the prompts share the same actual date prompt, the order books for both prompts will be merged. TOM and Cash prompts will never merge.

The merging of order books only affects execution and market data publication. Both prompt dates in the merged order book will be available for order entry.

One prompt date in the merged order book will take precedence over the other. A rolling prompt date will always take precedence over a single prompt date. The instrument identifier of the prompt with precedence will be used by the Market Data service.

GTC and GTD orders will be merged into the order book with precedence and will return to the order book into which they were entered when the order books are no longer merged.

Strategy order books that include a leg in a merged order book will also merge.


Trading day 1
Orders submitted in individual order books
New Order Single (12)
Client Order ID = 1000
Security ID = 745845255 (Jun24)
...
Time in Force = '1' GTC
...
Order book update for Jun24
Client
Execution Report (8)
GW
New Order Single (12)
Client Order ID = 1001
Security ID = 64353256 (3M)
...
Time in Force = '1' GTC
...
Order book update for 3M
Execution Report (8)
Trading day 2
Order books merge, GTCs are restated in the merged order book
Orders and trades are reported in the order book in which they were submitted
Execution Report (8)
Client Order ID = 1000
Order ID = 2000
Security ID = 745845255 (Jun24)
...
Time in Force = '1' GTC
...
Exec Type = 'D' Restated
Order Status = '0' New
...
Execution Report (8)
Client Order ID = 1001
Order ID = 2001
Security ID = 64353256 (3M)
...
Time in Force = '1' GTC
...
Exec Type = 'D' Restated
Order Status = '0' New
...
Client
GW
Order book update for 3M
New Order Single (12)
Client Order ID = 20001
Security ID = 64353256 (3M)
...
Execution Report (8)
New Order Single (12)
Client Order ID = 20008
Security ID = 745845255 (Jun24)
...
Execution Report (8)
Execution Report (8)
Client Order ID = 2008
Order ID = 354545
Security ID = 745845255 (Jun24)
...
Exec Type = 'F' Trade
Order Status = '2' Filled
...
Trading day 3
Order books separate
Execution Report (8)
Client Order ID = 1000
Order ID = 2000
Security ID = 745845255 (Jun24)
...
Time in Force = '1' GTC
...
Exec Type = 'D' Restated
Order Status = '0' New
...
Order book update for Jun24
Client
Execution Report (8)
GW
Client Order ID = 1001
Order ID = 2001
Security ID = 64353256 (3M)
...
Time in Force = '1' GTC
...
Exec Type = 'D' Restated
Order Status = '0' New
...
Order book update for 3M

3.20 Self Execution Prevention (SEP)

A member can guard against traders in their organisation executing orders with each other.

A member can use SEP functionality without configuring a SEP handling action in which case the Exchange configured response type would be triggered to cancel the incoming order. Alternatively a member can configure SEP identifiers and specify the action to be taken if two orders with an identical SEP ID could execute.

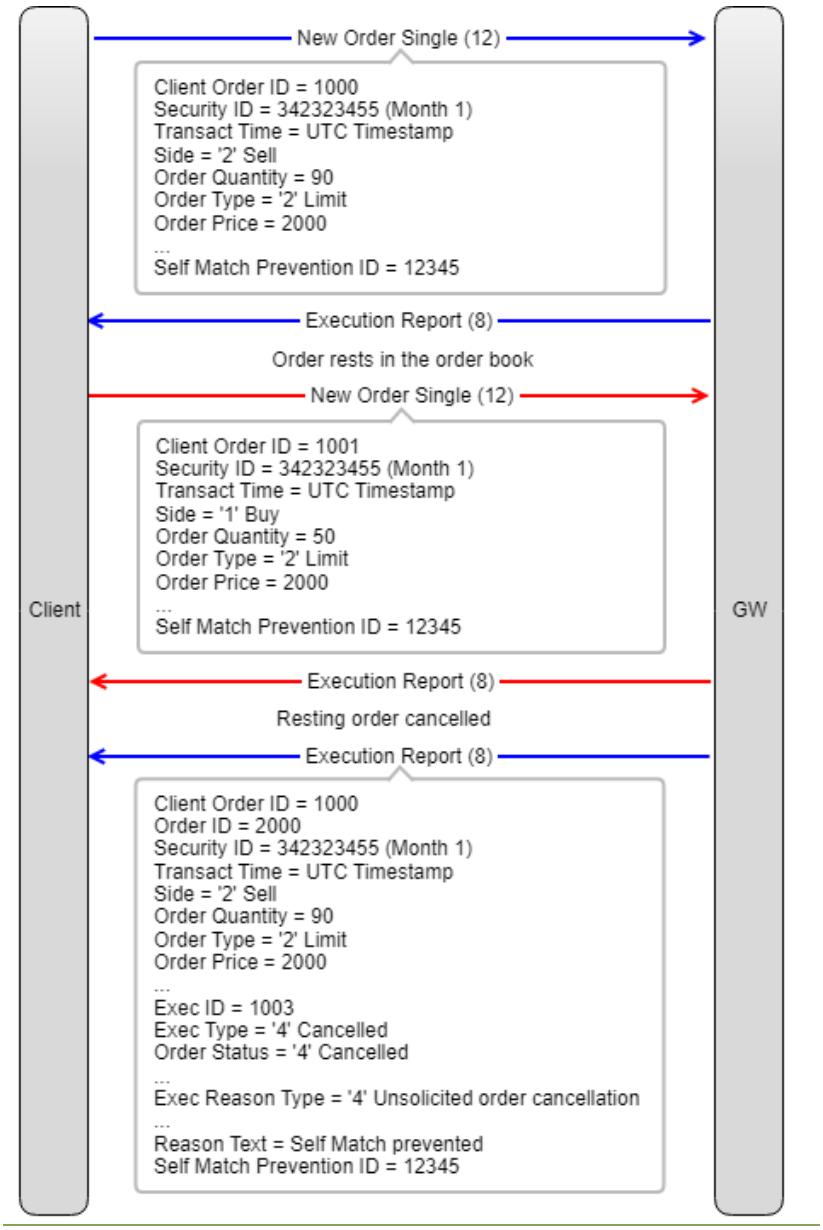
A SEP ID will be specified as a maximum of 9 digits. A Member Risk Manager can use the Risk Management interface to define the SEP configuration as described in the Risk Management Gateway FIX Specification. This configuration will be effective from the next trading day.

A SEP ID can be entered in the Self Match Prevention ID on order submission. If orders with an identical SEP ID from the same member firm can cross the SEP handling action that has been configured is triggered to cancel either the incoming or resting order or both (incoming and resting).

The Execution Report sent for the cancelled order will contain Reason Text = Self Match prevented.

The availability of SEP functionality will be determined by the Exchange. If an order is submitted with the Self Match Prevention ID populated and SEP is not available for the Security ID specified, the order will be rejected. The Execution Report (8) sent will contain Reason Text = Self Match Prevention not configured for the tradable instrument.

Self Execution Prevention triggered – resting order cancelled



3.21 Market Maker Protection (MMP)

Market Maker Protection will be available to permissioned trading users. A Member Risk Manager will use the Risk Management interface to specify the level of protection that should apply to a trading user in a particular contract as described in the Risk Management Gateway FIX Specification.

The Member Risk Manager will specify the protection type and protection limit measured over a configured time period which is defined in seconds. This time period defines the length of the rolling time interval for MMP recalculation which is used to determine if the quantity limit has been reached.

The following protection types can be configured:

- *Cumulative percent over time - Total percentage of orders executed within the configured time period*
- *Volume over time - Total count of volume executed within the configured time period*
- *Number of tradable instruments traded over time - Total count of option strikes within the configured time period.*

If an MMP limit is breached the protection response is triggered to pull orders and reject further orders until MMP is explicitly reset by the trading user using an MMP Reset Request (30). The MMP reset will only affect the MMP limit that has been breached.

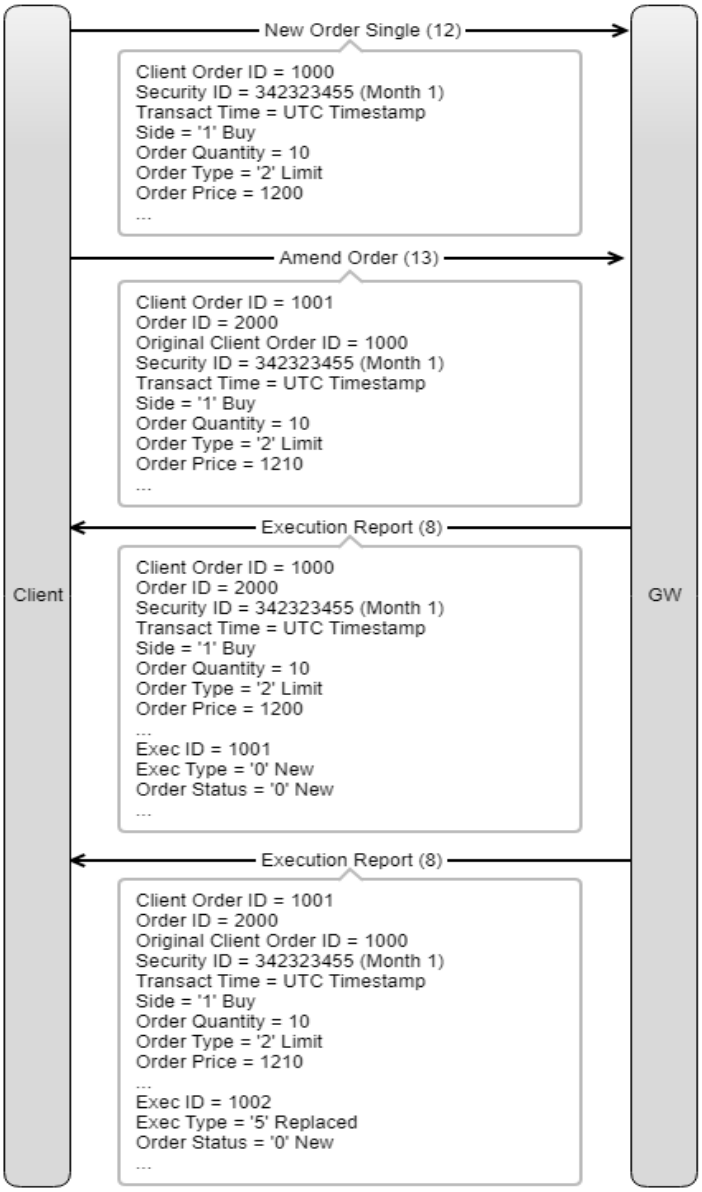
Note, whenever a protection response is triggered, the corresponding trading user will be notified by a News (40) message.

Once MMP is reset order and quote submission can resume.

3.22 Inflight Order Processing

The gateway will accept a single inflight amend or cancellation request whilst processing a new order. The amend request is queued until the preceding request has been processed. Multiple inflight messages will be rejected.

For example, a New Order Single (12) is submitted followed immediately afterwards by an Amend Order (13). An Execution Report (8) is returned for the order submission and then the amendment.



Refer to Appendix A: Inflight Order Handling and Appendix B: Speed Bump Inflight Order Handling in the Order Entry Gateway FIX Specification for more examples.

3.23 Trade Reporting

When an outright order matches, a trade half will be assigned an identifier which will be reported in the Trade ID on the Execution Report (8).

A strategy trade half will be reported in a single Execution Report including the leg details. The legs of strategy trade will be assigned a Leg Allocation ID which will be shared with either the Leg Allocation ID of another strategy trade or the Trade ID of an outright trade.

3.24 Client ID Usage

Of the following client identifiers only two can be specified for an order *or Mass Quote*:

- Client ID Short Code
- Legal Entity ID
- Proprietary Client ID.

4 Message Definitions

4.1 Inbound Messages

- Logon (5)
- Heartbeat (0)
- Test Request (1)
- Resend Request (2)
- Sequence Reset (4)
- Logout (6)
- Security Definition Request (10)
- New Order Single (12)
- Amend Order (13)
- Cancel Order (15)
- Mass Cancel Request (17)
- *Mass Quote (22)*
- *Quote Request (20)*
- *MMP Reset Request (30).*

4.2 Outbound Messages

- Logon (5)
- Heartbeat (0)
- Test Request (1)
- Resend Request (2)
- Sequence Reset (4)
- Logout (6)
- Reject (3)
- Business Message Reject (7)
- News (40)
- Security Definition (11)
- Order Amend Rejected (14)
- Order Cancel Rejected (16)
- Execution Report (8)
- Mass Cancel Report (18)
- *Mass Quote Ack (23)*

- *Quote Request ~~Rejected~~Ack (21)*
- *MMP Reset Ack (31).*

4.3 Data Types

Data Type	Size (bytes)	Format
Char	1	ASCII Alphanumeric Permitted ASCII characters are A-Z, a-z, 0-9, underscore('_') and space(' ')
String	*(n)*	Fixed length. These fields use standard Char bytes. All fields of this data type will be null terminated and the length of the field will include this null character. If the field value does not occupy the full length of the field, data after the null termination should be discarded. For incoming messages to the gateway: • If the field value occupies the full length of the field and does not include null character, the gateway will override the last character with null value. • In case the field is empty, the first byte will be null filled. This is to indicate that the field is not applicable / not filled in.
UInt8	1	Unsigned integer. Minimum value = 0 Maximum value = 254 Null value = 255
Int8	1	Signed integer. Minimum value = -127 Maximum value = 127 Null value = -128
UInt16	2	Little Endian encoded unsigned integer. Minimum value = 0 Maximum value = 65,534 Null value = 65,535
Int16	2	Little Endian encoded signed integer. Minimum value = -32,767

Data Type	Size (bytes)	Format
		Maximum value = 32,767 Null value = -32,768
UInt32	4	Little Endian encoded unsigned integer Minimum value = 0 Maximum value = 4,294,967,294 Null value = 4,294,967,295
Int32	4	Little Endian encoded signed integer. Minimum value = -2,147,483,647 Maximum value = 2,147,483,647 Null value = -2,147,483,648
UInt64	8	Little Endian encoded 64 bits signed integer Minimum value = 0 Maximum value = 18,446,744,073,709,551,614 Null value = 18,446,744,073,709,551,615 Note: Timestamps will be represented as UTC up to microsecond precision with the nanosecond element being represented by trailing zeros.
Int64	8	Little Endian encoded 64 bits unsigned integer Minimum value = -9,223,372,036,854,775,807 Maximum value = 9,223,372,036,854,775,807 Null value = -9,223,372,036,854,775,808 Note: Prices will support 6 implied decimals.
Bitmap Fixed Length	32	Bitmap Fixed Length provides up to 256 representation options. To indicate availability, set 1 to the applicable bit position and 0 for unavailability. Each bit in the presence map will represent a field and the sequence in which the fields should be included into the message will be based on the bit position (starting from the most significant bit position).
Bitmap Variable Length	*(n)*	Bitmap Variable Length is used to indicate the presence of fields and nested repeating blocks in a repeating block. To indicate availability set 1 to the applicable bit position and 0 for unavailability. The length of the bitmaps used for different repeating blocks may vary.

4.4 Message Composition

Each message comprises of the following logical components:

1. Header
2. Body
3. Trailer

Fields within a message are formed in the same order as the composition given above.

The header has a Fields Presence Map which is used to indicate the fields that are present in the body of the message.

Fields that are part of the header and the trailer are considered mandatory.

4.4.1 Fields Presence Map

The binary protocol provides a concept of fields presence maps per each message type where using these bitmap fields present in the message, the message sender could indicate actual fields that are present within the message in a dynamic nature.

Each bit in the presence map will represent a field and the sequence in which the fields should be included into the message will be based on the bit position (starting from the most significant bit position). All fields applicable to a particular presence map should be included in the message immediately following the applicable presence map.

For example, consider an 8 bit presence map. 1st, 2nd and 3rd positions indicate Security ID, Client Order ID and Order Quantity respectively where rest of the positions have not been assigned to a field.

To indicate the presence of the fields Security ID and Order Quantity the presence map will be set as shown below:

Bit Position (BP)	0	1	2	3	4	5	6
Represented field	Security ID	Client Order ID	Order Quantity	N/A	N/A	N/A	N/A
Bit value (presence)	1	0	1	0	0	0	0

Message view:

Preceding fields of the message							
Presence Map	1	0	1	0	0	0	0
Security ID	12345						
Order Quantity	1000						

Succeeding fields of the message

The applicable data types and lengths of the body fields are provided in each message. Based on the available fields as indicated by the fields presence map, the recipient of the message is expected to decode the message accordingly.

Bit position for a field that commonly appears in multiple messages may be different; each message may have its own bit position for individual fields present in that message.

4.4.2 Repeating Blocks and Nested Repeating Blocks

The binary message protocol supports repeating blocks within the message body while also allowing nested repeating blocks within a repeating block.

When indicating a repeating block, the fields presence map will only indicate the presence of the repeating block. Based on the repeating block construct, the receiving party is expected to evaluate the field contents and the numbers of repeating blocks.

This specification describes the repeating block construct and the relevant field information such as the data types required to identify the message contents and also to calculate header and trailer information such as message length and checksums.

Each repeating block construct will have a repeating block header field which is immediately followed by a fields presence map which will indicate the presence of the applicable fields in that repeating block and any nested repeating blocks included within.

For example, consider an 8-bit fields presence map included in the message header of a Mass Quote:

BP	0	1	2	3	4	5	6
Represented field	Quote ID	Transaction Time	No Quote Sets	N/A	N/A	N/A	N/A
Bit value (presence)	1	1	1	0	0	0	0

Position 2 indicates a repeating block which indicates the number of quote sets (i.e. number of quote pairs) present in the Mass Quote message.

A sample No Quote Sets repeating block construct is given below:

No Quote Sets	Number of Quote Set repeating blocks. Valid values are 1 or n. (Repeating block header field)
Quote Set Repeating Group Fields Presence Map	This will indicate the fields/nested repeating blocks present in this repeating block

0	Security ID		Tradable Instrument identifier.
1	No Quote Entries		Number of Quote Entry repeating blocks. Valid values are 1 to 3. (Repeating block header field)
	Quote Entry Fields Presence Map		This will indicate the fields/nested repeating blocks present in this repeating block
	0	Quote Entry ID	
	1	Quote Price Level	
	2	Bid Size	
	3	Offer Size	
	4	Bid Price	
	5	Offer Price	

In the above message construct, No Quote Entries is a nested repeating block within the No Quote Sets repeating block.

4.5 Required Fields

The following conventions are used for fields in the message definitions:

Y	Mandatory
C	Conditionally required based on a specified condition or presence of another field
N	Not required / optional

4.6 Message Header

Seq	Field Name	Req	Data Type	Description
1	Start of Message	Y	UInt8	Indicates the starting point of a message. Always set to the ASCII STX character (0x02).
2	Length	Y	UInt16	Length of the message including all the fields in the message (i.e. length of all header, body and trailer fields)
3	Message Type	Y	UInt8	Defines the message type.

Seq	Field Name	Req	Data Type	Description
4	Sequence Number	Y	UInt32	Message sequence number.
5	PossDup	Y	UInt8	Indicates whether the message was previously transmitted with the same sequence number: Valid values: 0 = No (original transmission) 1 = Yes (possible duplicate)
6	PossResend	Y	UInt8	Indicates whether the message was previously transmitted under a different sequence number: Valid values: 0 = No (original transmission) 1 = Yes (possible resend)
7	Comp ID	Y	String (11)	Identifies the sender of the message.
8	Sending Time	Y	UInt64	Time the message is transmitted.
9	Original Sending Time	Y	UInt64	Time the message was originally transmitted. Applicable only if PossDup 1 = Yes (possible duplicate). If the original time is not available, this will be the same value as Sending Time. 0 value otherwise.
10	Body Fields Presence Map	Y	Bitmap Fixed Length	Indicates the list of fields that would be present immediately after this Body Fields Presence Map field.

4.7 Message Trailer

Seq	Field Name	Req	Data Type	Description
1	Checksum	Y	UInt32	CRC32C based checksum.

4.8 Administrative Messages

4.8.1 Logon (5)

The Logon request and response are used to authenticate the client and agree on the sequence numbers.

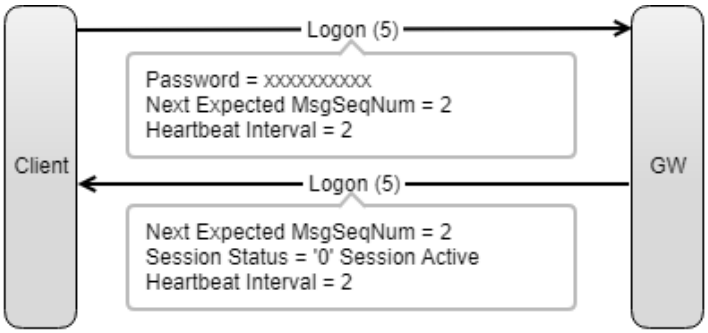
On initial logon the status of persisted orders is communicated by the publication of Execution Reports for all open orders.

The list of available tradable instruments for the current trading day will be published by the Market Data service independently.

BP	Field Name	Req	Data Type	Description
0	Password	C	String (450)	Encrypted Password assigned to the Comp ID. Conditionally required in the Logon message initiated by the client. Absent in the Logon message sent by the gateway.
1	New Password	N	String (450)	New encrypted Password for the Comp ID. May be present only in the Logon message initiated by the client
2	Next Expected MsgSeqNum	Y	UInt32	Next expected message sequence number to be received.
3	Session Status	C	UInt8	Status of the binary session. Valid values: 0 = Session active 1 = Session password change Conditionally required in the Logon message ~~initiated~~sent by the gateway.
4	Heartbeat Interval	Y	UInt32	Heartbeat interval in seconds.

Example Message Flow

Initial Logon



4.8.2 Heartbeat (0)

Heartbeat is sent at the interval specified in Logon (5). It is also sent in response to a Test Request (1).

BP	Field Name	Req	Data Type	Description
0	Reference Test Request ID	C	String (21)	Conditionally required if the Heartbeat is in response to a Test Request. The value in this field will echo the Test Request ID received in the Test Request.

4.8.3 Test Request (1)

Test Request can be sent by either the client or gateway to verify a connection is active. The recipient responds with a Heartbeat (0).

BP	Field Name	Req	Data Type	Description
0	Test Request ID	Y	String (21)	Identifier included in Test Request message to be returned in resulting Heartbeat.

4.8.4 Resend Request (2)

Resend Request is used to initiate the retransmission of messages if a sequence number gap is detected.

To request a single message. The Start Sequence and End Sequence should be the same.

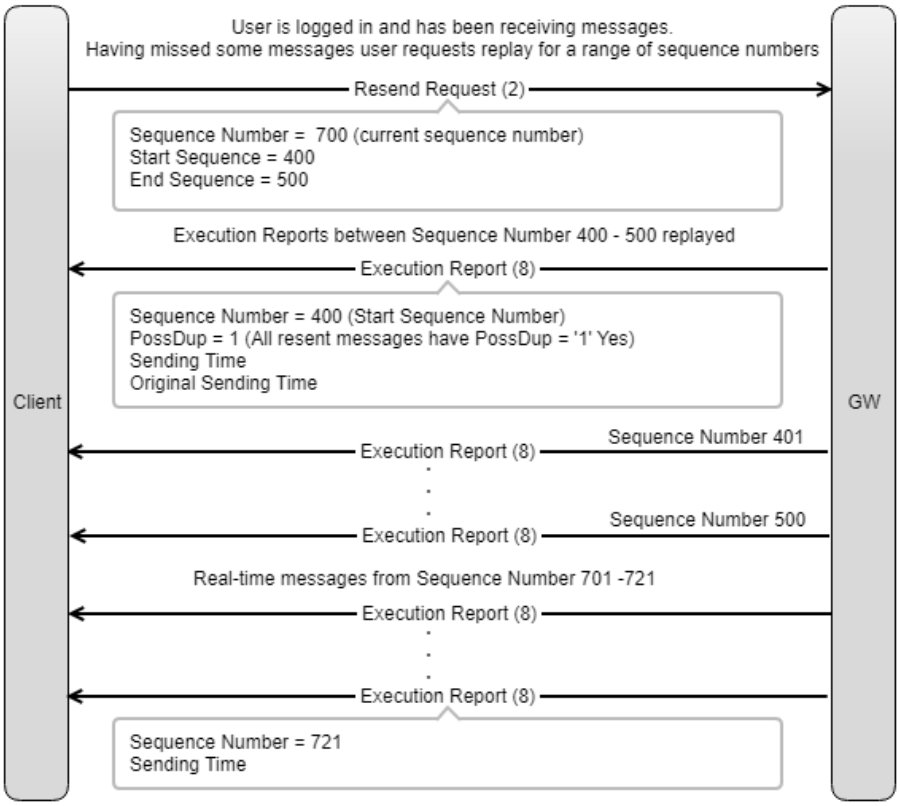
To request a specific range of messages. The Start Sequence should be the first message of the range and the End Sequence should be the last of the range.

To request all messages after a particular message. The Start Sequence should be the sequence number immediately after that of the last processed message and the End Sequence should be zero (0)

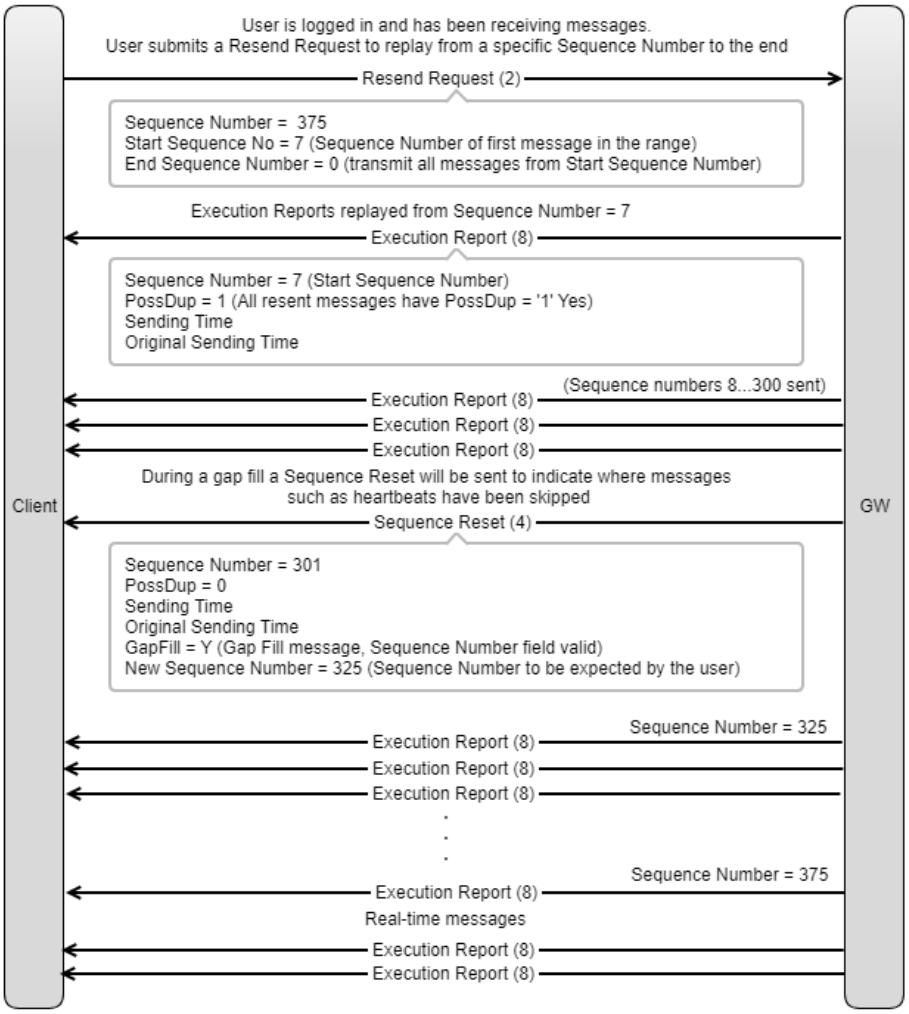
BP	Field Name	Req	Data Type	Description
0	Start Sequence	Y	UInt32	Sequence number of the first message expected to be resent.
1	End Sequence	Y	UInt32	Sequence number of the last message expected to be resent. This may be set to 0 to request the sender to transmit ALL messages starting from Start Sequence Number.

Example Message Flows

Resend Request for a range of messages

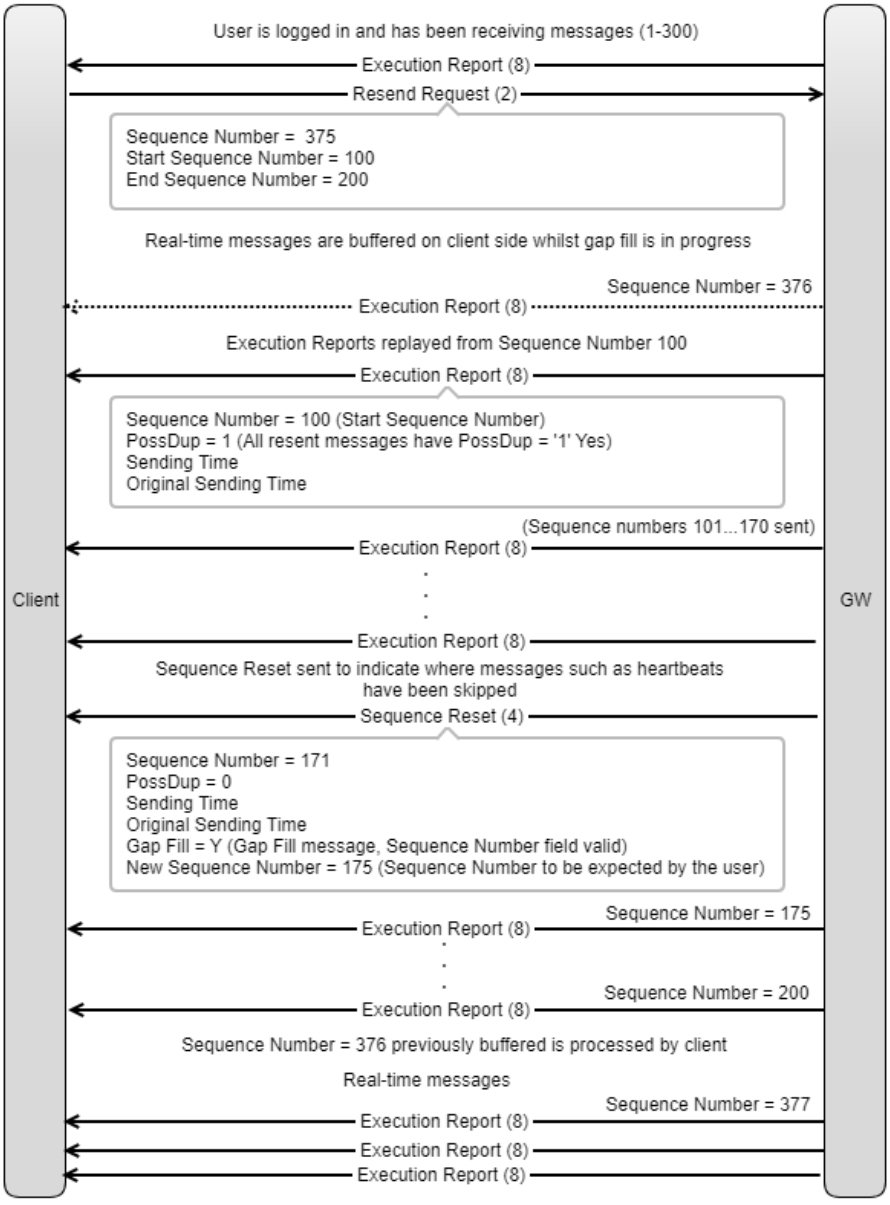


Resend Request for all messages after a particular message



Resend Request - incoming message buffered by Client

A Resend Request is submitted but before gap fill messages have been transmitted an incoming message is received. The client will hold the message until all the gap fill messages have been received and then process the buffered message. All messages should be processed in sequence number order.



4.8.5 Sequence Reset (4)

Sequence Reset allows the client or the gateway to increase the expected incoming sequence number of the other party.

In a Gap Fill it is sent as notification of the next sequence number to be transmitted.

BP	Field Name	Req	Data Type	Description
0	Gap Fill	N	Char	Indicates whether the sequence number is to be interpreted in Reset mode or Gap Fill mode: Valid values: N = Reset (ignore Sequence Number) Y = Gap Fill (Sequence Number valid)
1	New Sequence Number	Y	UInt32	Sequence number of the next message to be transmitted.

4.8.6 Logout (6)

Logout initiates or confirms the termination of a client session. Clients should terminate their sessions gracefully by logging out.

If a user is disabled by LME Market Operations while logged in then a Logout message will be sent to the user and the session will be disconnected.

If a user has their password reset by LME Market Operations and attempts to login with their previous password, the user will receive a Logout with Session Status = '100' Password change is required.

BP	Field Name	Req	Data Type	Description
0	Session Status	C	UInt8	Status of the binary session. Valid values: 3 = New session password does not comply with the policy 4 = Session logout complete 5 = Invalid username or password 6 = Account locked 7 = Logons are not allowed at this time 8 = Password expired 100 = Password change is required 101 = Other Conditionally required only if the message is generated by the gateway.
1	Logout Text	C	String (76)	Reason for the Logout.

BP	Field Name	Req	Data Type	Description
				Conditionally required if Session Status = '101' Other

4.8.7 Reject (3)

Reject will be sent when a message is received but cannot be properly processed by the gateway due to a session level rule violation.

BP	Field Name	Req	Data Type	Description
0	Message Reject Code	Y	UInt16	Code specifying the reason for the session level rejection: Valid values: 1 = Required field missing 2 = Field not defined for this message 3 = Undefined field 4 = Field specified without a value 5 = Value is incorrect for this field 6 = Incorrect data format for value 9 = Comp ID problem 10 = Sending Time Accuracy problem 11 = Invalid message type 13 = Field appears more than once 99 = Other
1	Reference Message Type	N	UInt8	Message type of the rejected message.
2	Reference Field Name	N	String (50)	Name of the field which caused the rejection.
3	Reference Sequence Number	Y	UInt32	Sequence number of the message which caused the rejection.
4	Reason Text	N	String (76)	Text specifying the reason for the rejection.

4.9 Other Messages

4.9.1 Business Message Reject (7)

Once an application level message passes validation at session level it will then be validated at business level. If business level validation detects an error condition then a rejection should be issued. Many business level messages have specific fields for rejection handling where a specific field is not available the Business Message Reject message will be returned.

BP	Field Name	Req	Data Type	Description
0	Business Reject Code	Y	UInt16	Code specifying the reason for the rejection of the message: Valid values: 0 = Other ~~1 = Unknown ID~~ 2 = Unknown Security 3 = Unspecified Message Type ~~4 = Application not available~~ 5 = Conditionally required field missing 8 = Throttle limit exceeded 9 = Throttle limit exceeded, session will be disconnected
1	Reason Text	N	String (76)	Text specifying the reason for the rejection.
2	Reference Message Type	Y	UInt8	Message type of the rejected message.
3	Reference Field Name	N	String (50)	Name of the field which caused the rejection.
4	Reference Sequence Number	N	UInt32	Sequence number of the message which caused the rejection.
5	Business Reject Reference ID	N	String (21)	Client specified unique identifier on the message that was rejected. For example, for a New Order Single this would be the client specified identifier in the Client Order ID.

4.9.2 News (40)

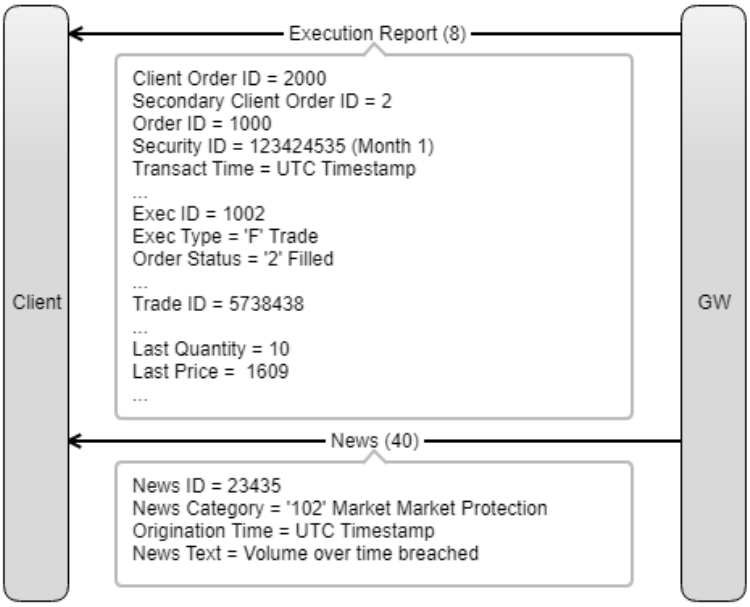
A News message is a general free format message from the exchange. *A News message is also sent in response to a market maker protection breach, see* 3.19 Market Maker Protection (MMP).

BP	Field Name	Req	Data Type	Description
0	News ID	Y	String (21)	Unique identifier assigned for the News message.
1	News Category	Y	UInt8	Category of the News. Valid values: 101 = Market message

BP	Field Name	Req	Data Type	Description
				102 = Market Maker Protection
2	Origination Time	Y	UInt64	Time of message origination. UTC Timestamp
3	News Text	Y	String (251)	Free text field for Market message *or one of the following for Market Maker Protection:* • *Cumulative percent over time breached* • *Volume over time breached* • *Number of tradable instruments traded over time breached*

Example Message Flow

Market Maker Protection breached



4.10 Application Messages

4.10.1 Security Definition Request (10)

Security Definition Request is used to request the creation of either an option strike or a strategy.

BP	Field Name	Req	Data Type	Description
0	Security Request ID	Y	String (19)	Client specified unique identifier of the Security Definition Request.
1	Security Exchange	Y	String (5)	The market which is used to identify the security. XLME
2	Product Complex	Y	String (5)	Identifies an entire suite of products for a given market. Valid values: LME = Base ~~LMEP~~LPM = Precious
3	Symbol	Y	String (21)	Symbol for the LME contract code e.g. CADF (Copper Future) or OCDF (Copper Monthly Average Future).
4	Security Type	Y	UInt8	Indicates the type of security whether outright or strategy. Valid values: *2 = Option* 3 = Multi-leg instrument
5	Security Sub Type	Y	UInt8	Indicates the type of instrument to be created. Valid values: *0 = Outright* 1 = Carry *2 = Custom (Futures)* *3 = 3 Months Average* *4 = 6 Months Average* *5 = 12 Months Average* *6 = Carry Average* *7 = Call Spread* *8 = Put Spread* *9 = Custom (Delta Hedge)* *10 = Custom (Options)*

BP	Field Name		Req	Data Type	Description
6	*Maturity Date*		*C*	*UInt32*	*Expiration date for options.* *Conditionally required for Security Type = '2' Option.*
7	*Strike Price*		*C*	*Int64*	*Strike price for an option.* *Conditionally required for Security Type = '2' Option.*
8	*Put or Call*		*C*	*UInt8*	*Used to express option right* *Valid values:* *0 = Put* *1 = Call* *Conditionally required for Security Type = '2' Option.*
9	No Legs		C	UInt8	Number of legs repeating blocks. Cannot be less than *2 or more than 5. Note this will only be 1 for a 3 Month Average, 6 Month Average and 12 Month Average.* Conditionally required for Security Type = '3' Multi-leg instrument.
	Legs Body Fields Presence Map		C	Bitmap Variable Length (1)	Conditionally required if No Legs > 0 where each repeating group represents a leg in the multi-leg instrument.
	0	Leg Security ID	Y	UInt64	Security ID of the leg. *For an Average strategy, only the Security ID of the first leg of the strategy is provided as the other months are consecutive.*
	1	Leg Side	Y	UInt8	The side of this individual leg. Valid values: 1 = Buy 2 = Sell
	2	Leg Ratio	Y	UInt32	With 3 implied decimals. *For a delta hedge custom strategy, this is the delta used to determine the covering quantity.*

BP	Field Name		Req	Data Type	Description
					For all other strategies and also for an options leg in a delta hedge custom strategy this is the ratio of quantity for this individual leg relative to the entire multi-leg instrument. *For example, for a custom strategy such as a Butterfly the leg ratio would be 1:2:1, for the first leg Leg Ratio = 1 (buy near contract month), second leg Leg Ratio = 2 (sell two contracts in far month) and third leg Leg Ratio = 1 (buy one contract in yet farther month).* *For a Carry Average the front leg must include a ratio for the number of average legs. For example, 3M-3Q (Jul/Aug/Sep) Carry Average, 3M leg Leg Ratio = 3, legs 2/3/4 would have Leg Ratio = 1.*
	3	*Leg Price*	*C*	*Int64*	*Used to specify an anchor price for a leg. Not used for execution price.* *Conditionally required for the futures legs of Security Sub Type = '9' Custom (Delta Hedge) to specify the underlying futures price.*

4.10.2 Security Definition (11)

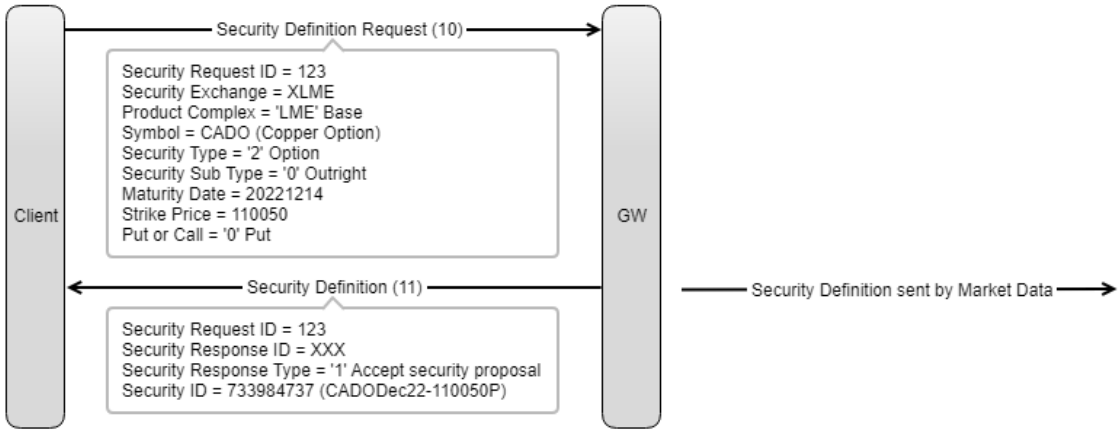
Security Definition will be returned to the originator of the Security Definition Request (10) to accept, accept with revisions or reject the creation of a tradable instrument. Market participants will be notified of a newly created instrument by the Market Data service.

BP	Field Name	Req	Data Type	Description
0	Security Request ID	Y	String (19)	Client specified unique identifier of the Security Definition Request.
1	Security Response ID	Y	String (21)	Unique ID assigned to Security Definition message.
2	Security Response Type	Y	UInt8	Type of Security Definition message response. Valid values: 1 = Accept security proposal

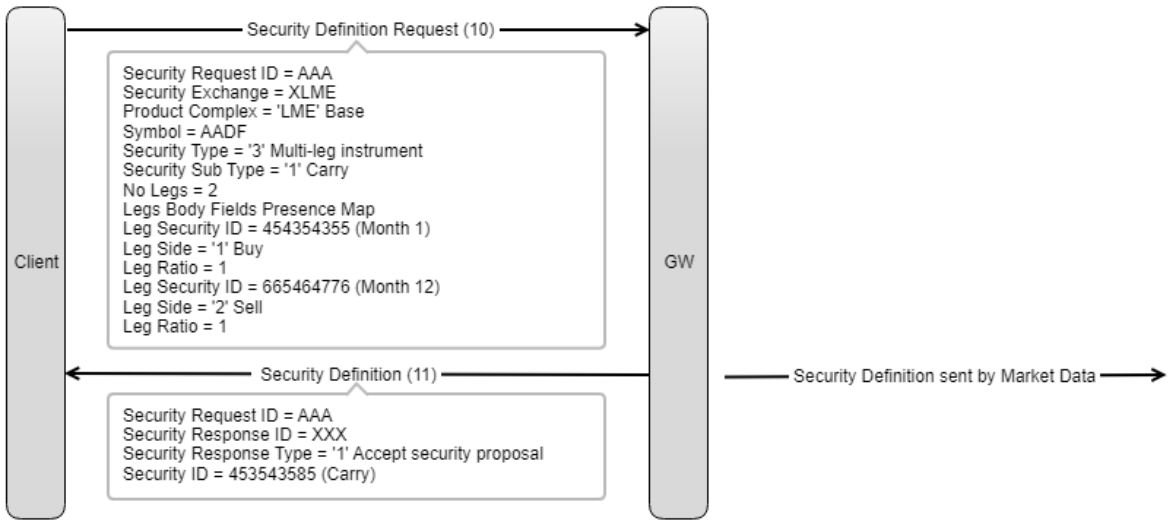
BP	Field Name	Req	Data Type	Description
				2 = Accept security proposal with revisions as indicated in the message 5 = Reject security proposal
3	Security Reject Reason	C	UInt8	Identifies the reason a security definition request is being rejected. Valid values: 12 = Invalid instrument structure specified 99 = Other 101 = Throttle limit exceeded 102 = Invalid strike price 103 = LegSecurityID (602) does not exist 104 = Invalid prompt date 105 = Invalid SecuritySubType.(762) Conditionally required if Security Response Type = '5' Reject security proposal.
4	Security ID	C	UInt64	Tradable Instrument identifier. Conditionally required if Security Response Type = '1' Accept security proposal or '2' Accept security proposal with revisions as indicated in the message.
5	Reason Text	C	String (76)	Text specifying the reason for the rejection. Conditionally required if Security Reject Reason = '99' Other.

Example Message Flows

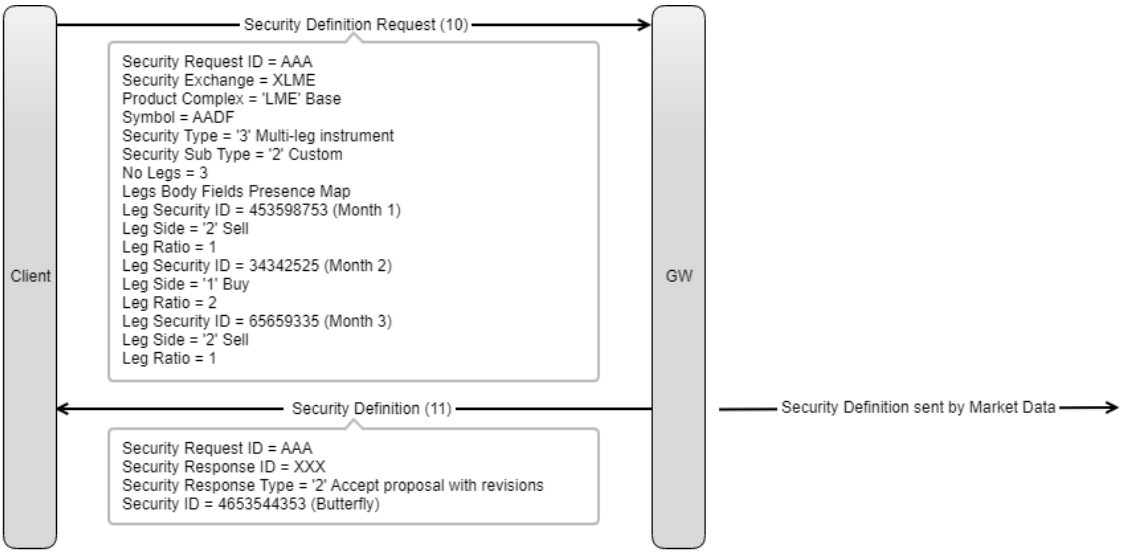
Option Strike Request



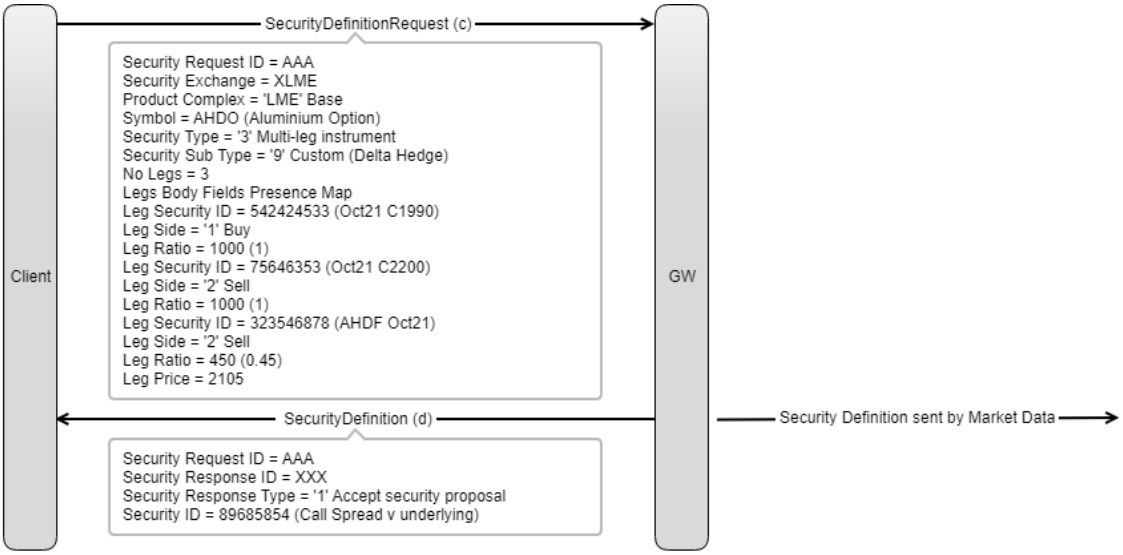
Futures Strategy Request



Inverse Custom Strategy Request



Delta Hedge Strategy Request – Call Spread versus underlying



A denominator of 1000 is applied to the Leg Ratio value which is shown in brackets.

4.10.3 New Order Single (12)

New Order Single is used to submit a new order for execution. An Execution Report (8), Reject (3) or Business Message Reject (7) is sent in response.

BP	Field Name	Req	Data Type	Description
0	Client Order ID	Y	String (19)	Client specified identifier of the order.
4	Security ID	Y	UInt64	Tradable Instrument identifier.
5	Transact Time	Y	UInt64	Time when the message was generated.
6	Side	Y	UInt8	Side of the order. Valid values: 1 = Buy 2 = Sell
7	Order Quantity	Y	Int32	Total quantity of the order.
8	Order Type	Y	UInt8	Order type applicable to the order. Valid values: 2 = Limit *3 = Stop Loss* 4 = Stop Limit *10 = Market to Limit* 11 = Iceberg

BP	Field Name	Req	Data Type	Description
				12 = Post Only *13 = One Cancels Other Market to Limit* *14 = One Cancels Other Limit*
9	Order Price	Y	Int64	Price of the order. *Must be null value if:* *Order Type = '3' Stop Loss* *Order Type = '10' Market to Limit.*
10	Time in Force	Y	UInt8	Specifies how long the order remains in effect. Valid values: 0 = Day 1 = Good Till Cancel (GTC) 3 = Immediate Or Cancel (IOC) *4 = Fill Or Kill (FOK)* 6 = Good Till Date (GTD)
11	Order Restrictions	Y	Char	Restrictions associated with an order. Valid values: D = Non-algorithmic (human) E = Algorithmic (algo)
12	Order Capacity	Y	Char	Indicates the trading capacity. Valid values: A (agency) = AOTC P (principal) = DEAL R (riskless principal) = MTCH
13	Account Type	Y	UInt8	Specifies the type of account associated with the order. Valid values: 1 = Client ISA 3 = House 8 = Joint back office account (JBO) = Gross OSA 101 = Client OSA For contracts assigned to the T4 booking model only 3 = House is valid whereas for the T2 booking model all account types are valid.

BP	Field Name	Req	Data Type	Description
~~14~~	~~Executing Firm~~	~~N~~	~~String (4)~~	~~Identifier of the executing firm.~~
15	Client ID Short Code	C	UInt64	Client short code identifier If there is no client this can be populated with the value '0' = No Client for Account Type = 3~~. If that is not available, should be populated with the value '0' = No Client~~. Not valid if populated with either 1, 2 or 3. Conditionally required for client orders i.e. Account Type = 1, 8 or 101.
16	Legal Entity ID	C	String (41)	LEI. ~~Conditionally required for client orders i.e. Account Type = 1, 8 or 101.~~
17	Proprietary Client ID	C	String (41)	Proprietary or Custom Client ID as assigned by the member. Conditionally required for client orders i.e. Account Type = 1, 8 or 101.
18	Entering Firm	N	String (4)	Identifier of the entering firm, a member mnemonic.
19	Origination Trader	Y	String (41)	Order origination trader.
20	Customer Account	C	String (~~6~~31)	Identification of the client account code where the Account Type = 1, 8 or 101.
21	Correspondent Broker	N	String (4)	~~ID of the firm or trader.~~ ~~Used for order routing. Identifier of the trader who submits an order for a member through another member.~~A 3 character broker code (Member mnemonic).
~~22~~	~~Liquidity Provider~~	~~N~~	~~String (41)~~	~~This should be submitted if the trader qualifies for a Liquidity Provider initiative.~~
23	Market Maker	N	~~String (41)~~Char	This should be ~~submitted~~set to Y if the trader qualifies for a Market Maker~~Liquidity Provider~~ initiative.
24	Decision Maker	C	UInt64	Decision maker short code, required on client orders to identify the investment

BP	Field Name	Req	Data Type	Description
				decision maker. Also used under the power of representation clause where the investment decision maker may be a third party in accordance with Article 8 of Commission Delegated Regulation (EU) …/… 22 on transaction reporting under Article 26 of Regulation EU No 600/2014. Conditionally required for client orders i.e. Account Type = 1, 8 or 101.
25	Investment Decision within Firm (IDM)	N	UInt64	Short code to identify the individual who is responsible for the investment decision.
26	Execution Decision within Firm (EDM)	Y	UInt64	Short code to identify the execution decision maker with the firm.
27	Investment Decision Country ~~Code~~	N	String (3)	ISO country code of the branch responsible for the person making the investment decision.
28	Execution Decision Country ~~Code~~	N	String (3)	ISO country code of the branch responsible for the person making the execution decision.
29	Client Branch Country ~~Code~~	C	String (3)	ISO country code to identify the branch that received the client order or made an investment decision for a client. Conditionally required for client orders i.e. Account Type = 1, 8 or 101.
30	Broker Client ID	Y	String (17)	Identifier of the entity in a risk group.
31	Text	N	String (51)	Free text
32	*Self Match Prevention ID*	*N*	*UInt32*	*Identifies an order that should not be matched to an opposite order if both buy and sell orders for the trade contain the same Self Match Prevention ID and are submitted by the same member.*
33	Display Quantity	C	UInt32	Visible quantity of an Iceberg order. Conditionally required if Order Type = '11' Iceberg.

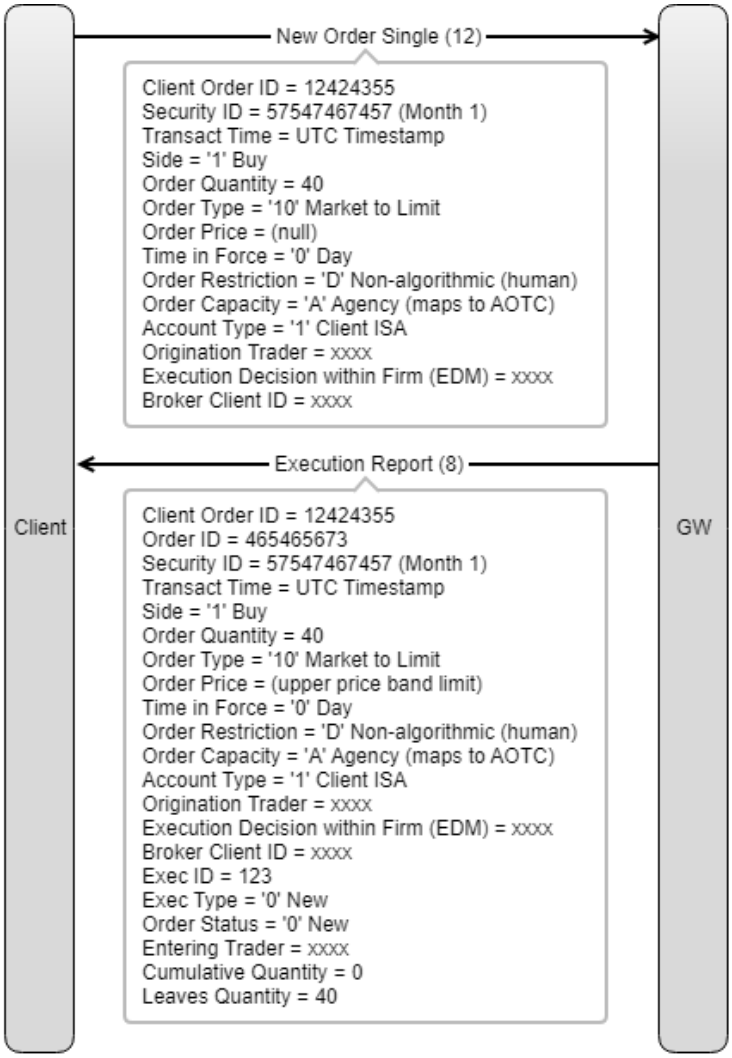
BP	Field Name	Req	Data Type	Description
				If present, must be < Order Quantity.
34	Expiry Date	C	UInt32	The expiry date of an order. Conditionally required if Time In Force = '6' Good 'til Date. Format is YYYYMMDD.
35	Trigger Price	C	Int64	Trigger price for stop orders. Conditionally required if: *Order Type = '3' Stop Loss* Order Type = '4' Stop Limit *Order Type = '13' OCO Market to Limit* *Order Type = '14' OCO Limit*
36	Trigger Price Type	C	UInt8	Type of price event that triggers the stop order: Valid values: 2 = Last Trade 4 = Best Bid or Last Trade 5 = Best Offer or Last Trade Conditionally required if: *Order Type = '3' Stop Loss* Order Type = '4' Stop Limit *Order Type = '13' OCO Market to Limit* *Order Type = '14' OCO Limit*
37	Trigger Type	C	UInt8	Trigger prompt for stop order elements. Valid value: 4 = Price Movement
38	*Trigger New Price*	*C*	*Int64*	*Limit order price of the stop once triggered.* *Conditionally required if Order Type = '14' OCO Limit.*
40	Cancel on Disconnect	N	Char	Specifies whether the order should be cancelled on system disconnection: Valid values: Y = Yes N = No (default)

BP	Field Name	Req	Data Type	Description
41	Direct Electronic Access	N	Char	Signifies order received from a direct access or sponsored access (the trader has direct electronic access). Valid value: Y = Yes Absence of this field infers No (default)
42	Aggregated Order	N	Char	In the context of ESMA RTS 24 Article 2(3), it signifies that the order consists of several orders aggregated together. This maps to ESMA RTS value "AGGR". Valid value: Y = Yes Absence of this field infers No (default). Not valid if Client ID Short Code ~~= '0' No Client~~has been populated.
43	Pending Allocation Order	N	Char	In the context of ESMA RTS 24 Article 2(2), it signifies that the order submitter "is authorized under the legislation of a Member State to allocate an order to its client following submission of the order to the trading venue and has not yet allocated the order to its client at the time of the submission of the order". This maps to ESMA RTS value "PNAL". Valid value: Y = Yes Absence of this field infers No (default). Not valid if Client ID Short Code ~~= '0' No Client~~has been populated.
44	Liquidity Provision Order	N	Char	In the context of ESMA RTS 24 Article 3, it signifies that the order was submitted "as part of a market making strategy pursuant to Articles 17 and 18 of Directive 2014/65/EU, or is submitted as part of another activity in accordance with Article 3" (of RTS 24). Valid value: Y = Yes

BP	Field Name	Req	Data Type	Description
				Absence of this field infers No (default).
45	Risk Reduction Order	N	Char	In the context of ESMA RTS 22 Article 4(2)(i), it signifies that the commodity derivative order is a transaction "to reduce risk in an objectively measurable way in accordance with Article 57 of Directive 2014/65/EU". Valid value: Y = Yes Absence of this field infers No (default).

Example Message Flow

Market to Limit order



4.10.4 Amend Order (13)

Amend Order is used to change the parameters of an existing order. If successful an Execution Report (8) is returned to confirm replacement of the order otherwise an Order Amend Rejected (14) is returned if the request is rejected and the order remains unchanged.

BP	Field Name	Req	Data Type	Description
0	Client Order ID	Y	String (19)	Client specified identifier of the order.
2	Order ID	N	UInt64	Unique order identifier assigned by the trading system.
3	Original Client Order ID	Y	String (19)	Original order identified as the order to be amended.
4	Security ID	Y	UInt64	Tradable Instrument identifier. Must be the same as the original order.
5	Transact Time	Y	UInt64	Time when the message was generated.
6	Side	Y	UInt8	Side of the order. Valid values: 1 = Buy 2 = Sell Must be the same as the original order.
7	Order Quantity	Y	Int32	Total quantity of the order.
8	Order Type	Y	UInt8	Order type applicable to the order. Valid values: 2 = Limit *3 = Stop Loss* 4 = Stop Limit *10 = Market to Limit* 11 = Iceberg *12 = Post Only* *13 = One Cancels Other Market to Limit* *14 = One Cancels Other Limit.* Must be the same as the original order.
9	Order Price	Y	Int64	Price of the order. *Must be null value if:* *Order Type = '3' Stop Loss*

BP	Field Name	Req	Data Type	Description
				Order Type = '10' Market to Limit.
10	Time in Force	Y	UInt8	Specifies how long the order remains in effect. Valid values: 0 = Day 1 = Good Till Cancel (GTC) 3 = Immediate Or Cancel (IOC) *4 = Fill Or Kill (FOK)* 6 = Good Till Date (GTD). Must be the same as the original order.
11	Order Restrictions	Y	Char	Restrictions associated with an order. Valid values: D = Non-algorithmic (human) E = Algorithmic (algo)
12	Order Capacity	Y	Char	Indicates the trading capacity. Valid values: A (agency) = AOTC P (principal) = DEAL R (riskless principal) = MTCH
13	Account Type	Y	UInt8	Specifies the type of account associated with the order. Valid values: 1 = Client ISA 3 = House 8 = Joint back office account (JBO) = Gross OSA 101 = Client OSA. Must be the same as the original order.
~~14~~	~~Executing Firm~~	~~C~~	~~String (4)~~	~~Identifier of the executing firm.~~ ~~Present only if specified on the original order and must be the same as the original order.~~
15	Client ID Short Code	C	UInt64	Client short code identifier. If there is no client this can be populated with the value '0' = No Client for Account Type = 3~~If that is~~

BP	Field Name	Req	Data Type	Description
				~~not available, should be populated with the value '0' = No Client.~~ Not valid if populated with either 1, 2 or 3 Conditionally required for client orders i.e. Account Type = 1, 8 or 101. Present only if specified on the original order and must be the same as the original order.
16	Legal Entity ID	C	String (41)	LEI. ~~Conditionally required for client orders i.e. Account Type = 1, 8 or 101.~~ Present only if specified on the original order and must be the same as the original order.
17	Proprietary Client ID	C	String (41)	Proprietary or Custom Client ID as assigned by the member. Conditionally required for client orders i.e. Account Type = 1, 8 or 101. Present only if specified on the original order and must be the same as the original order.
18	Entering Firm	C	String (4)	Identifier of the entering firm, a member mnemonic. Present only if specified on the original order and must be the same as the original order.
19	Origination Trader	Y	String (41)	Order origination trader. Must be the same as the original order.
20	Customer Account	C	String (~~6~~31)	Identification of the client account code where the Account Type = 1, 8 or 101. Present only if specified on the original order and must be the same as the original order.

BP	Field Name	Req	Data Type	Description
21	Correspondent Broker	C	String (4)	A 3 character broker code (Member mnemonic).~~ID of the firm or trader.~~ ~~Used for order routing. Identifier of the trader who submits an order for a member through another member.~~ Present only if specified on the original order and must be the same as the original order.
~~22~~	~~Liquidity Provider~~	~~C~~	~~String (41)~~	~~This should be submitted if the trader qualifies for a Liquidity Provider initiative.~~ ~~Present only if specified on the original order and must be the same as the original order.~~
23	Market Maker	C	~~String (41)~~Char	This should be ~~submitted~~set to Y if the trader qualifies for a ~~Liquidity Provider~~Market Maker initiative. Present only if specified on the original order and must be the same as the original order.
24	Decision Maker	C	UInt64	Decision maker short code, required on client orders to identify the investment decision maker. Also used under the power of representation clause where the investment decision maker may be a third party in accordance with Article 8 of Commission Delegated Regulation (EU) …/… 22 on transaction reporting under Article 26 of Regulation EU No 600/2014. Conditionally required for client orders i.e. Account Type = 1, 8 or 101. Present only if specified on the original order and must be the same as the original order.
25	Investment Decision within Firm (IDM)	N	UInt64	Short code to identify the individual who is responsible for the investment decision.
26	Execution Decision within Firm (EDM)	Y	UInt64	Short code to identify the execution decision maker with the firm.

BP	Field Name	Req	Data Type	Description
27	Investment Decision Country ~~Code~~	N	String (3)	ISO country code of the branch responsible for the person making the investment decision.
28	Execution Decision Country ~~Code~~	N	String (3)	ISO country code of the branch responsible for the person making the execution decision.
29	Client Branch Country ~~Code~~	C	String (3)	ISO country code to identify the branch that received the client order or made an investment decision for a client. Conditionally required for client orders i.e. Account Type = 1, 8 or 101.
30	Broker Client ID	Y	String (17)	Identifier of the entity in a risk group. Must be the same as the original order.
31	Text	N	String (51)	Free text
32	*Self Match Prevention ID*	*C*	*UInt32*	*Identifies an order that should not be matched to an opposite order if both buy and sell orders for the trade contain the same Self Match Prevention ID and are submitted by the same member.* *Present only if specified on the original order and must be the same as the original order.*
33	Display Quantity	C	UInt32	Visible quantity of an Iceberg order. Conditionally required if Order Type = '11' Iceberg. If present, must be < Order Quantity.
34	Expiry Date	C	UInt32	The expiry date of an order. Conditionally required if Time In Force = '6' Good 'til Date. Format is YYYYMMDD.
35	Trigger Price	C	Int64	Trigger price for stop orders. Conditionally required if: *Order Type = '3' Stop Loss*

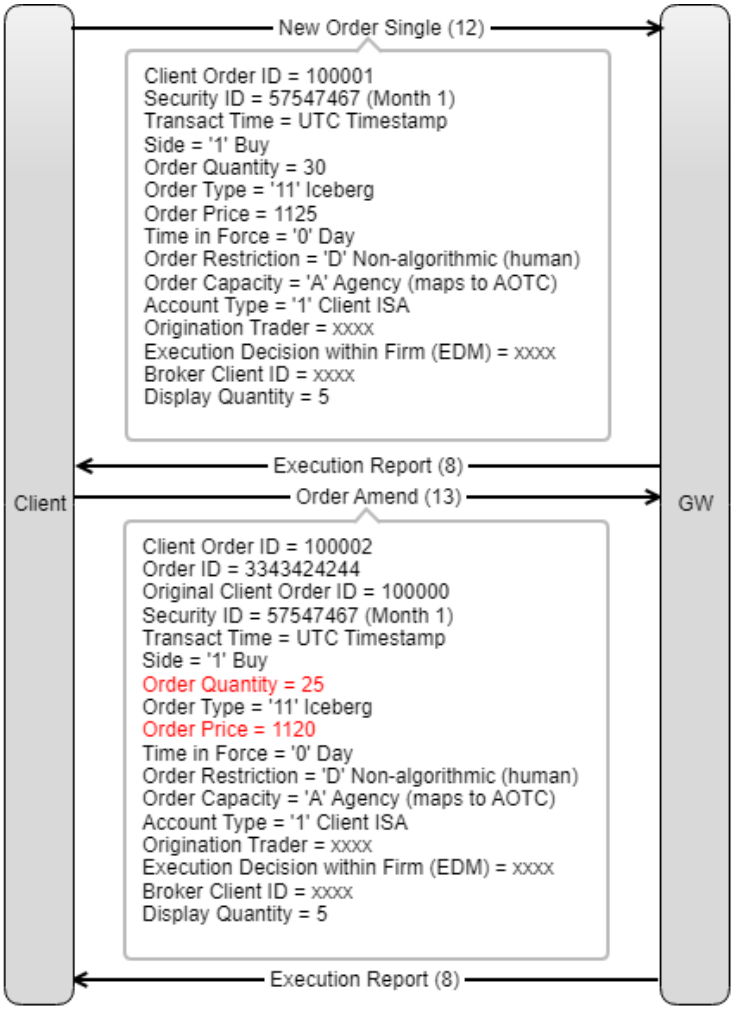
BP	Field Name	Req	Data Type	Description
				Order Type = '4' Stop Limit *Order Type = '13' OCO-M2L* *Order Type = '14' OCO-Limit*
36	Trigger Price Type	C	UInt8	Type of price event that triggers the stop order: Valid values: 2 = Last Trade 4 = Best Bid or Last Trade 5 = Best Offer or Last Trade Conditionally required if: *Order Type = '3' Stop Loss* Order Type = '4' Stop Limit *Order Type = '13' OCO-M2L* *Order Type = '14' OCO-Limit.* Present only if specified on the original order and must be the same as the original order.
37	Trigger Type	C	UInt8	Trigger prompt for stop order elements. Valid value: 4 = Price Movement. Present only if specified on the original order and must be the same as the original order.
38	*Trigger New Price*	*C*	*Int64*	*Limit order price of the stop once triggered.* *Conditionally required if Order Type = '14' OCO-Limit.*
40	Cancel on Disconnect	N	Char	Specifies whether the order should be cancelled on system disconnection: Valid values: Y = Yes N = No Default N = No. Present only if specified on the original order and must be the same as the original order.

BP	Field Name	Req	Data Type	Description
41	Direct Electronic Access	N	Char	Signifies order received from a direct access or sponsored access (the trader has direct electronic access). Valid value: Y = Yes Absence of this field infers No (default)
42	Aggregated Order	N	Char	In the context of ESMA RTS 24 Article 2(3), it signifies that the order consists of several orders aggregated together. This maps to ESMA RTS value "AGGR". Valid value: Y = Yes Absence of this field infers No (default). Not valid if Client ID Short Code ~~= '0' No Client~~has been populated.
43	Pending Allocation Order	N	Char	In the context of ESMA RTS 24 Article 2(2), it signifies that the order submitter "is authorized under the legislation of a Member State to allocate an order to its client following submission of the order to the trading venue and has not yet allocated the order to its client at the time of the submission of the order". This maps to ESMA RTS value "PNAL". Valid value: Y = Yes Absence of this field infers No (default). Not valid if Client ID Short Code ~~= '0' No Client~~has been populated.
44	Liquidity Provision Order	N	Char	In the context of ESMA RTS 24 Article 3, it signifies that the order was submitted "as part of a market making strategy pursuant to Articles 17 and 18 of Directive 2014/65/EU, or is submitted as part of another activity in accordance with Article 3" (of RTS 24). Valid value: Y = Yes

BP	Field Name	Req	Data Type	Description
				Absence of this field infers No (default).
45	Risk Reduction Order	N	Char	In the context of ESMA RTS 22 Article 4(2)(i), it signifies that the commodity derivative order is a transaction "to reduce risk in an objectively measurable way in accordance with Article 57 of Directive 2014/65/EU". Valid value: Y = Yes Absence of this field infers No (default).

Example Message Flow

Amend Order



4.10.5 Order Amend Rejected (14)

Order Amend Rejected is returned when an Amend Order (13) request ~~or an amendment instruction for a quote side in a Mass Quote (22)~~ is rejected.

BP	Field Name	Req	Data Type	Description
0	Client Order ID	Y	String (19)	Client specified identifier of the order. ~~For a quote this is the Quote ID in a Mass Quote (22).~~
~~1~~	~~Secondary Client Order ID~~	~~N~~	~~UInt8~~	~~Quote Entry ID in a Mass Quote (22).~~
2	Order ID	N	UInt64	Unique order identifier assigned by the trading system.
3	Original Client Order ID	Y	String (19)	Original order identified as the order to be amended. ~~Not populated for a Mass Quote.~~
4	Transact Time	Y	UInt64	Time when the message was generated.
5	Order Status	Y	UInt8	Order status as at the time of rejection: Valid values: 0 = New 1 = Partially Filled 2 = Filled 3 = Done for Day 4 = Cancelled 6 = Pending Cancel 8 = Rejected ~~10~~A = Pending New ~~12~~C = Expired ~~14~~E = Pending ~~Amend~~Replace
6	Amend Reject Code	N	UInt16	Code that identifies the reason for the rejection. Valid values: 0 = Too late to amend 1 = Unknown order 3 = Order already in Pending Cancel or Pending Replace Status 6 = Duplicate Client Order ID received 18 = Invalid price increment

BP	Field Name	Req	Data Type	Description
				99 = Other
7	Reason Text	C	String (76)	Text specifying the reason for the rejection. Conditionally required if Amend Reject Code = '99' Other

4.10.6 Cancel Order (15)

Cancel Order is used to cancel the remaining quantity of an existing order. An Execution Report (8) is returned to confirm cancellation or an Order Cancel Rejected (16) if the cancel is rejected.

BP	Field Name	Req	Data Type	Description
0	Client Order ID	Y	String (19)	Unique identifier set by the entering firm.
2	Order ID	N	UInt64	Unique order identifier assigned by the trading system.
3	Original Client Order ID	Y	String (19)	Original order identified as the order to be cancelled.
4	Security ID	Y	UInt64	Tradable Instrument identifier. Must be the same as the original order.
5	Transact Time	Y	UInt64	Time when the message was generated.
6	Side	Y	UInt8	Side of the order. Valid values: 1 = Buy 2 = Sell Must be the same as the original order.

4.10.7 Order Cancel Rejected (16)

Order Cancel Rejected is returned when a Cancel Order (15) request or a cancellation instruction for a quote side in a Mass Quote (22) is rejected.

BP	Field Name	Req	Data Type	Description
0	Client Order ID	Y	String (19)	Client specified identifier of the order.
1	*Secondary Client Order ID*	*N*	*UInt8*	*Quote Entry ID in a Mass Quote (22).*

BP	Field Name	Req	Data Type	Description
2	Order ID	N	UInt64	Unique order identifier assigned by the trading system.
3	Original Client Order ID	Y	String (19)	Original order identified as the order to be cancelled. ~~Not populated~~*Null for a Mass Quote.*
4	Transact Time	Y	UInt64	Time when the message was generated.
5	Order Status	Y	UInt8	Order status as at the time of rejection: Valid values: 0 = New 1 = Partially Filled 2 = Filled 3 = Done for Day 4 = Cancelled 6 = Pending Cancel 8 = Rejected ~~10~~A = Pending New ~~12~~C = Expired ~~14~~E = Pending Replace ~~Amend~~
6	Cancel Reject Code	Y	UInt16	Code that identifies the reason for the rejection. Valid values: 0 = Too late to amend 1 = Unknown order 3 = Order already in Pending Cancel or Pending Replace Status 6 = Duplicate Client Order ID received 99 = Other
7	Reason Text	C	String (76)	Text specifying the reason for the rejection. Conditionally required if Cancel Reject Code = '99' Other
8	*Side*	*N*	*UInt8*	*Optional qualifier to indicate the quote side in a Mass Quote (22).*

4.10.8 Execution Report (8)

Execution Report is used to:

- confirm the receipt of an order submitted using New Order Single or *Mass Quote*

- confirm changes to an existing order (i.e. accept cancel and replace requests)
- confirm or convey an order cancellation or expiration
- convey order or trade cancellation by Market Operations
- convey triggering of a stop order
- convey fill information
- reject orders
- *convey speed bump processing*
- convey information about restated persisted orders carried from one trading day to the next.

Exec Type identifies the purpose of the execution report message and Order Status conveys the current state of the order.

The attributes that can be returned in an Execution Report for each execution type are listed in the Execution Report Matrix.

BP	Field Name	Req	Data Type	Description
0	Client Order ID	Y	String (19)	The client specified identifier in the message that caused this Execution Report. *For quotes this is mapped to Quote ID in a Mass Quote (22)*
1	*Secondary Client Order ID*	*N*	*UInt8*	*Quote Entry ID in a Mass Quote (22).* *Conditionally required according to the Execution Report Matrix.*
2	Order ID	Y	UInt64	Unique order identifier assigned by the trading system.
3	Original Client Order ID	C	String (19)	Client Order ID of the previous order (NOT the initial order of the day) as assigned by the institution. Identifies the previous order in cancel and cancel/replace requests. Conditionally required according to the Execution Report Matrix.
4	Security ID	Y	UInt64	Tradable Instrument identifier.
5	Transact Time	Y	UInt64	Time when the message was generated.
6	Side	Y	UInt8	Side of the order. Valid values:

BP	Field Name	Req	Data Type	Description
				1 = Buy 2 = Sell
7	Order Quantity	Y	Int32	Total quantity of the order.
8	Order Type	Y	UInt8	Order type applicable to the order. Valid values: 2 = Limit *3 = Stop Loss* 4 = Stop Limit *10 = Market to Limit* 11 = Iceberg *12 = Post Only* *13 = One Cancels Other Market to Limit* *14 = One Cancels Other Limit*
9	Order Price	Y	Int64	Price of the order.
10	Time in Force	Y	UInt8	Specifies how long the order remains in effect. Valid values: 0 = Day 1 = Good Till Cancel (GTC) 3 = Immediate Or Cancel (IOC) *4 = Fill Or Kill (FOK)* 6 = Good Till Date (GTD)
11	Order Restrictions	Y	Char	Restrictions associated with an order. Valid values: D = Non-algorithmic (human) E = Algorithmic (algo)
12	Order Capacity	Y	Char	Indicates the trading capacity. Valid values: A (agency) = AOTC P (principal) = DEAL R (riskless principal) = MTCH
13	Account Type	Y	UInt8	Specifies the type of account associated with the order. Valid values: 1 = Client ISA 3 = House

BP	Field Name	Req	Data Type	Description
				8 = Joint back office account (JBO) = Gross OSA 101 = Client OSA For contracts assigned to the T4 booking model only 3 = House is valid whereas for the T2 booking model all account types are valid.
14	Executing Firm	YC	String (4)	Identifier of the executing firm, a member mnemonic. Conditionally required according to the Execution Report Matrix.
15	Client ID Short Code	C	UInt64	Client short code identifier. If there is no client this can be populated with the value '0' = No Client for Account Type = 3If that is not available, should be populated with the value '0' = No Client. Not valid if populated with either 1, 2 or 3 Conditionally required for client orders i.e. Account Type = 1, 8 or 101. Conditionally required according to the Execution Report Matrix.
16	Legal Entity ID	C	String (41)	LEI. Conditionally required for client orders i.e. Account Type = 1, 8 or 101. Conditionally required according to the Execution Report Matrix.
17	Proprietary Client ID	C	String (41)	Proprietary or Custom Client ID as assigned by the member. Conditionally required for client orders i.e. Account Type = 1, 8 or 101. Conditionally required according to the Execution Report Matrix.
18	Entering Firm	C	String (4)	Identifier of the entering firm, a member mnemonic.

BP	Field Name	Req	Data Type	Description
				Conditionally required according to the Execution Report Matrix.
19	Origination Trader	Y	String (41)	Order origination trader.
20	Customer Account	C	String (~~6~~31)	Identification of the client account code where the Account Type = 1, 8 or 101. Conditionally required according to the Execution Report Matrix.
21	Correspondent Broker	C	String (4)	A 3 character broker code (Member mnemonic).~~ID of the firm or trader.~~ ~~Used for order routing. Identifier of the trader who submits an order for a member through another member.~~ Conditionally required according to the Execution Report Matrix.
~~22~~	~~Liquidity Provider~~	~~C~~	~~String (41)~~	~~This should be submitted if the trader qualifies for a Liquidity Provider initiative.~~ ~~Conditionally required according to the Execution Report Matrix.~~
23	Market Maker	C	~~String (41)~~Char	This should be ~~submitted~~set to Y if the trader qualifies for a ~~Liquidity Provider~~Market Maker initiative. Conditionally required according to the Execution Report Matrix.
24	Decision Maker	C	UInt64	Decision maker short code, required on client orders to identify the investment decision maker. Also used under the power of representation clause where the investment decision maker may be a third party in accordance with Article 8 of Commission Delegated Regulation (EU) …/… 22 on transaction reporting under Article 26 of Regulation EU No 600/2014. Conditionally required for client orders i.e. Account Type = 1, 8 or 101. Conditionally required according to the Execution Report Matrix.

BP	Field Name	Req	Data Type	Description
25	Investment Decision within Firm (IDM)	C	UInt64	Short code to identify the individual who is responsible for the investment decision. Conditionally required according to the Execution Report Matrix.
26	Execution Decision within Firm (EDM)	Y	UInt64	Short code to identify the execution decision maker with the firm.
27	Investment Decision Country ~~Code~~	C	String (3)	ISO country code of the branch responsible for the person making the investment decision. Conditionally required according to the Execution Report Matrix.
28	Execution Decision Country ~~Code~~	C	String (3)	ISO country code of the branch responsible for the person making the execution decision. Conditionally required according to the Execution Report Matrix.
29	Client Branch Country ~~Code~~	C	String (3)	ISO country code to identify the branch that received the client order or made an investment decision for a client. Conditionally required for client orders i.e. Account Type = 1, 8 or 101. Conditionally required according to the Execution Report Matrix.
30	Broker Client ID	Y	String (17)	Identifier of the entity in a risk group.
31	Text	C	String (51)	Free text. Conditionally required according to the Execution Report Matrix.
32	*Self Match Prevention ID*	*N*	*UInt32*	*Identifies an order that should not be matched to an opposite order if both buy and sell orders for the trade contain the same Self Match Prevention ID and are submitted by the same member.* *Conditionally required according to the Execution Report Matrix.*

BP	Field Name	Req	Data Type	Description
33	Display Quantity	C	UInt32	Visible quantity of an Iceberg order. Conditionally required if Order Type = '11' Iceberg. If present, must be < Order Quantity.
34	Expiry Date	C	UInt32	The expiry date of an order. Conditionally required if Time In Force = '6' Good 'til Date. Format is YYYYMMDD.
35	Trigger Price	C	Int64	Trigger price for stop orders. Conditionally required if: *Order Type = '3' Stop Loss* Order Type = '4' Stop Limit *Order Type = '13' OCO Market to Limit* *Order Type = '14' OCO Limit*
36	Trigger Price Type	C	UInt8	Type of price event that triggers the stop order: Valid values: 2 = Last Trade 4 = Best Bid or Last Trade 5 = Best Offer or Last Trade Conditionally required if: *Order Type = '3' Stop Loss* Order Type = '4' Stop Limit *Order Type = '13' OCO Market to Limit* *Order Type = '14' OCO Limit*
37	Trigger Type	C	UInt8	Trigger prompt for stop order elements. Valid value: 4 = Price Movement
38	*Trigger New Price*	*C*	*Int64*	*Limit order price of the stop once triggered.* *Conditionally required if Order Type = '14' OCO Limit.*
40	Cancel on Disconnect	N	Char	Specifies whether the order should be cancelled on system disconnection:

BP	Field Name	Req	Data Type	Description
				Valid values: Y = Yes N = No Default N = No
41	Direct Electronic Access	N	Char	Signifies order received from a direct access or sponsored access (the trader has direct electronic access). Valid value: Y = Yes Absence of this field infers No (default)
42	Aggregated Order	N	Char	In the context of ESMA RTS 24 Article 2(3), it signifies that the order consists of several orders aggregated together. This maps to ESMA RTS value "AGGR". Valid value: Y = Yes Absence of this field infers No (default). Not valid if Client ID Short Code ~~= '0' No Client~~has been populated.
43	Pending Allocation Order	N	Char	In the context of ESMA RTS 24 Article 2(2), it signifies that the order submitter "is authorized under the legislation of a Member State to allocate an order to its client following submission of the order to the trading venue and has not yet allocated the order to its client at the time of the submission of the order". This maps to ESMA RTS value "PNAL". Valid value: Y = Yes Absence of this field infers No (default). Not valid if Client ID Short Code ~~= '0' No Client~~has been populated.
44	Liquidity Provision Order	N	Char	In the context of ESMA RTS 24 Article 3, it signifies that the order was submitted "as part of a market making strategy pursuant to Articles 17 and 18 of Directive 2014/65/EU,

BP	Field Name	Req	Data Type	Description
				or is submitted as part of another activity in accordance with Article 3" (of RTS 24). Valid value: Y = Yes Absence of this field infers No (default).
45	Risk Reduction Order	N	Char	In the context of ESMA RTS 22 Article 4(2)(i), it signifies that the commodity derivative order is a transaction "to reduce risk in an objectively measurable way in accordance with Article 57 of Directive 2014/65/EU". Valid value: Y = Yes Absence of this field infers No (default).
46	*Quote Price Level*	*N*	*UInt8*	*Indicates the price level being quoted.* *Valid values are 1, 2 or 3 (i.e. up to 3 price levels).*
65	Exec ID	Y	String (21)	Unique identifier assigned by the trading system to the execution message.
66	Exec Ref ID	C	String (21)	Reference identifier used with Trade Cancel execution type. Conditionally required if Exec Type = 'H' Trade Cancel.
67	Exec Type	Y	Char	Describes the specific Execution Report. Valid values: 0 = New 3 = Done 4 = Cancelled 5 = Replaced 6 = Pending Cancel 8 = Rejected C = Expired D = Restated E = Pending Replace F = Trade H = Trade Cancel L = Triggered or Activated by the System

BP	Field Name	Req	Data Type	Description
68	Order Status	Y	UInt8	Identifies current status of order. Valid values: 0 = New 1 = Partially Filled 2 = Filled 3 = Done for day 4 = Cancelled 6 = Pending Cancel 8 = Rejected A = Pending New C = Expired E = Pending Replace
69	Entering Trader	Y	String (11)	Identifier of the trader entering the order.
70	Clearing Firm	C	String (4)	Identifier of the clearing firm, a member mnemonic. Conditionally required if Exec Type = 'F' Trade.
71	Trade ID	C	UInt64	Identifier assigned ~~to a trade~~ by the trading system ~~for the trade half~~ which joins buy and sell half trades. Conditionally required if Exec Type = 'F' Trade.
72	Exec Restatement Reason	C	UInt16	The reason for restatement. Valid values: 1 = GT renewal / restatement *99 = Other. See Exec Type Reason for speed bump handling.* Conditionally required if Exec Type = 'D' Restated.
73	Exec Type Reason	C	UInt8	The initiating event for the Execution Report. Conditionally required to report unsolicited cancellation and *order status in speed bump processing.* Valid values: 4 = Unsolicited order cancellation

BP	Field Name	Req	Data Type	Description
				101 = Order accepted but speed bump applied *102 = Order added after speed bump* *103 = Order cancelled whilst in speed bump delay* *104 = Original order is in speed bump enforced delay* *105 = Order updated after speed bump delay* *106 = Amend is in speed bump delay* *107 = Order amended after speed bump delay* *108 = Order rejected after speed bump delay* *109 = Unsolicited cancel while in speed bump*
74	Order Category	C	UInt8	Defines the type of interest behind a trade (fill or partial fill). Valid value: 7 = Implied Order Conditionally required for a trade from an implied order when Exec Type = 'F' Trade.
75	Aggressor Indicator	C	Char	Indicates if a matching order is an aggressor or not in the trade. Y = Aggressor N = Passive Conditionally required if Exec Type = 'F' Trade.
76	Order Reject Reason	N	UInt16	Code to identify reason for order rejection. Valid values: 6 = Duplicate Order 15 = Unknown Account(s) 18 = Invalid price increment 99 = Other. Conditionally required if Exec Type = '8' Rejected.
77	Reason Text	N	String (76)	Text specifying the reason for the rejection.

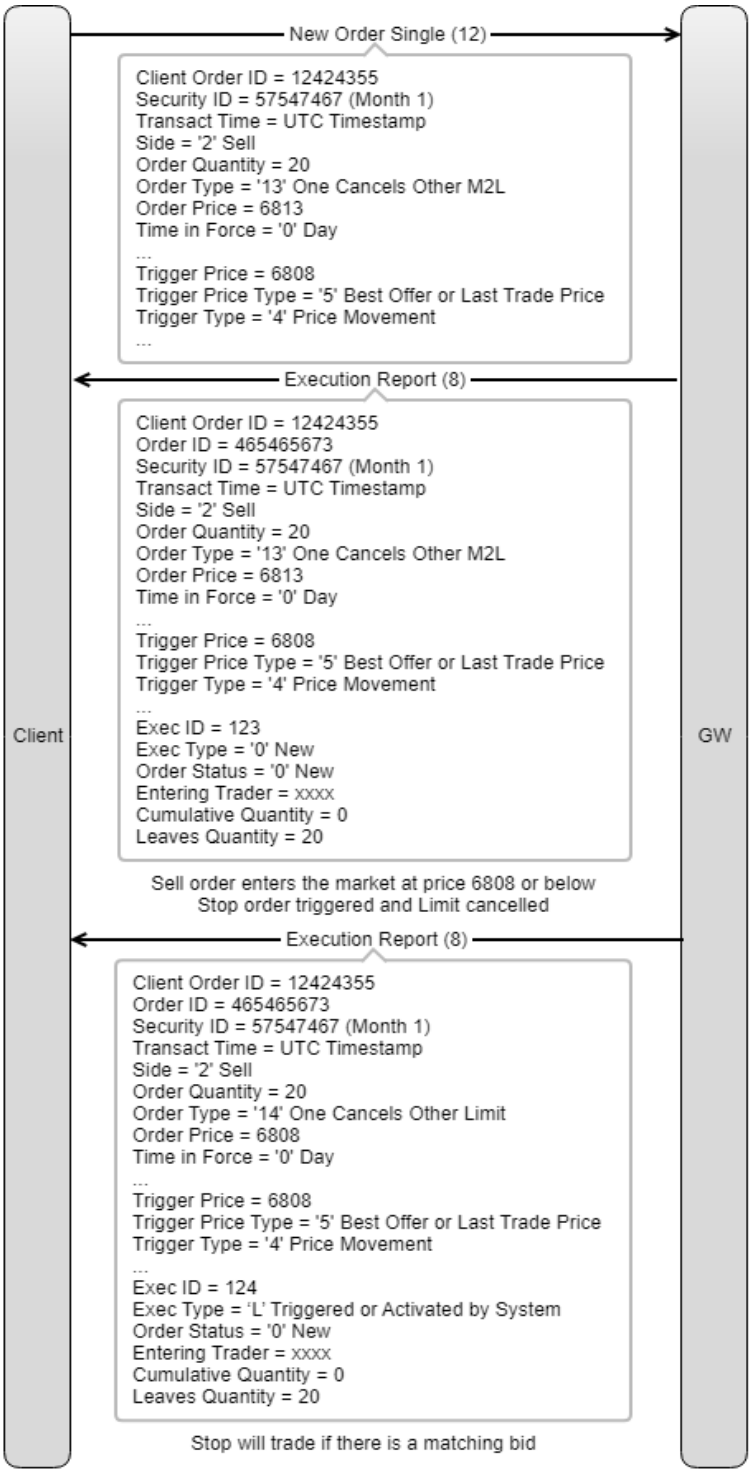
BP	Field Name		Req	Data Type	Description
					Conditionally required if Exec Type Reason = '4' Unsolicited order cancellation or Order Reject Reason = '99' Other.
78	Last Quantity		C	UInt32	The total volume of this trade. Conditionally required if Exec Type = 'F' Trade.
79	Last Price		C	UInt64	The price of this trade. Conditionally required if Exec Type = 'F' Trade.
80	Cumulative Quantity		Y	UInt32	The quantity of the order that has been executed so far.
81	Leaves Quantity		Y	UInt32	The quantity open for further execution.
82	No Legs		C	UInt8	Number of Instrument Leg repeating group instances. Conditionally required if Exec Type = 'F' Trade on a multileg tradable instrument.
	Legs Body Fields Presence Map		C	Bitmap Variable Length (1)	Conditionally required if No Legs > 0 where each repeating group represents a leg in the multileg instrument.
	0	Leg Security ID	Y	UInt64	Multileg tradable instrument's individual Security ID.
	1	Leg Side	Y	UInt8	The side of this individual leg (multileg security). Valid values: 1 = Buy 2 = Sell
	2	Leg Alloc ID	Y	UInt64	Strategy leg trade identifier assigned by the trading system which is shared by half trades.
	3	Leg Last Price	Y	Int64	Execution price assigned to the leg of the multileg tradable instrument.

BP	Field Name		Req	Data Type	Description
	4	Leg Last Quantity	Y	UInt32	Fill quantity for the instrument leg.

Example Message Flows

OCO submitted, Stop triggered and Limit cancelled

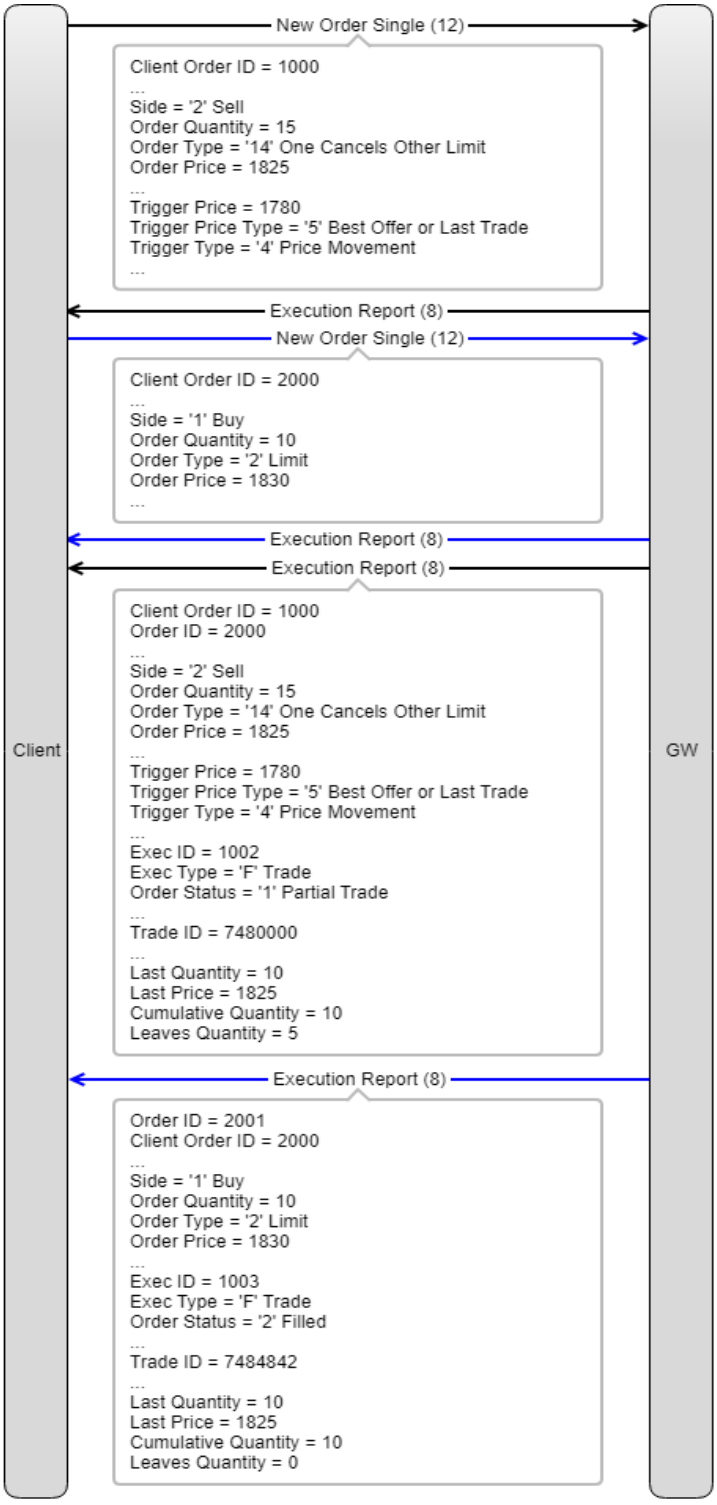
An OCO order is submitted as a Limit offer with a Market Stop trigger price of 6808, an incoming offer triggers the Stop order and cancels Limit element of the OCO. An Execution Report is not sent for cancellation. The triggered Market Stop is converted to a Limit order at a trigger new price of 6808.



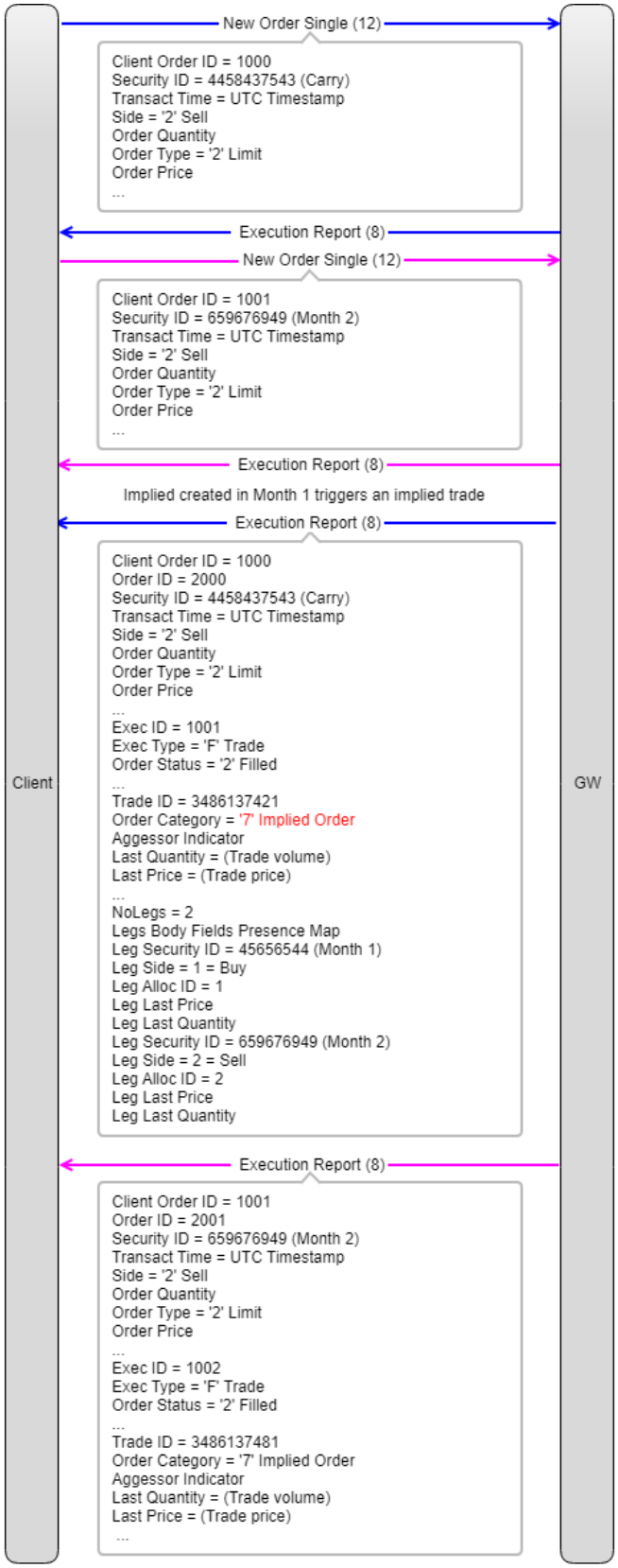
OCO Partial Trade

OCO is submitted as Limit offer for 15 lots at 1825 with a Market Stop Loss trigger price of 1780.

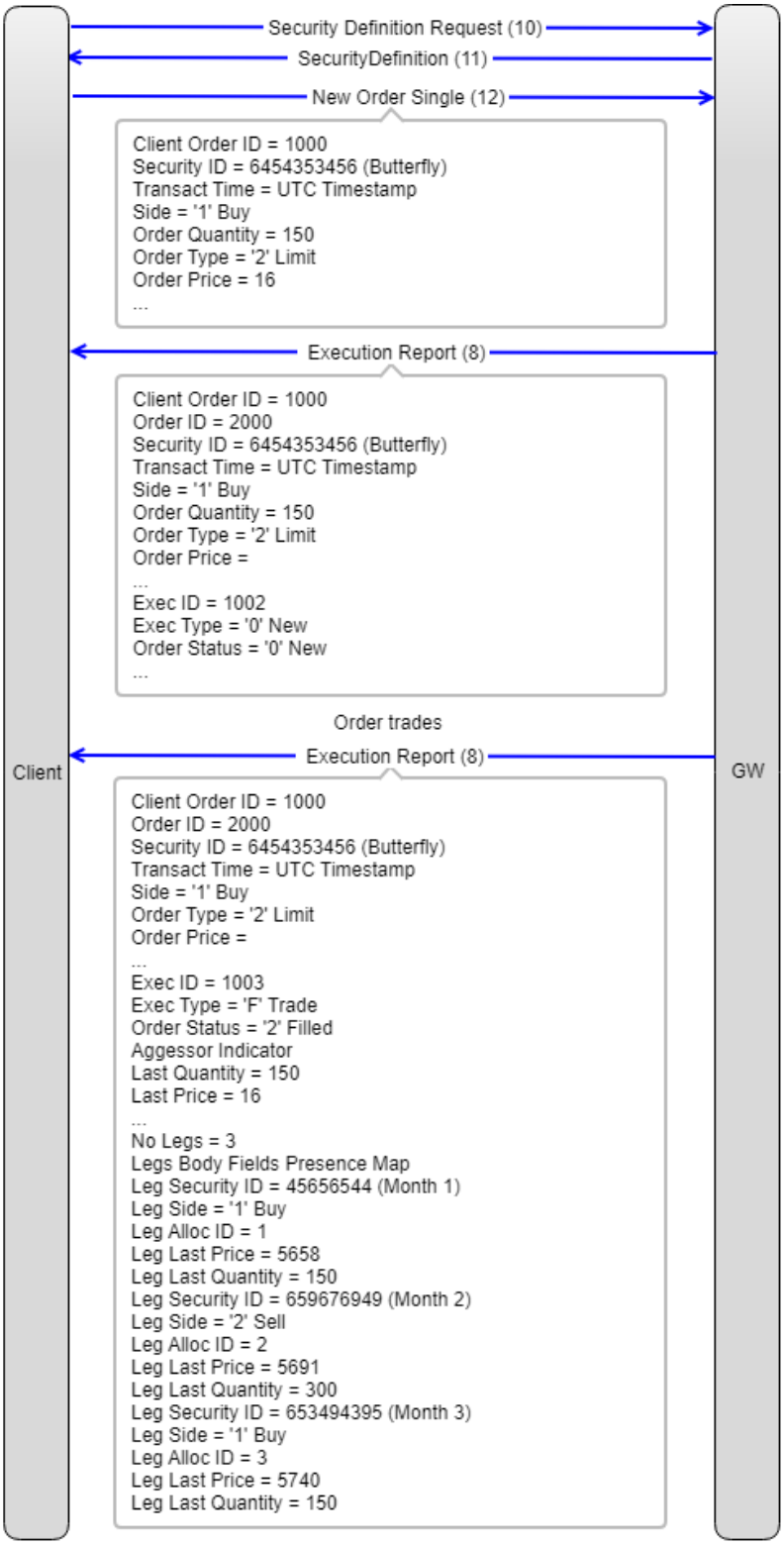
A Limit bid is submitted at 1830 for 10 lots. The OCO order is not triggered but trades 10 lots with the incoming order. The OCO remains in the order book with a residual volume of 5 lots



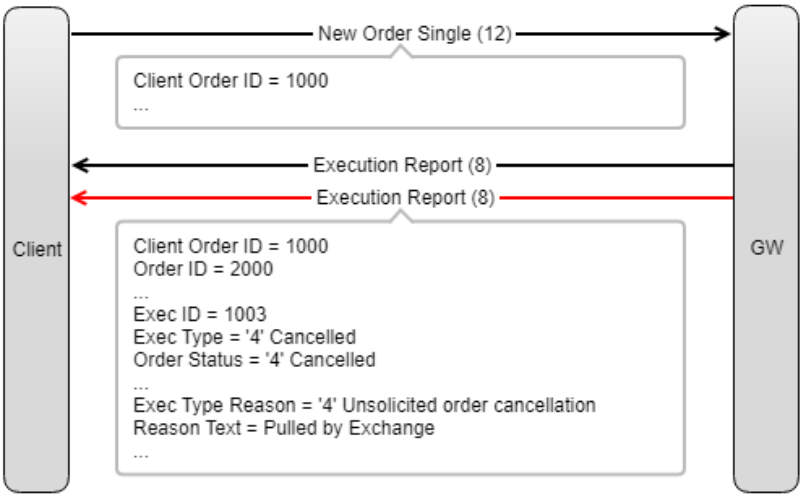
Implied trade



Custom strategy Butterfly trades



Order cancellation by Exchange



4.10.8.1 Execution Report Matrix

An Execution Report can be returned in response to a request e.g. New Order Single (12) or unsolicited in response to a particular action.

The fields that can be included are contingent on the purpose of the message and any mandatory or conditionally supplied tags specified by the originator in the initiating request or returned response to a particular action.

Legend:

M = Mandatory

C = Conditional

O = Optional

P = Returned in first outbound message if present in original message.

The following table indicates the field that will be returned for specific execution types:

Field Name	Order Accepted	Order Restated (GTC/GTD)	Order Triggered	*Order Restated (Speed Bump)*	Order Replaced	Order Pending Replace	*Order Replaced (after Speed Bump)*	Order Cancelled (Solicited)	Order Cancelled (Unsolicited)	Order Expired	Done for Day	Outright Filled	Strategy Filled	Trade Busted	Order Rejected
Executing Firm	MP	MP	M	*M*	M	M	*M*	M	M	M	M	MP	MP	MP	M
Client ID Short Code	P	P										P	P	P	
Legal Entity ID	P	P										P	P	P	
Proprietary Client ID	P	P										P	P	P	
Entering Firm	P	P										P	P	P	

Field Name	Order Accepted	Order Restated (GTC/GTD)	Order Triggered	*Order Restated (Speed Bump)*	Order Replaced	Order Pending Replace	*Order Replaced (after Speed Bump)*	Order Cancelled (Solicited)	Order Cancelled (Unsolicited)	Order Expired	Done for Day	Outright Filled	Strategy Filled	Trade Busted	Order Rejected
Origination Trader	M	M	M	*M*	M	M	*M*	M	M	M	M	M	M	M	M
Customer Account	P	P										P	P	P	
Correspondent Broker	P	P										P	P	P	
~~Liquidity Provider~~	~~P~~	~~P~~										~~P~~	~~P~~	~~P~~	
Market Maker	P	P										P	P	P	
Broker Client ID	M	M	M	*M*	M	M	*M*	M	M	M	M	M	M	M	M
Decision Maker	P	P										P	P	P	
Investment Decision within Firm (IDM)	P	P			P		*P*					P	P	P	
Execution Decision within Firm (EDM)	M	M	M	*M*	M	M	*M*	M	M	M	M	M	M	M	M
Investment Decision Country ~~Code~~	P	P			P		*P*					P	P	P	
Execution Decision Country ~~Code~~	P	P			P		*P*					P	P	P	

Field Name	Order Accepted	Order Restated (GTC/GTD)	Order Triggered	*Order Restated (Speed Bump)*	Order Replaced	Order Pending Replace	*Order Replaced (after Speed Bump)*	Order Cancelled (Solicited)	Order Cancelled (Unsolicited)	Order Expired	Done for Day	Outright Filled	Strategy Filled	Trade Busted	Order Rejected
Client Branch Country ~~Code~~	P	P			P		*P*					P	P	P	
Order Capacity	M	M	M	*M*	M	M	*M*	M	M	M	M	M	M	M	M
Order Restrictions	M	M	M	*M*	M	M	*M*	M	M	M	M	M	M	M	M
Direct Electronic Access	P	P			P		*P*					P	P	P	P
Aggregated Order	P	P			P		*P*					P	P	P	P
Pending Allocation Order	P	P			P		*P*					P	P	P	P
Liquidity Provision Order	P	P			P		*P*					P	P	P	P
Risk Reduction Order	P	P			P		*P*					P	P	P	P
Entering Trader	M	M	M	*M*	M	M	*M*	M	M	M	M	M	M	M	M
Clearing Firm												M	M	M	
Account Type	M	M	M	*M*	M	M	*M*	M	M	M	M	M	M	M	M
Order ID	M	M	M	*M*	M	M	*M*	M	M	M	M	M	M	M	M
Client Order ID	M	M	M	*M*	M	M	*M*	M	M	M	M	M	M	M	M

Field Name	Order Accepted	Order Restated (GTC/GTD)	Order Triggered	*Order Restated (Speed Bump)*	Order Replaced	Order Pending Replace	*Order Replaced (after Speed Bump)*	Order Cancelled (Solicited)	Order Cancelled (Unsolicited)	Order Expired	Done for Day	Outright Filled	Strategy Filled	Trade Busted	Order Rejected
Secondary Client Order ID	O				O			O	O			O	O	O	O
Original Client Order ID					M	M	*M*	M							
Security ID	M	M	M	*M*	M	M	*M*	M	M	M	M	M	M	M	M
Side	M	M	M	*M*	M	M	*M*	M	M	M	M	M	M	M	M
Order Quantity	M	M	M	*M*	M	M	*M*	M	M	M	M	M	M	M	M
Order Type	M	M	M	*M*	M	M	*M*	M	M	M	M	M	M	M	M
Order Price	C	C	M	*C*	C	C	*C*	C	C	C	C	M	M	M	C
Expiry Date	C	C	C	*C*	C	C	*C*	C	C	C	C	C	C	C	C
Cancel on Disconnect	C	C	C	*C*	C	C	*C*	C	C	C	C	C	C	C	C
Transact Time	M	M	M	*M*	M	M	*M*	M	M	M	M	M	M	M	M
Display Quantity	C	C		*C*	C	C	*C*	C	C	C	C	C	~~C~~	C	C
Text	P	P			P		*P*					P	P	P	P
Time in Force	M	M	M	*M*	M	M	*M*	M	M	M	M	M	M	M	M

Field Name	Order Accepted	Order Restated (GTC/GTD)	Order Triggered	*Order Restated (Speed Bump)*	Order Replaced	Order Pending Replace	*Order Replaced (after Speed Bump)*	Order Cancelled (Solicited)	Order Cancelled (Unsolicited)	Order Expired	Done for Day	Outright Filled	Strategy Filled	Trade Busted	Order Rejected
Self Match Prevention ID	*P*	*P*							*P*						*P*
Trigger Type	C	C	C	*C*	C	C	*C*	C	C	C	C	C	C	C	C
Trigger Price	C	C	C	*C*	C	C	*C*	C	C	C	C	C	C	C	C
Trigger Price Type	C	C	C	*C*	C	C	*C*	C	C	C	C	C	C	C	C
Trigger New Price	*C*	*C*	*C*	*C*	*C*	*C*	*C*	*C*	*C*	*C*	*C*	*C*	*C*	*C*	*C*
Quote Price Level	*C*														*C*
Trade ID												M	M	M	
Exec ID	M	M	M	*M*	M	M	*M*	M	M	M	M	M	M	M	M
Exec Ref ID														M	
Exec Type	M	M	M	*M*	M	M	*M*	M	M	M	M	M	M	M	M
Exec Type Reason	C			*M*		M	*M*	C	M						
Order Status	M	M	M	*M*	M	M	*M*	M	M	M	M	M	M	M	M
Order Reject Reason															M

Field Name	Order Accepted	Order Restated (GTC/GTD)	Order Triggered	*Order Restated (Speed Bump)*	Order Replaced	Order Pending Replace	*Order Replaced (after Speed Bump)*	Order Cancelled (Solicited)	Order Cancelled (Unsolicited)	Order Expired	Done for Day	Outright Filled	Strategy Filled	Trade Busted	Order Rejected
Exec Restatement Reason		M		*M*											
Order Category												C	C	C	
Aggressor Indicator												M	M	M	
Last Quantity												M	M	M	
Last Price												M	M	M	
Leaves Quantity	M	M	M	*M*	M	M	*M*	M	M	M	M	M	M	M	M
Cumulative Quantity	M	M	M	*M*	M	M	*M*	M	M	M	M	M	M	M	M
Reason Text									C	C					C
No Legs													M	C	
Legs Body Fields Presence Map													M	C	
Leg Security ID													M	C	
Leg Side													M	C	
Leg Alloc ID													M	C	

Field Name	Order Accepted	Order Restated (GTC/GTD)	Order Triggered	*Order Restated (Speed Bump)*	Order Replaced	Order Pending Replace	*Order Replaced (after Speed Bump)*	Order Cancelled (Solicited)	Order Cancelled (Unsolicited)	Order Expired	Done for Day	Outright Filled	Strategy Filled	Trade Busted	Order Rejected
Leg Last Price													M	C	
Leg Last Quantity													M	C	

4.10.9 Mass Cancel Request (17)

Mass Cancel Request is used to cancel the remaining quantity of a group of orders *and/or quotes* matching criteria specified within the message. Persisted orders will be included in the cancellation request.

If the request is accepted, an Execution Report (8) will be sent for each order cancelled followed by Mass Cancel Report (18).

If the request is rejected, Mass Cancel Report (18) will be returned indicating the reason.

BP	Field Name	Req	Data Type	Description
0	Client Order ID	Y	String (19)	Unique ID of Mass Cancel Request as assigned by the institution.
2	Mass Cancel Request Type	Y	UInt8	Specifies the type of cancellation requested. Valid values: 1 = Cancel orders/*quotes* for a Security ID (tradable instrument) 3 = Cancel orders/*quotes* for a Product (contract e.g. CADF - Copper Future) 7 = Cancel all orders/*quotes* *101 = Cancel quotes specified in Quote ID*
3	Mass Cancel Scope	Y	UInt8	Specifies the scope of the cancellation requested. Valid values: 1 = Cancel orders only *2 = Cancel quotes only* *3 = Cancel orders and quotes.* *Must be 2 = Cancel quotes if Mass Cancel Request Type = '101' Cancel quotes specified in Quote ID.*
5	Transact Time	Y	UInt64	Time when the message was generated.
7	Security Exchange	C	String (5)	Market which is used to identify the security: XLME Conditionally required if Symbol is specified.
8	Product Complex	C	String (5)	Identifies an entire suite of products for a given market.

BP	Field Name	Req	Data Type	Description
				Valid values: LME = Base ~~LMEP~~LPM = Precious Conditionally required if Symbol is specified.
9	Symbol	C	String (21)	Symbol for the LME contract code e.g. CADF (Copper Future) or OCDF (Copper Monthly Average Future). Conditionally required if Mass Cancel Request Type = '3' Cancel orders/*quotes* for a Product.
10	Security ID	C	UInt64	Tradable Instrument identifier. Conditionally required if Mass Cancel Request Type = '1' Cancel orders/*quotes* for a Security ID.
11	*Quote ID*	*C*	*String (19)*	*Mass Quote Identifier.* *Conditionally required if Mass Cancel Request Type = '101' Cancel quotes specified in Quote ID.*
12	Broker Client ID	N	String (17)	Identifier of the entity in a risk group. Can be used with Mass Cancel Request Type = '7' Cancel all orders/*quotes*.
13	Side	N	UInt8	Optional qualifier to indicate the side of the market for which orders are to be cancelled. Can be used if Mass Cancel Request Type = '3' Cancel orders/*quotes* for a Product. Absence of this field indicates that orders/*quotes* are to be cancelled regardless of side.

4.10.10 Mass Cancel Report (18)

Mass Cancel Report is returned in response to a Mass Cancel Request (17).

Each affected order that is cancelled is acknowledged with a separate Execution Report (8).

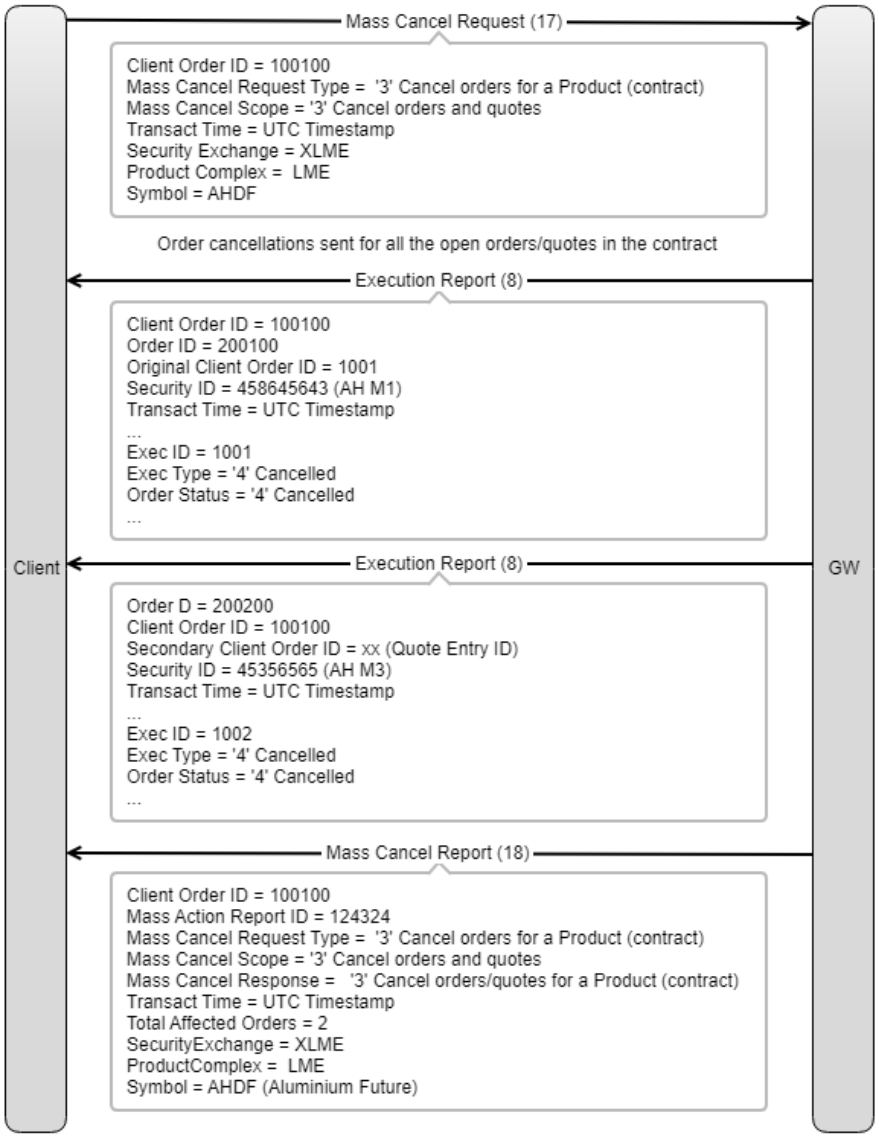
BP	Field Name	Req	Data Type	Description
0	Client Order ID	N	String (19)	Unique ID of Mass Cancel Request as assigned by the institution.
1	Mass Action Report ID	Y	String (21)	Unique Identifier for the Order Mass Cancel Report assigned by the system.
2	Mass Cancel Request Type	Y	UInt8	Specifies the type of cancellation requested. Valid values: 1 = Cancel orders/*quotes* for a Security ID (tradable instrument) 3 = Cancel orders/*quotes* for a Product (contract e.g. CADF - Copper Future) 7 = Cancel all orders/*quotes* *101 = Cancel quotes specified in Quote ID*
3	Mass Cancel Scope	Y	UInt8	Specifies the scope of the cancellation requested. Valid values: 1 = Cancel orders only *2 = Cancel quotes only* *3 = Cancel orders and quotes.* *Must be 2 = Cancel quotes if Mass Cancel Request Type = '101' Cancel quotes specified in Quote ID.*
4	Mass Cancel Response	Y	UInt8	Indicates the action taken on the cancel request: Valid values: 0 = Cancel Request Rejected 1 = Cancel orders/*quotes* for a Security ID 3 = Cancel orders/*quotes* for a Product 7 = Cancel all orders/*quotes* *101 = Cancel quotes specified in Quote ID*
5	Transact Time	Y	UInt64	Time when the message was generated.
6	Total Affected Orders	Y	UInt32	Indicates the total number of orders affected by the Mass Cancel Request.

BP	Field Name	Req	Data Type	Description
7	Security Exchange	C	String (5)	Market which is used to identify the security: XLME Conditionally required if Symbol is specified.
8	Product Complex	C	String (5)	Identifies an entire suite of products for a given market. Valid values: LME = Base ~~LMEP~~LPM = Precious Conditionally required if Symbol is specified.
9	Symbol	C	~~UInt64~~String (21)	Tradable Instrument identifier. Conditionally required if Mass Cancel Request Type = '1' Cancel orders/*quotes* for a Security ID.
10	Security ID	C	UInt64	Tradable Instrument identifier. Conditionally required if Mass Cancel Request Type = '1' Cancel orders/*quotes* for a Security ID.
11	*Quote ID*	*C*	*String (19)*	*Mass Quote Identifier.* *Conditionally required if Mass Cancel Request Type = '101' Cancel quotes specified in Quote ID.*
12	Broker Client ID	C	String (17)	Identifier of the entity in a risk group. Can be used with Mass Cancel Request Type = '7' Cancel all orders/*quotes*.
13	Side	C	UInt8	Optional qualifier to indicate the side of the market for which orders are to be cancelled. Can be used if Mass Cancel Request Type = '3' Cancel orders/*quotes* for a Product. Absence of this field indicates that orders/*quotes* are to be cancelled regardless of side.
14	Mass Cancel Reject Reason	C	UInt16	Code specifying the reason for the rejection. Valid values:

BP	Field Name	Req	Data Type	Description
				1 = Invalid or Unknown Security 3 = Invalid or Unknown Product 99 = Other. Conditionally required if Mass Cancel Response = '0' Cancel Request Rejected
15	Reason Text	C	String (76)	Text specifying the reason for the rejection. Conditionally required if Mass Cancel Reject Reason = '99' Other.

Example Message Flow

Mass cancel orders and quotes in a contract



4.10.11 Mass Quote (22)

Mass Quote is used to enter and manage (amend and/or cancel) multiple orders submitted as quotes in a single contract.

BP	Field Name	Req	Data Type	Description
0	Quote ID	Y	String (19)	Client specified unique identifier for the Mass Quote. This maps to the Client Order ID in the Execution Report.
1	Transact Time	Y	UInt64	Time when the message was generated.
2	Account Type	Y	UInt8	Specifies the type of account associated with the order. Valid values: 1 = Client ISA 3 = House 8 = Joint back office account (JBO) = Gross OSA 101 = Client OSA For contracts assigned to the T4 booking model only 3 = House is valid whereas for the T2 booking model all account types are valid.
3	Order Restrictions	Y	Char	Restrictions associated with an order. Valid values: D = Non-algorithmic (human) E = Algorithmic (algo)
4	Order Capacity	Y	Char	Indicates the trading capacity. Valid values: A (agency) = AOTC P (principal) = DEAL R (riskless principal) = MTCH
~~5~~	~~Executing Firm~~	~~N~~	~~String (4)~~	~~Identifier of the executing firm.~~
6	Client ID Short Code	C	UInt64	Client short code identifier. If there is no client this can be populated with the value '0' = No Client for Account Type = 3~~If that is not available, should be populated with the value '0' = No Client.~~ Not valid if populated with either 1, 2 or 3.

BP	Field Name	Req	Data Type	Description
				Conditionally required for client orders i.e. Account Type = 1, 8 or 101.
7	Legal Entity ID	C	String (41)	LEI. ~~Conditionally required for client orders i.e. Account Type = 1, 8 or 101.~~
8	Proprietary Client ID	C	String (41)	Proprietary or Custom Client ID as assigned by the member. Conditionally required for client orders i.e. Account Type = 1, 8 or 101.
9	Entering Firm	N	String (4)	Identifier of the entering firm, a member mnemonic.
10	Origination Trader	Y	String (41)	Order origination trader.
11	Customer Account	C	String (~~6~~31)	Identification of the client account code where the Account Type = 1, 8 or 101.
12	Correspondent Broker	N	String (4)	A 3 character broker code (Member mnemonic).~~ID of the firm or trader.~~ ~~Used for order routing. Identifier of the trader who submits an order for a member through another member.~~
~~13~~	~~Liquidity Provider~~	~~N~~	~~String (41)~~	~~This should be submitted if the trader qualifies for a Liquidity Provider initiative.~~
14	Market Maker	N	~~String (41)~~Char	This should be ~~submitted~~set to Y if the trader qualifies for a ~~Liquidity Provider~~Market Maker initiative.
15	Decision Maker	C	UInt64	Decision maker short code, required on client orders to identify the investment decision maker. Also used under the power of representation clause where the investment decision maker may be a third party in accordance with Article 8 of Commission Delegated Regulation (EU) …/… 22 on transaction reporting under Article 26 of Regulation EU No 600/2014 .

BP	Field Name	Req	Data Type	Description
				Conditionally required for client orders i.e. Account Type = 1, 8 or 101.
16	*Investment Decision within Firm (IDM)*	*N*	*UInt64*	*Short code to identify the individual who is responsible for the investment decision.*
17	*Execution Decision within Firm (EDM)*	*Y*	*UInt64*	*Short code to identify the execution decision maker with the firm.*
18	*Investment Decision Country ~~Code~~*	*N*	*String (3)*	*ISO country code of the branch responsible for the person making the investment decision.*
19	*Execution Decision Country ~~Code~~*	*N*	*String (3)*	*ISO country code of the branch responsible for the person making the execution decision.*
20	*Client Branch Country ~~Code~~*	*C*	*String (3)*	*ISO country code to identify the branch that received the client order or made an investment decision for a client.* *Conditionally required for client orders i.e. Account Type = 1, 8 or 101.*
21	*Broker Client ID*	*Y*	*String (17)*	*Identifier of the entity in a risk group.*
22	*Self Match Prevention ID*	*N*	*UInt32*	*Identifies an order that should not be matched to an opposite order if both buy and sell orders for the trade contain the same Self Match Prevention ID and are submitted by the same member.*
23	*Direct Electronic Access*	*N*	*Char*	*Signifies order received from a direct access or sponsored access (the trader has direct electronic access).* *Valid value:* *Y = Yes* *Absence of this field infers No (default)*
24	*Total Quote Entries*	*Y*	*UInt8*	*Total number of Quote entries (pairs) in this message.* *Must be ≥ 1 and ≤ 20.*

BP	Field Name		Req	Data Type	Description
25	No Quote Sets		Y	UInt8	The number of sets of quotes contained in the message. Must be ≥ 1 but ≤ Total Quote Entries.
	Quote Sets Fields Presence Map		Y	Bitmap Variable Length (1)	This will indicate the fields/nested repeating blocks present in this repeating block.
	1	Security ID	Y	UInt64	Tradable Instrument identifier.
	2	No Quote Entries	Y	UInt8	Number of Quote Entry repeating blocks. Must be ≥ 1 but ≤ 3.
		Quote Entry Fields Presence Map	Y	Bitmap Variable Length (1)	This will indicate the fields/nested repeating blocks present in this repeating block.
		0 Quote Entry ID	Y	UInt8	Quote pair ID in a Mass Quote message. Starts from 1 and incremented by 1 for each quote pair in the message. The last value must be the same as Total Quote Entries in this message.
		1 Quote Price Level	Y	UInt8	Indicates the price level that is being submitted or modified. Valid values are 1, 2 or 3. ~~Value of zero indicates this is a dummy quote to make up for minimum quote size in a mass quote.~~
		2 Bid Size	Y	Int32	Bid quantity. Value 0 indicates cancellation of this side. Value -1 indicates no change to this side or a dummy quote. ~~Null value indicates dummy quote.~~
		3 Offer Size	Y	Int32	Offer quantity. Value 0 indicates cancellation of this side. Value -1 indicates no change to this side or a dummy quote.

BP	Field Name			Req	Data Type	Description
						~~Null value indicates dummy quote.~~
		4	Bid Price	Y	Int64	Bid price. Null value indicates dummy quote or a cancelled price.
		5	Offer Price	Y	Int64	Offer price. Null value indicates dummy quote or a cancelled price.

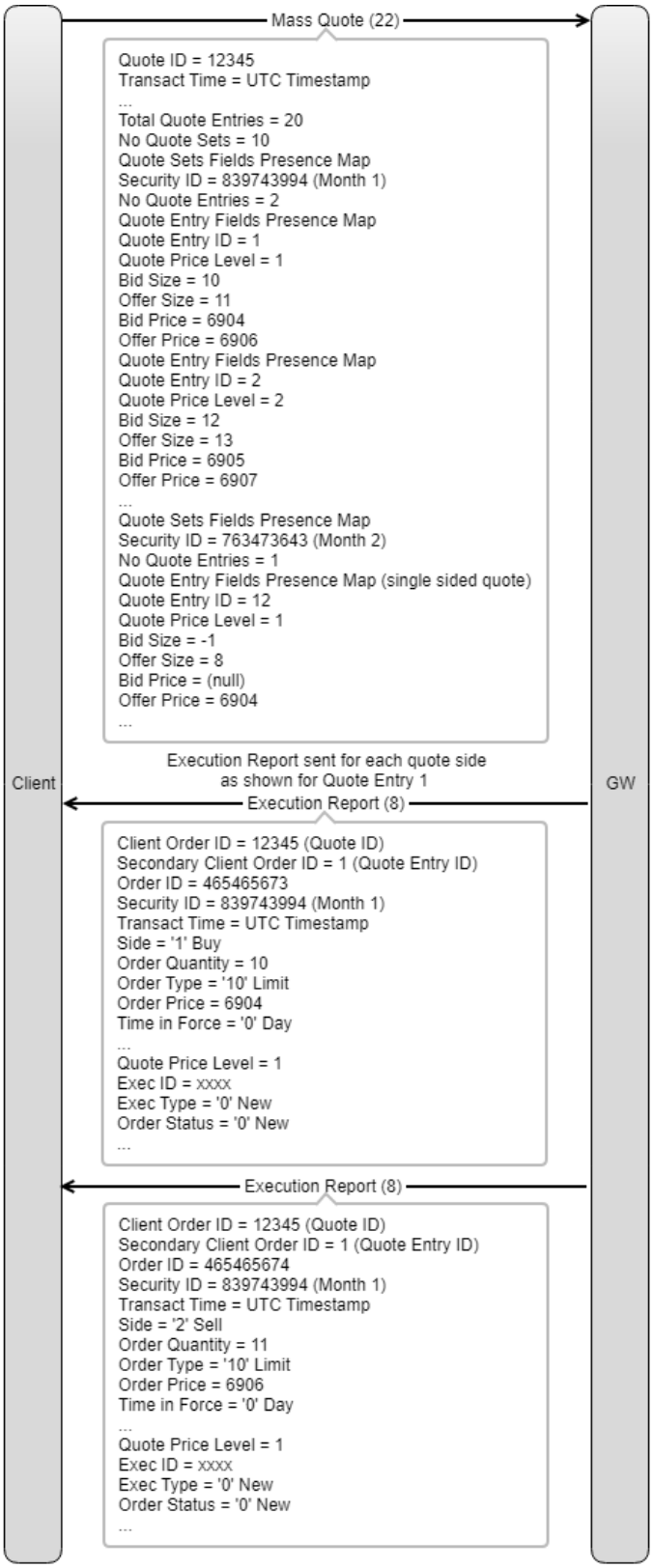
4.10.12 Mass Quote Ack (23)

Mass Quote Ack reports the rejection of a Mass Quote (22) at message level.

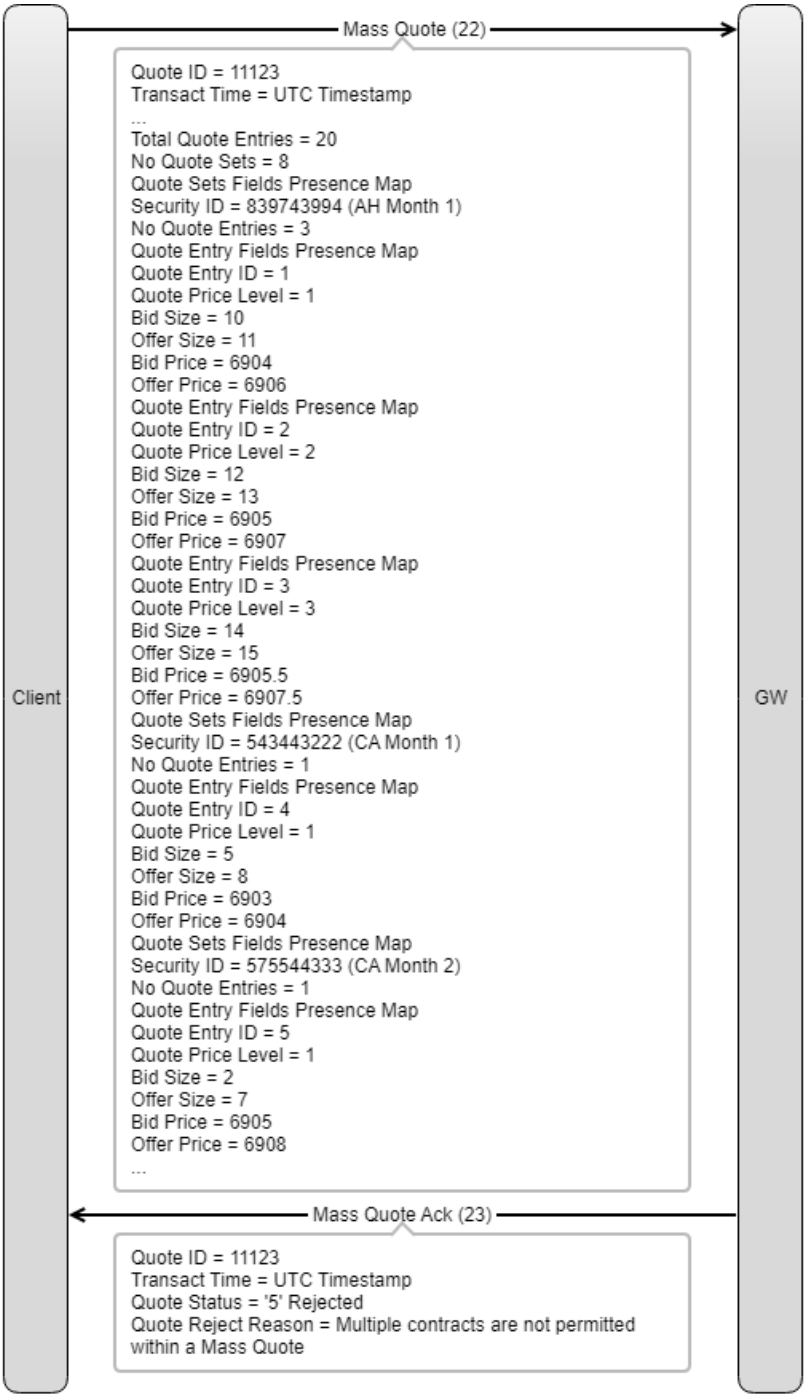
BP	Field Name	Req	Data Type	Description
0	Quote ID	Y	String (19)	Client specified unique identifier for the Mass Quote. This maps to the Client Order ID in the Execution Report.
1	Transact Time	Y	UInt64	Time when the message was generated.
2	Quote Status	Y	UInt8	Status of the Mass Quote acknowledgement. Valid value: 5 = Rejected
3	Quote Reject Reason	Y	UInt16	Code specifying the reason for the rejection. Valid values: 6 = Duplicate quote 9 = Not authorized to quote security ~~11 = Quote Locked - Unable to Update/Cancel~~ 99 = Other
4	Reason Text	C	String (76)	Text specifying the reason for the rejection. Conditionally required if Quote Reject Reason = '99' Other.

Example Message Flows

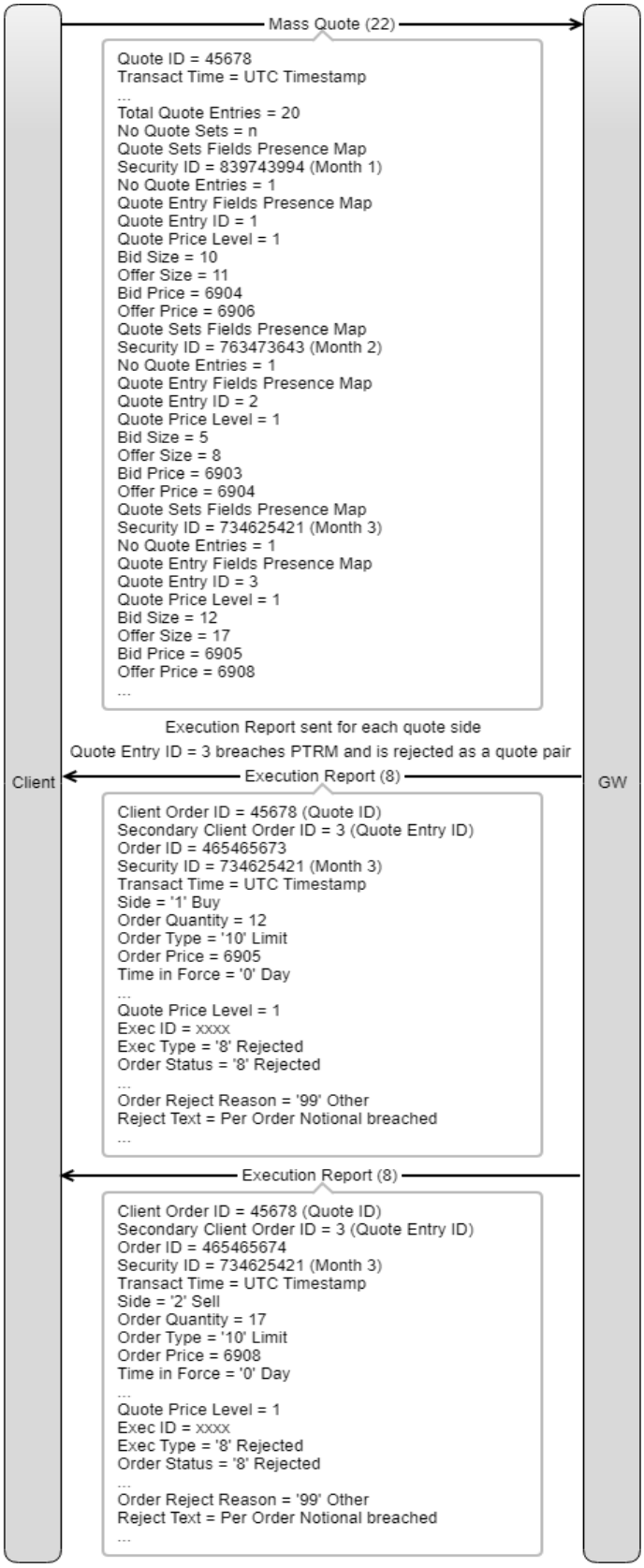
Mass Quote submission



Entire Mass Quote rejected

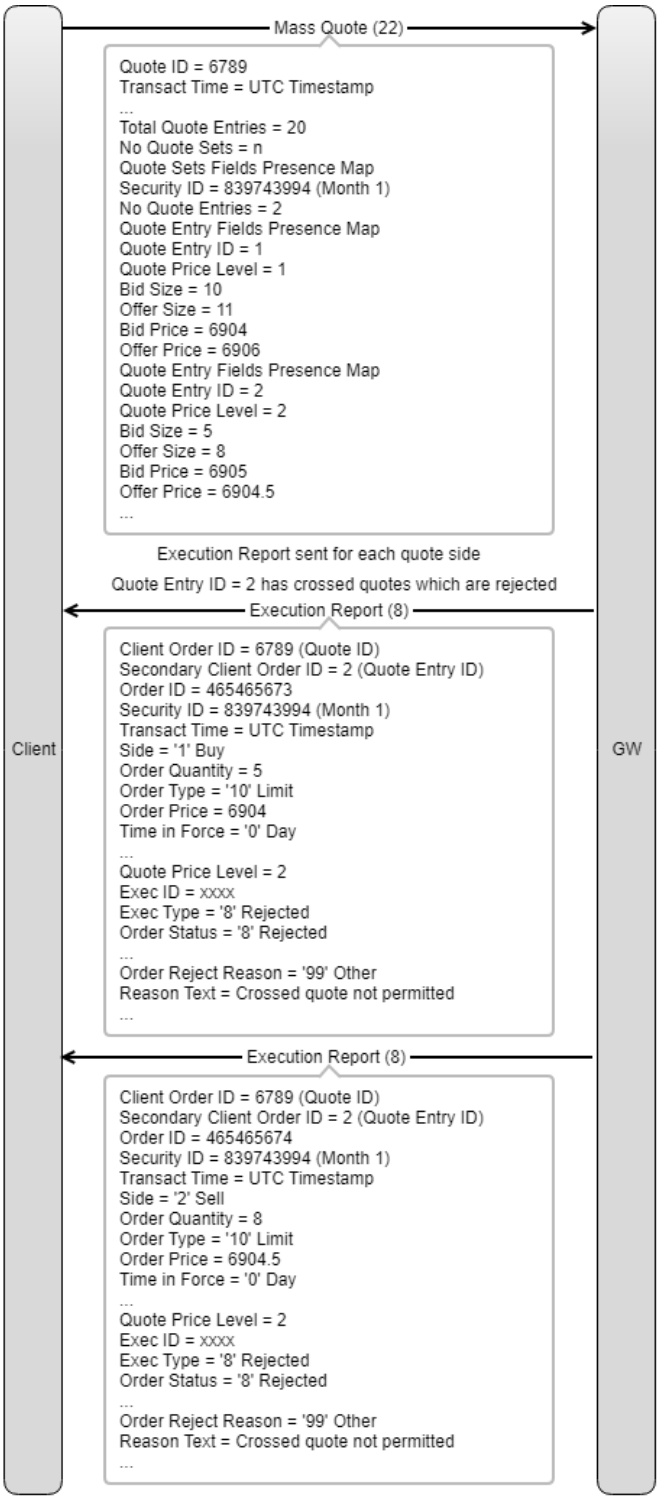


Mass Quote – individual quote rejections

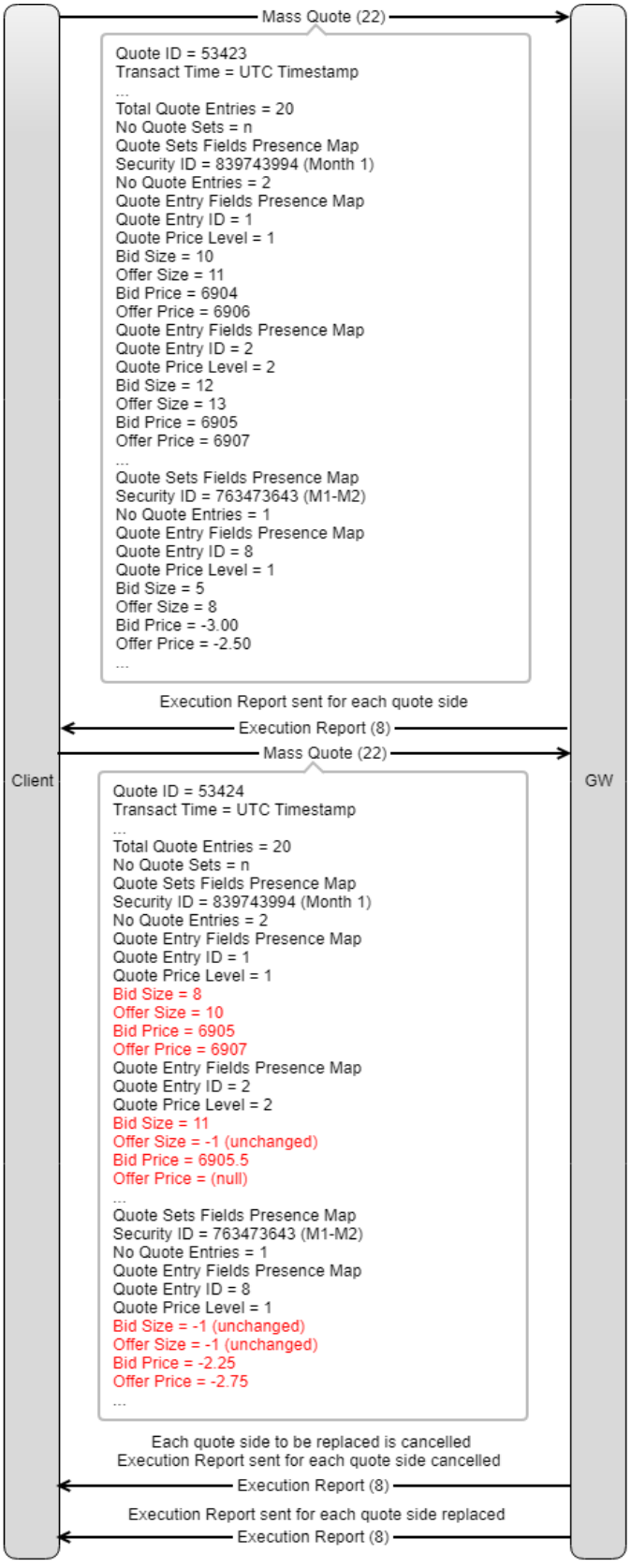


Crossed quotes within a Quote Entry rejected

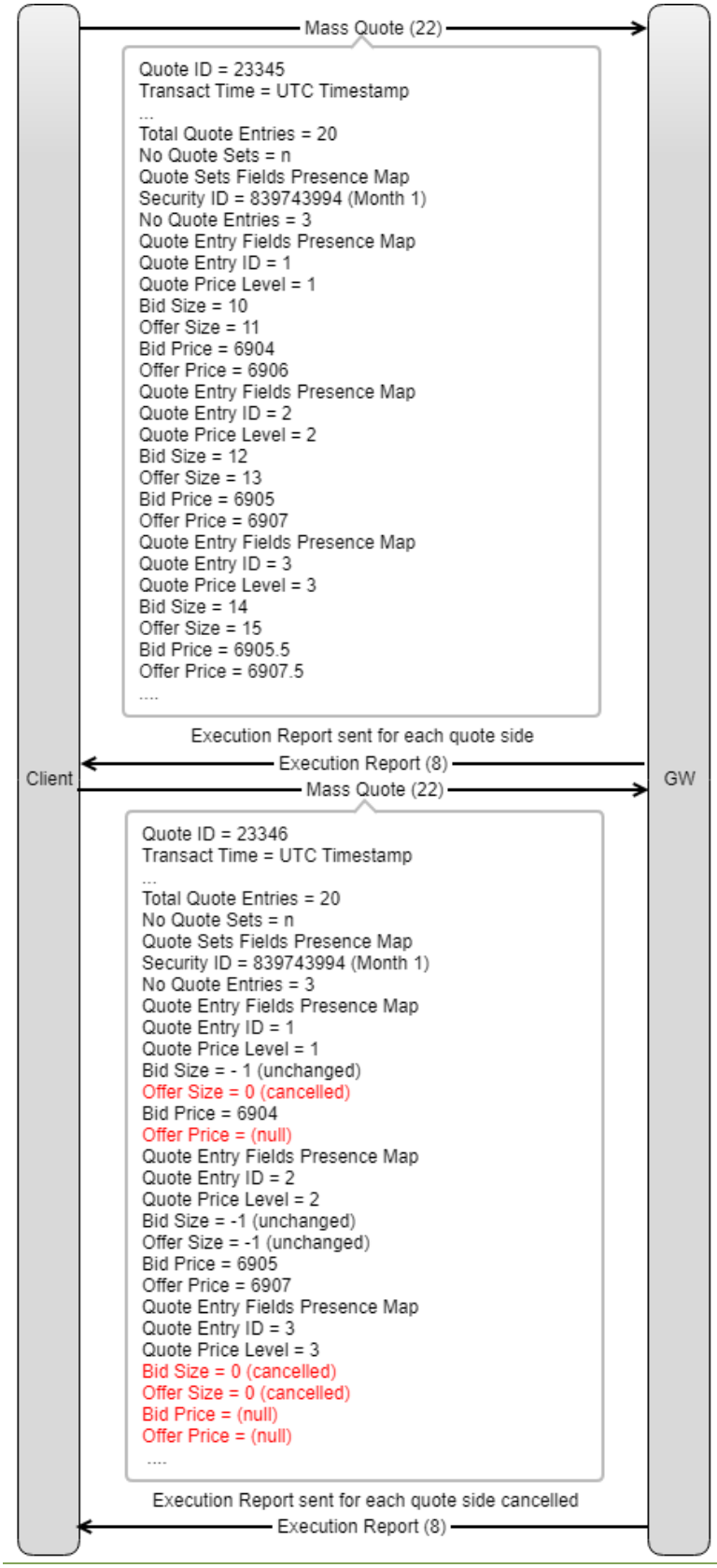
The bid ~~at~~for 6905 for 5 lots and offer at 6904.5 for 8 lots in Quote Entry ID = 2 can trade with each other. The matching engine will reject a cross quote pair within a single Quote Entry. The originator of the Mass Quote will be notified of the rejection by an Execution Report for each quote pair.



Mass Quote entries replaced ~~amended~~



Individual quotes cancelled



4.10.13 Quote Request (20)

Quote Request is used to requests prices from market participants.

The Quote Request is disseminated via the Market Data service to market participants. ~~The originator of the Quote Request will not receive any acknowledgement unless the quote request is rejected.~~

BP	Field Name	Req	Data Type	Description
0	Quote Request ID	Y	String (19)	Client specified identifier for the quote request.
1	Security ID	Y	UInt64	Tradable Instrument identifier.
2	Quote Request Type	Y	UInt8	Indicates the type of Quote Request being generated. Valid values: 1 = Manual - used to indicate a single quote request 2 = Automatic - used to indicate a streaming quote request
3	Transact Time	Y	UInt64	Time when the message was generated.
4	Side	N	UInt8	Side of order. If not defined indicates a two-sided quote is required. Valid values: 1 = Buy 2 = Sell
5	Quantity	N	Int32	Order quantity. If not entered, a volume of 0 will be published.

4.10.14 Quote Request Ack (21)

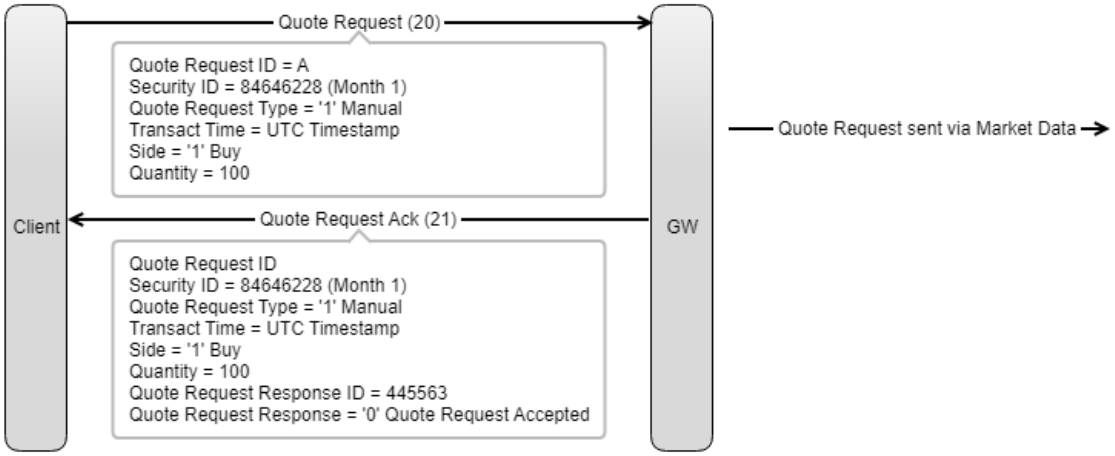
Quote Request Ack (21) is returned by the gateway in response to a Quote Request (20).

BP	Field Name	Req	Data Type	Description
0	Quote Request ID	Y	String (19)	Client specified identifier for the quote request.
1	Security ID	Y	UInt64	Tradable Instrument identifier.

BP	Field Name	Req	Data Type	Description
2	Quote Request Type	Y	UInt8	Indicates the type of Quote Request being generated. Valid values: 1 = Manual - used to indicate a single quote request 2 = Automatic - used to indicate a streaming quote request
3	Transact Time	Y	UInt64	Time when the message was generated.
4	Side	C	UInt8	Side of order. If not defined indicates a two-sided quote is required. Valid values: 1 = Buy 2 = Sell. Conditionally required if specified on the original message.
5	Quantity	C	Int32	Order quantity. Conditionally required if specified on the original message.
6	Quote Request Response ID	Y	String (21)	Unique identifier for the Quote Request Ack assigned by the system.
7	Quote Request Response	Y	UInt8	Indicates the action taken as a result of the Quote Request. Valid values: 0 = Quote Request Accepted 1 = Quote Request Rejected
8	Quote Request Reject Reason	C	UInt8	Code that identifies the reason for the rejection. Valid value: 99 = Other Conditionally required if Quote Request Response = '1' Quote Request Rejected.
9	Reason Text	C	String (76)	Text specifying the reason for the rejection. Conditionally required if Quote Request Reject Reason = '99' Other

Example Message Flow

RFQ Submission



~~4.10.14 Quote Request Rejected (21)~~

~~Quote Request Rejected notifies the originator that their Quote Request (20) has been rejected.~~

~~BP~~	~~Field Name~~	~~Req~~	~~Data Type~~	~~Description~~
~~0~~	~~Quote Request ID~~	~~Y~~	~~String (19)~~	~~Client specified identifier for the quote request.~~
~~1~~	~~Security ID~~	~~Y~~	~~UInt64~~	~~Tradable Instrument identifier.~~
~~2~~	~~Quote Request Type~~	~~Y~~	~~UInt8~~	~~Indicates the type of Quote Request being generated.~~ ~~Valid values:~~ ~~1 = Manual - used to indicate a single quote request~~ ~~2 = Automatic - used to indicate a streaming quote request~~
~~3~~	~~Transact Time~~	~~Y~~	~~UInt64~~	~~Time when the message was generated.~~
~~4~~	~~Side~~	~~C~~	~~UInt8~~	~~Side of order. If not defined indicates a two-sided quote is required.~~ ~~Valid values:~~ ~~1 = Buy~~ ~~2 = Sell.~~ ~~Conditionally required if specified on the original message.~~

BP	Field Name	Req	Data Type	Description
5	Quantity	C	Int32	Order quantity. Conditionally required if specified on the original message.
6	Quote Request Reject Reason	Y	UInt8	Code that identifies the reason for the rejection. Valid value: 99 = Other
7	Reason Text	C	String (76)	Text specifying the reason for the rejection. Conditionally required if Quote Request Reject Reason = '99' Other

4.10.15 MMP Reset Request (30)

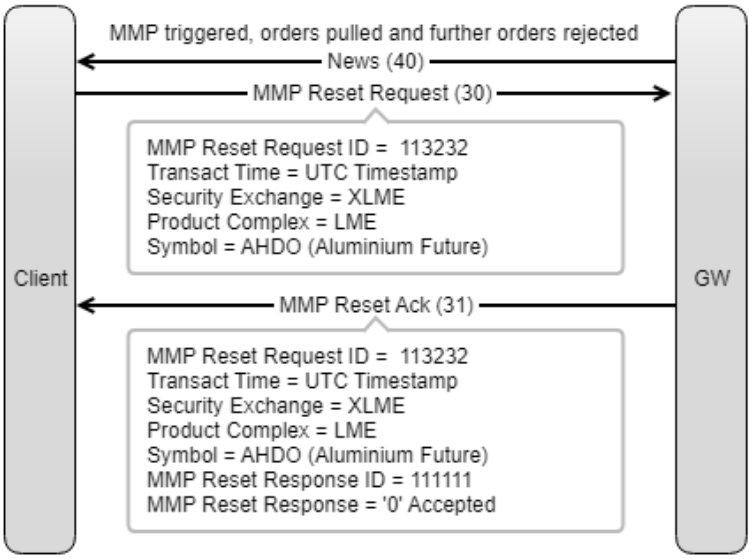
MMP Reset Request is used to reinstate a trader (i.e. Comp ID) that has breached a Market Maker Protection type.

BP	Field Name	Req	Data Type	Description
0	MMP Reset Request ID	Y	String (1921)	Client specified identifier for the MMP Reset request.
1	Transact Time	Y	UInt64	Time when the message was generated.
2	Security Exchange	Y	String (5)	Market which is used to identify the security: XLME
3	Product Complex	Y	String (5)	Identifies an entire suite of products for a given market. Valid values: LME = Base LMEPLPM = Precious
4	Symbol	Y	String (21)	Symbol for the LME contract code e.g. CADF (Copper Future) or OCDF (Copper Monthly Average Future).

4.10.16 MMP Reset Ack (31)

MMP Reset Ack is returned in response to an MMP Reset Request (30).

BP	Field Name	Req	Data Type	Description
0	MMP Reset Request ID	Y	String (19~~21~~)	Client specified identifier for the MMP Reset request.
1	Transact Time	Y	UInt64	Time when the message was generated.
2	Security Exchange	Y	String (5)	Market which is used to identify the security: XLME
3	Product Complex	Y	String (5)	Identifies an entire suite of products for a given market. Valid values: LME = Base ~~LMEP~~LPM = Precious
4	Symbol	Y	String (21)	Symbol for the LME contract code e.g. CADF (Copper Future) or OCDF (Copper Monthly Average Future).
5	MMP Reset Response ID	Y	String (21)	Unique identifier for the MMP Reset Response Ack assigned by the system.
6	MMP Reset Response	Y	UInt8	Specifies the action taken as a result of the MMP Reset Request message. Valid values: 0 = Accepted 2 = Rejected
7	MMP Reset Reject Reason	C	UInt8	Code that identifies the reason for the rejection. Conditionally required when MMP Reset Response = '2' Rejected Valid values: ~~0 = Invalid party or parties~~ 98 = Not authorised 99 = Other
8	Reason Text	C	String (76)	Text specifying the reason for the rejection. Conditionally required if MMP Reset Reject Reason = '99' Other.

Example Message Flow

MMP Reset requested