Order Entry Gateway
FIX Specification

Please respond to:

newtradingplatform@lme.com

Table of Contents

Document History

Version	Date	Change Description
1.0	31/12/2019	Initial draft
1.1	09/04/2020	Updated following internal review

Preface

This document describes the LME implementation of the FIX protocol based on FIX 5.0 SP2 Specification with relevant extension packs.

The document assumes the reader has an understanding of the FIX protocol, see http://www.fixprotocol.org/.

1 Session Management

1.1 Authentication

1.1.1 Comp ID

A FIX session is established by sending a Logon (35=A) request which includes the sender and the target in the Message Header:

- SenderCompID (49) – the party initiating the session.
- TargetCompID (56) – the acceptor of the session as per configuration.

The client should use the Comp ID provided by the Exchange. A single client may have multiple connections to the Gateway i.e., multiple FIX sessions, each with its own Comp ID.

1.1.2 Password Encryption

The client should specify their password in EncryptedPassword (1402) in the Logon request.

To encrypt the password, the client is expected to use a 2048-bit RSA (http://en.wikipedia.org/wiki/RSA_(algorithm)) public key circulated (through a different medium) by the Exchange. 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 Gateway authenticates the participant's Logon (35=A) request and sends a Logon (35=A) response containing SessionStatus (1409) which indicates whether the logon attempt was successful or not.

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

A password change can be made in a Logon (35=A) request. The client should specify the new encrypted password in EncryptedNewPassword (1404) and the current encrypted password in EncryptedPassword (1402).

The status of the new password (i.e. whether it is accepted or rejected) will be specified in the SessionStatus (1409) response from the Gateway. The new password, if accepted, will be effective for subsequent logins.

1.2 Establishing a FIX Session

The client must wait for a successful Logon response before sending additional messages. If additional messages are received from the client before the exchange of Logon messages, the TCP/IP connection with the client will be disconnected.

If a Logon (35=A) attempt fails for the following reasons, the Gateway will send a Logout (35=5) or a Reject (35=3) and then terminate the session:

- Password failure
- Comp ID is locked
- Logon is not permitted during this time

For all other reasons, including the following, the Gateway will terminate the session without sending a Logout or Reject:

- Invalid Comp ID

If during the logon of a client (i.e., a Comp ID), the Gateway receives a second connection attempt while a valid FIX session is already underway for that same Comp ID, the Gateway will terminate both connections without sending a Logout (35=5) or Reject (35=3).

Inbound message sequence number will not be incremented if the connection is abruptly terminated due to the logon failure.

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 (43) is not set to Y = Yes, then the Gateway will send a Logout (35=5) and terminate the FIX session. In this scenario the inbound sequence number will not be incremented.

1.3 Message Sequence Numbers

As outlined in the FIX protocol, the client and Gateway will each maintain a separate and independent set of incoming and outgoing message sequence numbers. Sequence numbers should be initialized to 1 (one) at the start of the day and be incremented throughout the session. Either side of a FIX session will track the:

- NextExpectedMsgSeqNum (789) (starting at 1)
- Next To Be Sent Message Sequence number (starting at 1); with respect to the contra-party.

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

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

If any message sent by one side of a FIX session contains a sequence number that is LESS than the NextExpectedMsgSeqNum (789) then the other side of this session is expected to send a Logout message and terminate the FIX connection immediately, unless the PossDup flag is set to Y = Yes

A FIX 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 (35=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 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 (35=1) is not received within a reasonable transmission time (recommended being an elapsed time equivalent to 3 heartbeat intervals), the Gateway will send a Logout (35=5) 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 FIX Session

Session termination can be initiated by either the Gateway or the client by sending a Logout (35=5). Upon receiving the Logout request, the contra party will respond with a Logout 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 or a Logout reply, the Logout initiator should wait for 60 seconds prior to terminating the connection.

The client is expected to terminate each FIX connection at the end of each trading day before the Gateway is shut down. Any open FIX connections will be terminated by the Gateway sending a Logout when the service is shut down. Under exceptional circumstances, for example, a slow consumer, the Gateway may initiate the termination of a connection during the trading day by sending a Logout.

If, during the exchange of Logout messages, the client or the Gateway detects a sequence gap, it should send a Resend Request.

1.6 Re-establishing a FIX Session

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

Once the FIX session is re-established, the message sequence numbers will continue from the last message successfully transmitted prior to the termination.

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 FIX session the Gateway or the client may use the Sequence Reset (35=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 client side or on Gateway side, the client is expected to be able to continue the same session.

In case of a catastrophic scenario, the Gateway will restart from a higher sequence number considering the previous session or may start from sequence number 1.

If the sequence number is reset to 1 by the Gateway, all previous messages will not be available for the client side.

The client and the Gateway are expected to negotiate on the NextExpectedMsgSeqNum (789) 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, 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.

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 NextExpectedMsgSeqNum (789) during Logon or the MsgSeqNum (34) at other times.

The Resend Request will indicate the BeginSeqNo (7) and EndSeqNo (16) of the message gap identified and when replying to a Resend Request, the messages are expected to be sent strictly honouring the sequence.

If messages are received outside of the BeginSeqNo and EndSeqNo, 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 Sequence number 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 can specify “0” in the EndSeqNo to indicate that they expect the sender to send ALL messages starting from the BeginSeqNo.

In this scenario, if the recovering party receives messages with a sequence greater than the BeginSeqNo, out of sequence, the message will be ignored.

Administrative messages such as Sequence Reset, Heartbeat and Test Request which can be considered irrelevant for a retransmission could be skipped using the Sequence Reset message in gap-fill mode.

Note that the Gateway expects the client to skip Sequence Reset messages when replying to a Resend Request at all times.

When resending messages, the Gateway would use either PossDup or PossResend flag to indicate whether the messages were retransmitted earlier.

If PossDup flag is set to Y = Yes, 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 flag is set to Y = Yes, 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 ExecID (17) may be used for this purpose.

2.2 Resend Request

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

1. To request a single message. The BeginSeqNo and EndSeqNo should be the same.
2. To request a specific range of messages. The BeginSeqNo should be the first message of the range and the EndSeqNo should be the last of the range.

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

2.3 Logon Message Processing – Next Expected Message Sequence

The session initiator should supply the NextExpectedMsgSeqNum (789) the value next expected from the session acceptor in MsgSeqNum (34). The session acceptor should validate the logon request including that NextExpectedMsgSeqNum (789) does not represent a gap. It then constructs its logon response with NextExpectedMsgSeqNum (789) containing the value next expected from the session initiator in MsgSeqNum (34) 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 NextExpectedMsgSeqNum (789) 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 NextExpectedMsgSeqNum (789) 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 based on MsgSeqNum (34) 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 from the client. If a new Resend Request is received during this time, the Gateway will terminate the session immediately without sending the Logout message.

Note that indicating the NextExpectedMsgSeqNum (789) in the Logon (35=A) is mandatory.

2.4 Possible Duplicates

The Gateway handles possible duplicates according to the FIX protocol. The client and the Gateway use the PossDupFlag (43) field to indicate that a message may have been previously transmitted with the same MsgSeqNum (34).

2.5 Possible Resends

The Gateway does not handle possible resends for the client-initiated messages (e.g. New Order, etc.) and the message will be processed without considering the value in the PossResend (97) field. Any message with duplicate ClOrdID (11) will be rejected based on the Client Order ID uniqueness check and messages which conform to the uniqueness check will be processed as normal messages.

The Gateway may use the PossResend (97) field to indicate that an application message may have already been sent under a different MsgSeqNum (34). The client should validate the contents (e.g. ExecID (17)) 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

The Execution Report, Order Mass Cancel Report, Business Message Reject, Reject and News messages generated during a period when a client is disconnected from the Gateway will be sent to the client when it next reconnects on the same business day. In the unlikely event the disconnection was due to a Gateway outage, some messages may not be retransmitted and the messages which will be retransmitted will include a PossResend (97) set to Y = Yes.

3 Service Description

3.1 Security Identification

Each tradable instruments will be identified using a SecurityID (48) which can be a maximum of 19 digits. It is required to specify SecurityIDSource (22) as '8' Exchange Symbol in conjunction with the SecurityID (48).

3.2 Security Creation

A Security Definition Request (35=c) can be submitted to create a new tradable instrument:

Instrument Request Type	FIX Tag
Option strike	MaturityDate (541) StrikePrice (202) PutOrCall (201)
Strategy	SecurityType (167) = MLEG SecuritySubType (762) LegSecurityID (602) LegSecurityIDSource (603) LegRatioQty (623) LegSide (624)

Strategy legs must be specified in order of expiry, Security Definition Requests containing strategy legs that are not in chronological order will not be permitted. The Security Definition (35=d) response will contain the SecurityResponseType (323) = '5' Reject security proposal and the SecurityRejectReason (1607) = '12' Invalid instrument structure.

The accepted strategy definition is the buy side of the strategy. If a Strategy Definition Request is submitted which is the inverse (sell side) of an already created strategy the request will be accepted. The Security Definition returned will contain the SecurityResponseType (323) = '2' Accept security proposal with revisions as indicated in the message and the Security ID of the existing strategy.

3.2.1 Defined Strategy Types

The following defined strategy types are supported:

Futures Strategies

SecuritySubType (762)	Strategy Name	Definition (from buy perspective)
1	Carry	Buy near leg, Sell far leg
3	Average 3M	Buying 3 consecutive (monthly) legs

SecuritySubType (762)	Strategy Name	Definition (from buy perspective)
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).

For an Average strategy only the front leg needs to be specified as the remaining legs will be consecutive.

Options Strategies

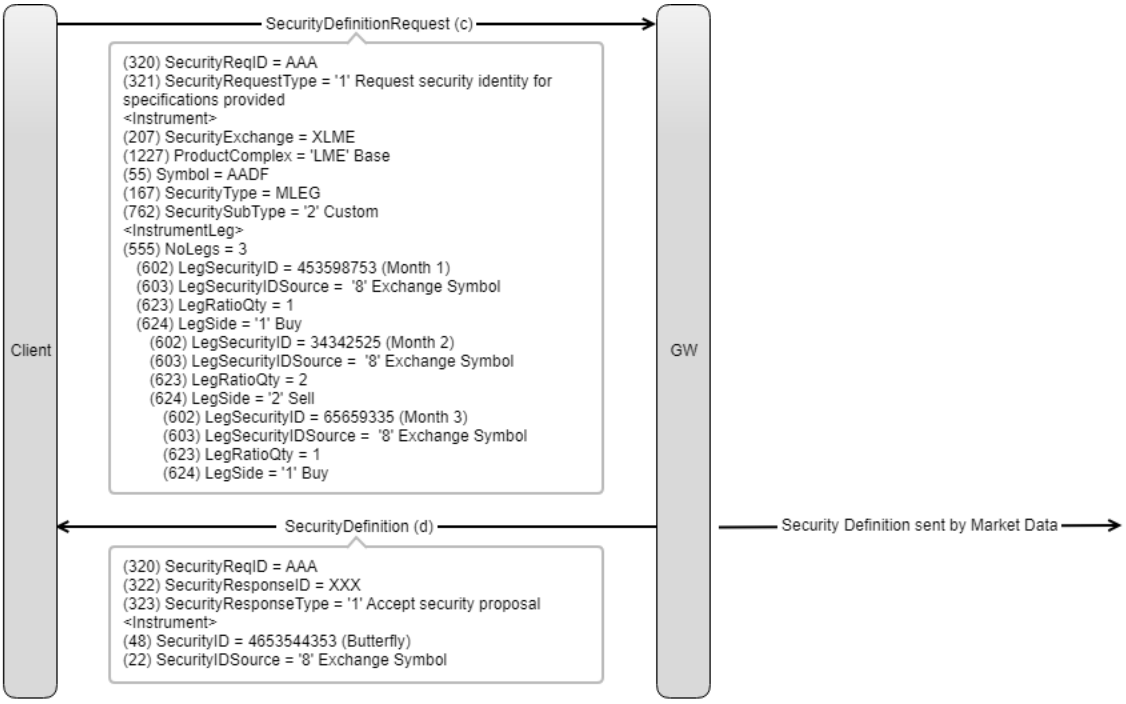
SecuritySubType (762)	Strategy Name	Definition (from buy perspective)
7	Call Spread	Buy a (call) strike, sell a (call) higher strike within the same option expiry
8	Put Spread	Buy a (put) strike, sell a (put) lower strike within the same option expiry

3.2.2 Custom Strategies

A non-Exchange defined strategy can be submitted in a Security Definition Request (35=c) with SecuritySubType (762) = ‘2’ Custom which can consist up to four legs in a Futures contract or premium quoted Option. Each leg in the strategy must be in the same contract.

The custom strategy is defined by specifying the Security ID of each of the legs, the leg ratio and leg side.

For example, a Futures Butterfly can be 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 (35=D) which includes a unique Client Order identifier in the ClOrdID (11).

Multiple orders can be submitted using Mass Quotes available in the Binary protocol.

3.4 Order Management

3.4.1 Order Types

The following order types are supported:

Order Type	FIX Tag and Value
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.	OrdType (40) = 2 Price (44)

Order Type	FIX Tag and Value
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.	OrdType (40) = K
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.	OrdType (40) = 3 StopPx (99) TriggerType (1100) = 4 TriggerPriceType (1107)
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.	OrdType (40) = 4 Price (44) StopPx (99) TriggerType (1100) = 4 TriggerPriceType (1107)
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.	OrdType (40) = 2 Price (44) DisplayQty (1138) OrderQty (38)
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 or if the revision request would result in the order executing it is rejected and the original order remains.	OrdType (40) = 2 Price (44) ExecInst (18) = 6
One Cancels Other (OCO) 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.	OrdType (40) = 2 Price (44) TriggerType (1100) = 4 TriggerPrice (1102) TriggerPriceType (1107) TriggerNewPrice (1110) TriggerOrderType (1111)

3.4.2 Order Validity Conditions

Validity Condition	FIX Tag and Value
Day An order that will expire at the end of the day.	TimeInForce (59) = 0 (default)
Good Till Cancelled (GTC) An order that is valid until it is either cancelled or matched.	TimeInForce (59) = 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.	TimeInForce (59) = 3
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.	TimeInForce (59) = 4
Good Till Date (GTD) The order is valid until the end of the trading date specified.	TimeInForce (59) = 6 ExpireDate (432)

3.4.3 Order Types and Permitted Order Validity Conditions

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

3.5 Order Identification

On order submission a ClOrdID (11) is specified by the originator. The client should comply with the FIX protocol and ensure the Client Order IDs are unique for the duration of the trading day and has

not been used already for a currently persisted order. When an order is accepted, the system assigns an OrderID (37) that is unique for all orders and quotes. When modifying or deleting an order the OrigClOrdID (41) is used to identify the order.

3.6 Order Expiry

No Execution Report will be sent for Day orders when they expire at the end of the trading day.

The originator will receive an Execution Report with ExecType (150) and OrdStatus (39) = ‘C’ Expired for expiring orders submitted with TimeInForce (59) = ‘6’ Good til Date.

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

An Execution Report will be sent for GTC orders if the tradable instrument expires at the end of the trading day.

3.7 Order Restatement

Persisted orders are saved when an instrument enters the Close state. The originator is notified by Execution Report with ExecType (150) and OrdStatus (39) = ‘3’ Done for Day.

On initial logon on the next trading day, Execution Reports are sent for persisted orders that have been returned with ExecType (150) = ‘D’ Restated, OrdStatus (39) = ‘0’ New or ‘1’ Partially Filled and ExecRestatementReason (378) = ‘1’ GT renewal / restatement.

3.8 Order Revision

An order can be amended by using an Order Cancel Replace Request (35=G) and specifying the OrigClOrdID (41). The client can optionally specify the OrderID (37) in the Order Cancel Replace Request message. If the OrderID (37) is specified, the system will validate whether the OrderID is associated with the correct order as identified using the OrigClOrdID (41). The Order Cancel Replace Request will be rejected if the specified OrderID (37) is invalid based on this validation.

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

- Price (44)
- StopPx (99)
- OrderQty (38)
- DisplayQty (1138)
- ExpireDate (432)
- TriggerPrice (1102)
- TriggerNewPrice (1110)
- OrderOrigination (1724)
- OrderCapacity (528)
- OrderAttributeType (2594)
- Text (58)

The following MiFID Party Roles (452) can be amended if specified on the original order:

- 300 Investment Decision Within Firm
- 301 Execution Decision Within Firm
- 302 Investment Decision Country Code
- 303 Execution Decision Country Code
- 304 Client Branch Country Code

If an attribute is not present in the Order Cancel Replace Request it will retain its original value. If value(s) have been previously specified for OrderOrigination (1724) and OrderAttributeType (2594) are not specified in the Order Cancel Replace Request it indicates that the values have been removed.

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

The StopPx (99), TriggerPrice (1102) or TriggerNewPrice (1110) cannot be amended if the Stop order has been triggered.

If the client sends an Order Cancel Replace Request for an order for which an Order Cancel Replace Request is being processed the second Order Cancel Replace Request is rejected. If an Order Cancel Request is submitted for an Order Cancel Replace Request that is being processed, the incoming Order Cancel Request will be accepted.

3.9 Order Cancellation

An individual order can be cancelled using Order Cancel Request (35=F) by specifying the OrigClOrdID (41).

The client can optionally specify the OrderID (37) in the Order Cancel Request. If the OrderID (37) is specified, the system will validate whether the OrderID is associated with the correct order as identified by the OrigClOrdID. The Order Cancel Request will be rejected if the specified OrderID is invalid based on this validation.

A successful cancellation will return an Execution Report (35=8). If the cancel is rejected, an Order CancelReject (35=9) is sent containing CxlRejReason (102).

3.10 Mass Cancellation

Multiple orders can be cancelled using Order Mass Cancel Request (35=q) by specifying which orders are to be cancelled:

Cancellation Type	FIX Tag and Value
All orders for a FIX Comp ID	MassCancelRequestType (530) = 7
All orders for a tradable instrument	MassCancelRequestType (530) = 1
All orders for a specified contract	MassCancelRequestType (530) = 3

Cancellation Type	FIX Tag and Value
All orders for an end client account	MassCancelRequestType (530) = 7 PartyRole (452) = '81' Broker Client ID
All orders for a specific contract and side of the market	MassCancelRequestType (530) = 3 Side (54)

If the Mass Cancel Request is accepted, Execution Reports will be sent for each order cancellation and reference the ClOrdID (11) provided on the Order Mass Cancel Request. An Order Mass Cancel Report (35=r) will follow and specify the TotalAffectedOrders (533) in the MassCancelResponse (531).

If the Mass Cancel Request is rejected, an Order Mass Cancel Report will be sent with the MassCancelResponse (531) = '0' Cancel Request Rejected and will include the MassCancelRejectReason (532).

3.11 Cancel on Disconnect

The Gateway will not automatically cancel a user's non-persisted orders in the event of a Logout. A user should explicitly cancel such orders prior to Logout using an Order Mass Cancel Request (35=q).

On order submission a user can specify whether non-persisted orders should be cancelled on system disconnection either as a result of a network issue or in the event of inactivity such as too many missed heartbeats by using ExecInst (18) = 'o' Cancel on Connection Loss.

On detection of a loss of connectivity, the Gateway will use ExecInst (18) = 'o' Cancel on Connection Loss to determine whether a user's non-persisted orders are cancelled.

3.12 Speed Bumps

Exchange contracts may be configured with speed bumps. A speed bump will only be applicable to New Order Single and Order Cancel Replace Requests.

Passive orders, Order Cancel Requests, Order Mass Cancel Requests, Mass Quotes and Post Only orders will be exempt.

The status of an order in a speed bump will be reported in ExecTypeReason (2431) in the Execution Report:

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

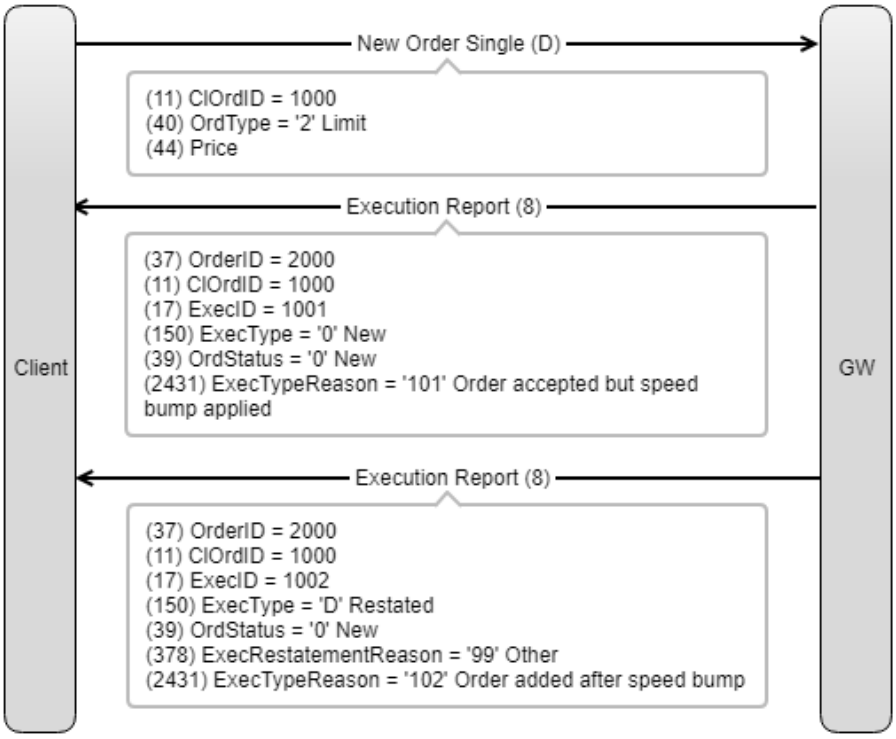
108 = Order rejected after speed bump delay

109 = Unsolicited cancel while in speed bump

Order submission is speed bumped

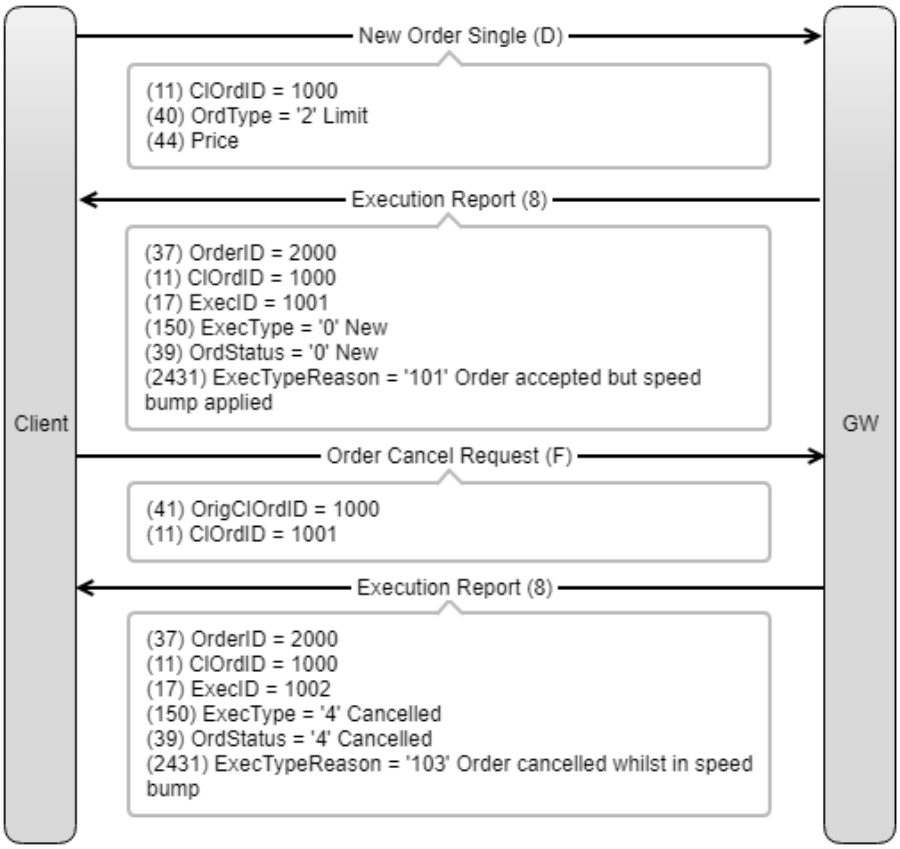
If an order is submitted but is subject to a speed bump, the order is held and not added to the order book until the order has been released from the speed bump. The Execution Report sent in acknowledgement includes an ExecTypeReason (2431) = ‘101’ Order accepted but speed bump applied.

The Execution Report sent once the order has cleared the speed bump and is added to the order book includes ExecType (150) = ‘D’ Restated and ExecTypeReason (2431) = ‘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 Order Cancel Request is not subject to speed bump conditions. The Execution Report sent in response to the cancellation includes ExecTypeReason (2431) = ‘103’ Order cancelled whilst in speed bump delay.

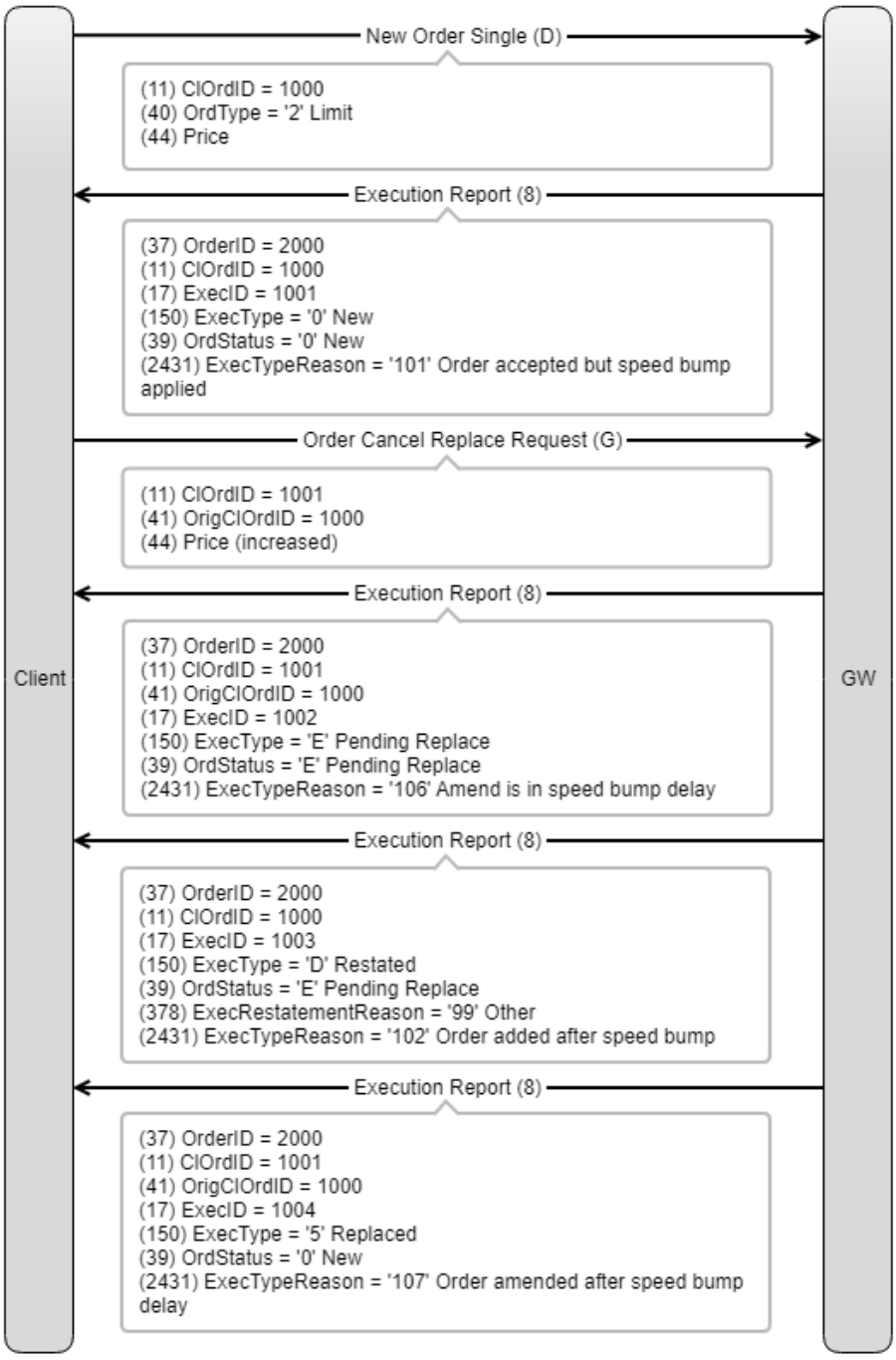


Executable order revision for a speed bumped order will be speed bumped

An order is submitted which is subject to a speed bump. An Order Cancel Replace Request is submitted while the order submission is in the speed bump queue. The revised order is executable and therefore speed bumped. The Execution Report for the order revision includes ExecTypeReason (2431) = ‘106’ Amend is in speed bump delay. The Order Cancel Replace Request 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 ExecType (150) = ‘D’ Restated, OrdStatus (39) = ‘E’ Pending Replace and ExecTypeReason (2431) = ‘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 ExecType (150) = ‘5’ Replaced and ExecTypeReason (2431) = ‘107’ Order amended after speed bump delay.

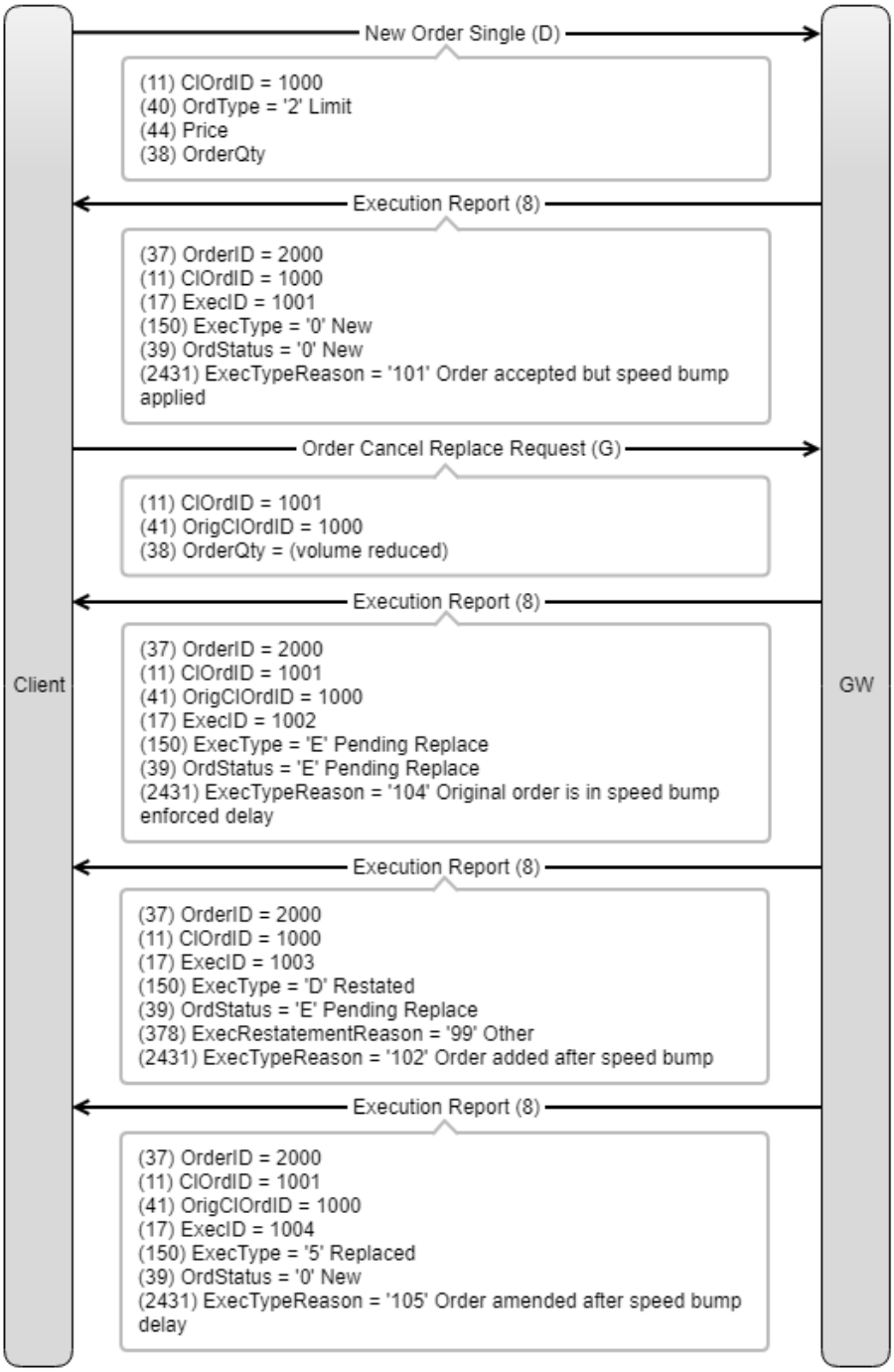


Non-executable order revision for a speed bumped order will not be speed bumped

An order is submitted which is subject to a speed bump. An Order Cancel Replace Request is submitted while the order submission is in the speed bump queue. The revised order will rest in the order book and is therefore not subject to speed bump conditions. The Order Cancel Replace Request will not be processed until the original order has cleared the speed bump therefore the Execution Report for the revision includes ExecType (150) = ‘E’ Pending Replace and ExecTypeReason (2431) = ‘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 ExecType (150) = ‘D’ Restated and ExecTypeReason (2431) = ‘102’ Order added after speed bump.

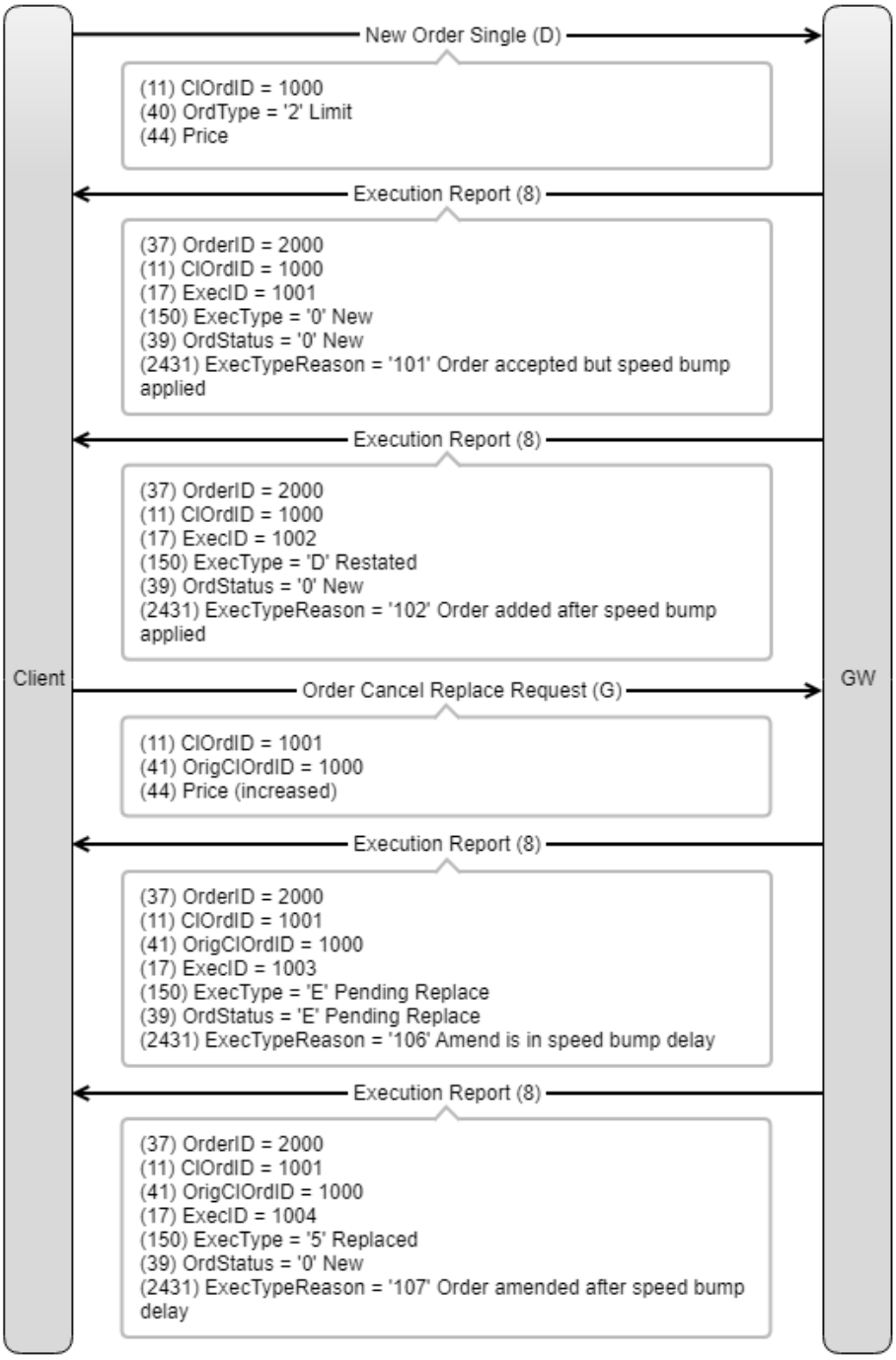
When the order is replaced the Execution Report includes ExecType (150) = ‘5’ Replaced and ExecTypeReason (2431) = ‘105’ Order updated after speed bump delay.



Executable order revision for a resting order will be speed bumped

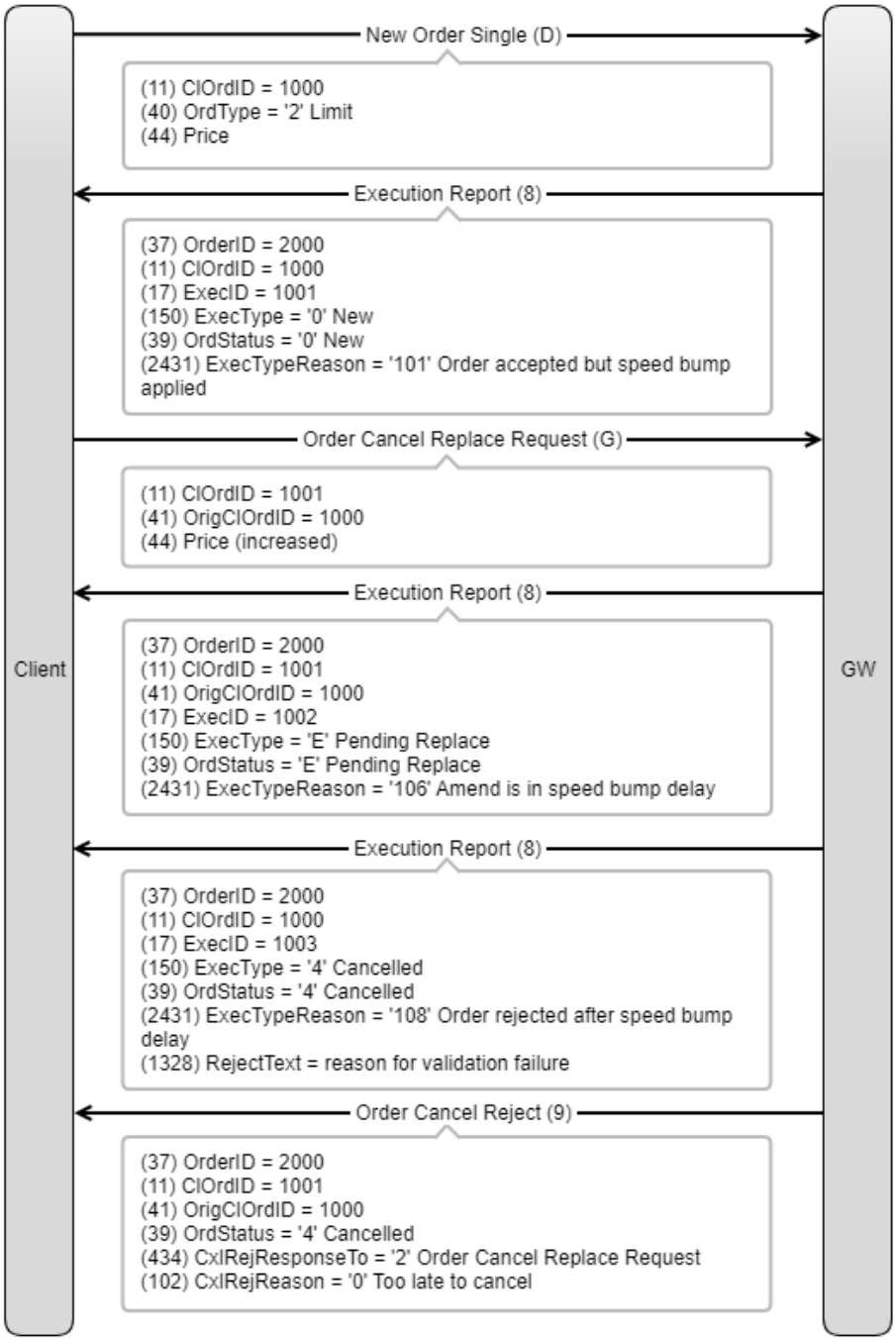
An order revision is submitted for a resting order that was previously speed bumped. The Order Cancel Replace Request is speed bumped as the revision will not provide liquidity. The Execution Report for the revision includes ExecType (150) = ‘E’ Pending Replace with and ExecTypeReason (2431) = ‘106’ Amend is in speed bump delay.

When the order is replaced the Execution Report includes ExecType (150) = ‘5’ Replaced and ExecTypeReason (2431) = ‘107’ Order amended after speed bump delay.



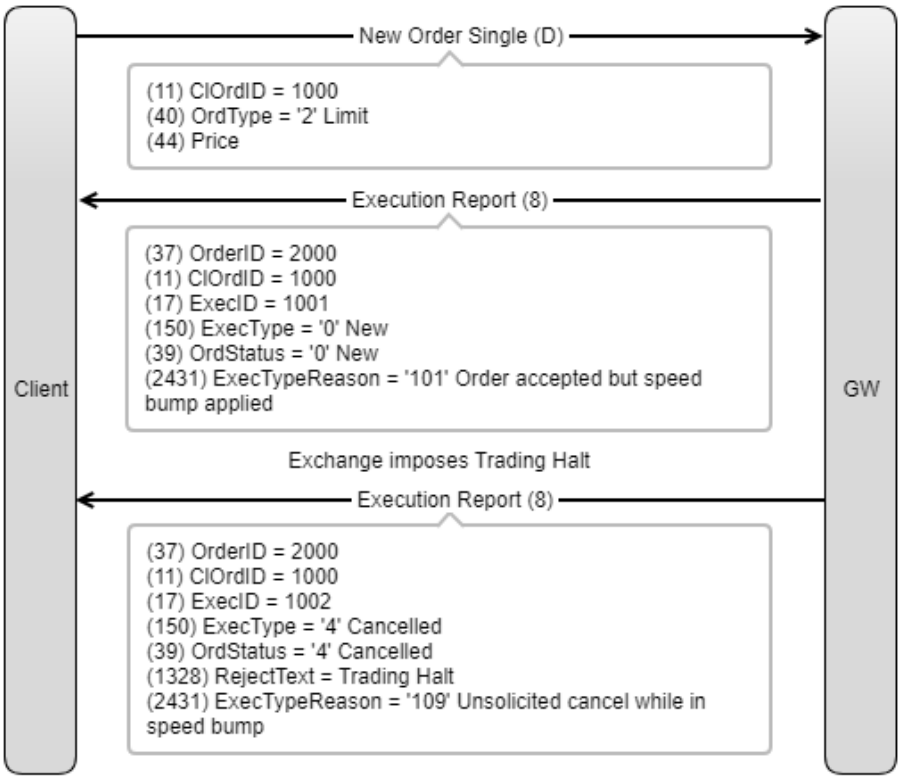
Speed bumped order is cancelled due to validation failure (inflight speed bumped revision also cancelled)

An order is submitted which is subject to a speed bump. An Order Cancel Replace 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 ExecType (150) = '4' Cancelled and ExecTypeReason (2431) = '108' Order rejected after speed bump delay with the reason for the business validation failure in RejectText (1328). An Order Cancel Reject is sent for the order revision.



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 ExecType (150) = '4' Cancelled with ExecTypeReason (2431) = '109' Unsolicited cancel while in speed bump and RejectText (1328) = Trading Halt.



3.13 Message Throttling

The LME implements a message throttle which limits the maximum number of order submissions, revisions, Quote Requests (35=R) and Security Definition Requests (35=c) that can be submitted per second by a FIX Comp ID. Messages submitted in excess of the throttle limit in any given second will result in those messages being rejected by the gateway and will be notified by a Business Message Reject (35=j).

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

Note, order cancellation messages are exempt from throttling.

3.14 Security Definition Throttle

The number of Security Definition Requests (35=c) that can be submitted by a FIX Comp ID are set at per day rate and also included in the per second message throttle. A user breaching the daily limit will have further message submissions rejected by the gateway.

3.15 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 or a member can configure SEP identifiers and specify the action to be taken if two orders with an identical SEP ID could execute.

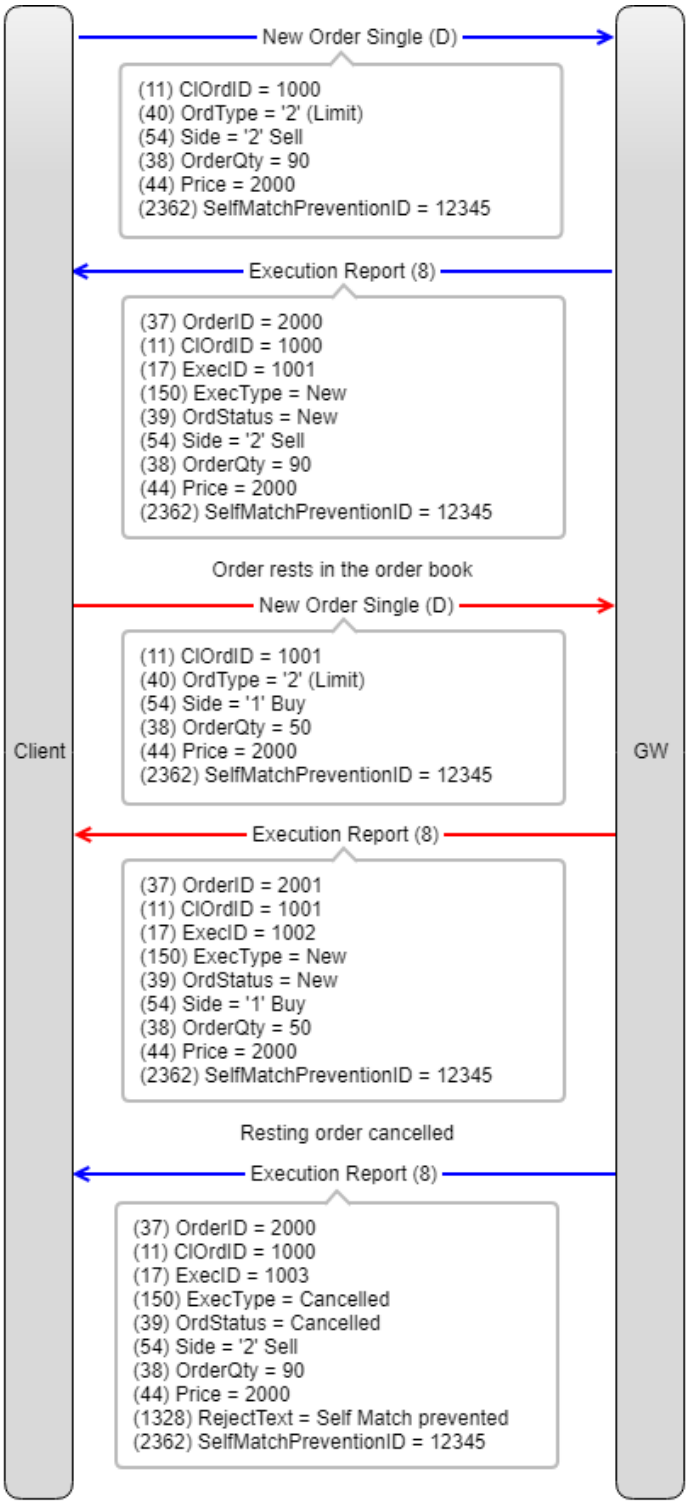
A SEP ID will be specified as a maximum of 9 digits. A member will use a Party Entitlements Definition Request (35=DA) submitted via the Risk Management Gateway to define the SEP configuration. This configuration will be effective from the next trading day.

A SEP ID can be entered in the SelfMatchPreventionID (2362) on order submission. If orders with an identical SEP ID can cross the SEP handling action that has been configured is triggered to cancel either the incoming or resting order.

The Execution Report sent for the cancelled order will contain RejectText (1328) = Self Match prevented.

The availability of SEP functionality will be determined by the Exchange. If an order is submitted with the SelfMatchPreventionID (2362) populated and SEP is not available the order will be rejected. The Execution Report sent will contain RejectText (1328) = Self Match Prevention not configured for the tradable instrument.

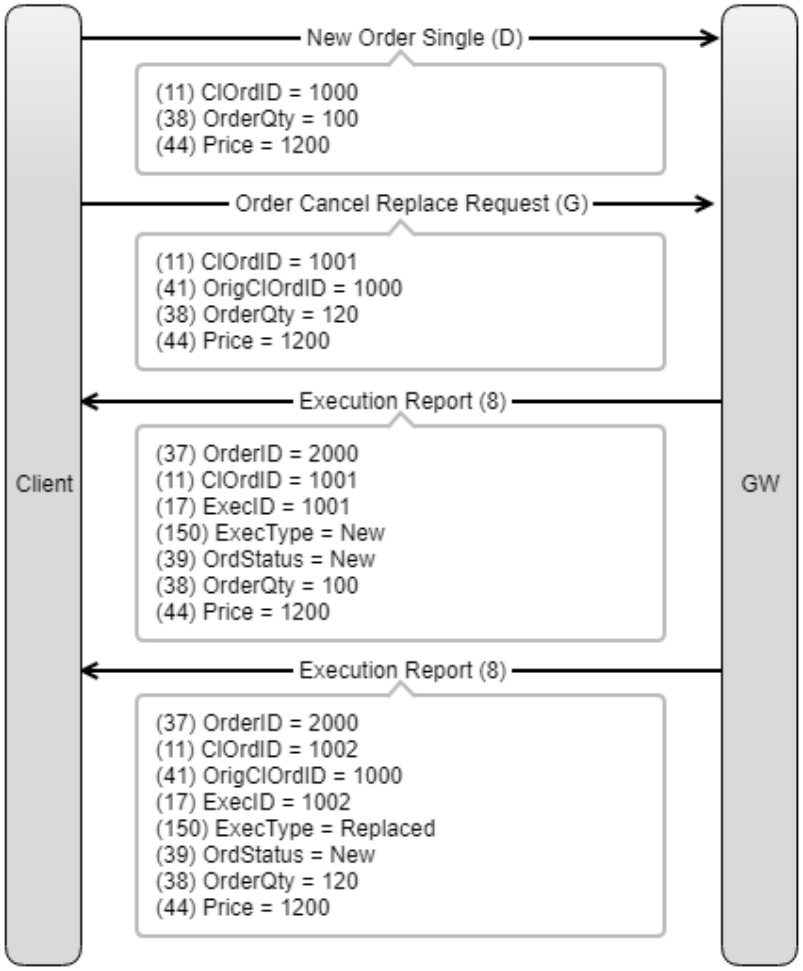
Self Execution Prevention triggered – resting order cancelled



3.16 Inflight Order Processing

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

For example, a New Order Single is submitted followed immediately afterwards by an Order Cancel Replace Request. An Execution Report is returned for the order submission and then the revision.



3.17 Trade Reporting

When an order matches the trade half will be assigned an identifier which will be reported in the TrdMatchID (880) on the Execution Report (35=8). A strategy trade will be reported in a single Execution Report including the leg details. The legs of strategy trade will be assigned a LegAllocID (1366).

4 Message Definitions

4.1 Supported Messages by Protocol

Message	FIX 5.0	Binary
Logon (A) Establishes a FIX session	✓	✓
Heartbeat (0) Used to check connectivity	✓	✓
Test Request (1) Used to verify a connection is still active	✓	✓
Resend Request (2) Request the retransmission of messages	✓	✓
Reject (3) Issued when a message is received but cannot be properly processed due to a session-level rule violation	✓	✓
Sequence Reset (4) Indicates there is a gap in the message sequence numbers	✓	✓
Logout (5) Terminates a FIX session	✓	✓
Business Message Reject (j) Reject an application level message which fulfils session level rules	✓	✓
News (B) Disseminates text information	✓	✓
Security Definition Request (c) Request the creation of a tradable instrument	✓	✓
Security Definition (d) Response to a Security Definition Request	✓	✓

Message	FIX 5.0	Binary
New Order Single (D) Submit a new order for execution	✓	✓
Order Cancel Replace Request (G) Revise an existing order	✓	✓
Order Cancel Request (F) Request the cancellation of all the remaining quantity of an existing order	✓	✓
Order Cancel Reject (9) Reports that an Order Cancel Request or Order Cancel Replace Request has been rejected	✓	✓
Execution Report (8) Sent in response to order and fill related client messages	✓	✓
Order Mass Cancel Request (q) Cancel multiple orders	✓	✓
Order Mass Cancel Report (r) Acknowledgement of the Order Mass Cancel Request	✓	✓
Quote Request (R) Requests quotes from market participants	✓	✓
Quote Request Reject (AG) Reports that a Quote Request has been rejected	✓	✓
Mass Quote Allows multiple orders to be submitted as quote pairs in multiple tradable instruments in a single contract		✓
Mass Quote Ack Reports the rejection of a Mass Quote at message level		✓
Quote Cancel Cancels Mass Quote submissions		✓

Message	FIX 5.0	Binary
Quote Status Report Acknowledgement of the Quote Cancel		✓
Party Action Request Reinstate after Market Maker Protection limit breach		✓
Party Action Report Acknowledgement of the Party Action Request		✓

4.2 Inbound Messages

- Logon (A)
- Heartbeat (0)
- Test Request (1)
- Resend Request (2)
- Sequence Reset (4)
- Logout (5)
- Security Definition Request (c)
- New Order Single (D)
- Order Cancel Replace Request (G)
- Order Cancel Request (F)
- Order Mass Cancel Request (q)
- Quote Request (R)

4.3 Outbound Messages

- Logon (A)
- Heartbeat (0)
- Test Request (1)
- Resend Request (2)
- Sequence Reset (4)
- Logout (5)
- Reject (3)
- Business Message Reject (j)
- News (B)

- Security Definition (d)
- Order Cancel Reject (9)
- Execution Report (8)
- Order Mass Cancel Report (r)
- Quote Request Reject (AG)

4.4 Data Types

Data Types used are based on the published standard FIX specifications.

Data Type	Format
UTCTimestamp (27)	YYYYMMDD-HH:mm:ss.SSSSSSSSS
Price (20)	Can be up to 12 significant digits before the decimal point (with provision for a negative value) and at the most 6 decimal places For example, 1234567891234.567891 -123456789123.456789

4.5 Required Fields

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

Y - Required by FIX

Y* - Required by LME

C - Conditionally required by FIX

C* - Conditionally required by LME

N - Not required/Optional.

4.6 Message Header

Tag	Field Name	Req	Data Type	Description
8	BeginString	Y	String (8)	Always set to FIXT1.1
9	BodyLength	Y	Length	Message length, in bytes, forward to the CheckSum field.
35	MsgType	Y	String (3)	Defines message type.
1128	ApplVerID	N	String (1)	Version of FIX used in the message:

Tag	Field Name	Req	Data Type	Description
				9 = FIX50SP2 Returned by the Gateway
49	SenderCompID	Y	String (10)	Identifies the sender of the message.
56	TargetCompID	Y	String (10)	Identifies the receiver of the message. LME
34	MsgSeqNum	Y	SeqNum (9)	Message sequence number.
43	PossDupFlag	N	Boolean	Indicates whether the message was previously transmitted with the same MsgSeqNum (34). Absence of this field is interpreted as original transmission (N).
97	PossResend	N	Boolean	Indicates whether the message was previously transmitted under a different MsgSeqNum (34). Absence of this field is interpreted as original transmission (N).
52	SendingTime	Y	UTCTimestamp	Time the message was transmitted.
122	OrigSendingTime	C	UTCTimestamp	Conditionally required for messages sent as a result of a Resend Request (2). If the original time is not available, this should be the same value as SendingTime (52).

4.7 Message Trailer

Tag	Field Name	Req	Data Type	Description
10	CheckSum	Y	String (7)	Standard check sum described by FIX protocol. Always last field in the message; i.e. serves, with the trailing <SOH>, as the end-of-message delimiter. Always defined as three characters.

4.8 Administrative Messages

4.8.1 Logon (A)

The first messages exchanged in a FIX session are the Logon request and the Logon response. The main purposes of the Logon request and response are:

- To authenticate the client.
- To agree on the sequence numbers.

On initial logon the status of persisted orders is communicated to the FIX session 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 of the FIX Logon request.

Tag	Field Name	Req	Data Type	Description
98	EncryptMethod	Y	Int	Method for encryption. Valid value is: 0 = None
108	HeartBtInt	Y	Int	Heartbeat interval in seconds.
789	NextExpectedMsgSeqNum	Y*	SeqNum (9)	Next expected MsgSeqNum (34) value to be received.
1400	EncryptedPasswordMethod	N	Int	Enumeration defining the encryption method used to encrypt password fields: 101 = RSA
1402	EncryptedPassword	Y	Data	Encrypted password – encrypted via the method specified in EncryptedPasswordMethod (1400)

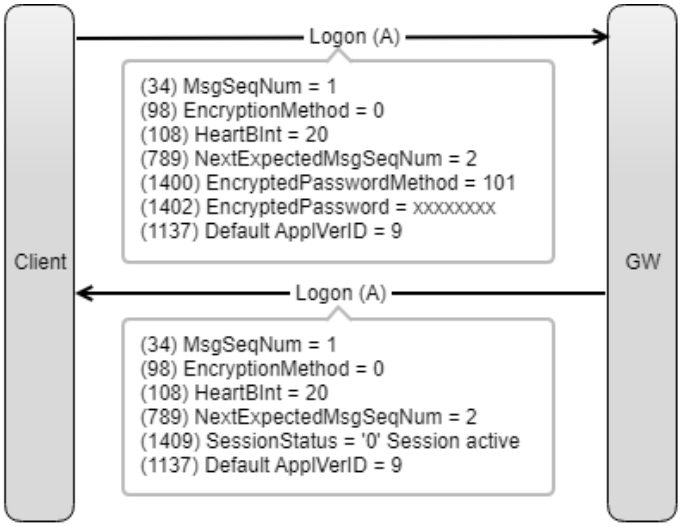
Tag	Field Name	Req	Data Type	Description
1404	EncryptedNewPassword	N	Data	Encrypted new password – encrypted via the method specified in EncryptedPasswordMethod (1400)
1137	DefaultApplVerID	Y	String (1)	The default version of FIX being used in this session. 9 = FIX50SP2

A Logon message is returned in response to an incoming Logon message to initiate a FIX session. The SessionStatus (1409) indicates whether the logon attempt was successful or not.

Tag	Field Name	Req	Data Type	Description
98	EncryptMethod	Y	Int	Method for encryption. Valid value is: 0 = None
108	HeartBtInt	Y	Int	Heartbeat interval in seconds.
789	NextExpectedMsgSeqNum	Y*	SeqNum (9)	Next expected MsgSeqNum (34) value to be received.
1409	SessionStatus	N	Int	Status of the FIX session. Valid values: 0 = Session active 1 = Session password changed
1137	DefaultApplVerID	Y	String (1)	The default version of FIX being used in this session. 9 = FIX50SP2

Example Message Flow

Initial Logon



4.8.2 Heartbeat (0)

Heartbeat (35=0) is sent at the interval specified in HeartBtInt (108) in Logon (35=A). It is also sent in response to a Test Request (35=1).

Tag	Field Name	Req	Data Type	Description
112	TestReqID	C	String (20)	Conditionally required if the heartbeat is a response to a Test Request (1). The value in this field should echo the TestReqID (112) received in the Test Request.

4.8.3 Test Request (1)

Test Request (35=1) can be sent by either the Client or Gateway to verify a connection is active. The recipient responds with a Heartbeat (35=0).

Tag	Field Name	Req	Data Type	Description
112	TestReqID	Y	String (20)	Identifier included in Test Request message to be returned in resulting Heartbeat (0).

4.8.4 Resend Request (2)

Resend Request (35=2) is used to initiate the retransmission of messages if a sequence number gap is detected.

To request a single message. The BeginSeqNo and EndSeqNo should be the same.

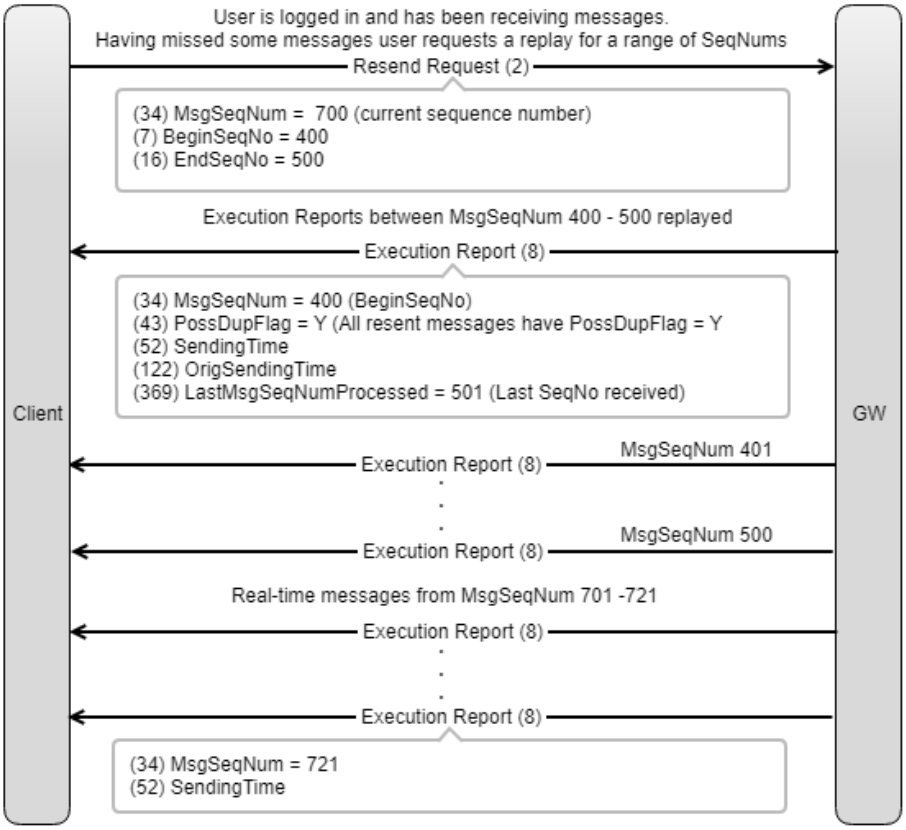
To request a specific range of messages. The BeginSeqNo should be the first message of the range and the EndSeqNo should be the last of the range.

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

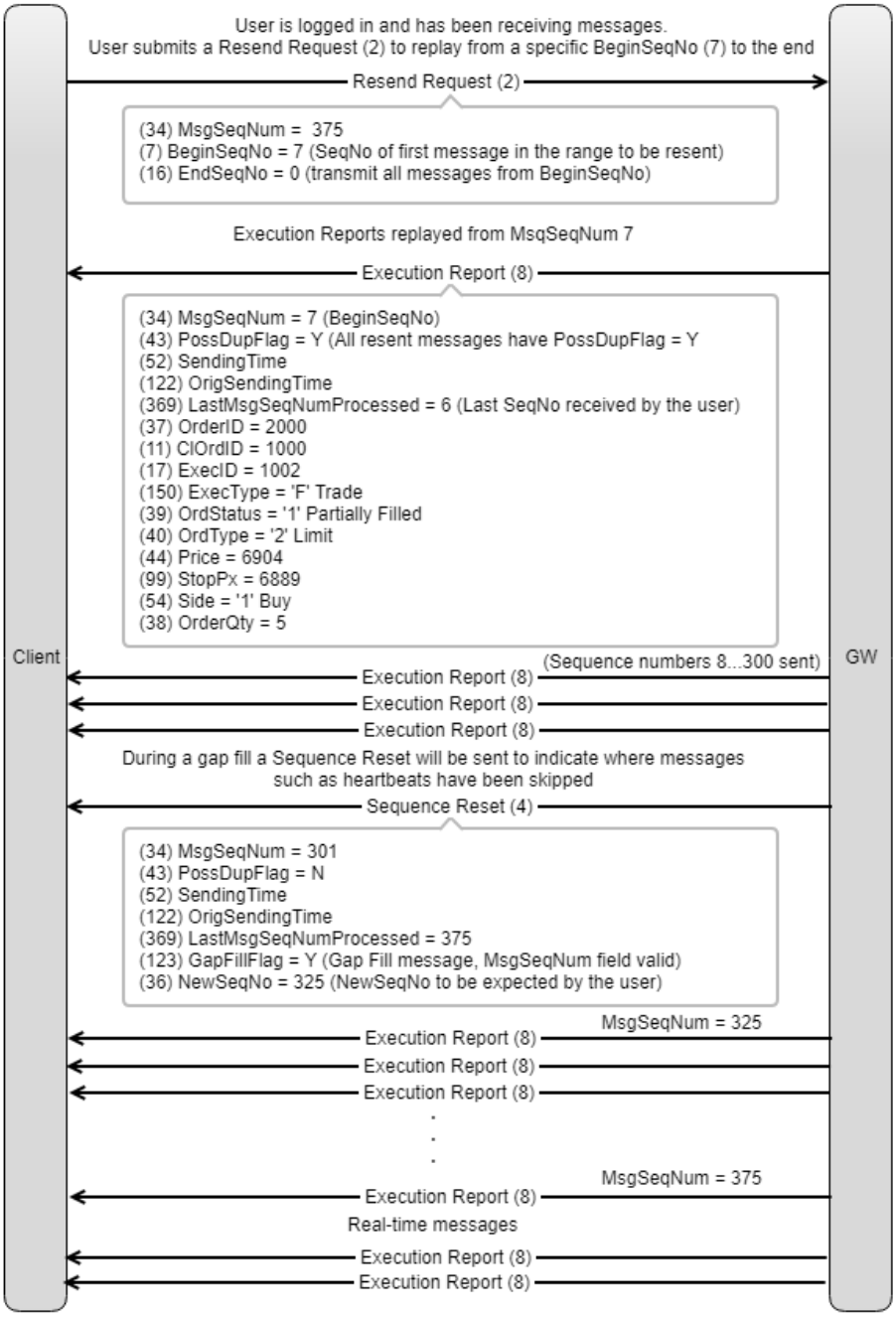
Tag	Field Name	Req	Data Type	Description
7	BeginSeqNo	Y	SeqNum (9)	Message sequence number of the first message in the range to be resent.
16	EndSeqNo	Y	SeqNum (9)	Sequence number of the last message expected to be resent. This may be set to 0 to request the sender to transmit ALL messages starting from BeginSeqNo (7).

Example Message Flow

Resend Request for a range of messages

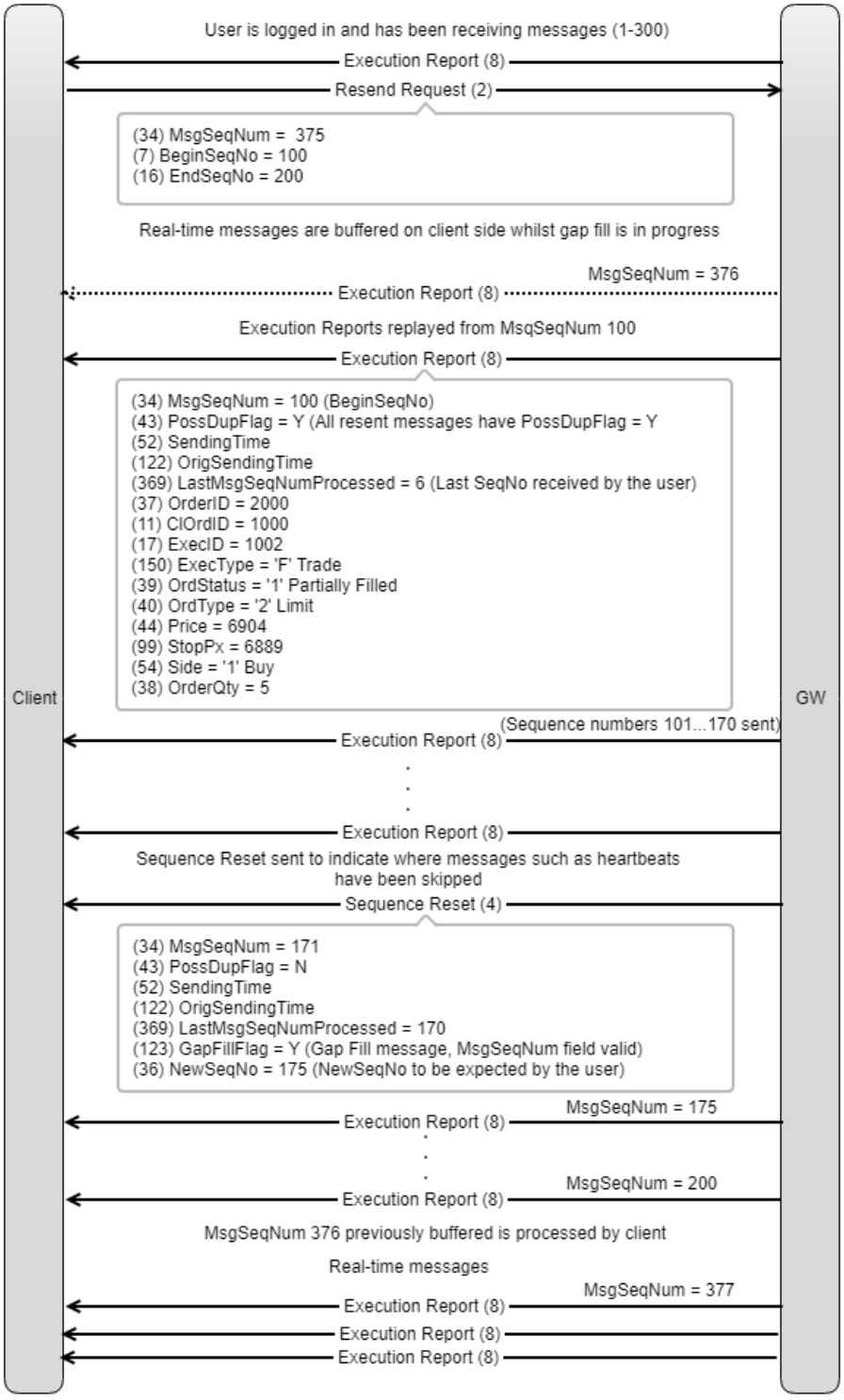


Resend Request for all messages after a particular message



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 (35=4) 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.

Tag	Field Name	Req	Data Type	Description
123	GapFillFlag	N	Boolean	Indicates that the Sequence Reset message is replacing administrative or application messages which will not be resent. Valid value: Y = Gap Fill message, MsgSeqNum (34) field valid. N = Sequence Reset, ignore MsgSeqNum (tag 34). If omitted default value is N.
36	NewSeqNo	Y	SeqNum (9)	Sequence number of the next message to be transmitted.

4.8.6 Logout (5)

Logout (35=5) initiates or confirms the termination of a FIX session. FIX clients should terminate their sessions gracefully by logging out.

If a FIX 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 FIX user has their password reset by LME Market Operations and attempts to login with their previous password, the user will receive a Logout with SessionStatus (1409) = Password change is required.

Tag	Field Name	Req	Data Type	Description
1409	SessionStatus	N	Int	Session status at time of logout. Valid values: 3 = New session password does not comply with policy 4 = Session logout complete 5 = Invalid username or password 6 = Account locked 7 = Logons are not allowed at this time 100 = Password change is required 101 = Other
58	Text	N	String (50)	Reason for logout.

4.8.7 Reject (3)

Reject (35=3) will be sent when a message is received but cannot be properly processed by the Gateway due to a session level rule violation. For example, a message missing a mandatory tag.

Tag	Field Name	Req	Data Type	Description
45	RefSeqNum	Y	SeqNum (9)	Sequence number of the message which caused the rejection.
371	RefTagID	N	Int	If a message is rejected due to an issue with a particular field its tag number will be indicated.
372	RefMsgType	N	String (2)	Message type of the rejected message.
373	SessionRejectReason	N	Int	Code specifying the reason for the session level rejection: Valid values: 0 = Invalid Tag Number 1 = Required Tag Missing 2 = Tag not defined for this message 3 = Undefined tag 4 = Tag specified without a value 5 = Value is incorrect (out of range) for this tag 6 = Incorrect data format for value 9 = CompID problem 10 = Sending Time Accuracy problem 11 = Invalid Msg Type 13 = Tag appears more than once 15 = Repeating group fields out of order 16 = Incorrect NumInGroup count for repeating group 18 = Invalid/Unsupported Application Version 99 = Other.
58	Text	N	String (50)	Text specifying the reason for the rejection.

4.8.8 Business Message Reject (j)

Once an application level message passes validation at FIX 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 tags for rejection handling where a specific tag is not available the Business Message Reject message (35=j) will be returned.

Tag	Field Name	Req	Data Type	Description
45	RefSeqNum	N	SeqNum (9)	Sequence number of the message which caused the rejection.

Tag	Field Name	Req	Data Type	Description
372	RefMsgType	Y	String (2)	Message type of the rejected message.
379	BusinessRejectRefID	N	String (21)	The value of the business-level "ID" field on the message being referenced For example, SecurityReqID (320) of a Security Definition Request.
380	BusinessRejectReason	Y	Int	Code specifying the reason for the rejection of the message. Valid values: 0 = Other 1 = Unknown ID 2 = Unknown Security 3 = Unsupported Message Type 4 = Application not available 5 = Conditionally required field missing 8 = Throttle limit exceeded 9 = Throttle limit exceeded, session will be disconnected.
58	Text	N	String (50)	Where possible, message to explain reason for rejection.

4.8.9 News (B)

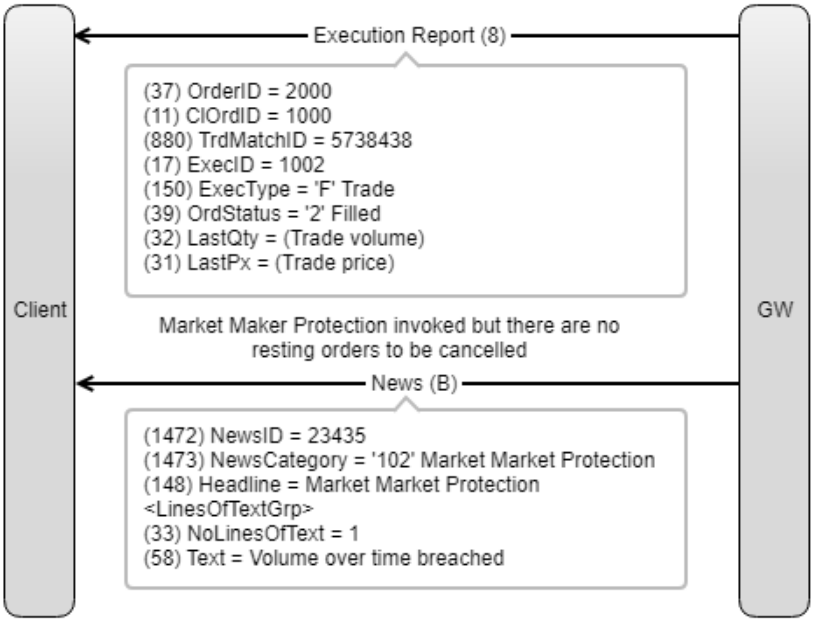
A News message (35=B) is a general free format message from the exchange.

Tag	Field Name	Req	Data Type	Description
1472	NewsID	Y*	String (20)	Unique identifier for News message.
1473	NewsCategory	Y*	Int	Category of News message. Valid values: 101 = Market message 102 = Market Maker Protection
42	OrigTime	Y*	UTCTimestamp	Time of message origination
148	Headline	Y	String (75)	Specifies the headline text either Market message or Market Maker Protection
Component Block <LinesOfTextGrp>				

Tag	Field Name	Req	Data Type	Description
33	NoLinesOfText	Y	NumInGrp (1)	Specifies the number of repeating lines of text specified. This value is always set to 1.
58	Text	Y	String (250)	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
End Component Block				

Example Message Flow

Market Maker Protection breached



4.9 Parties Component Block

Tag	Field Name	Req	Data Type	Description
453	NoPartyIDs	Y*	NumInGrp (2)	Number of parties specified.
>448	PartyID	Y*	String	Party identifier/code.

Tag	Field Name	Req	Data Type	Description
			See PartyRole Usage	Required if NoPartyIDs (453) > 0.
>447	PartyIDSource	Y*	Char	Source of the PartyID (448) value. Required if NoPartyIDs (453) > 0. Valid values: P = Client Short Code D = Proprietary/Custom E = ISO Country Code (i.e., two letter ISO country code) N = Legal Entity ID - LEI
>452	PartyRole	Y*	Int	Role of the specified PartyID (448). Required if NoPartyIDs (453) > 0. Valid values: 1 = Executing Firm 3 = Client ID 4 = Clearing Firm 7 = Entering Firm 11 = Order Origination Trader 24 = Customer Account 26 = Correspondent broker 35 = Liquidity provider 36 = Entering Trader 66 = Market Maker 81 = Broker Client ID 122 = Decision Maker 300 = Investment Decision Within Firm 301 = Execution Decision Within Firm 302 = Investment Decision Country Code 303 = Execution Decision Country Code 304 = Client Branch Country Code

4.9.1 PartyRole Usage

PartyRole (452) values used by LME are described below:

PartyRole		PartyID format	Description	Required
1	Executing Firm	Char (3)	Identifier of the executing firm. Required for MiFID as agent relationships are not captured in the LME participant structure.	Used for Transaction Reporting and Order Record Keeping
3	Client ID	Integer (8 bytes) for Client Short Code Alphanumeric (<=40) for Client ID or Custom	Client identifier. If that is not available, PartyID (448) should be populated with the value 0 = 'No Client'. Note: If PartyIDSource (447) set to 'N' Legal Entity ID = Client ID If PartyIDSource (447) set to 'P' Client Short Code If PartyIDSource (447) set to 'D' Proprietary/Custom = Alphanumeric Client ID is not validated by the system but should be set correctly on New Order Single requests. Required only for Client Orders i.e. AccountType (581) = 1, 8 or 101. Up to two instances of PartyRole (452) = '3' Client ID can be specified for the T4 booking model. For any contracts assigned to the T2 booking model, two instances of PartyRole (452) = '3' Client ID are mandatory and should include PartyIDSource (447) = 'P' Client Short Code and PartyIDSource (447) = 'D' Proprietary/Custom.	Mandatory for Client orders Used for Transaction Reporting and Order Record Keeping

PartyRole		PartyID format	Description	Required
4	Clearing Firm	Char (3)	Identifier of the clearing firm. A 3 character broker code (Member mnemonic). Cannot be entered in requests but is returned in Execution Reports for all fills.	Used for Transaction Reporting and Order Record Keeping
7	Entering Firm	Char (3)	Identifier of the entering firm. A 3 character broker code (Member mnemonic). Required for MiFID as agent relationships are not captured in the LME participant structure.	Used for Transaction Reporting and Order Record Keeping
11	Order Origination Trader	String (<=40)	Order Origination Trader (associated with Order Origination Firm e.g. trader who initiates/submits the order). Required as could be more than one individual under a FIX Comp ID. Required in New Order Single and will be returned in Execution Reports.	Mandatory for House and Client orders
24	Customer Account	Alphanumeric (<=60)	Identification of the Client Account Code where the AccountType (581) = 1, 8 or 101.	Mandatory for Client orders
26	Correspondent broker - Non-executing broker	Char (3)	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).	Used for Order Record Keeping
35	Liquidity Provider	String (<=40)	This should be submitted if the trader qualifies for a Liquidity Provider initiative.	Used for Order Record Keeping

PartyRole		PartyID format	Description	Required
			Not validated by the system but should be set correctly on New Order Single requests.	
36	Entering Trader	String (<=10)	Identifier of the trader entering the order. Cannot be entered in requests but will be returned in Execution Reports.	
66	Market Maker	String (<=40)	This should be submitted if the trader qualifies for a Liquidity Provider initiative. Not validated by the system but should be set correctly on New Order Single requests.	Optional
81	Broker Client ID	String (<=16)	Identifier of the entity in a risk group. Only valid if PartyIDSource (447) = 'D' Proprietary/Custom. Required in New Order Single and will be returned in Execution Reports.	Mandatory for Risk Management
122	Decision Maker	Integer (8 bytes)	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. Required only for Client Orders i.e. AccountType (581) = 1, 8 or 101.	Mandatory for Client orders Used for Transaction Reporting
300	Investment Decision Within Firm	Integer (8 bytes)	Short code to identify the individual who is responsible for the investment decision.	Used for Transaction Reporting and Order Record Keeping

PartyRole		PartyID format	Description	Required
301	Execution Decision Within Firm	Integer (8 bytes)	Short code to identify the execution decision maker with the firm. Required in New Order Single and will be returned in Execution Reports.	Mandatory for House and Client orders Used for Transaction Reporting and Order Record Keeping
302	Investment Decision Country Code	Char (2)	ISO Country Code of the branch responsible for the person making the investment decision.	Used for Transaction Reporting
303	Execution Decision Country Code	Char (2)	ISO Country Code of the branch responsible for the person making the execution decision.	Used for Transaction Reporting
304	Client Branch Country Code	Char (2)	ISO Country Code to identify the branch that received the client order or made an investment decision for a client. Required for Client Orders i.e., AccountType (581) = 1, 8 or 101	Mandatory for Client orders Used for Transaction Reporting

4.10 Application Messages

4.10.1 Security Definition Request (c)

Security Definition Request (35=c) is used to request the creation of either an Option strike or a strategy market. A Security Definition (35=d) will be sent in response to the request.

Tag	Field Name	Req	Data Type	Description
320	SecurityReqID	Y	String (18)	Unique ID of a Security Definition Request.
321	SecurityRequestType	Y	String (1)	Type of Security Definition Request. 1 = Request security identity for the specifications provided
Component Block <Instrument>				
207	SecurityExchange	Y*	Exchange (4)	Market which is used to identify the security: XLME
1227	ProductComplex	Y*	String (4)	Identifies an entire suite of products for a given market. Valid values: LME = Base LMEP = Precious
55	Symbol	Y*	String (20)	Symbol for the LME contract code e.g. CADF (Copper Future) or OCDF (Copper Monthly Average Future).
167	SecurityType	Y*	String (4)	Indicates the type of security whether outright or strategy e.g. MLEG for strategy. Valid values: OPT = Option MLEG = Multileg instrument
762	SecuritySubType	Y*	Int	Indicates the security sub type. Valid values: 0 = Outright 1 = Carry 2 = Custom 3 = 3 Month Average

Tag	Field Name	Req	Data Type	Description
				4 = 6 Month Average 5 = 12 Month Average 6 = Carry Average 7 = Call Spread 8 = Put Spread
541	MaturityDate	C*	LocalMktDate	Expiration date for options (YYYYMMDD) Conditionally required if SecurityType (167) = 'OPT' Option. Not required if SecurityType (167) = MLEG.
202	StrikePrice	C*	Price (20)	Strike Price for an Option. Conditionally required if SecurityType (167) = 'OPT' Option.
201	PutOrCall	C*	Int	Used to express option right Valid values: 0 = Put 1 = Call Conditionally required if SecurityType (167) = 'OPT' Option.
Component Block <InstrmtLegGrp>				
555	NoLegs	C*	NumInGrp (1)	Conditionally required if SecurityType (167) = 'MLEG' Number of InstrumentLeg repeating group instances. Cannot be more than 4 or less than 2. Note this will only be 1 for a 3 Month Average, 6 Month Average and 12 Month Average.
Component Block <InstrumentLeg>				
>602	LegSecurityID	C*	Int	SecurityID of the leg derived from the SecurityID (48) of the outright. Conditionally required if SecurityType (167) = MLEG.

Tag	Field Name	Req	Data Type	Description
				For an Average strategy only the LegSecurityID of the first leg of the strategy is provided as the other months are consecutive.
>603	LegSecurityIDSource	C*	String (1)	Identifies the source of the LegSecurityID value. Valid value: 8 = Exchange defined. Conditionally required when LegSecurityID (602) is specified.
>623	LegRatioQty	C*	Int	The ratio of quantity for this individual leg relative to the entire multileg security. For example, for a custom strategy such as a Butterfly would be 1:2:1, for the first leg LegRatioQty = 1 (buy near contract month), second leg LegRatioQty = 2 (sell two contracts in far month) and third leg LegRatioQty = 1 (buy one contract in yet farther month). Conditionally required if SecurityType (167) = MLEG.
>624	LegSide	C*	Char	The side of this individual leg. Valid values: 1 = Buy 2 = Sell. Conditionally required if SecurityType (167) = MLEG.
End Component Blocks				

4.10.2 Security Definition (d)

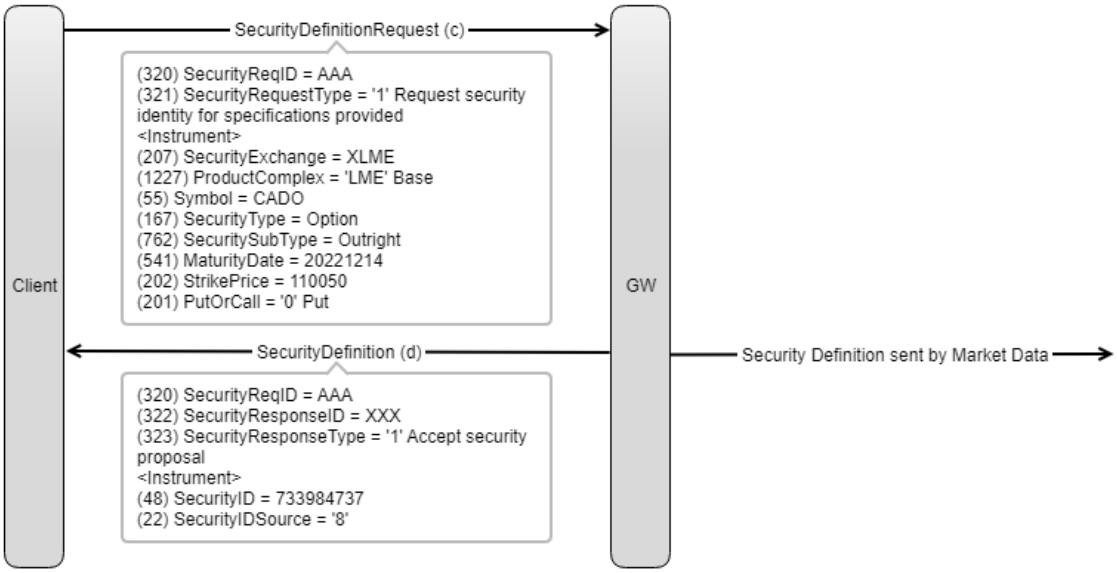
Security Definition (35=d) will be returned to the originator of the Security Definition Request (35=c) to accept, accept with revisions or reject the creation of a tradable instrument. Market participants will be notified of a newly created instrument by the Market Data service.

Tag	Field Name	Req	Data Type	Description
320	SecurityReqID	Y*	String (18)	Client generated ID supplied on the Security Definition Request.
322	SecurityResponseID	Y*	String (21)	Unique ID of a Security Definition Request (c) message.
323	SecurityResponseType	Y*	Int	Type of Security Definition message response. Valid values: 1 = Accept security proposal 2 = Accept security proposal with revisions as indicated in the message 5 = Reject security proposal
1607	SecurityRejectReason	C	Int	Conditionally required to specify a rejection reason when SecurityResponseType (323) = ‘5’ Reject security proposal. Valid values: 1 = Invalid instrument requested 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)
Component Block <Instrument>				
48	SecurityID	C*	Int	Tradable instrument identifier Conditionally required if SecurityResponseType (323) = '1' Accept security proposal or ‘2’ Accept security proposal with revisions as indicated in the message
22	SecurityIDSource	C*	String (1)	Identifies the source of the SecurityID (48): 8 = Exchange Symbol

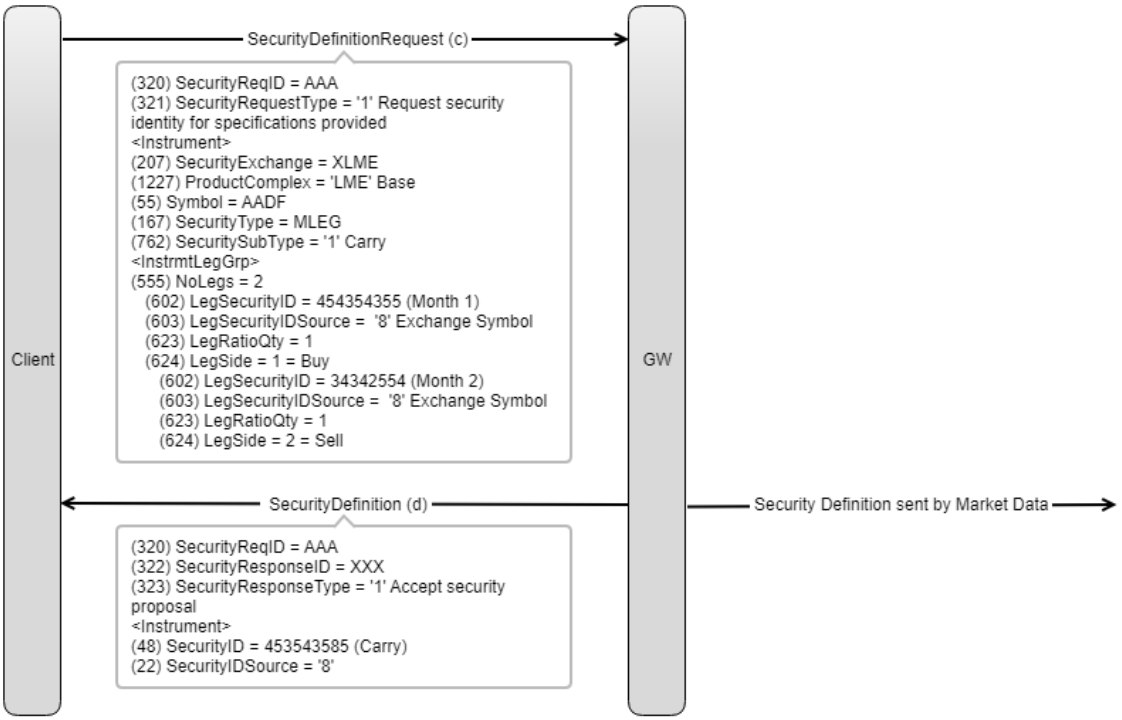
Tag	Field Name	Req	Data Type	Description
				Conditionally required when SecurityID (48) is specified.
End Component Block				
58	Text	C*	String (50)	Identifies the reason for rejection. Conditionally required if SecurityRejectReason (1607) = '99' Other

Example Message Flows

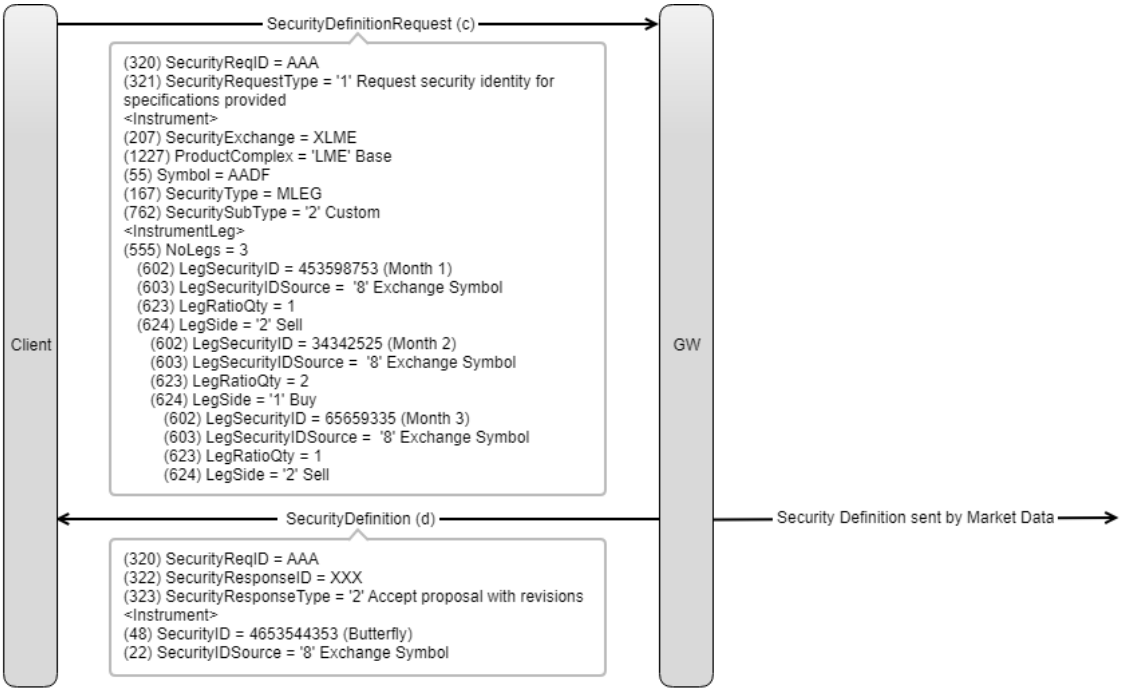
Option Strike Request



Futures Strategy Request



Inverse Custom Strategy Request



4.10.3 New Order Single (D)

New Order Single (35=D) is used to submit a new order for execution. An Execution Report (35=8), Reject (35=3) or Business Message Reject (35=j) is sent in response.

Tag	Field Name	Req	Data Type	Description
11	ClOrdID	Y	String (18)	Unique identifier set by the entering firm.
Component Block <Parties>		Y*	See Parties Component Block The following PartyRole (452) values are mandatory: '11' Order Origination Trade '81' Broker Client ID '301' Execution Decision Within Firm	
581	AccountType	Y*	Int	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
18	ExecInst	N	MultipleCharValue	Instructions for order handling. If more than one instruction is applicable to an order, this field can contain multiple instructions separated by space. Valid values: 6 = Participate but don't initiate (required for Post only order submission) o = Cancel on connection loss
Component Block <DisplayInstruction>				
1138	DisplayQty	C*	Qty	Visible quantity. Conditionally required for Iceberg orders. If present, must be < OrderQty (38)
End Component Block				

Tag	Field Name	Req	Data Type	Description
Component Block <Instrument>				
48	SecurityID	Y*	Int	Tradable instrument identifier
22	SecurityIDSource	C*	String (1)	Identifies the source of the SecurityID (48): 8 = Exchange Symbol Conditionally required when SecurityID (48) is specified.
End Component Block				
54	Side	Y	Char	Side of the order Valid values: 1 = Buy 2 = Sell
60	TransactTime	Y	UTCTimestamp	Timestamp when the message was generated.
Component Block <OrderQtyData>				
38	OrderQty	Y	Qty	Total order quantity of the order.
End Component Block				
40	OrdType	Y	Char	Order type applicable to the order. Valid values: 2 = Limit 3 = Stop / Stop Loss 4 = Stop Limit K = Market to Limit
44	Price	C	Price (20)	Order price. Conditionally required if OrdType (40) is 2 = Limit or 4 = Stop Limit For strategy orders a price is described as: Contango – represented by a negative price.

Tag	Field Name	Req	Data Type	Description
				Backwardation - represented by a positive price.
99	StopPx	C	Price (20)	The Stop loss trigger price. Conditionally required if OrdType (40): 3 = Stop/ Stop Loss 4 = Stop Limit TriggerPriceType (1107) is required if a Stop Price is specified.
Component Block <TriggeringInstruction>				
1100	TriggerType	C	Char	Trigger prompt for stop order elements. Conditionally required if any other Triggering tags are specified. Valid value: 4 = Price Movement
1102	TriggerPrice	C*	Price (20)	Stop order price of the OCO. Conditionally required for an OCO.
1107	TriggerPriceType	C*	Char	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 StopPx (99) or TriggerPrice (1102) is specified
1110	TriggerNewPrice	C*	Price (20)	Limit order price of the stop once triggered. Conditionally required if Trigger Order Type (1111) = '2' Limit
1111	TriggerOrderType	C*	Char	Order type of the stop once triggered. Valid values: K = Market to Limit

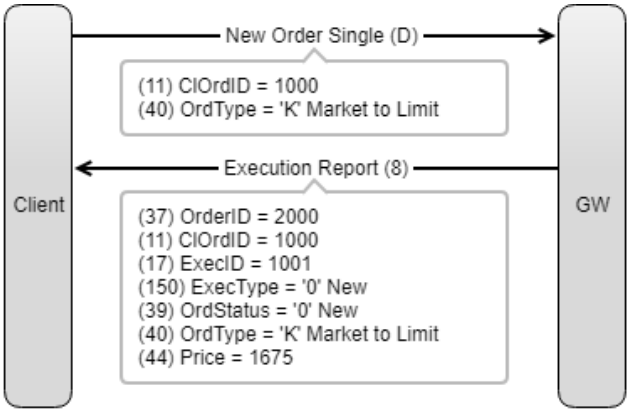
Tag	Field Name	Req	Data Type	Description
				2 = Limit Conditionally required for an OCO.
End Component Block				
59	TimeInForce	N	Char	Specifies how long the order remains in effect. Valid values: 0 = Day 1 = Good 'til Cancel 3 = Immediate or Cancel 4 = Fill or Kill 6 = Good 'til Date Absence of this field indicates Day.
432	ExpireDate	C	LocalMktDate	Conditionally required if TimeInForce (59) = Good 'til Date. Format is YYYYMMDD.
528	OrderCapacity	Y*	Char	Indicates the trading capacity. Valid values: A (agency) = AOTC P (principal) = DEAL R (riskless principal) = MTCH
529	OrderRestrictions	Y*	Char	Restrictions associated with an order. Valid values: D = Non-algorithmic (human) E = Algorithmic (algo)
58	Text	N	String (50)	Free text.
1724	OrderOrigination	N	Char	Identifies the origin of the order. Valid value: 5 = Order received from a direct access or sponsored access (the trader has direct electronic access – DEA). Absence of this field indicates DEA = false

Tag	Field Name	Req	Data Type	Description
2362	SelfMatchPreventionID	N	Int (9)	Identifies an order that should not be matched to an opposite order if both buy and sell orders for the trade contain the same SelfMatchPreventionID (2362) and submitted by the same member.
Component Block <OrderAttributeGrp>				
2593	NoOrderAttributes	N	NumInGrp (1)	Number of order attribute entries.
2594	OrderAttributeType	C*	Int	The type of order attribute. Conditionally required if NoOrderAttributes (2593) > 0. Valid values: 0 = Aggregated order. In the context of ESMA RTS 24 Article 2(3), when OrderAttributeValue (2595) = Y, it signifies that the order consists of several orders aggregated together. This maps to ESMA RTS value "AGGR". Not valid if PartyRole (452) = ‘3’ Client ID, PartyIDSource (447) = 'P' Client Short Code and PartyID (448) = ‘0’ No Client. 1 = Pending allocation. In the context of ESMA RTS 24 Article 2(2), when OrderAttributeValue (2595) = Y, 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". Not valid if PartyRole (452) = ‘3’ Client ID, PartyIDSource (447) = 'P'

Tag	Field Name	Req	Data Type	Description
				Client Short Code and PartyID (448) = '0' No Client. 2 = Liquidity Provision Order. In the context of ESMA RTS 24 Article 3, when OrderAttributeValue (2595) = Y, 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). 3 = Risk Reduction Order. In the context of ESMA RTS 22 Article 4(2)(i), when OrderAttributeValue (2595) = Y, 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".
2595	OrderAttributeValue	C*	String (1)	The value associated with the order attribute type specified in OrderAttributeType (2594). Conditionally required if NoOrderAttributes (2593) > 0. Valid value: Y = Yes
End Component Block				

Example Message Flow

Market to Limit order



4.10.4 Order Cancel Replace Request (G)

Order Cancel Replace Request (35=G) is used to change the parameters of an existing order. If successful an Execution Report (35=8) is returned to confirm replacement of the order otherwise an Order Cancel Reject (35=9) is returned if the request is rejected and the order remains unchanged.

The following tags will not be available on an Order Cancel Replace Request and will therefore retain the value supplied on order entry:

- AccountType (581)
- SelfMatchPreventionID (2362)
- TriggerPriceType (1100)
- TriggerOrder (1110).

Similarly the Party details that cannot be amended will also not be present and therefore remain unchanged.

Tag	Field Name	Req	Data Type	Description
37	OrderID	N	String (19)	A unique order identifier set by the trading system. This identifier is not changed by cancel/replace messages; it will remain the same for all chain of orders.
Component Block <Parties>		Y*	See Parties Component Block The following PartyRole (452) values can be modified: 300 = Investment Decision Within Firm 301 = Execution Decision Within Firm 302 = Investment Decision Country Code 303 = Execution Decision Country Code	

Tag	Field Name	Req	Data Type	Description
			304 = Client Branch Country Code	
41	OrigClOrdID	Y	String (18)	Original order identified as the order to be modified. It is the ID of the latest non-rejected order (not the initial order of the day).
11	ClOrdID	Y	String (18)	Unique identifier set by the entering firm.
18	ExecInst	C*	MultipleCharValue	Instructions for order handling. If more than one instruction is applicable to an order, this field can contain multiple instructions separated by space. Valid value: 6 = Participate but don't initiate (required for Post only order submission) o = Cancel on connection loss Conditionally required if specified on the original order and must match the previous submission.
Component Block <DisplayInstruction>				
1138	DisplayQty	C*	Qty	Visible quantity for an Iceberg order. Conditionally required if specified on the original order.
End Component Block				
Component Block <Instrument>				
48	SecurityID	Y*	Int	Tradable instrument identifier
22	SecurityIDSource	C*	String (1)	Identifies the source of the SecurityID (48): 8 = Exchange Symbol Conditionally required when SecurityID (48) is specified.
End Component Block				

Tag	Field Name	Req	Data Type	Description
54	Side	Y	Char	Must be the same value as original order. Valid values: 1 = Buy 2 = Sell
60	TransactTime	Y	UTCTimestamp	Timestamp when the message was generated.
Component Block <OrderQtyData>				
38	OrderQty	Y	Qty	New order quantity. Note: this is not the LeavesQty (151) but the new total quantity of the order.
End Component Block				
40	OrdType	Y	Char	The order type cannot change. Valid values: 2 = Limit 3 = Stop / Stop Loss 4 = Stop Limit K = Market to Limit
44	Price	C	Price (20)	For an order in a carry contract, a “contango” price is expressed as a negative price and a “backwardation” price as positive. Conditionally required for all Limit order types.
99	StopPx	C	Price (20)	The Stop loss trigger price. Conditionally required if OrdType (40): 3 = Stop/ Stop Loss 4 = Stop Limit.
Component Block <TriggeringInstruction>				
1100	TriggerType	C*	Char	Trigger prompt for stop order elements. Conditionally required if OCO Triggering tags are specified. Not required if StopPx (99) is specified.

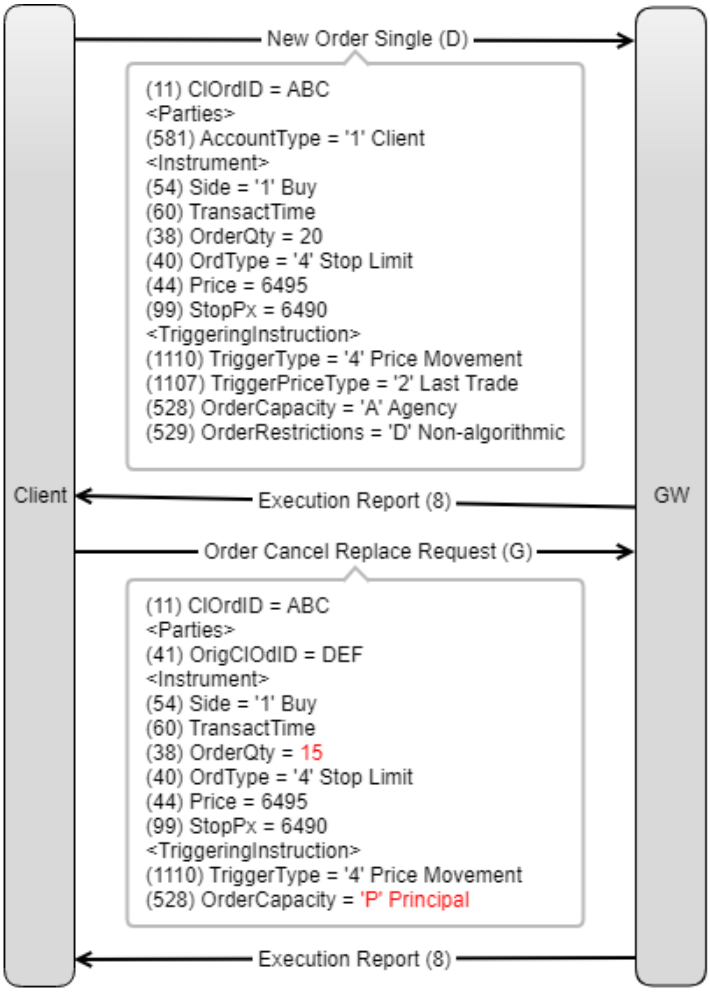
Tag	Field Name	Req	Data Type	Description
				Valid value: 4 = Price Movement
1102	TriggerPrice	C*	Price (20)	Stop order price of the OCO. Conditionally required if specified on the original order.
1110	TriggerNewPrice	C*	Price (20)	Limit order price of the stop once triggered. Conditionally required if specified on the original order.
End Component Block				
59	TimeInForce	C*	Char	Specifies how long the order remains in effect. Conditionally required if specified on the original order and must match with the previous submission. Valid values: 0 = Day 1 = Good 'til Cancel 3 = Immediate or Cancel 4 = Fill or Kill 6 = Good 'til Date Absence of this field indicates Day.
432	ExpireDate	C	LocalMktDate	Conditionally required if TimeInForce (59) = Good 'til Date is specified. Format is YYYYMMDD.
528	OrderCapacity	Y*	Char	Indicates the trading capacity. Valid values: A (agency) = AOTC P (principal) = DEAL R (riskless principal) = MTCH
529	OrderRestrictions	Y*	Char	Restrictions associated with an order. Valid values: D = Non-algorithmic (human) E = Algorithmic (algo)
58	Text	N	String (50)	Free text. Can be amended if supplied.

Tag	Field Name	Req	Data Type	Description
				If the field is not specified on the original order and is added this indicates an amendment. Absence of the field indicates that previously entered free text has been removed.
1724	OrderOrigination	N	Char	Identifies the origin of the order. Valid value: 5 = Order received from a direct access or sponsored access (the trader has direct electronic access – DEA) Absence of this field indicates DEA = false
Component Block <OrderAttributeGrp> An OrderAttributeType (2594) e.g. 3 = Risk Reduction Order specified on the original order and present on the amendment indicates that the attribute is unchanged. Absence of the attribute indicates that it has been removed. If an attribute is not specified on the original order and is added this indicates an amendment.				
2593	NoOrderAttributes	N	NumInGrp (1)	Number of order attribute entries.
2594	OrderAttributeType	C*	Char	The type of order attribute. Conditionally required if NoOrderAttributes (2593) > 0. Valid values: 0 = Aggregated order. In the context of ESMA RTS 24 Article 2(3), when OrderAttributeValue (2595) = Y, it signifies that the order consists of several orders aggregated together. This maps to ESMA RTS value "AGGR". Not valid if PartyRole (452) = ‘3’ Client ID, PartyIDSource (447) = 'P' Client Short Code and PartyID (448) = ‘0’ No Client. 1 = Pending allocation. In the context of ESMA RTS 24 Article 2(2), when OrderAttributeValue (2595) = Y, it

Tag	Field Name	Req	Data Type	Description
				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". Not valid if PartyRole (452) = ‘3’ Client ID, PartyIDSource (447) = 'P' Client Short Code and PartyID (448) = ‘0’ No Client. 2 = Liquidity Provision Order. In the context of ESMA RTS 24 Article 3, when OrderAttributeValue (2595) = Y, 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). 3 = Risk Reduction Order. In the context of ESMA RTS 22 Article 4(2)(i), when OrderAttributeValue (2595) = Y, 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".
2595	OrderAttributeValue	C*	String (1)	The value associated with the order attribute type specified in OrderAttributeType (2594). Conditionally required if NoOrderAttributes (2593) > 0. Valid value: Y = Yes
End Component Block				

Example Message Flow

Revise Order



4.10.5 Order Cancel Request (F)

Order Cancel Request (35=F) is used to cancel the remaining quantity of an existing order. An Execution Report (35=8) is returned to confirm cancellation or an Order Cancel Reject (35=9) if the cancel is rejected.

Tag	Field Name	Req	Data Type	Description
41	OrigClOrdID	Y	String (18)	Order identifier for the order to cancel. It is the ID of the latest non-rejected order (not the initial order of the day).
37	OrderID	N	String (19)	A unique order identifier set by the trading system. This identifier is not changed by

Tag	Field Name	Req	Data Type	Description
				cancel/replace messages; it will remain the same for all chain of orders.
11	ClOrdID	Y	String (18)	Unique identifier set by the entering firm.
Component Block <Instrument>				
48	SecurityID	Y*	Int	Tradable instrument identifier
22	SecurityIDSource	C*	String (1)	Identifies the source of the SecurityID (48): 8 = Exchange Symbol Conditionally required when SecurityID (48) is specified.
End Component Block				
54	Side	Y	Char	Side of the order. Valid values: 1 = Buy 2 = Sell
60	TransactTime	Y	UTCTimestamp	Timestamp when the message was generated.

4.10.6 Order Cancel Reject (9)

Order Cancel Reject (35=9) is returned in response to Order Cancel/Replace Request (35=G) or Order Cancel Request (35=F) that cannot be honoured. The unchanged order will remain in the order book if it has not already been cancelled or has expired.

Tag	Field Name	Req	Data Type	Description
37	OrderID	Y	String (19)	A unique order identifier set by the trading system. This identifier is not changed by cancel/replace messages; it will remain the same for all chain of orders. Set to NONE when OrdStatus (39) = Rejected and CxlRejReason (102) = '1' Unknown order.
11	ClOrdID	Y	String (18)	Unique identifier set by the entering firm.
41	OrigClOrdID	Y	String (18)	Original order identified for the order to modify. It is the ID of the latest non-

Tag	Field Name	Req	Data Type	Description
				rejected order (not the initial order of the day)
39	OrdStatus	Y	Char	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 C = Expired E = Pending Replace
60	TransactTime	Y	UTCTimestamp	Timestamp when the message was generated.
434	CxlRejResponseTo	Y	Char	Identifies the type of request that an Order Cancel Reject (9) is in response to. Valid values: 1 = Order Cancel Request (F) 2 = Order Cancel/Replace Request (G)
102	CxlRejReason	Y*	Int	Code that identifies the reason for the rejection of the cancellation. Valid values: 0 = Too late to cancel 1 = Unknown order 3 = Order already in Pending Cancel or Pending Replace Status 6 = Duplicate ClOrderID (11) received 18 = Invalid price increment 99 = Other
1328	RejectText	C*	String (75)	Conditionally required if CxlRejReason (102) = '99' Other. Reason description for rejecting the transaction.

4.10.7 Execution Report (8)

Execution Report (35=8) is used to:

- confirm the receipt of an order
- confirm changes to an existing order (i.e. accept cancel and replace requests)
- confirm or convey an order cancellation or expiration
- convey trade cancellation by Market Operations
- convey fill information on working orders
- reject orders
- convey information about restated long orders carried from one trading day to the next.

ExecType (150) identifies the purpose of the execution report message and OrdStatus (39) 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.

Tag	Field Name	Req	Data Type	Description
37	OrderID	Y	String (19)	A unique order identifier set by the trading system. This identifier is not changed by cancel/replace messages; it will remain the same for all chain of orders.
11	ClOrdID	Y*	String (18)	The identifier entered by the market participant in the message that caused this Execution Report.
41	OrigClOrdID	C	String (18)	ClOrdID (11) 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 Execution Report Matrix.
Component Block <Parties>		Y*	See Parties Component Block	
880	TrdMatchID	C*	String (19)	Identifier assigned to a trade by the matching engine for the trade half.

Tag	Field Name	Req	Data Type	Description
				Conditionally required if ExecType (150) = ‘F’ Trade.
17	ExecID	Y	String (19)	Unique identifier of execution message.
19	ExecRefID	C*	String (19)	Reference identifier used with Trade Cancel execution type. Conditionally required if ExecType (150) = ‘H’ Trade Cancel.
150	ExecType	Y	Char	Describes the specific Execution Report. Valid values: 0 = New 3 = Done 4 = Cancelled 5 = Replaced 8 = Rejected C = Expired D = Restated E = Pending Replace F = Trade H = Trade Cancel L = Triggered or Activated by the System
39	OrdStatus	Y	Char	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 C = Expired E = Pending Replace
103	OrdRejReason	C*	Int	Conditionally required if ExecType (150) = ‘8’ Rejected Valid values: 1 = Unknown Symbol

Tag	Field Name	Req	Data Type	Description
				2 = Exchange Closed 6 = Duplicate Order 15 = Unknown Account(s) 18 = Invalid price increment 99 = Other
378	ExecRestatementReason	C*	Int	Conditionally required if ExecType (150) = 'D' Restated. The reason for restatement. Valid values: 1 = GT renewal / restatement 99 = Other. See ExecTypeReason (2431) for speed bump handling.
581	AccountType	Y*	Int	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
1115	OrderCategory	C*	Char	Conditionally required for a trade from an implied order when OrderStatus (39) = '1' Partially Filled or '2' Filled Defines the type of interest behind a trade (fill or partial fill). Valid value: 7 = Implied Order
Component Block <Instrument>				
48	SecurityID	Y*	Int	Tradable instrument identifier
22	SecurityIDSource	C*	String (1)	Identifies the source of the SecurityID (48): 8 = Exchange Symbol Conditionally required when SecurityID (48) is specified.

Tag	Field Name	Req	Data Type	Description
End Component Block				
54	Side	Y	Char	Side of the order Valid values: 1 = Buy 2 = Sell
Component Block <OrderQtyData>				
38	OrderQty	Y*	Qty	Total order quantity of the order.
End Component Block				
40	OrdType	Y*	Char	Order type applicable to the order. Valid values: 2 = Limit 3 = Stop / Stop Loss 4 = Stop Limit K = Market to Limit
44	Price	C	Price (20)	The order price. Conditionally required if OrdType (40) = '2' Limit or '4' Stop Limit. For strategy orders, a "contango" price is expressed as a negative price and a "backwardation" price as a positive price.
99	StopPx	C*	Price (20)	The Stop loss trigger price. Conditionally required if OrdType (40) = '3' Stop / Stop Loss or '4' Stop Limit. TriggerPriceType (1107) is required if StopPx is specified.
Component Block <TriggeringInstruction>				
1100	TriggerType	C*	Char	Trigger prompt for the stop order elements. Conditionally required if any other Triggering tags are specified. Valid value:

Tag	Field Name	Req	Data Type	Description
				4 = Price Movement
1102	TriggerPrice	C*	Price (20)	Stop order price of the OCO. Conditionally required for an OCO.
1107	TriggerPriceType	C*	Char	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 StopPx (99) or TriggerPrice (1102) is specified
1110	TriggerNewPrice	C*	Price (20)	Limit order price of the stop once triggered. Conditionally required if TriggerOrderType (1111) = '2' Limit
1111	TriggerOrderType	C*	Char	Order type of the order once triggered. K = Market to Limit 2 = Limit Conditionally required for an OCO.
End Component Block				
59	TimeInForce	Y*	Char	Specifies how long the order remains in effect. Valid values: 0 = Day 1 = Good 'til cancel (GTC) 3 = Immediate or cancel (IOC) 4 = Fill or Kill 6 = Good 'til Date (GTD)
432	ExpireDate	C	LocalMktDate	Conditionally required if TimeInForce (59) = '6' Good 'til Date is not specified.

Tag	Field Name	Req	Data Type	Description
				Format is YYYYMMDD.
18	ExecInst	C*	MultipleCharValue	Instructions for order handling. If more than one instruction is applicable to an order, this field can contain multiple instructions separated by space. Valid values: 6 = Participate but don't initiate for Post Only orders o = Cancel on connection loss Conditionally required according to Execution Report Matrix.
1057	AggressorIndicator	C*	Boolean	Indicates if a matching order is an aggressor or not in the trade. Y = Aggressor N = Passive Conditionally required if ExecType (150) = 'F' Trade.
528	OrderCapacity	Y*	Char	Designates the capacity of the firm placing the order. Valid values: A (agency) = AOTC P (principal) = DEAL R (riskless principal) = MTCH
529	OrderRestrictions	Y*	MultipleCharValue	Indicates if the order is entered either by an algo trader or a human. Valid values: D = Non-algorithmic (human) E = Algorithmic (algo)
32	LastQty	C	Qty	Conditionally required if ExecType (150) = 'F' Trade. The total volume of this trade.
31	LastPx	C	Price (20)	Conditionally required if ExecType (150) = 'F' Trade.

Tag	Field Name	Req	Data Type	Description
				The price of this trade.
151	LeavesQty	Y	Qty	The quantity open for further execution. If OrdStatus (39) = '4' Cancelled, 'C' Expired or '8' Rejected then LeavesQty (151) could be 0 otherwise LeavesQty (151) will be OrderQty (38) - CumQty (14)
14	CumQty	Y	Qty	The quantity of the order that has been executed so far.
60	TransactTime	Y*	UTCTimestamp	Timestamp when the message was generated.
Component Block <DisplayInstruction>				
1138	DisplayQty	C*	Qty	Visible quantity for Iceberg orders. Conditionally required for an Iceberg order.
End Component Block				
58	Text	C*	String (50)	Contains the value supplied in this field on the order. Conditionally required according to Execution Report Matrix.
Component Block <InstrmtLegExecGrp>				
555	NoLegs	C*	NumInGrp (2)	Conditionally required if ExecType (150) = ‘F’ Trade on a multileg tradable instrument. Number of InstrumentLeg repeating group instances.
Component Block <InstrumentLeg> - Required if NoLegs (555) > 0.				
>602	LegSecurityID	C*	Int	Conditionally required if ExecType (150) = ‘F’ Trade on a multileg tradable instrument.

Tag	Field Name	Req	Data Type	Description
				Multileg tradable instrument's individual SecurityID.
>603	LegSecurityIDSource	C*	String (1)	Identifies the source of the SecurityID (48): 8 = Exchange Symbol Conditionally required when LegSecurityID (602) is specified.
>624	LegSide	C*	Char	Conditionally required if ExecType (150) = 'F' Trade on a multileg tradable instrument. The side of this individual leg (multileg security). Valid values: 1 = Buy 2 = Sell
End Component Block				
1366	LegAllocID	C*	String (19)	Strategy leg trade identifier. Conditionally required if ExecType (150) = 'F' Trade on a multileg tradable instrument.
637	LegLastPx	C*	Price (20)	Conditionally required if ExecType (150) = 'F' Trade on a multileg tradable instrument. Execution price assigned to the leg of the multileg tradable instrument.
1418	LegLastQty	C*	Qty	Conditionally required if ExecType (150) = 'F' Trade on a multileg tradable instrument. Fill quantity for the instrument leg.
End Component Block				
1328	RejectText	C*	String (75)	Identifies the reason for rejection. Conditionally required if ExecTypeReason (2431) = '4'

Tag	Field Name	Req	Data Type	Description
				Unsolicited order cancellation or OrdRejReason (103) = '99' Other.
1724	OrderOrigination	C*	Int	Origin of the order Valid value: 5 = Order received from a direct access or sponsored access (the trader has direct electronic access – DEA) Absence of this field indicates DEA = false. Conditionally required according to Execution Report Matrix.
2431	ExecTypeReason	C*	Int	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 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
2362	SelfMatchPreventionID	C*	Int (9)	Identifies an order that should not be matched to an opposite order if both buy and sell orders for the

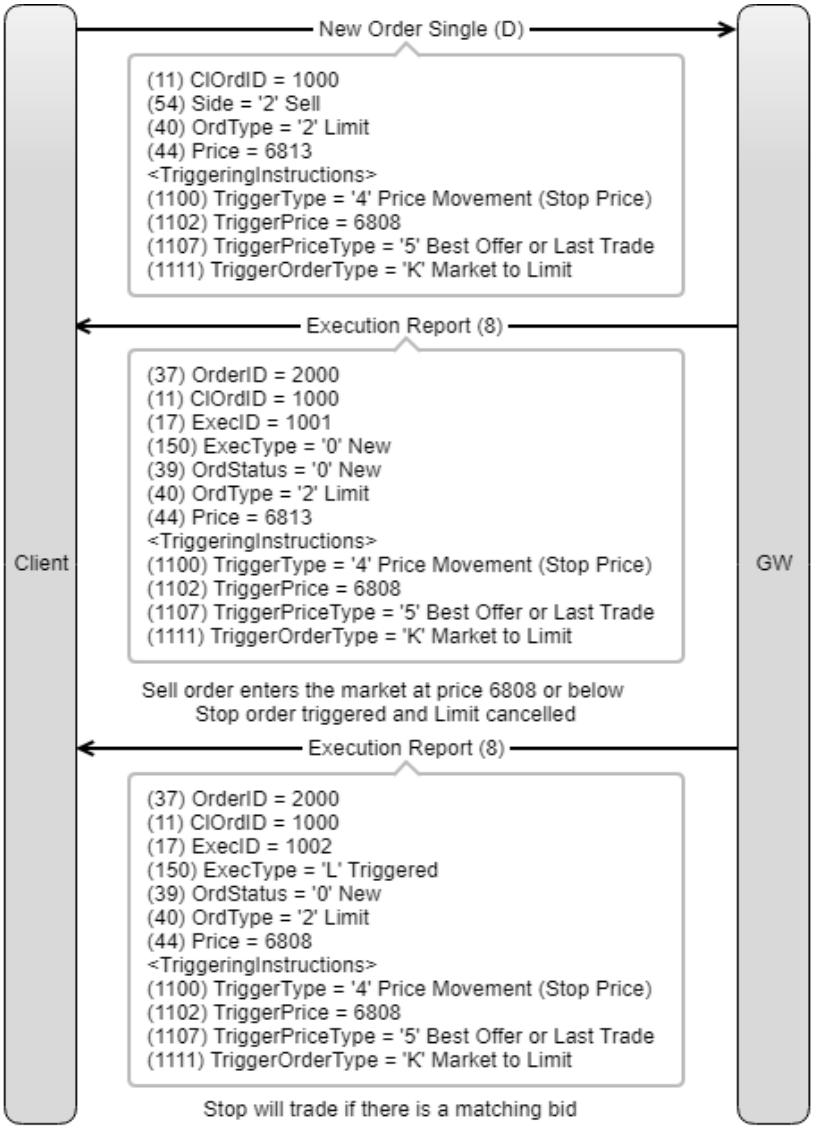
Tag	Field Name	Req	Data Type	Description
				trade contain the same SelfMatchPreventionID (2362) and submitted by the same member. Conditionally required according to Execution Report Matrix.
Component Block <OrderAttributeGrp> - Conditionally required if specified.				
2593	NoOrderAttributes	C*	NumInGrp (1)	Number of order attribute entries.
2594	OrderAttributeType	C*	Int	The type of order attribute. Conditionally required if NoOrderAttributes (2593) > 0. Valid values: 0 = Aggregated order. In the context of ESMA RTS 24 Article 2(3), when OrderAttributeValue (2595) = Y, it signifies that the order consists of several orders aggregated together. This maps to ESMA RTS value "AGGR". 1 = Pending allocation. In the context of ESMA RTS 24 Article 2(2), when OrderAttributeValue (2595) = Y, 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". 2 = Liquidity Provision Order. In the context of ESMA RTS 24 Article 3, when OrderAttributeValue (2595) = Y, 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

Tag	Field Name	Req	Data Type	Description
				part of another activity in accordance with Article 3" (of RTS 24). 3 = Risk Reduction Order. In the context of ESMA RTS 22 Article 4(2)(i), when OrderAttributeValue (2595) = Y, 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".
2595	OrderAttributeValue	C*	String (1)	The value associated with the order attribute type specified in OrderAttributeType (2594). Conditionally required if NoOrderAttributes (2593) > 0. Valid value: Y = Yes
End Component Block				

Example Message Flows

OCO submitted, Stop triggered and Limit cancelled

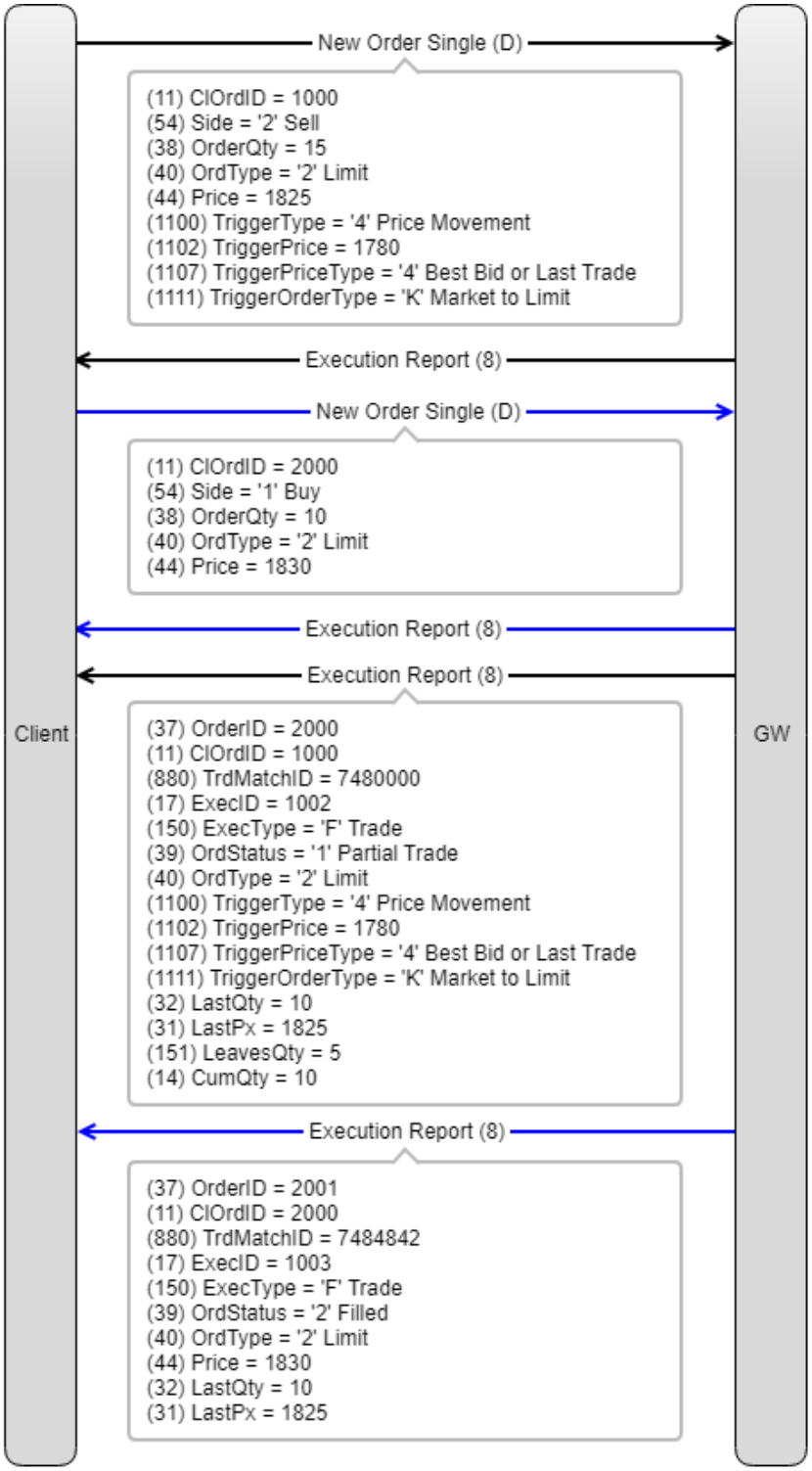
An OCO order is submitted as a Limit offer with a Market Stop trigger price of 6808, an incoming offer triggers the Stop order and cancels Limit element of the OCO. An Execution Report is not sent for cancellation. The triggered Market Stop is converted to a Limit order at a trigger new price of 6808.



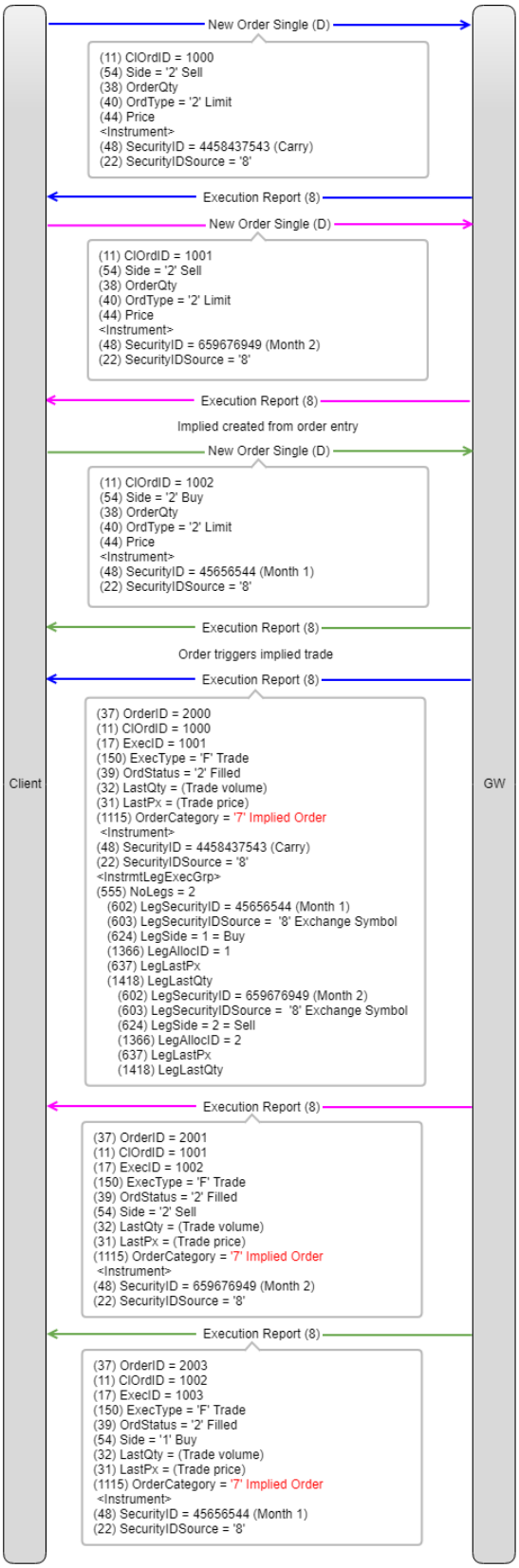
OCO Partial Trade

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

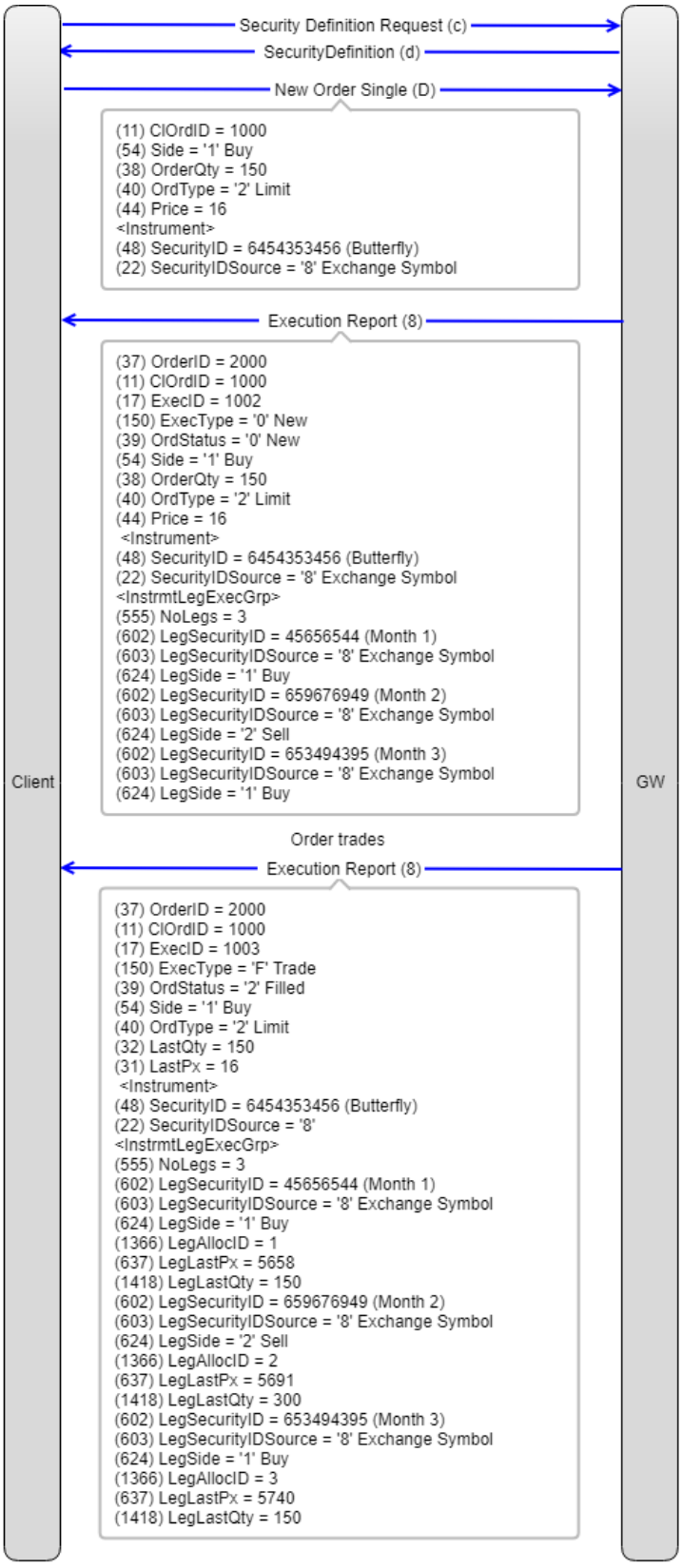
A Limit bid is submitted at 1830 for 10 lots. The OCO order is not triggered but trades 10 lots with the incoming order. The OCO remains in the order book with a residual volume of 5 lots


Client
GW
New Order Single (D)
(11) ClOrdID = 1000
(54) Side = '2' Sell
(38) OrderQty = 15
(40) OrdType = '2' Limit
(44) Price = 1825
(1100) TriggerType = '4' Price Movement
(1102) TriggerPrice = 1780
(1107) TriggerPriceType = '4' Best Bid or Last Trade
(1111) TriggerOrderType = 'K' Market to Limit
Execution Report (8)
New Order Single (D)
(11) ClOrdID = 2000
(54) Side = '1' Buy
(38) OrderQty = 10
(40) OrdType = '2' Limit
(44) Price = 1830
Execution Report (8)
Execution Report (8)
(37) OrderID = 2000
(11) ClOrdID = 1000
(880) TrdMatchID = 7480000
(17) ExecID = 1002
(150) ExecType = 'F' Trade
(39) OrdStatus = '1' Partial Trade
(40) OrdType = '2' Limit
(1100) TriggerType = '4' Price Movement
(1102) TriggerPrice = 1780
(1107) TriggerPriceType = '4' Best Bid or Last Trade
(1111) TriggerOrderType = 'K' Market to Limit
(32) LastQty = 10
(31) LastPx = 1825
(151) LeavesQty = 5
(14) CumQty = 10
Execution Report (8)
(37) OrderID = 2001
(11) ClOrdID = 2000
(880) TrdMatchID = 7484842
(17) ExecID = 1003
(150) ExecType = 'F' Trade
(39) OrdStatus = '2' Filled
(40) OrdType = '2' Limit
(44) Price = 1830
(32) LastQty = 10
(31) LastPx = 1825

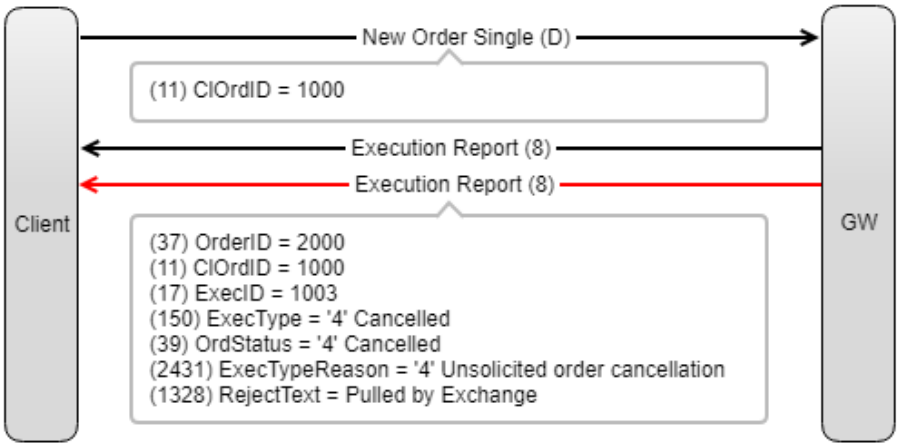
Implied trade



Custom strategy Butterfly trades



Order cancellation by Exchange



4.10.7.1 Execution Report Matrix

An Execution Report can be returned in response to a request e.g. New Order Single (35=D) or unsolicited in response to a particular action.

The tags 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 in returned response to a particular action.

Legend:

M = Mandatory

C = Conditional

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

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

Tag	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
OrderID (37)	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
ClOrdID (11)	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
OrigClOrdID (41)					M	M	M	M							
<Parties> PartyRole (452)															
1 = Executing Firm	P	P										P	P	P	
3 = Client ID	P	P										P	P	P	

Tag	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
4 = Clearing Firm												M	M	M	
7 = Entering Firm	P	P										P	P	P	
11 = Order Origination Trader	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
24 = Customer Account	P	P										P	P	P	
26 = Correspondent broker	P	P										P	P	P	
35 = Liquidity provider	P	P										P	P	P	
36 = Entering Trader	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
66 = Market Maker	P	P										P	P	P	
81 = Broker Client ID	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
122 = Decision Maker	P	P										P	P	P	
300 = Investment Decision Within Firm	P	P			P		P					P	P	P	
301 = Execution Decision Within Firm	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M

Tag	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
302 = Investment Decision Country Code	P	P			P		P					P	P	P	
303 = Execution Decision Country Code	P	P			P		P					P	P	P	
304 = Client Branch Country Code	P	P			P		P					P	P	P	
TrdMatchID (880)												M	M	M	
ExecID (17)	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
ExecRefID (19)														M	
ExecType (150)	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
OrdStatus (39)	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
OrdRejReason (103)															M
ExecRestatementReason (378)		M		M											
AccountType (581)	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
OrderCategory (1115)												C	C	C	

Tag	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
SecurityID (48)	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
SecurityIDSource (22)	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
Side (54)	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
OrderQty (38)	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
OrdType (40)	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
Price (44)	C	C	M	C	C	C	C	C	C	C	C	M	M	M	C
StopPx (99)	C	C	C		C	C	C	C	C	C	C	C	C	C	C
TriggerType (1100)	C	C	C		C	C	C	C	C	C	C	C	C	C	C
TriggerPrice (1102)	C	C	C		C	C	C	C	C	C	C	C	C	C	C
TriggerPriceType (1107)	C	C	C		C	C	C	C	C	C	C	C	C	C	C
TriggerNewPrice (1110)	C	C	C		C	C	C	C	C	C	C	C	C	C	C
TriggerOrderType (1111)	C	C	C		C	C	C	C	C	C	C	C	C	C	C
TimeInForce (59)	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
ExpireDate (432)	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C

Tag	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
ExecInst (18)	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
AggressorIndicator (1057)												M	M	M	
OrderCapacity (528)	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
OrderRestrictions (529)	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
LastQty (32)												M	M	M	
LastPx (31)												M	M	M	
LeavesQty (151)	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
CumQty (14)	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
TransactTime (60)	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
DisplayQty (1138)	C	C		C	C	C	C	C	C	C	C	C	C	C	C
Text (58)	P	P			P		P					P	P	P	P
NoLegs (555)													M	C	
LegSecurityID (602)													M	C	

Tag	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
LegSecurityIDSource (603)													M	C	
LegAllocID (1366)													M	C	
LegLastPx (637)													M	C	
LegLastQty (1418)													M	C	
RejectText (1328)									C	C					C
OrderOrigination (1724)	P	P			P		P					P	P	P	P
ExecTypeReason (2431)	C			M		M	M	C	M						
SelfMatchPreventionID (2362)	P	P							C						P
NoOrderAttributes (2593)	P	P			P		P					P	P	P	P
OrderAttributeType (2594)	P	P			P		P					P	P	P	P
OrderAttributeValue (2595)	P	P			P		P					P	P	P	P

4.10.8 Order Mass Cancel Request (q)

Order Mass Cancel Request (35=q) is used to cancel the remaining quantity of a group of orders matching criteria specified within the message. Persisted orders will be included in the cancellation request. An Execution Report will be sent for each order cancelled followed by the Order Mass Cancel Report (35=r).

Order Mass Cancel Report will be returned if the request is accepted or rejected.

Tag	Field Name	Req	Data Type	Description
11	ClOrdID	Y	String (18)	Unique ID of Order Mass Cancel Request as assigned by the institution. This identifier will be returned in ClOrdID (11) in the Execution Report of each order cancelled.
530	MassCancelRequestType	Y	Char	Specifies the type of cancellation requested Valid values: 1 = Cancel orders for a Security ID (tradable instrument e.g. CADec19) 3 = Cancel orders for a Product (contract e.g. CADF - Copper Future) 7 = Cancel all orders
Component Block <Parties>		C*		See Parties Component Block Conditionally required for MassCancelRequestType (530) = '7' (Cancel all orders) to specify the PartyID (448) when cancelling orders for a specific end client, PartyRole (452) = '81' Broker Client ID. If not specified, orders will be cancelled for the FIX Comp ID of the message originator.
Component Block <Instrument>				
207	SecurityExchange	C*	Exchange (4)	Market which is used to identify the security: XLME

Tag	Field Name	Req	Data Type	Description
				Conditionally required if Symbol (55) is specified.
1227	ProductComplex	C*	String (4)	Identifies an entire suite of products for a given market. Valid values: LME = Base LMEP = Precious Conditionally required if Symbol (55) is specified
55	Symbol	C*	String (20)	Symbol for the LME contract code e.g. CADF (Copper Future) or OCDF (Copper Monthly Average Future) Conditionally required if MassCancelRequestType (530) = '3' Cancel orders for a Product
48	SecurityID	C*	Int	Tradable instrument identifier. Conditionally required if MassCancelRequestType (530) = '1' Cancel orders for a Security ID.
22	SecurityIDSource	C*	String (1)	Identifies the source of the SecurityID (48): 8 = Exchange Symbol Conditionally required when SecurityID (48) is specified.
End Component Block				
54	Side	N	Char	Optional qualifier to indicate the side of the market for which orders are to be cancelled. Can be used if MassCancelRequestType (530) = '3' Cancel orders for a Product. Absence of this field indicates that orders are to be cancelled regardless of side.
60	TransactTime	Y	UTCTimestamp	Timestamp when the message was generated.

4.10.9 Order Mass Cancel Report (r)

Order Mass Cancel Report (35=r) is returned in response to an Order Mass Cancel Request (35=q).

Each affected order that is cancelled is acknowledged with a separate Execution Report (35=8).

Tag	Field Name	Req	Data Type	Description
11	ClOrdID	N	String (18)	ClOrdID provided on the Order Mass Cancel Request.
1369	MassActionReportID	Y	String (19)	Unique Identifier for the Order Mass Cancel Report assigned by the system
530	MassCancelRequestType	Y	Char	Specifies the type of cancellation required: Valid values: 1 = Cancel orders for a SecurityID 3 = Cancel orders for a Product (Symbol) 7 = Cancel all orders
531	MassCancelResponse	Y	Char	Indicates the action taken on the cancel request: Valid values: 0 = Cancel request rejected 1 = Cancel orders for a SecurityID 3 = Cancel orders for a Product (Symbol) 7 = Cancel all orders
532	MassCancelRejectReason	C*	Int	Indicates why the Order Mass Cancel Request was rejected. Conditionally required if MassCancelResponse (531) = '0' Cancel request rejected. Valid values: 1 = Invalid or Unknown Security 3 = Invalid or Unknown Product 99 = Other
533	TotalAffectedOrders	Y*	Int	Indicates the total number of orders affected by the Order Mass Cancel Request.

Tag	Field Name	Req	Data Type	Description
Component Block <Parties>		C*		See Parties Component Block Conditionally required for MassCancelRequestType (530) = 7 (Cancel all orders) to specify the PartyID (448) when cancelling orders for a specific end client, PartyRole (452) = '81' Broker Client ID.
Component Block <Instrument>				
207	SecurityExchange	C*	Exchange (4)	Market which is used to identify the security: XLME Conditionally required if Symbol (55) is specified.
1227	ProductComplex	C*	String (4)	Identifies an entire suite of products for a given market. Valid values: LME = Base LMEP = Precious Conditionally required if Symbol (55) is specified
55	Symbol	C*	String (20)	Symbol for the LME contract code e.g. CADF (Copper Future) or OCDF (Copper Monthly Average Future) Conditionally required if MassCancelRequestType (530) = '3' Cancel orders for a Product
48	SecurityID	C*	Int	Tradable instrument identifier. Conditionally required if MassCancelRequestType (530) = '1' Cancel orders for a Security ID.
22	SecurityIDSource	C*	String (1)	Identifies the source of the SecurityID (48): 8 = Exchange Symbol

Tag	Field Name	Req	Data Type	Description
				Conditionally required when SecurityID (48) is specified.
End Component Block				
54	Side	N	Char	Optional qualifier to indicate the side of the market for which orders are to be cancelled. Can be used if MassCancelRequestType (530) = '3' Cancel orders for a Product. Absence of this field indicates that orders are to be cancelled regardless of side.
60	TransactTime	Y*	UTCTimestamp	Timestamp when the message was generated.
58	Text	C*	String (50)	Identifies the reason for rejection. Conditionally required if MassCancelRejectReason (532) = '99' Other.

4.10.10 Quote Request (R)

Quote Request (35=R) is used to requests quotes from market participants.

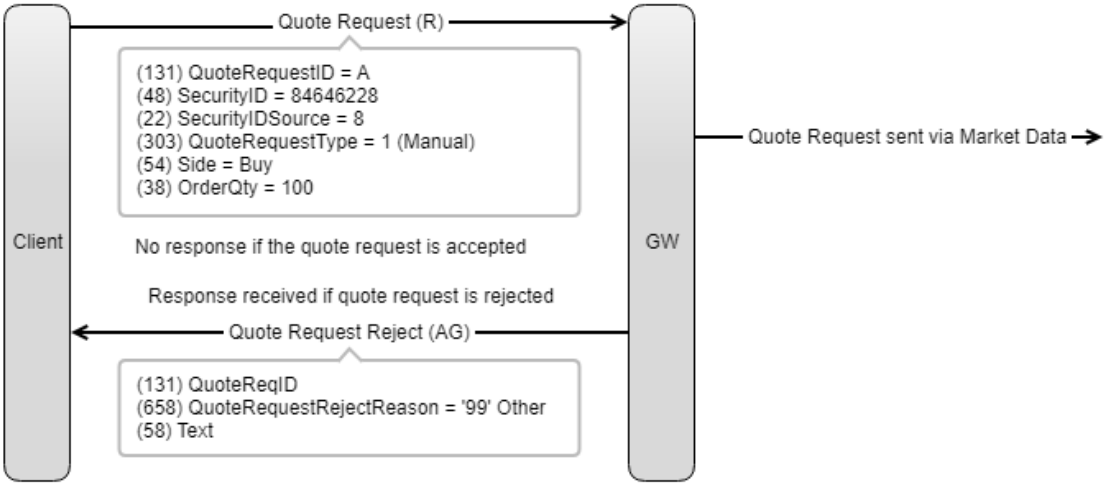
The Quote Request is disseminated via the Market Data service to market participants. The originator of the Quote Request will not receive any acknowledgement unless the quote request is rejected.

Tag	Field Name	Req	Data Type	Description
131	QuoteReqID	Y	String (18)	Unique identifier for quote request message.
Component Block <QuotReqGrp>				
146	NoRelatedSym	Y	NumInGrp (1)	Number of related symbols (instruments) in this request. The value can only be 1.
Component Block <Instrument>				
>48	SecurityID	Y*	Int	Tradable instrument identifier
>22	SecurityIDSource	C*	String (1)	Identifies the source of the SecurityID (48):

Tag	Field Name	Req	Data Type	Description
				8 = Exchange Symbol Conditionally required when SecurityID (48) is specified.
End Component Block				
303	QuoteRequestType	Y*	Int	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
54	Side	N	Char	Side of order. If not defined indicates a two-sided quote is required. Valid values: 1 = Buy 2 = Sell
Component Block <OrderQtyData>				
38	OrderQty	N	Qty	Order quantity. If not entered, a volume of 0 will be published
End Component Blocks				

Example Message Flow

RFQ Submission



4.10.11 Quote Request Reject (AG)

Quote Request Reject (35=AG) notifies the originator that their Quote Request (35=R) has been rejected.

Tag	Field Name	Req	Data Type	Description
131	QuoteReqID	Y	String (18)	Unique identifier for quote request message.
658	QuoteRequestRejectReason	Y	Int	Reason the Quote Request (R) was rejected. Valid value: 99 = Other. Text (58) will contain more specific information.
Component Block <QuotReqRjctGrp>				
146	NoRelatedSym	Y	NumInGrp (1)	Number of related symbols (instruments) in the Quote Request. The value can only be 1.
Component Block <Instrument>				
>48	SecurityID	Y*	Int	Tradable instrument identifier

Tag	Field Name	Req	Data Type	Description
>22	SecurityIDSource	C*	String (1)	Identifies the source of the SecurityID (48): 8 = Exchange Symbol Conditionally required when SecurityID (48) is specified.
End Component Block				
54	Side	C*	Char	Side of order. Valid values: 1 = Buy 2 = Sell Conditionally required if specified on the original message.
Component Block <OrderQtyData>				
38	OrderQty	C*	Qty	Total order quantity of the order. Conditionally required if specified on the original message.
End Component Blocks				
58	Text	C*	String (50)	Identifies the reason for rejection. Conditionally required if QuoteRequestRejectReason (658) = '99' Other.