Risk Management 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.

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

Any 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 Definition

3.1 User Roles

3.1.1 Member Risk Management User

A Member Risk Management user will be responsible for the reference data of their own organisation and that of any related entities.

A Member Risk Manager will be able to perform the following functions:

Function	Messages
Create Risk Groups using PartyRole '38' Position Account	Party Details Definition Request (CX) Party Details Definition Request Act (CY)
Define and manage end clients using PartyRole '81' Broker Client ID and assign to Risk Groups	Party Details Definition Request (CX) Party Details Definition Request Act (CY)
Request a snapshot of Risk Groups and end clients within each group	Party Details List Request (CF) Party Details List Report (CG)
Modify limit values assigned to Risk Groups	Party Risk Limits Definition Request (CS) Party Risk Limits Definition Request Ack (CT)
View limits on Risk Groups	Party Risk Limits Request (CL) Party Risk Limits Report (CM)
Receive current utilisation and utilisation alerts	Party Risk Limits Report (CM)
Initiate Kill Switch/Reinstate following a kill	Party Action Request (DH) Party Action Report (DI)
Maintain Self Execution Prevention parameters	Party Entitlements Definition Request (DA) Party Entitlements Definition Request Ack (DB)
View and maintain Market Marker Protection parameters	Party Risk Limits Definition Request (CS) Party Risk Limits Definition Request Ack (CT)

3.1.2 View Only Risk Management User

A View Only Risk Management user will be able to access the reference data of their own organisation and that of any related entities specifically to:

- View limit valuest set at Member and Risk Group level

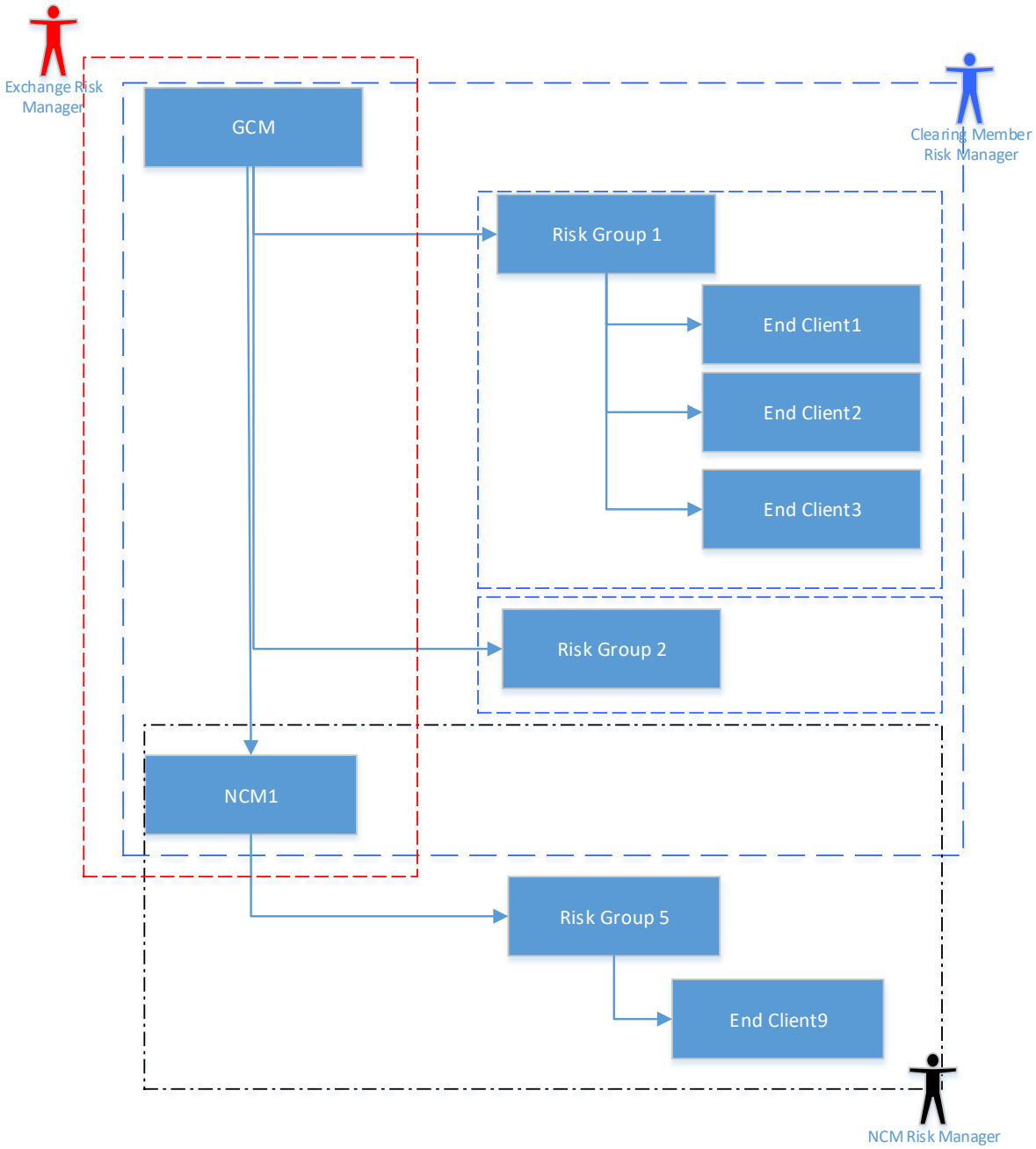
- Receive current utilisation and utilisation alerts
- View Market Maker Protection parameters.

3.2 Risk Groups

A Member Risk Manager will manage their limits using risk groups. The Member Risk Manager will allocate limits at Member and risk group level. End client entities will be assigned to a risk group using PartyRole ‘81’ Broker Client ID and will be mandatory for order submissions.

Initially the end clients will be assigned to a default risk group which has a limit value of zero. This means that any orders that are submitted by the participant will be rejected. The Member Risk Manager can then allocate the end client to an existing risk group or create a new risk group. Where the risk group has limits preconfigured, the end client will immediately be subject to the associated limits. End clients cannot be allocated to more than one group.

The following diagram shows the hierarchy at which limits can be set. Limits are set by the Exchange and Member Risk Managers. The Exchange will configure limits at Member level. All Members will set limits at both Member and risk group levels within their own firm. In addition GCMs will set limits at Member level for any NCMs that clear through them.



A Member Risk Manager can manage risk groups and end clients during the trading day but the request will be actioned at the end of the day so as not to affect utilisation.

3.3 Risk Limit Types

The limit types that are available to be set by Members are defined by the Exchange along with the level in the product hierarchy to which they apply. For all defined limits, the Exchange sets limit values on each Member at Member level. In addition to Exchange set limit values, Member Risk Managers will set a limit value for each limit set by the Exchange. If a limit value is not specified then the default value will be zero and orders will be rejected. The most stringent from the Exchange / Member set limits will apply.

The Exchange Risk Managers will be able to view all Member Risk Manager set limits but will not be able to adjust them on behalf of a Member. Similarly Member Risk Managers will be able to view all Exchange risk manager set limits using a Party Risk Limits Request (35=CL) and specifying RiskLimitRequestType (1760) = '1' Definitions. Using the same request, an NCM will be able to view the Exchange limits and their own Member limits. A GCM will not be able to view the limits that the NCM has set on themselves.

3.3.1 Per Order Quantity

Risk check to prevent inadvertent entry of large order quantity (fat finger error). This limit type is mandated through MiFID II under RTS 7 Article 20 Pre-trade and post-trade controls, (Article 48(4) and (6) of Directive 2014/65/EU).

3.3.2 Per Order Notional Value

Risk check to prevent the entry of an order that exceeds the notional value limit. The value of the order is calculated as the product of the order quantity, lot size and order price. This limit type is mandated through MiFID II under RTS 7 Article 20 Pre-trade and post-trade controls, (Article 48(4) and (6) of Directive 2014/65/EU).

3.3.3 Gross Long Quantity

Risk check on accumulated gross long quantity which is calculated as the sum of bid orders and buy trades.

3.3.4 Gross Short Quantity

Risk check on accumulated gross short quantity which is calculated as the sum of offer orders and sell trades.

3.3.5 Net Short Quantity

Risk check on accumulated net short quantity which is calculated as the sum of offer orders and sell trades minus buy trades.

3.3.6 Net Long Quantity

Risk check on accumulated net long quantity which is calculated as the sum of bid orders and buy trades minus sum of sell trades.

3.4 Risk Limit Alerts and Breaches

Alert thresholds will be configured by the Exchange to warn Member and Exchange Risk Managers when a cumulative limit utilisation[1] nears the limit value set. Alerts will not be applicable to Per Order Quantity and Per Order Notional Value.

[1] Cumulative limit utilisation is for the day only, limit values are fully available on the next day.

Alert threshold levels are based on a percentage of utilisation of the limit value set for each limit, for example:

- Warning Amber 75% and above
- Warning Red 90% or above
- Warning Limit Reached at 100%.

Member and Exchange Risk Managers will be notified when an order either triggers an alert or breaches a limit by an unsolicited Party Risk Limits Report (35=CM) which will include:

Details	FIX Tag
End Client identifier	PartyDetailSubID (1695)
Tradable instrument	InstrumentScope component block
Exchange or Member Limit Type	RiskLimitType (1530)
Participant level (Risk Group or Member)	PartyDetailRole (1693)
Participant identifier (Risk Group ID or Member mnemonic)	PartyDetailID (1691)
Order Identifier	Text (58)
Credit utilization percentage	RiskLimitUtilizationPercent (1765)
Breach time	TransactTime (60)

A Member Risk Manager can request credit utilisation against limits using a Party Risk Limits Request (35=CL) and specifying RiskLimitRequestType (1760) = ‘3’ Definitions and utilisations. Usage of this request type will be throttled.

3.5 Kill Switch

Exchange or Member Risk Managers can use a Party Action Request (35=DH) to suspend or halt trading activity at Member, risk group or end client level. PartyActionType (2329) = ‘0’ Suspend prevents order submission and order revision but allows order cancellation whereas PartyActionType (2329) = ‘1’ Halting trading prevents further order entry and pulls all orders in the market.

Enacting the kill switch affects the related entities at that level and below in the hierarchy. For example, a Clearing Member enacting a kill at Clearing Member level will also encompass all the GCMs risk groups and end clients and their NCMs. Once enacted the kill switch state will persist until explicitly lifted.

A successful Party Action Request (DH) will be acknowledged by a Party Action Report (35=DI) with PartyActionResponse (2332) = ‘0’ Accepted. Notification of the completion of the action will be sent unsolicited with PartyActionResponse (2332) = ‘1’ Completed.

A trader with orders in the market will be notified that the kill switch has been enacted to halt trading activity by unsolicited order cancellations. RejectText (1328) will specify the originator of the instruction e.g. Member Kill Switch enacted.

A Member Risk Manager can submit a Party Details List Request (35=CF) for a snapshot of risk groups and end clients. Party Details List Report (35=CG) will include PartyDetailStatus (1672) which can be either '0' Active, '1' Suspended or '2' Halted.

Following a kill, a Member Risk Manager can specify whether to reinstate at the level of the kill including or excluding lower levels in the risk hierarchy for example, a kill at risk group level, reinstate the risk group but not the end clients in the risk group.

If a kill is applied at a Member level and the request to reinstate is at lower level in the risk hierarchy than the Member level e.g. risk group or end client, this will be rejected as levels below the kill cannot be reinstated until all parent entities have been reinstated.

The following table contains the application of a Party Action Request (DH) by a member using a specific PartyRole to initiate a kill or reinstate and the affected level in the risk hierarchy:

Initiating PartyRole and Target	Kill level	Reinstate level
GCM/ICM/NCM using: PartyRole (452) = '118' Operator	Member and all risk groups and all end clients Not permitted for a GCM	Member and all risk groups and all end clients (excluding NCMs)
GCM/ICM/NCM using: PartyRole (452) = '118' Operator Target: RelatedPartyDetailRole (1565) = '118' Operator	Not permitted	Only the Member level but not risk groups or end clients (or NCMs)
GCM/ICM/NCM using: PartyRole (452) = '118' Operator Target: RelatedPartyDetailRole (1565) = '38' Position Account	Not permitted	Specific risk group within the Member but not end clients in the risk group.
GCM/ICM/NCM using: PartyRole (452) = '118' Operator Target: RelatedPartyDetailRole (1565) = '38' Position Account with PartyRelationship (1515) = '4001' Include lower levels	Specific risk group within the Member and all end clients within the risk group	Specific risk group within the Member and all end clients within the risk group

Initiating PartyRole and Target	Kill level	Reinstate level
GCM/ICM/NCM using: PartyRole (452) = '118' Operator Target: RelatedPartyDetailRole (1565) = '81' Broker Client ID	Specific end client in any risk group	Specific end client in any risk group
GCM using: PartyRole (452) = '4' Clearing Member	All GCM risk groups and end clients and all NCMs	All GCM risk groups and end clients and any NCMs that clear through the GCM
GCM using: PartyRole (452) = '4' Clearing Member Target: RelatedPartyDetailRole (1565) = '1' Executing Firm	Specific NCM	Specific NCM

Reinstatement can only be performed by the party (Exchange or Member) that enacted the kill. The Exchange cannot reinstate a Member enacted kill and a Member lift a kill enacted by the Exchange.

The Risk Manager can reinstate at any level in the hierarchy assuming that the level above in the hierarchy is active. For a Member level kill, the Risk Manager can reinstate the Member and all the related risk groups and end clients or reinstate just the Member but not the risk groups and end clients. Similarly a kill at risk group level can be reinstated to include or exclude the end clients of that risk group.

A successful request to reinstate will also be acknowledged by a Party Action Report (DI) using PartyActionResponse (2332) = '0' Accepted and subsequently confirmed with PartyActionResponse (2332) = '1' Completed.

3.6 Self Execution Prevention (SEP)

Participants can use Self Execution Prevention functionality to prevent orders or quotes from crossing with submissions made by traders in their organisation.

A Member Risk Manager can use a Party Entitlements Definition Request (35=DA) to specify SEP Match identifiers and SEP response should two orders with an identical SEP Match IDs be able to execute.

SEP parameters must be specified together for additions and modifications as follows:

FIX Tag	Values
EntitlementAttribType (1778)	4001 = SEP Match ID

FIX Tag	Values
	4002 = SEP Response
EntitlementAttribValue (1780)	SEP Match ID value (maximum 9 digits) SEP response value either: 1 = Cancel incoming order 2 = Cancel resting order

Additions and modifications to SEP parameters will not take effect until the next trading day. Matching SEP IDs whose configuration has not yet taken effect will have the exchange default protection response applied.

The SEP Match ID value will be submitted in the SelfMatchPreventionID (2362) on order submission.

3.7 Market Maker Protection (MMP)

Market Maker Protection provides a mechanism to prevent too many simultaneous trade executions on orders and quotes within a specific time period.

A Member Risk Manager can use a Party Limits Definition Request (35=CS) to specify the level of protection that should apply to a permitted trading user (CompID) in a particular contract.

The Risk Manager will specify the protection type and protection limit measured over a configured time period which is defined in seconds. This time period defines the length of the rolling time interval for MMP recalculation which is used to determine if the quantity limit has been reached. If the limit threshold is breached the protection response is invoked to pull orders and reject further orders until MMP is explicitly reset by the trading user. The Member Risk Manager cannot perform an MMP reset on a trader's behalf.

The following protection types can be set:

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

The relevant FIX tags for MMP are as follows:

FIX Tag	Values
RiskLimitType (1530)	Market Maker Protection Type: 301 = Cumulative percent over time 302 = Volume over time 303 = Number of Tradable Instruments traded over time

FIX Tag	Values
RiskLimitAmount (1531)	Protection limit count
RiskLimitVelocityPeriod (2336)	Protection timeframe
RiskLimitVelocityUnit (2337)	Unit of time S = Second

A Member Risk Manager can use a Party Limits Definition Request (CS) to retrieve the MMP parameters set for a CompID.

A Party Limits Definition Request Ack (35=CT) will be returned in response to a request or unsolicited following a change to an MMP floor by the Exchange.

4 Message Definitions

4.1 Supported Message Types

- Logon (A)
- Heartbeat (0)
- Test Request (1)
- Resend Request (2)
- Sequence Reset (4)
- Logout (5)
- Reject (3)
- Business Message Reject (j)
- News (B)
- Party Details Definition Request (CX)
- Party Details Definition Request Ack (CY)
- Party Details List Request (CF)
- Party Details List Report (CG)
- Party Risk Limits Definition Request (CS)
- Party Risk Limits Definition Request Ack (CT)
- Party Risk Limits Request (CL)
- Party Risk Limits Report (CM)
- Party Action Request (DH)
- Party Action Report (DI)
- Party Entitlements Definition Request (DA)
- Party Entitlements Definition Request Ack (DB)

4.1.1 Inbound Messages

- Logon (A)
- Heartbeat (0)
- Test Request (1)
- Resend Request (2)
- Sequence Reset (4)
- Logout (5)
- Party Details Definition Request (CX)
- Party Details List Request (CF)
- Party Risk Limits Definition Request (CS)
- Party Risk Limits Request (CL)
- Party Action Request (DH)
- Party Entitlements Definition Request (DA)

4.1.2 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)
- Party Details Definition Request Ack (CY)
- Party Details List Report (CG)
- Party Risk Limits Definition Request Ack (CT)
- Party Risk Limits Report (CM)
- Party Action Report (DI)
- Party Entitlements Definition Request Ack (DB)

4.2 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.

4.3 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, forwarded to CheckSum (10).
35	MsgType	Y	String (3)	Defines message type.
1128	ApplVerID	N	String (1)	Version of FIX used in the message: 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. RMLME
34	MsgSeqNum	Y	SeqNum (9)	Message sequence number.

Tag	Field Name	Req	Data Type	Description
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).

4.4 Message Trailer

Tag	Field Name	Req	Data Type	Description
10	CheckSum	Y	String (3)	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.

4.5 Administrative Messages

4.5.1 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.

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)
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.

Tag	Field Name	Req	Data Type	Description
789	NextExpectedMsgSeqNum	C	SeqNum (9)	Next expected MsgSeqNum (34) value to be received. Conditionally required when reconnecting intraday or logging on after a failover.
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

4.5.2 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 (1).

4.5.3 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 (1) to be returned in resulting Heartbeat (0).

4.5.4 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).

4.5.5 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.

4.5.6 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 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.

4.5.7 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

Tag	Field Name	Req	Data Type	Description
				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.

4.5.8 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.
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.
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.

4.5.9 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
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>				
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
End Component Block				

4.6 Common Component Blocks

4.6.1 RiskInstrumentScope

Repeating group of InstrumentScope components. Used to specify the product level to which a request applies. Required if RiskInstrument is specified.

Mandatory for Party Risk Limits Definition Request (CS), Party Risk Limits Definition Request Ack (CT) and Party Risk Limits Report (CM).

Conditionally required for Party Risk Limits Request (CL).

Tag	Field Name	Req	Data Type	Description
1534	NoRiskInstrumentScopes	Y*	NumInGrp (1)	Number of risk instrument scopes. This value can only be 1.
1535	InstrumentScopeOperator	Y*	Int	Operator to perform on the instrument(s) specified. Valid value: 1 = Include

4.6.2 RiskInstrument

Used to specify the product level to which a request applies. Instrument reference data will be available on the Market Data feed.

Mandatory for Party Risk Limits Definition Request (CS), Party Risk Limits Definition Request Ack (35=CT) and Party Risk Limits Report (CM).

Conditionally required for Party Risk Limits Request (CL).

Tag	Field Name	Req	Data Type	Description
1616	InstrumentScopeSecurityExchange	Y*	Exchange (4)	Market which is used to identify the security: XLME
1544	InstrumentScopeProductComplex	Y*	String (4)	Identifies an entire suite of products for a given market. Valid values: LME = Base LMEP = Precious
1545	InstrumentScopeSecurityGroup	Y*	String (2)	An exchange specific name assigned to a group of related securities which may be concurrently affected by market events and actions e.g. AH for Aluminium
1547	InstrumentScopeSecurityType	Y*	String (4)	Required for risk limits. Indicates the security type. Valid values: FUT = Future OPT = Option
1548	InstrumentScopeSecuritySubType	N	String (3)	Optional for risk limits. Indicates the security sub type. Valid values: 0 = Outright 1 = Carry (Futures only) 101 = TomNext (Futures only)
1536	InstrumentScopeSymbol	C*	String (20)	Conditionally required for MMP. Not required for risk limits.

Tag	Field Name	Req	Data Type	Description
				Symbol for the LME contract code e.g. CAFDF (Copper Future) or OCDF (Copper Monthly Average Future).

4.7 Application Messages

4.7.1 Party Details Definition Request (CX)

Party Details Definition Request (35=CX) is used to:

- Add, modify or remove a risk group
- Add an end client to a risk group
- Remove an end client from a risk group
- Move an end client between risk groups.

Note: A risk group cannot be removed if it contains end clients.

Tag	Field Name	Req	Data Type	Description
1505	PartyDetailsListRequestID	Y	String	Unique identifier for Party Details List Request
Component Block <PartyDetailsUpdateGrp> - Required				
1676	NoPartyUpdates	Y*	NumInGrp (1)	Number of party updates. The value can only be 1.
>1324	ListUpdateAction	Y*	Char	Action to be performed. Valid values: A = Add M = Modify D = Delete
Component Block <PartyDetailGrp>				
1671	NoPartyDetails	Y*	NumInGrp (1)	Number of party details. The value can only be 1.
>1691	PartyDetailID	Y*	String (16)	Party identifier. End client identifier or Risk Group identifier A value '0' is reserved for the Member default risk group
>1692	PartyDetailIDSource	Y*	Char	Used to identify source of PartyID value. Valid value: D = Proprietary/Custom (default)
>1693	PartyDetailRole	Y*	Int	Identifies the type of PartyID.

Tag	Field Name	Req	Data Type	Description
				Valid values: '38' Position Account = Risk Group '81' Broker Client ID = End Client
Component Block <RelatedPartyDetailGrp>				
1562	NoRelatedPartyDetailID	C*	NumInGrp (1)	Number of related party detail identifiers. Not required if adding a new end client to the default risk group. Conditionally required if ListUpdateAction (1324) = 'M' Modify and PartyDetailRole (1693) = '81' Broker Client ID = End Client to move an end client from one risk group to another.
>1563	RelatedPartyDetailID	C	String (16)	Party identifier for the party related to the party specified in PartyDetailID (1691). Conditionally required when NoRelatedPartyDetailID (1562) > 0. Risk Group identifier A value '0' is reserved for the default risk group
>1564	RelatedPartyDetailIDSource	C	Char	Identifies the source of the RelatedPartyDetailID (1563). Conditionally required when NoRelatedPartyDetailID (1562) > 0. Valid value: D = Proprietary/Custom (default)
>1565	RelatedPartyDetailRole	C	Int	Identifies the type or role of the RelatedPartyDetailID (1563) specified. Conditionally required when NoRelatedPartyDetailID (1562) > 0. Valid value: '38' Position Account = Risk Group

Tag	Field Name	Req	Data Type	Description
>1675	RelatedPartyDetailRoleQualifier	C*	Int	Used to move an end client from one risk group to another Conditionally required when NoRelatedPartyDetailID (1562) > 0. Valid values: 18 = Current (Risk Group) 19 = New (Risk Group)
End Component Blocks				

4.7.2 Party Details Definition Request Ack (CY)

Party Details Definition Request Ack (35=CY) is sent in response to a Party Details Definition Request (35=CX). The request can be accepted or rejected.

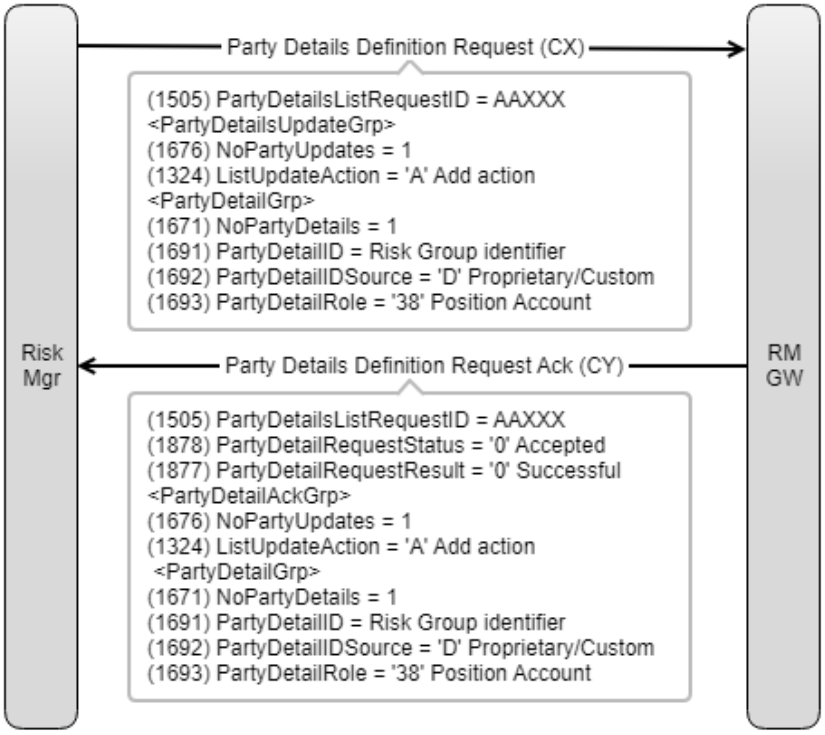
Tag	Field Name	Req	Data Type	Description
1505	PartyDetailsListRequestID	Y	String	Unique identifier for Party Details List Request.
1878	PartyDetailRequestStatus	Y	Int	Status of Party Details Definition Request. Valid values: 0 = Accepted 2 = Rejected
1877	PartyDetailRequestResult	Y*	Int	Result of Party Details Definition Request. Valid values: 0 = Successful (default) 1 = Invalid party/parties 2 = Invalid related party/parties 98 = Not authorised 99 = Other
Component Block <PartyDetailAckGrp>				
1676	NoPartyUpdates	Y*	NumInGrp (1)	Number of party updates. The value can only be 1.
>1324	ListUpdateAction	C	Char	Conditionally required when NoPartyUpdates (1676) > 0 Valid values:

Tag	Field Name	Req	Data Type	Description
				A = Add M = Modify D = Delete
Component Block <PartyDetailGrp>				
1671	NoPartyDetails	Y*	NumInGrp (1)	Number of party details.
>1691	PartyDetailID	Y*	String (16)	Party identifier. Risk Group identifier End client identifier
>1692	PartyDetailIDSource	Y*	Char	Used to identify source of PartyID value. Valid value: D = Proprietary/Custom (default)
>1693	PartyDetailRole	Y*	Int	Identifies the type of PartyID. Valid values: '38' Position Account = Risk Group '81' Broker Client ID = End Client
Component Block <RelatedPartyDetailGrp>				
1562	NoRelatedPartyDetailID	C*	NumInGrp (1)	Number of related party detail identifiers. The value can be 1 or 2. Conditionally required if ListUpdateAction (1324) = ‘M’ Modify and PartyDetailRole (1693) = '81' Broker Client ID = End Client
>1563	RelatedPartyDetailID	C	String (16)	Party identifier for the party related to the party specified in PartyDetailID (1691). Conditionally required when NoRelatedPartyDetailID (1562) > 0. Risk Group identifier

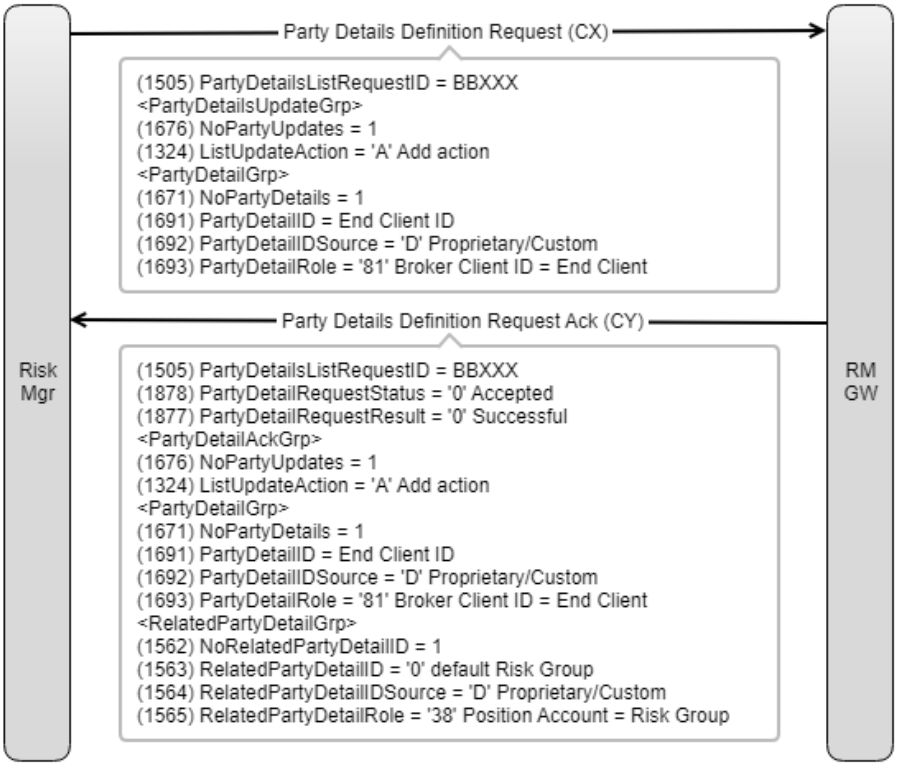
Tag	Field Name	Req	Data Type	Description
				A value '0' is reserved for the Member default risk group
>1564	RelatedPartyDetailIDSource	C	Char	Identifies the source of the RelatedPartyDetailID (1563). Conditionally required when NoRelatedPartyDetailID (1562) > 0. Valid value: D = Proprietary/Custom (default)
>1565	RelatedPartyDetailRole	C	Int	Identifies the type or role of the RelatedPartyDetailID (1563) specified. Conditionally required when NoRelatedPartyDetailID (1562) > 0. Valid value: '38' Position Account = Risk Group
>1675	RelatedPartyDetailRoleQualifier	C*	Int	Used to move an end client from one risk group to another. Conditionally required when NoRelatedPartyDetailID (1562) > 0. Valid values: 18 = Current (Risk Group) 19 = New (Risk Group)
End Component Blocks				
58	Text	C*	String (50)	Identifies the reason for rejection. Conditionally required if PartyDetailRequestResult (1877) = '99' Other

Example Message Flows

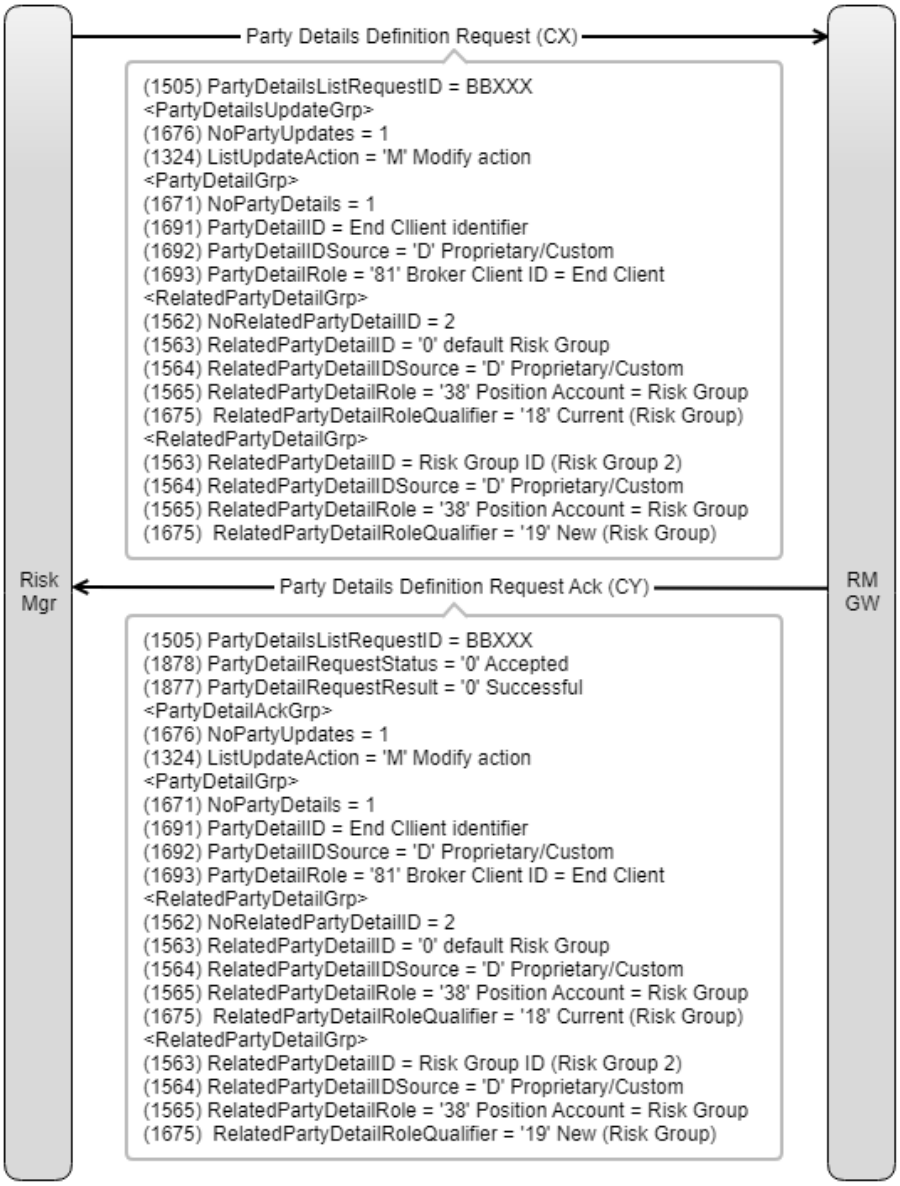
Add a Risk Group



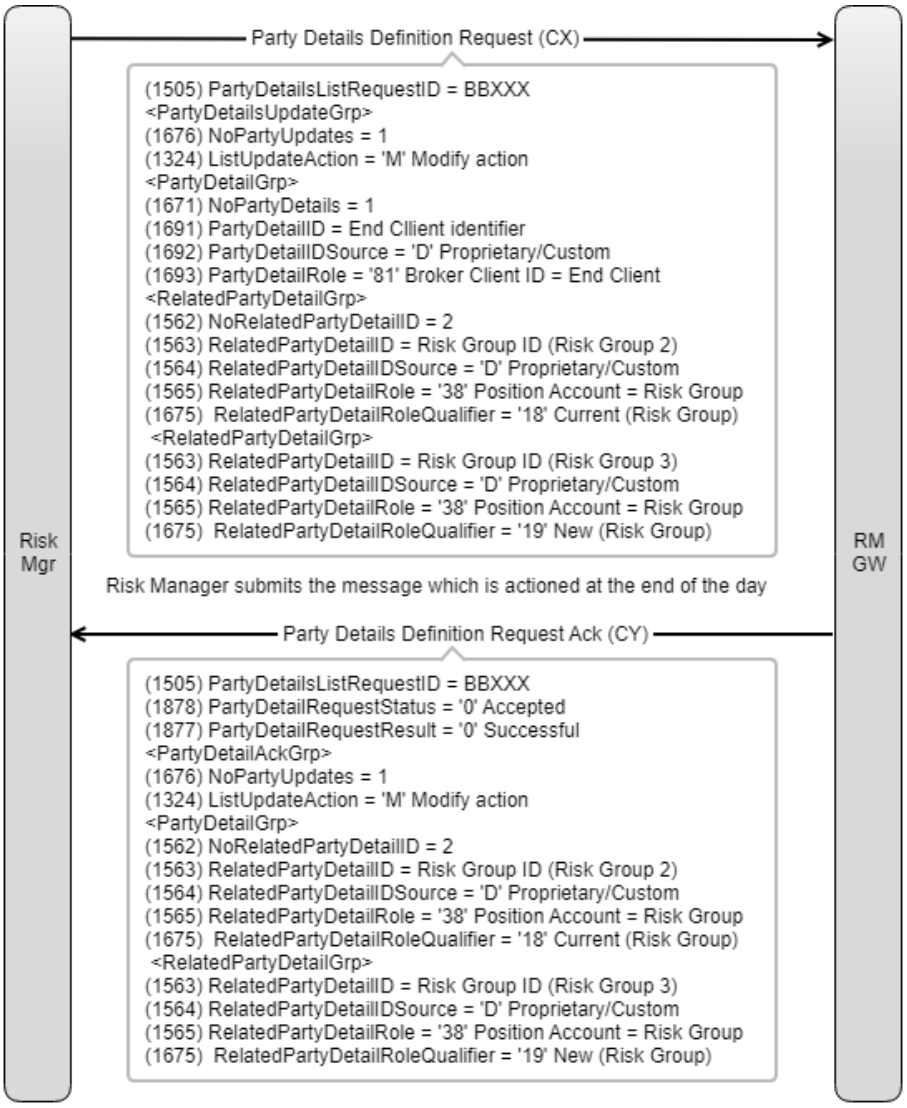
Add a new End Client and allocate to the default Risk Group



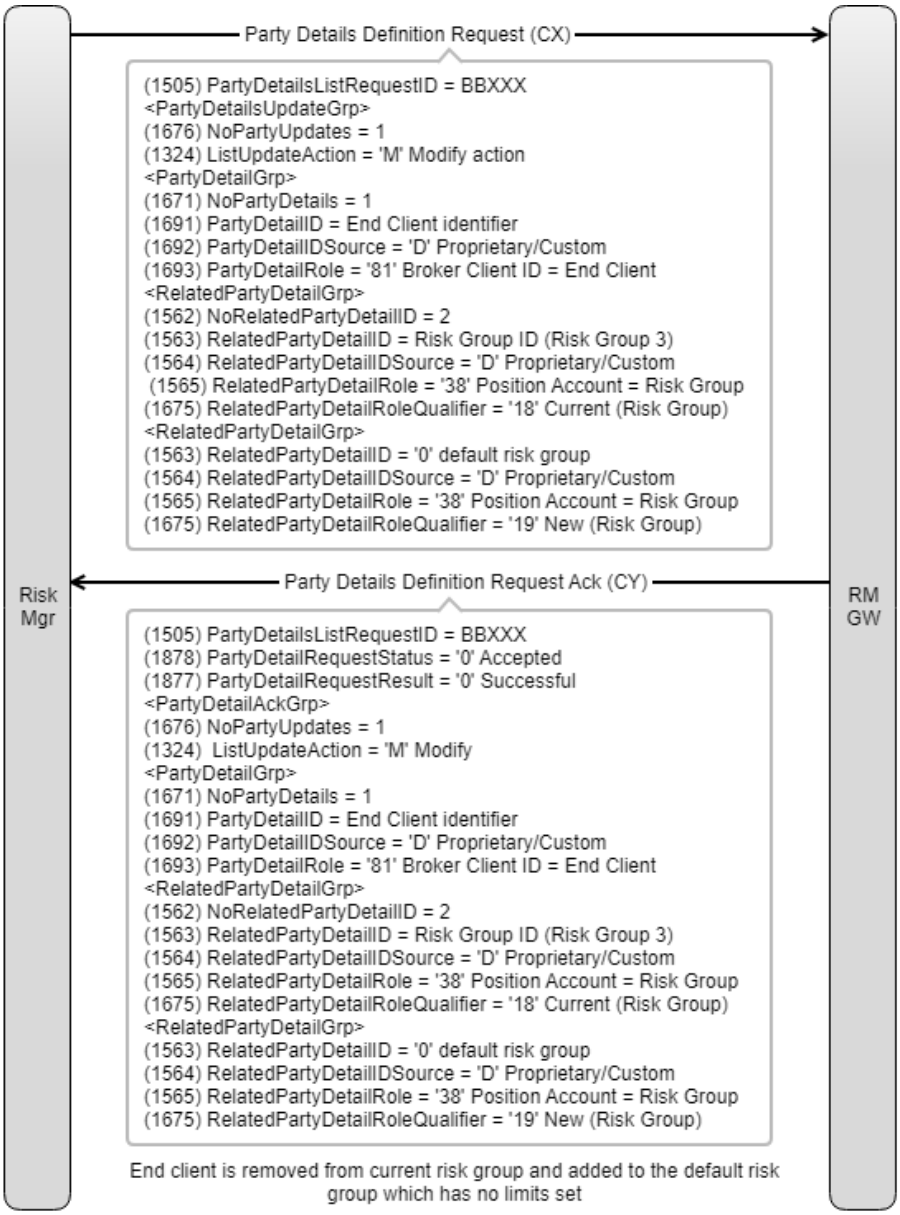
Move an End Client from the default Risk Group to a specific Risk Group



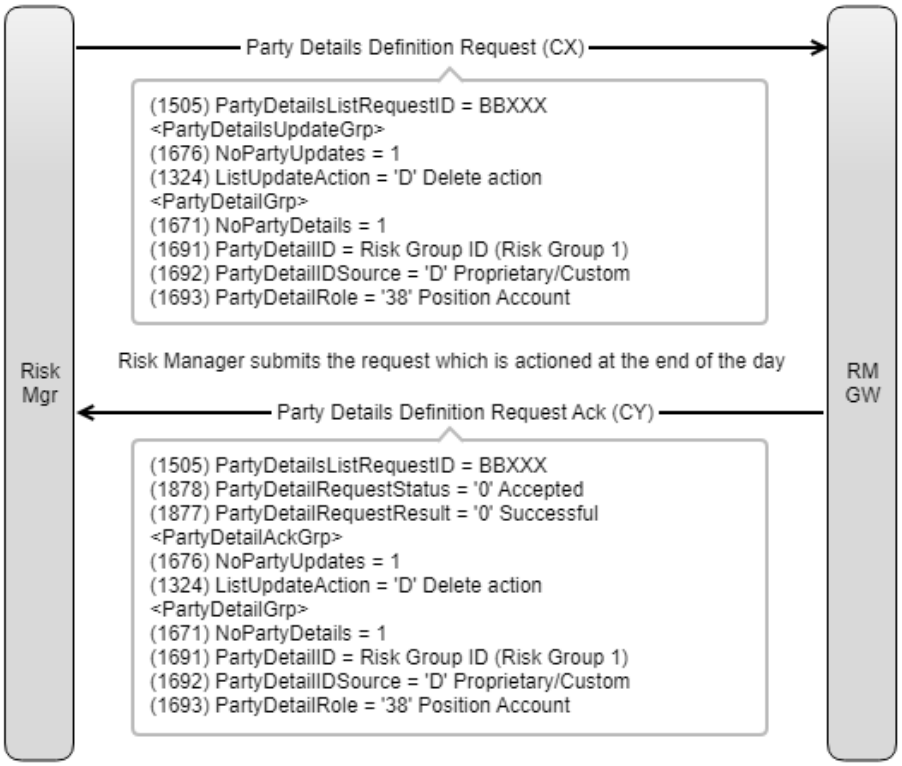
Move an End Client from one Risk Group to another



Move an End Client from a Risk Group to the default Risk Group

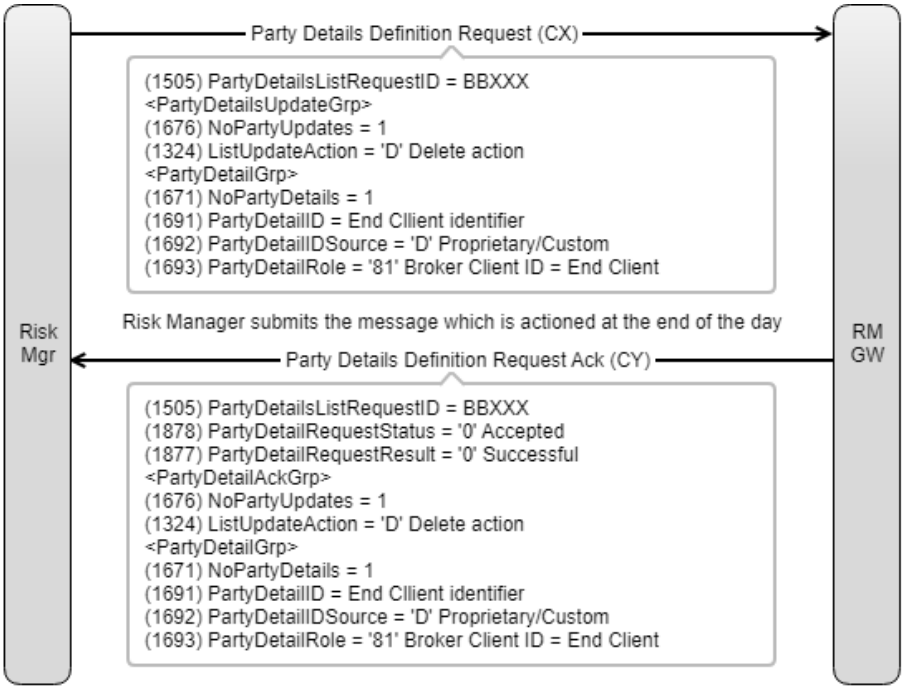


Remove a Risk Group from a Member



Remove an End Client

An end client can be deleted once it has been moved to the default risk group



4.7.3 Party Details List Request (CF)

Party Details List Request (35=CF) is used to request a snapshot of risk groups and end client reference data.

A GCM can request their own risks groups and a list of the NCMs that clear through them.

Tag	Field Name	Req	Data Type	Description
1505	PartyDetailsListRequestID	Y	String	Unique identifier of Party Details List Request.
Component Block <Parties>				
453	NoPartyIDs	N	NumInGrp (1)	Number of party updates. If specified this value can only be 1. If the Parties component block is not specified the request will be for all risks groups (and their associated end clients) and list NCMs cleared by a GCM.
>448	PartyID	C	String (16)	Party identifier. Conditionally required if NoPartyIDs (453) > 0. If this is not specified then the request is for all roles (in 452). A value '0' is reserved for the default risk group
>447	PartyIDSource	C	Char	Used to identify source of PartyID value. Conditionally required if NoPartyIDs (453) > 0. Valid value: D = Proprietary/Custom
>452	PartyRole	C	Int	Identifies the type of PartyID. Conditionally required if NoPartyIDs (453) > 0. Valid values: '1' Executing Firm = NCM '38' Position Account = Risk Group '81' Broker Client ID = End Client

Tag	Field Name	Req	Data Type	Description
End Component Block				

4.7.4 Party Details List Report (CG)

Party Details List Report (35=CG) is sent in response to a Party Details List Request (35=CF) to return a current snapshot of risk group and end client reference data. It also returns the status of the party either Active, Suspended or Halted.

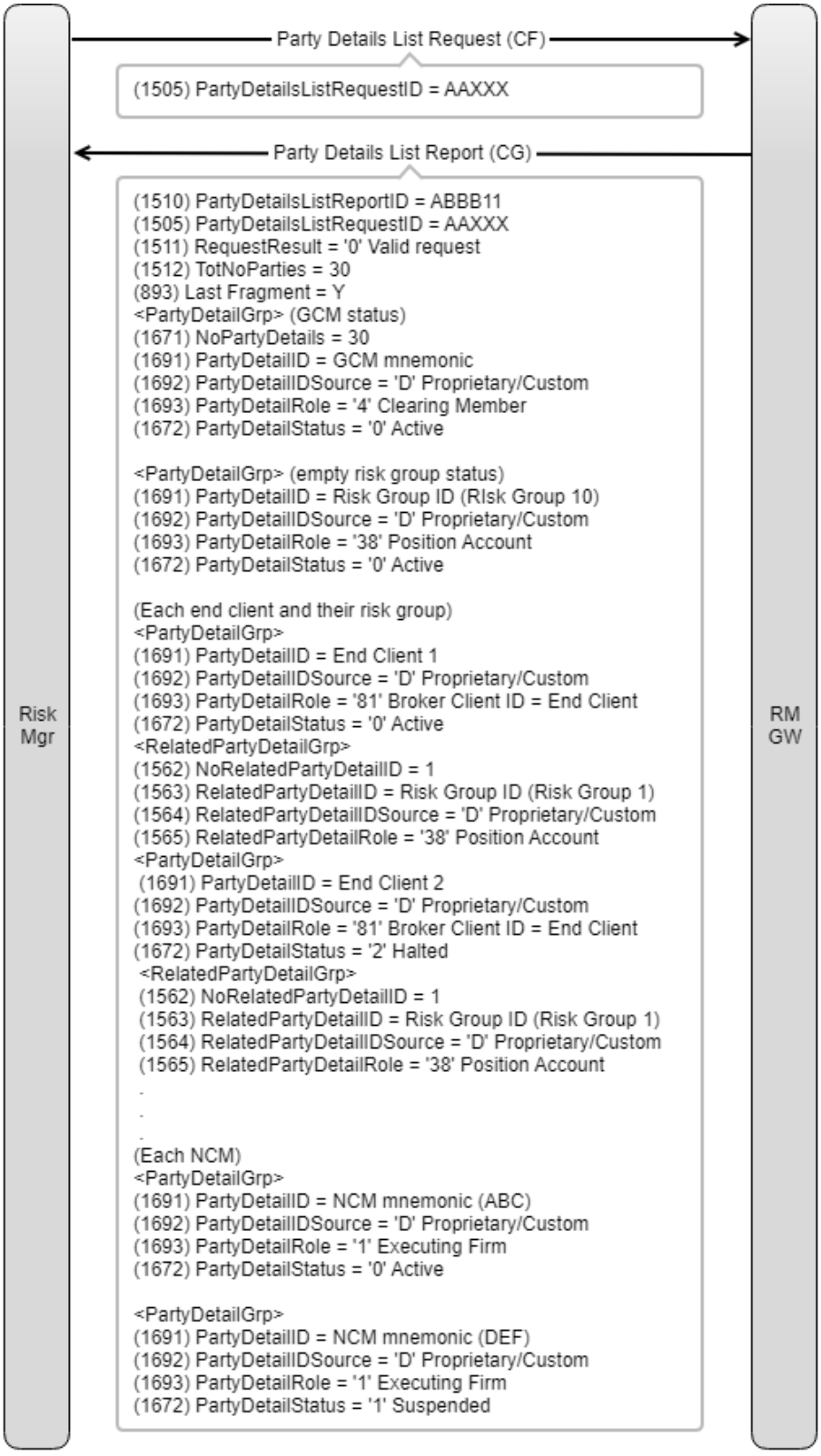
Tag	Field Name	Req	Data Type	Description
1510	PartyDetailsListReportID	Y	String	Identifier for the Party Details List Report.
1505	PartyDetailsListRequestID	Y*	String	Unique identifier of Party Details List Request.
1511	RequestResult	Y*	Int	Result of party detail list request. Valid values: 0 = Valid request 99 = Other
1512	TotNoParties	N	Int	Total number of PartyDetailGrp to be returned.
893	LastFragment	Y*	Boolean	Indicates whether this message is the last in a sequence of messages. Valid values: N = Not Last Message Y = Last Message
Component Block <PartyDetailGrp>				
1671	NoPartyDetails	Y*	NumInGrp (2)	Number of party details.
>1691	PartyDetailID	C	String (16)	Party identifier. Conditionally required when NoPartyDetails (1671) > 0. Member mnemonic Risk Group identifier End client identifier
>1692	PartyDetailIDSource	C	Char	Used to identify source of PartyID value. Conditionally required when NoPartyDetails (1671) > 0.

Tag	Field Name	Req	Data Type	Description
				Valid value: D = Proprietary/Custom (default)
>1693	PartyDetailRole	C	Int	Identifies the type of PartyID. Conditionally required when NoPartyDetails (1671) > 0. Valid values: '1' Executing Firm = NCM '4' Clearing Member = GCM or ICM '38' Position Account = Risk Group '81' Broker Client ID = End Client
1672	PartyDetailStatus	C*	Int	Indicates the status of the party identified with PartyDetailID (1691). Conditionally required when NoPartyDetails (1671) > 0. Valid values: '0' Active '1' Suspended '2' Halted
Component Block <RelatedPartyDetailGrp>				
1562	NoRelatedPartyDetailID	C*	NumInGrp (1)	Number of related party detail identifiers. The value can only be 1. Conditionally required if PartyDetailRole (1693) = '81' Broker Client ID = End Client
>1563	RelatedPartyDetailID	C	String (16)	Party identifier for the party related to the party specified in PartyDetailID (1691). Conditionally required when NoRelatedPartyDetailID (1562) > 0. Risk Group identifier
>1564	RelatedPartyDetailIDSource	C	Char	Identifies the source of the RelatedPartyDetailID (1563). Conditionally required when NoRelatedPartyDetailID (1562) > 0. Valid value: D = Proprietary/Custom (default)

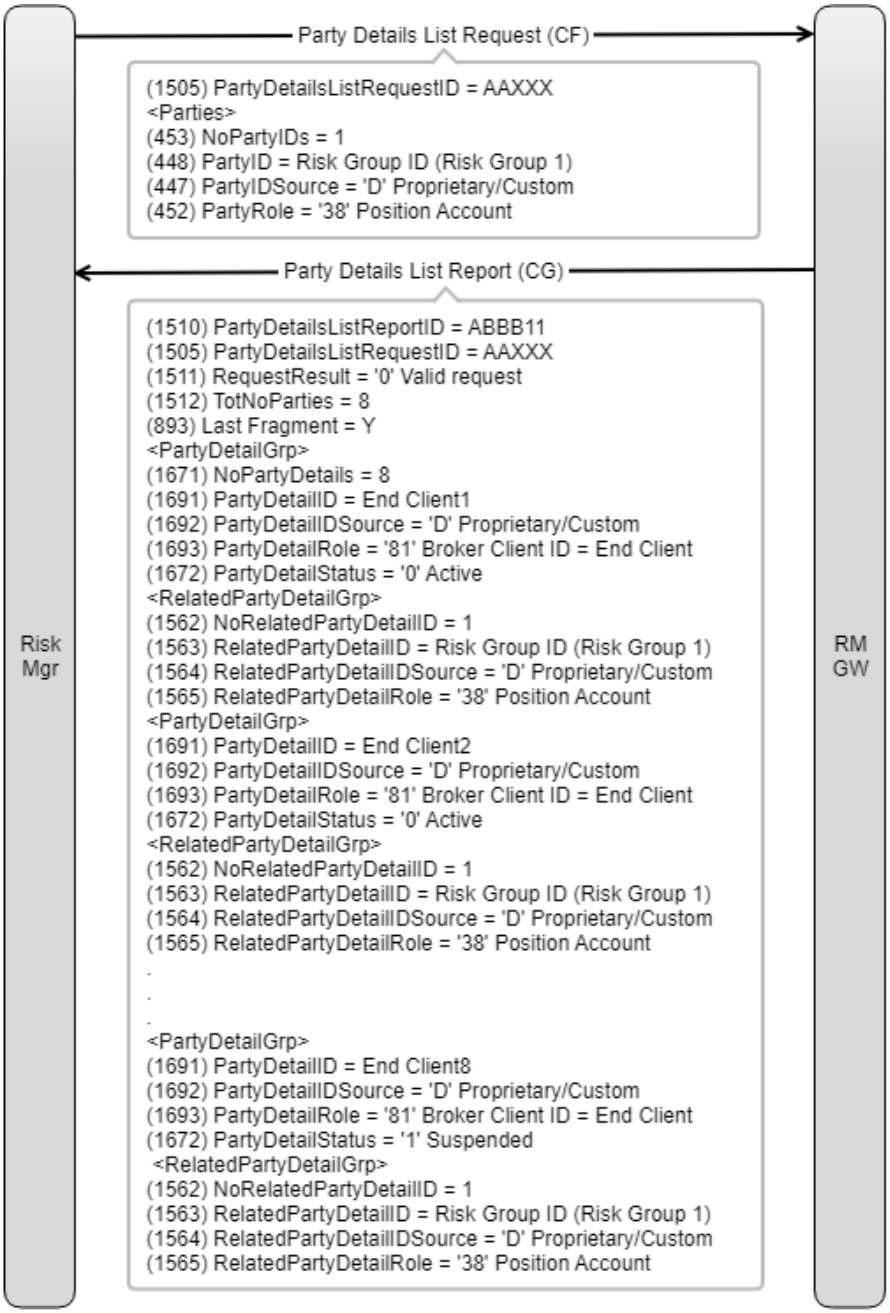
Tag	Field Name	Req	Data Type	Description
>1565	RelatedPartyDetailRole	C	Int	Identifies the type or role of the RelatedPartyDetailID (1563) specified. Conditionally required when NoRelatedPartyDetailID (1562) > 0. Valid value: '38' Position Account = Risk Group
End Component Block				
1328	RejectText	C*	String (75)	Identifies the reason for rejection. Conditionally required if RequestResult (1511) = '99' Other

Example Message Flows

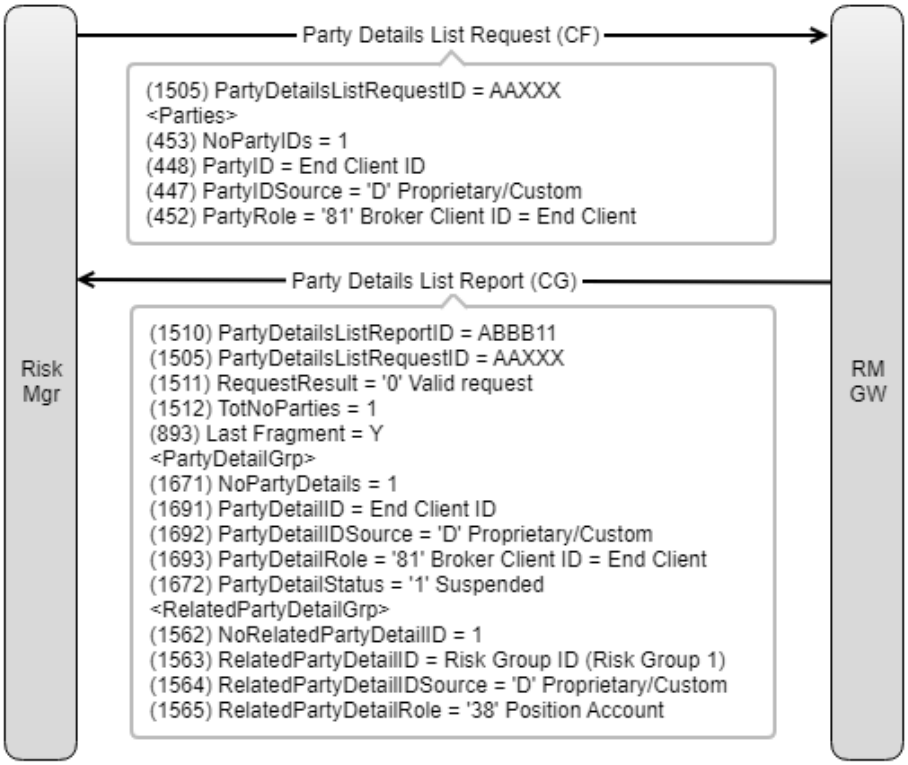
Snapshot of Risk Groups and End Clients requested by GCM



Snapshot of End Clients in a Risk Group



Snapshot of End Client



4.7.5 Party Risk Limits Definition Request (CS) for Risk Limit Configuration

Party Risk Limits Definition Request (35=CS) is used by a Member Risk Manager to modify risk limits.

Member risk limits will be predefined and set to zero by default on creation by the Exchange. The Member Risk Manager will then set their own limits and those for their NCMs using the Modify action.

Tag	Field Name	Req	Data Type	Description
1666	RiskLimitRequestID	Y	String	Unique identifier for the Party Risk Limits Request.
Component Block <PartyRiskLimitsUpdateGrp>				
1677	NoPartyRiskLimits	Y*	NumInGrp (1)	Number of party risk limits. This value must be set to 1.
>1324	ListUpdateAction	Y*	Char	Action to be performed. Valid values: M = Modify
Component Block <PartyDetailGrp>				

Tag	Field Name	Req	Data Type	Description
>1671	NoPartyDetails	Y*	NumInGrp (1)	Number of party details. The value can only be 1.
>>1691	PartyDetailID	Y*	String (16)	Party identifier. Member Mnemonic Risk Group identifier
>>1692	PartyDetailIDSource	Y*	Char	Used to identify source of PartyID value. Valid value: D = Proprietary/Custom (default)
>>1693	PartyDetailRole	Y*	Int	Identifies the type of PartyID. Valid values: '1' Executing Firm = NCM '4' Clearing Member = GCM or ICM '38' Position Account = Risk Group
Component Block <RiskLimitsGrp>				
>1669	NoRiskLimits	Y*	NumInGrp (1)	Number of risk limits for different instrument scopes. This value can only be 1.
Component Block <RiskLimitTypesGrp>				
>>1529	NoRiskLimitTypes	Y*	NumInGrp (1)	Number of risk limits with associated configuration. This value can only be 1.
>>>1530	RiskLimitType	Y*	Int	Used to specify the type of risk limit. Valid values: 201 = Member Per Order Quantity 202 = Member Per Order Notional Value 203 = Member Gross Short Quantity 204 = Member Gross Long Quantity 205 = Member Net Short Quantity 206 = Member Net Long Quantity
>>>1531	RiskLimitAmount	Y*	Amt	Specifies the risk limit amount. This amount can be a notional value or a quantity.

Tag	Field Name	Req	Data Type	Description
>>>1532	RiskLimitCurrency	C*	Currency	Specifies the risk limit currency. Conditionally required if RiskLimitAmount (1531) refers to notional value, used when RiskLimitType (1530) = '202' Member Per Order Notional Value. Valid values as supplied on Market Data security information
> Component Block <RiskInstrumentScopeGrp>		Y*	See RiskInstrumentScope	
>>Component Block <InstrumentScope>		Y*	See RiskInstrument	
End Component Blocks				

4.7.6 Party Risk Limits Definition Request Ack (CT) for Risk Limit Configuration

Party Risk Limits Definition Request Ack (35=CT) is used as a response to a Party Risk Limits Definition Request (35=CS) to accept or reject the definition of risk limits.

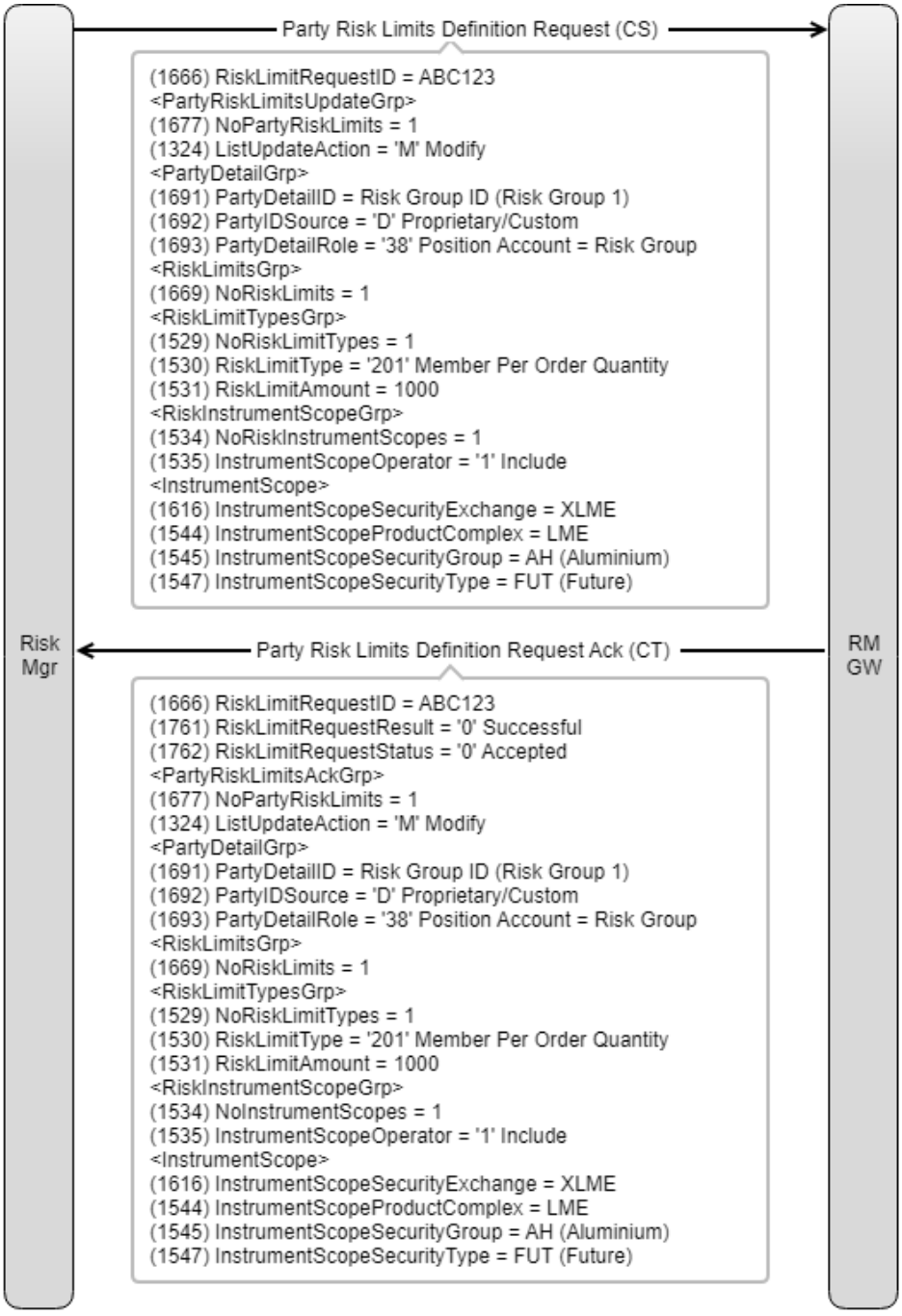
Tag	Field Name	Req	Data Type	Description
1666	RiskLimitRequestID	Y	String	Unique identifier for the Party Risk Limits Request.
1761	RiskLimitRequestResult	Y*	Int	Result of risk limit definition request. Valid values: 0 = Successful 1 = Invalid party 2 = Invalid related party 3 = Invalid risk limit type 5 = Invalid risk limit amount 7 = Invalid risk instrument scope 8 = Risk limit actions not supported 11 = Risk instrument scope not supported 13 = Risk limit already defined for party 98 = Not authorised 99 = Other

Tag	Field Name	Req	Data Type	Description
1762	RiskLimitRequestStatus	Y	Char	Status of risk limit definition request. Valid values: 0 = Accepted 2 = Rejected
Component Block <PartyRiskLimitsAckGrp>				
1677	NoPartyRiskLimits	Y*	NumInGrp (1)	Number of party risk limits. This value can only be 1.
>1324	ListUpdateAction	Y*	Char	Action to be performed. Valid value: M = Modify
Component Block <PartyDetailGrp>				
>1671	NoPartyDetails	Y*	NumInGrp (1)	Number of party details. The value can only be 1.
>>1691	PartyDetailID	Y*	String (16)	Party identifier. Member mnemonic Risk Group identifier
>>1692	PartyDetailIDSource	Y*	Char	Used to identify source of PartyID value. Valid value: D = Proprietary/Custom
>>1693	PartyDetailRole	Y*	Int	Identifies the type of PartyID. Valid values: '1' Executing Firm = NCM '4' Clearing Member = GCM or ICM '38' Position Account = Risk Group
Component Block <RiskLimitsGrp>				
>1669	NoRiskLimits	Y*	NumInGrp (1)	Number of risk limits for different instrument scopes.
Component Block <RiskLimitTypesGrp>				

Tag	Field Name	Req	Data Type	Description
>>1529	NoRiskLimitTypes	Y*	NumInGrp (1)	Number of risk limits with associated configuration.. This value can only be 1.
>>>1530	RiskLimitType	Y*	Int	Used to specify the type of risk limit. Valid values: 201 = Member Per Order Quantity 202 = Member Per Order Notional Value 203 = Member Gross Short Quantity 204 = Member Gross Long Quantity 205 = Member Net Short Quantity 206 = Member Net Long Quantity
>>>1531	RiskLimitAmount	Y*	Amt	Specifies the risk limit amount. This amount can be a notional value or a quantity.
>>>1532	RiskLimitCurrency	C*	Currency	Specifies the risk limit currency. Conditionally required if RiskLimitAmount (1531) refers to notional value, used when RiskLimitType (1530) = '202' Member Per Order Notional Value. Valid value: USD = US Dollars (default)
>Component Block <RiskInstrumentScopeGrp>		Y*	See RiskInstrumentScope	
>>Component Block <InstrumentScope>		Y*	See RiskInstrument	
End Component Blocks				
58	Text	C*	String (50)	Identifies the reason for rejection. Conditionally required if RiskLimitRequestResult (1761) = '99' Other

Example Message Flow

Modify Risk Limit



4.7.7 Party Risk Limits Definition Request (CS) for MMP Configuration

Used to enter, manage and retrieve Market Marker Protection parameters for a CompID.

Tag	Field Name	Req	Data Type	Description
1666	RiskLimitRequestID	Y	String	Unique identifier for the Party Risk Limits Request.
Component Block <PartyRiskLimitsUpdateGrp>				
1677	NoPartyRiskLimits	Y*	NumInGrp (1)	Number of party risk limits. This value must be 1.
>1324	ListUpdateAction	Y*	Char	Action to be performed. Valid values: A = Add D = Delete M = Modify S = Snapshot
Component Block <PartyDetailGrp>				
>1671	NoPartyDetails	Y*	NumInGrp (1)	Number of party details. The value can only be 1.
>>1691	PartyDetailID	Y*	String (10)	Party identifier. CompID of the Market Maker to which the MMP configuration applies
>>1692	PartyDetailIDSource	Y*	Char	Used to identify source of PartyID value. Valid value: D = Proprietary/Custom (default)
>>1693	PartyDetailRole	Y*	Int	Identifies the type of PartyID. Valid value: 35 = Liquidity Provider
Component Block<RiskLimitsGrp>				
>1669	NoRiskLimits	Y*	NumInGrp (1)	Number of Market Maker Protection types.

Tag	Field Name	Req	Data Type	Description
<RiskLimitTypesGrp>				
>>1529	NoRiskLimitTypes	Y*	NumInGrp (1)	Number of Market Maker Protection types with associated configuration. This value can only be 1.
>>>1530	RiskLimitType	Y*	Int	Market Maker Protection type. Valid values: 301 = Cumulative percent over time 302 = Volume over time 303 = Number of Tradable Instruments traded over time
>>>1531	RiskLimitAmount	C*	Amt	Protection limit for the Market Maker Protection type. Conditional required if LimitUpdateAction (1324) = 'A' Add or 'M' Modify. Not required if LimitUpdateAction (1324) = 'D' Delete or 'S' Snapshot
>>>2336	RiskLimitVelocityPeriod	C*	Int	The timeframe as a rolling window in which the protection type is counted and validated against the protection limit. The time unit of the timeframe is expressed in RiskLimitVelocityUnit (2337). Conditional required if LimitUpdateAction (1324) = 'A' Add or 'M' Modify. Not required if LimitUpdateAction (1324) = 'D' Delete or 'S' Snapshot
>>>2337	RiskLimitVelocityUnit	C*	String (1)	Unit of time in which RiskLimitVelocityPeriod (2336) is expressed. Conditionally required if RiskLimitVelocityPeriod (2336) is present. Valid value: S = Second

Tag	Field Name	Req	Data Type	Description
>Component Block <RiskInstrumentScopeGrp>		Y*	See RiskInstrumentScope	
>>Component Block <InstrumentScope>		Y*	See RiskInstrument	
End Component Blocks				

4.7.8 Party Risk Limits Definition Request Ack (CT) for MMP Configuration

Party Risk Limits Definition Request Ack (35=CT) is used as a response to a Party Risk Limits Definition Request (35=CS) to accept or reject the definition of Market Maker Protection parameters.

Party Risk Limits Definition Request Ack is sent unsolicited in response to a change by the Exchange to an MMP floor.

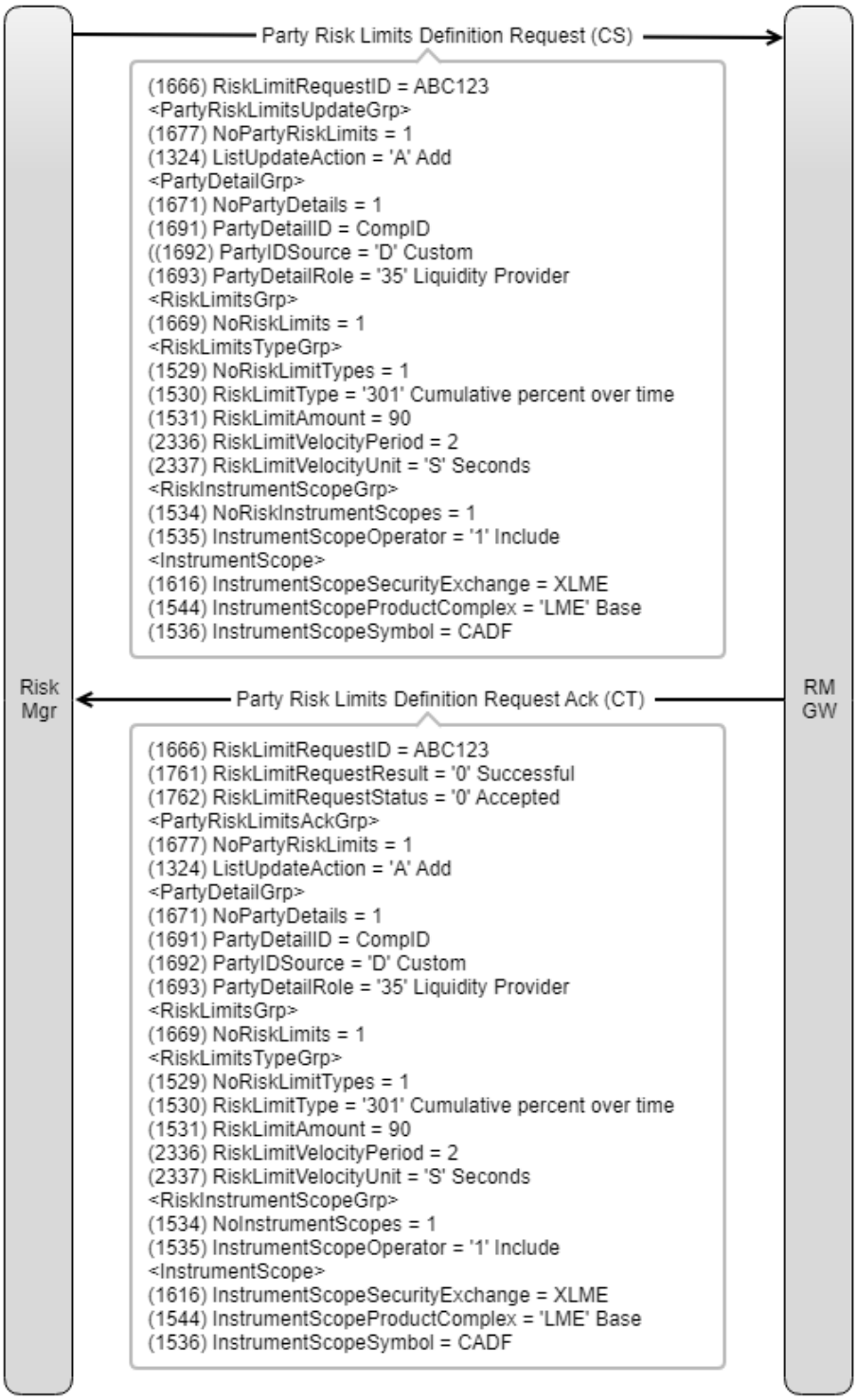
Tag	Field Name	Req	Data Type	Description
1666	RiskLimitRequestID	Y	String	Unique identifier for the Party Risk Limits Request.
1761	RiskLimitRequestResult	Y*	Int	Result of risk limit definition request. Valid values: 0 = Successful 1 = Invalid party 2 = Invalid related party 3 = Invalid risk limit type 5 = Invalid risk limit amount 7 = Invalid risk instrument scope 8 = Risk limit actions not supported 11 = Risk instrument scope not supported 13 = Risk limit already defined for party 98 = Not authorised 99 = Other
1762	RiskLimitRequestStatus	Y	Char	Status of risk limit definition request. Valid values: 0 = Accepted 2 = Rejected
Component Block <PartyRiskLimitsAckGrp>				

Tag	Field Name	Req	Data Type	Description
1677	NoPartyRiskLimits	Y*	NumInGrp (1)	Number of party risk limits. The value can only be 1.
>1324	ListUpdateAction	Y*	Char	Required if NoPartyRiskLimits (1677) > 0 Valid values: A = Add M = Modify D = Delete S = Snapshot
Component Block <PartyDetailGrp>				
>1671	NoPartyDetails	Y*	NumInGrp (1)	Number of party details. The value can only be 1.
>>1691	PartyDetailID	Y*	String (16)	Party identifier. CompID of the Market Maker to which the MMP
>>1692	PartyDetailIDSource	Y*	Char	Used to identify source of PartyID value. Valid value: D = Proprietary/Custom (default)
>>1693	PartyDetailRole	Y*	Int	Identifies the type of PartyID. Valid value: 35 = Liquidity Provider
Component Block <RiskLimitsGrp>				
>1669	NoRiskLimits	Y*	NumInGrp (1)	Number of Market Maker Protection types.
Component Block <RiskLimitTypesGrp>				
>>1529	NoRiskLimitTypes	Y*	NumInGrp (1)	Number of Market Maker Protection types with associated configuration. This value can only be 1.
>>>1530	RiskLimitType	Y*	Int	Market Maker Protection type.

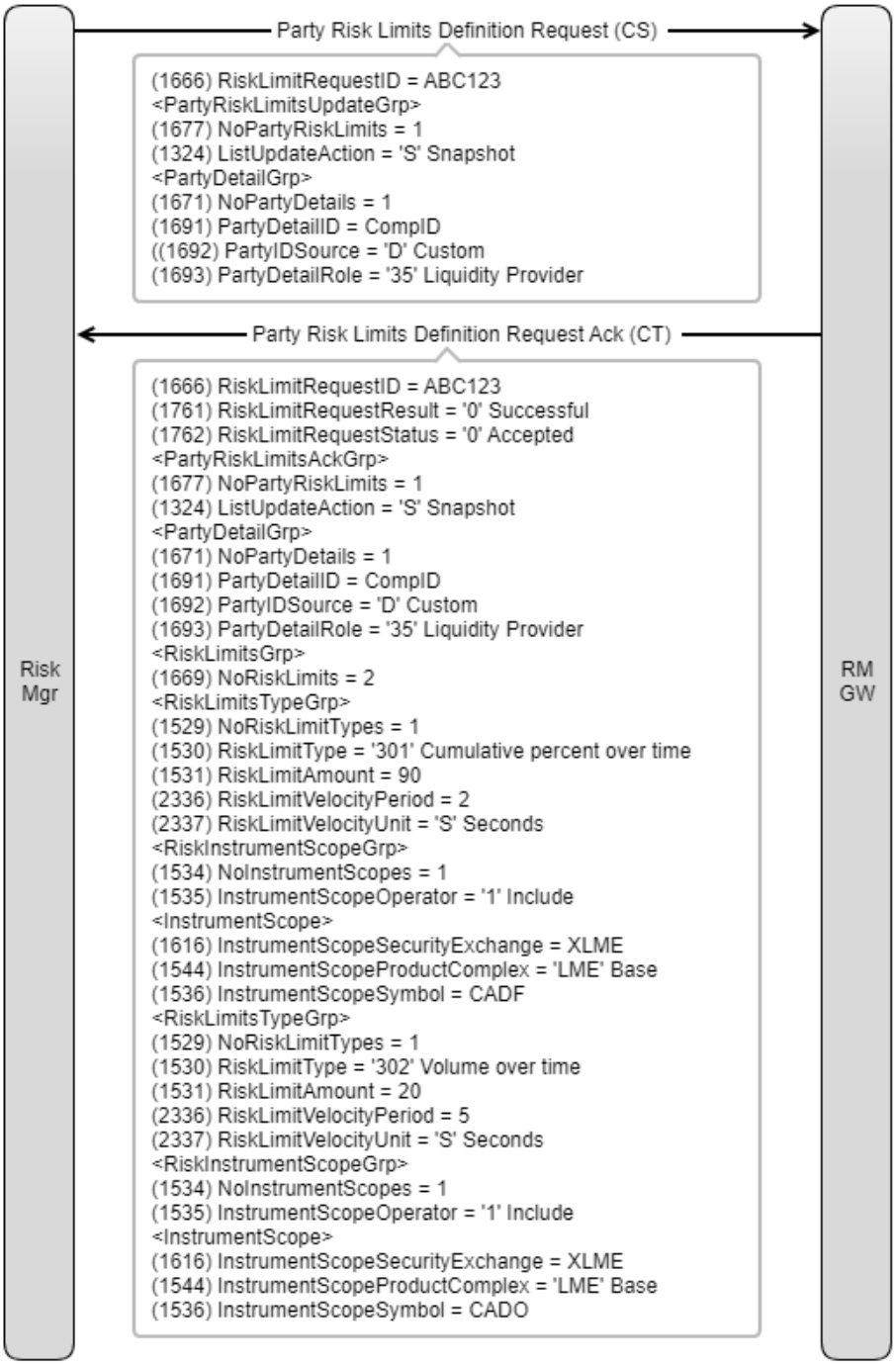
Tag	Field Name	Req	Data Type	Description
				Required if NoRiskLimitTypes (1529) > 0. Valid values: 301 = Cumulative percent over time 302 = Volume over time 303 = Number of Tradable Instruments traded over time
>>>1531	RiskLimitAmount	Y*	Amt	Protection limit for the Market Maker Protection type.
>>>2336	RiskLimitVelocityPeriod	Y*	Int	The timeframe as a rolling window in which the protection type is counted and validated against the protection limit. The time unit of the timeframe is expressed in RiskLimitVelocityUnit (2337).
>>>2337	RiskLimitVelocityUnit	Y*	String (1)	Unit of time in which RiskLimitVelocityPeriod (2336) is expressed. Valid value: S = Second
>Component Block <RiskInstrumentScopeGrp>		Y*	See RiskInstrumentScope	
>>Component Block <InstrumentScope>		Y*	See RiskInstrument	
End Component Blocks				
58	Text	C*	String (50)	Identifies the reason for rejection. Conditionally required if RiskLimitRequestResult (1761) = '99' Other

Example Message Flows

Set Market Maker Protection



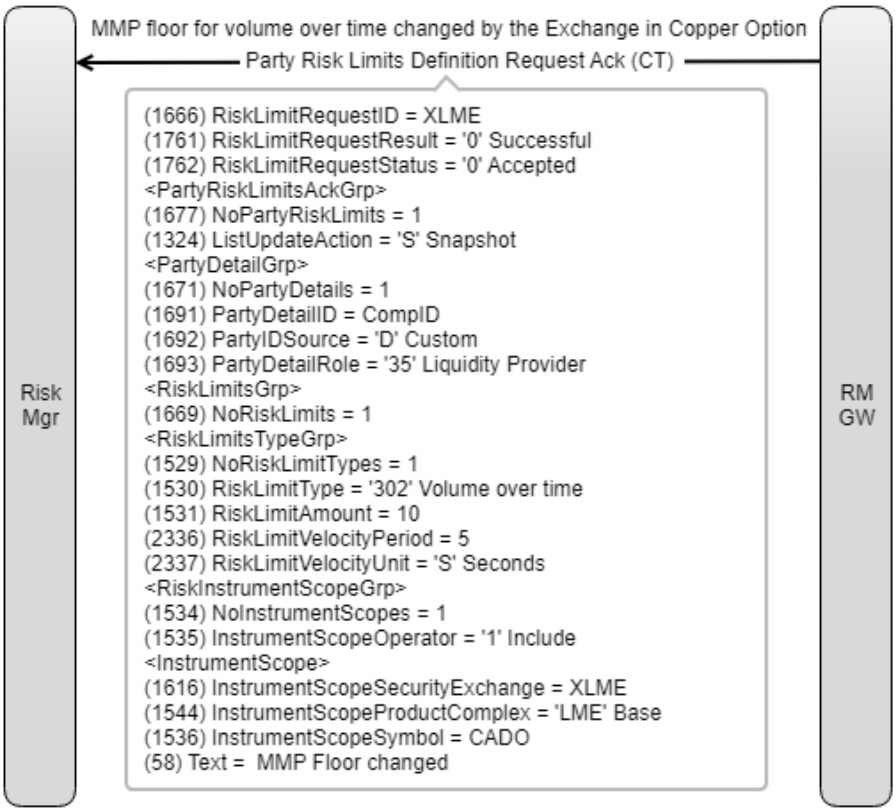
Retrieve Market Maker Protection Parameters



Notification of a change to Market Maker Protection Floors

The Exchange sets the MMP floor for Copper Options to be 20. The Member Risk Manager configures the Volume over time to also be 20 for a time period of 5 seconds.

The Exchange updates the MMP floor setting for Copper Options to be 10 which triggers a notification of the new floor setting to Risk Managers.



4.7.9 Party Risk Limits Request (CL)

Party Risk Limits Request (35=CL) is used to obtain information about risk limits.

The request can be for definitions or current utilisation (consumption) of the risk limits.

Tag	Field Name	Req	Data Type	Description
1666	RiskLimitRequestID	Y	String	Unique identifier for the Party Risk Limits Request.
1760	RiskLimitRequestType	Y*	Char	Type of risk limit information. Valid values: 1 = Definitions 3 = Definitions and utilisations
Component Block <Parties>				
453	NoPartyIDs	C*	NumInGrp (1)	Number of parties specified. Optional for RiskLimitRequestType (1760) = '1' Definitions

Tag	Field Name	Req	Data Type	Description
				Conditionally required for RiskLimitRequestType (1760) = '3' Definitions and utilisations
>448	PartyID	C	String (16)	Party identifier. Conditionally required if NoPartyIDs (453) > 0. Member mnemonic Risk Group identifier
>447	PartyIDSource	C	Char	Source of PartyID value. Conditionally required if NoPartyIDs (453) > 0. Valid value: D = Proprietary/Custom
>452	PartyRole	C	Int	Role of the specified PartyID. Conditionally required if NoPartyID's (453) > 0. Valid values: 1 = Executing Firm (NCM) 4 = Clearing Firm (GCM or ICM) 38 = Position account (for risk group)
End Component Block				
Component Block <RiskInstrumentScopeGrp>		C*	Conditionally required for RiskLimitRequestType (1760) = '3' Definitions and utilisations. See RiskInstrumentScope	
Component Block <InstrumentScope>		C*	Conditionally required for RiskLimitRequestType (1760) = '3' Definitions and utilisations. See RiskInstrument	
End Component Blocks				

4.7.10 Party Risk Limits Report (CM)

Party Risk Limits Report (35=CM) is used to communicate party risk limits and will be sent in response to a Party Risk Limits Request (35=CL) or unsolicited if an order reaches or breaches a risk limit threshold.

If an order simultaneously breaches the lower and upper threshold, only one message is sent. Similarly, if a strategy order breaches both long and short limit types, only one message notifying of the breach will be sent. The message will contain both the percentage utilisation and the upper limit threshold value.

Risk limit threshold levels will be set at 75%, 90% and 100%.

If an order breaches the risk limit, the order will be rejected and notification sent to the order originator in an Execution Report.

Tag	Field Name	Req	Data Type	Description
1667	RiskLimitReportID	Y	String	Identifier for the Party Risk Limits Report
1666	RiskLimitRequestID	C	String	Conditionally required when responding to Party Risk Limits Request.
1760	RiskLimitRequestType	C*	Int	Conditionally required when responding to a Party Risk Limits Request. Scope of risk limit information Valid values: 1 = Definitions (not sent unsolicited) 3 = Definitions and utilisations
1511	RequestResult	C	Int	Conditionally required when responding to Party Risk Limits Request. Valid values: 0 = Valid request 99 = Other
325	UnsolicitedIndicator	C*	Boolean	Conditionally required if.the message is sent unsolicited as an alert. Y = Message is being sent unsolicited.
1512	TotNoParties	Y*	Int	Specifies total number of parties being returned.
893	LastFragment	Y*	Boolean	Indicates whether this message is the last in a sequence of messages. Valid values: N = Not Last Message Y = Last Message

Tag	Field Name	Req	Data Type	Description
Component Block <PartyRiskLimitsGrp>				
1677	NoPartyRiskLimits	Y*	NumInGrp (2)	Number of party risk limits.
Component Block <PartyDetailGrp>				
1671	NoPartyDetails	Y*	NumInGrp (1)	Number of party details. This value can only be 1.
>1691	PartyDetailID	Y*	String (16)	Party identifier. Member Mnemonic Risk Group identifier
>1692	PartyDetailIDSource	Y*	Char	Used to identify source of PartyID value. Valid value: D = Proprietary/Custom (default)
>1693	PartyDetailRole	Y*	Int	Identifies the type of PartyID. Valid values: '1' Executing Firm = NCM '4' Clearing Member = GCM or ICM '38' Position Account = Risk Group
Component Block PartyDetailSubGrp				
1694	NoPartyDetailSubIDs	C*	NumInGrp (1)	Number of party detail sub-identifiers. Conditionally required when reporting a warning or a breach affecting an end client.
>1695	PartyDetailSubID	C	String (16)	Sub-identifier for the party specified in PartyDetailID (1691). Conditionally required when NoPartyDetailSubIDs (1694) > 0. End Client Identifier.
>1696	PartyDetailISubIDType	C	Int	Type of PartyDetailSubID (1695) value.

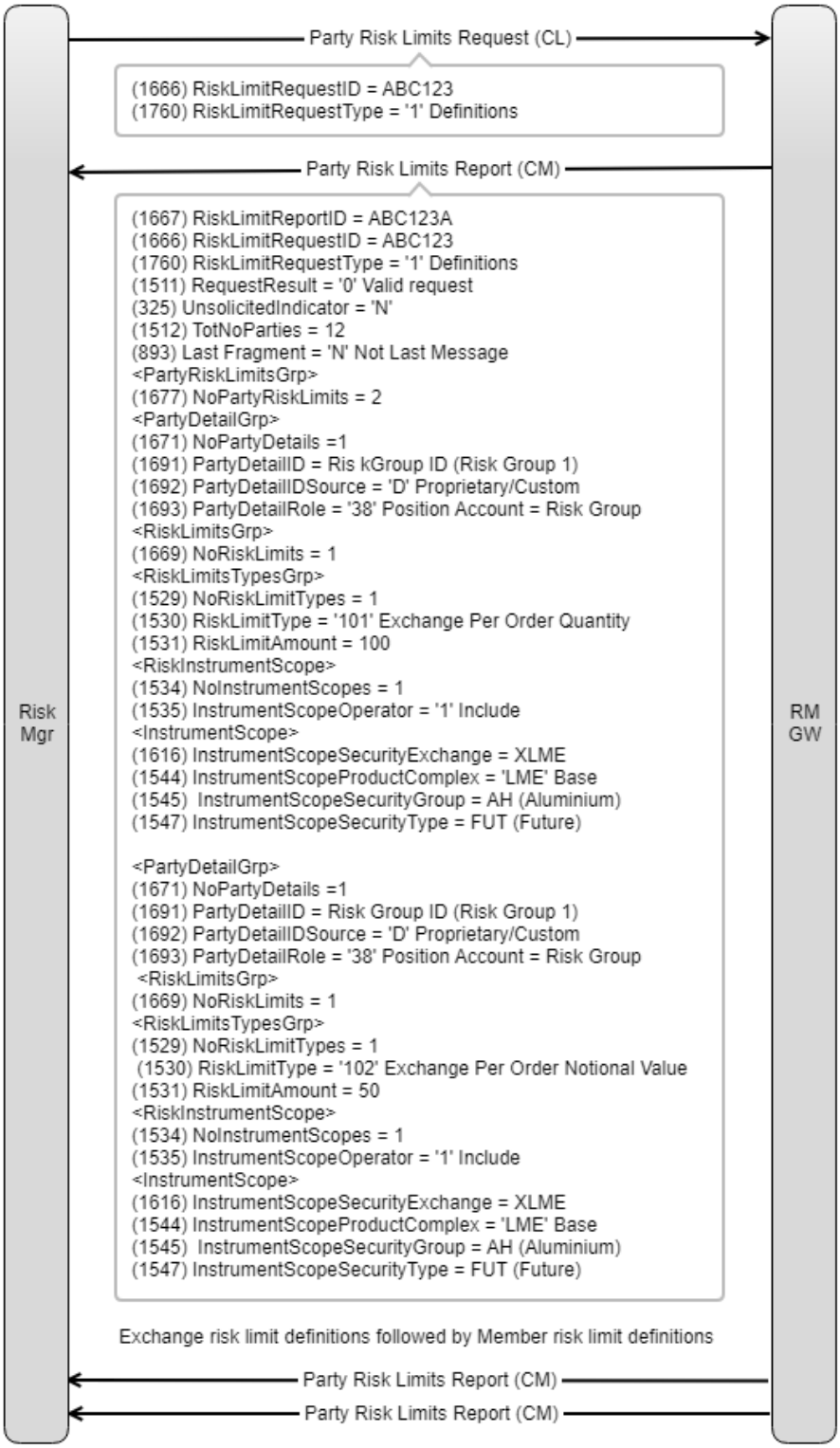
Tag	Field Name	Req	Data Type	Description
				Conditionally required when NoPartyDetailSubIDs (1694) > 0. Valid value: '81' Broker Client ID = End Client
Component Block <RiskLimitsGrp>				
1669	NoRiskLimits	Y*	NumInGrp (1)	Number of risk limits for different instrument scopes. This value can only be 1.
Component Block <RiskLimitTypesGrp>				
1529	NoRiskLimitTypes	Y*	NumInGrp (1)	Number of risk limits with associated warning levels. This value can only be 1.
>1530	RiskLimitType	Y*	Int	Used to specify the type of risk limit. Valid values: 101 = Exchange Per Order Quantity 102 = Exchange Per Order Notional Value 103 = Exchange Gross Short Quantity 104 = Exchange Gross Long Quantity 105 = Exchange Net Short Quantity 106 = Exchange Net Long Quantity 201 = Member Per Order Quantity 202 = Member Per Order Notional Value 203 = Member Gross Short Quantity 204 = Member Gross Long Quantity 205 = Member Net Short Quantity

Tag	Field Name	Req	Data Type	Description
				206 = Member Net Long Quantity
>1531	RiskLimitAmount	Y*	Amt	Specifies the risk limit amount. This amount can be a notional value or a quantity.
>1532	RiskLimitCurrency	C*	Currency	Specifies the risk limit currency. Conditionally required if RiskLimitAmount (1531) refers to notional value. Used when RiskLimitType (1530) is either '102' Exchange Per Order Notional Value or '202' Member Per Order Notional Value.
>1767	RiskLimitAction	C*	Int	Identifies the action to take or risk model to assume should risk limit be exceeded or breached for the specified party. Valid values: 2 = Reject 4 = Warning Conditionally required when reporting a warning or a breach.
>1765	RiskLimitUtilizationPercent	C*	Percentage	Percentage of utilisation of a party's set risk limit. Conditionally required for utilisations and when reporting a warning or a breach
Component Block RiskWarningLevelGrp				
>1559	NoRiskWarningLevels	C*	NumInGrp (1)	Number of risk warning levels. Value can only be 1. Conditionally required if RiskLimitAction (1767) = '4' Warning
>>1769	RiskWarningLevelAction	C	Int	Action to take should warning level be exceeded. Valid value:

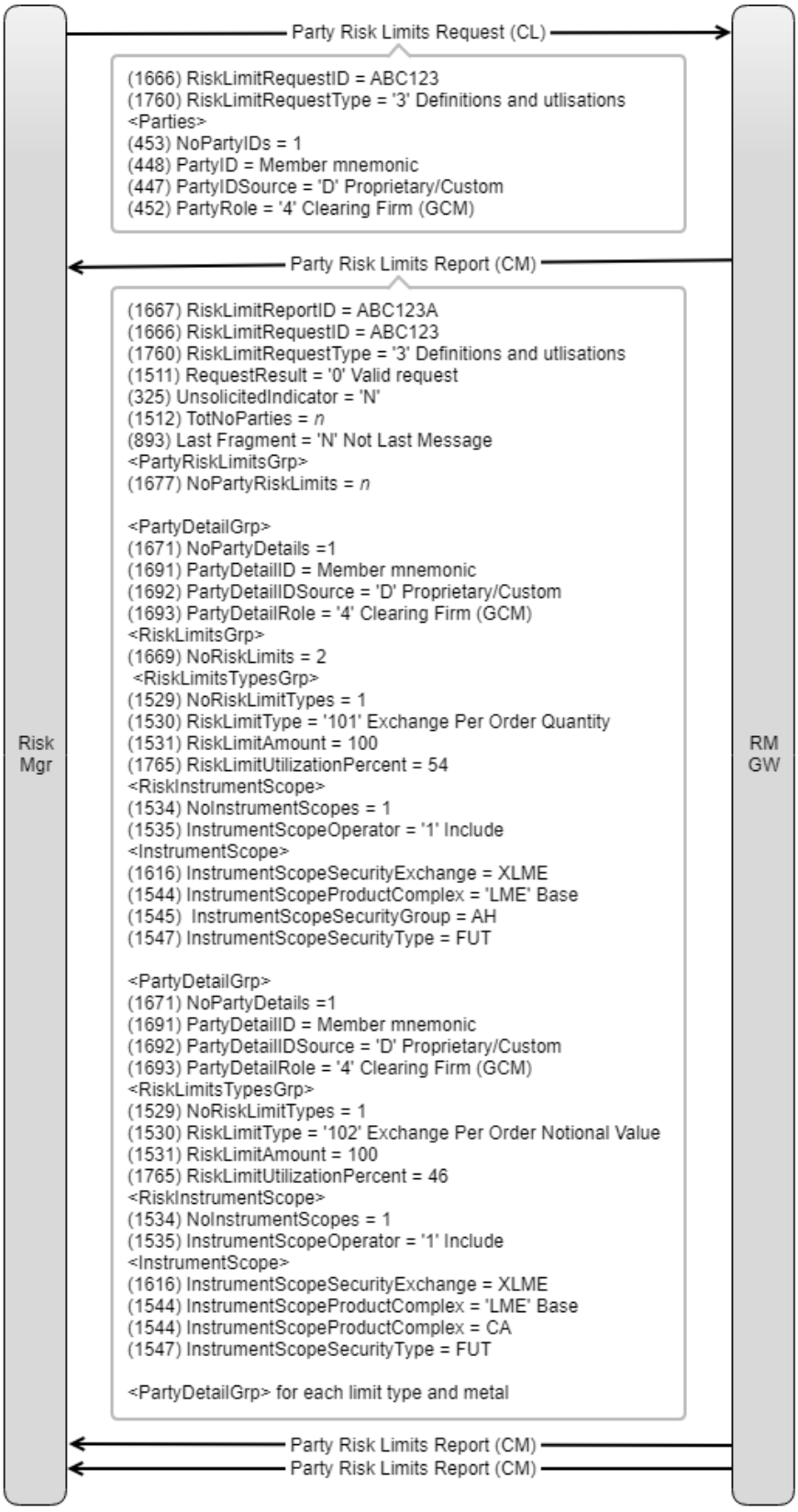
Tag	Field Name	Req	Data Type	Description
				4 = Warning Conditionally required if NoRiskWarningLevels (1559) > 0.
>>1560	RiskWarningLevelPercent	C	Percentage	Conditionally required if NoRiskWarningLevels (1559) > 0.
>>1561	RiskWarningLevelName	C*	String (1)	Name or error message associated with the risk warning level. Valid values: 1 = Warning Amber Breach 2 = Warning Red Breach 3 = Warning Limit Reached Conditionally required if NoRiskWarningLevels (1559) > 0.
Component Block <RiskInstrumentScopeGrp>		Y*	See RiskInstrumentScope	
Component Block <InstrumentScope>		Y*	See RiskInstrument	
End Component Blocks				
58	Text	C*	String (50)	Identifier of the order that breached the limit. Conditionally required if RiskLimitAction (1767) = '2' Reject
1328	RejectText	C*	String (75)	Identifies the reason for rejection. Conditionally required if RequestResult (1511) = '99' Other
60	TransactTime	C*	UTCTimestamp	Timestamp when the message was generated. Conditionally required when RiskLimitAction (1767) is specified.

Example Message Flows

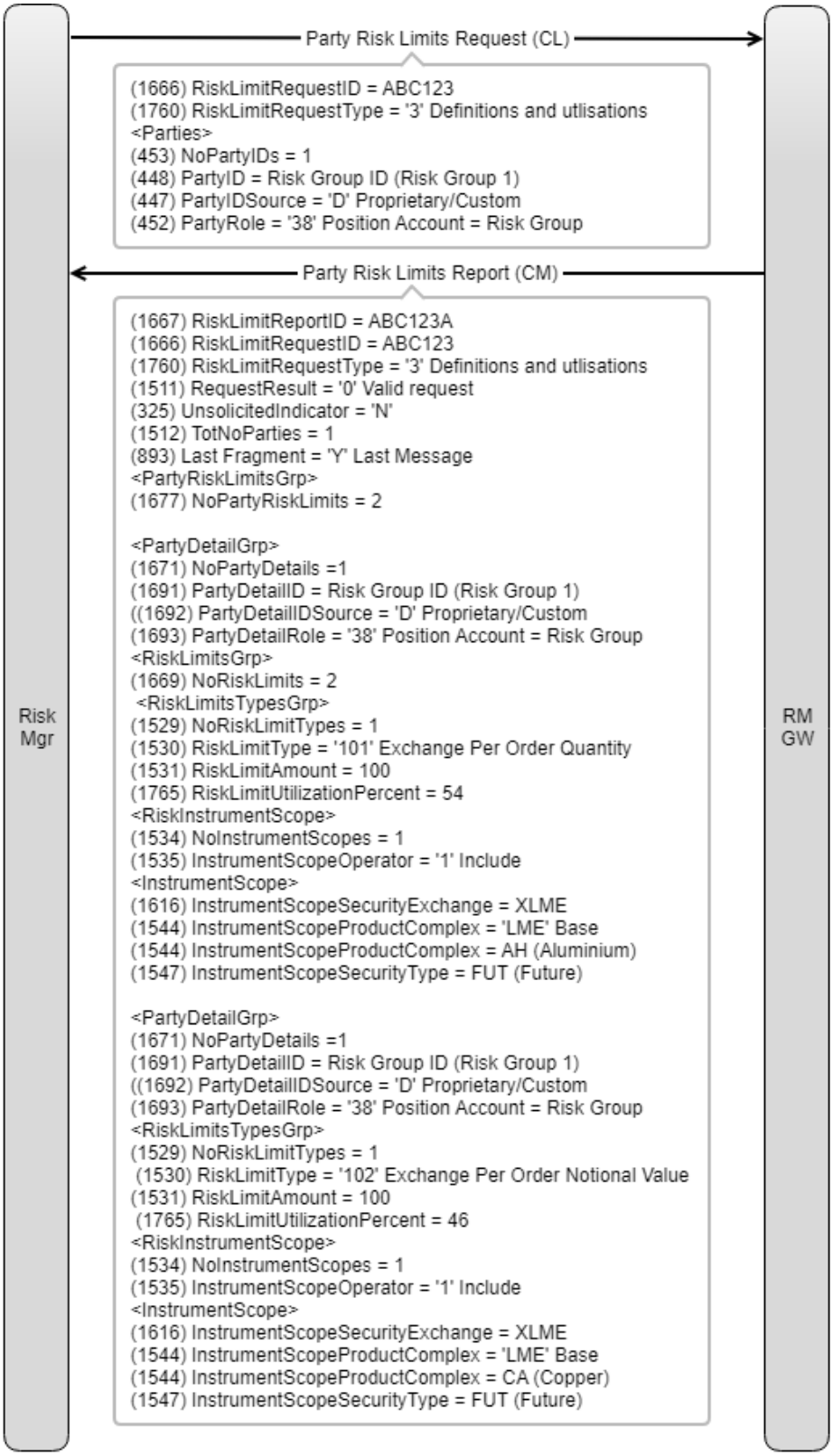
Snapshot of Risk Limit Definitions



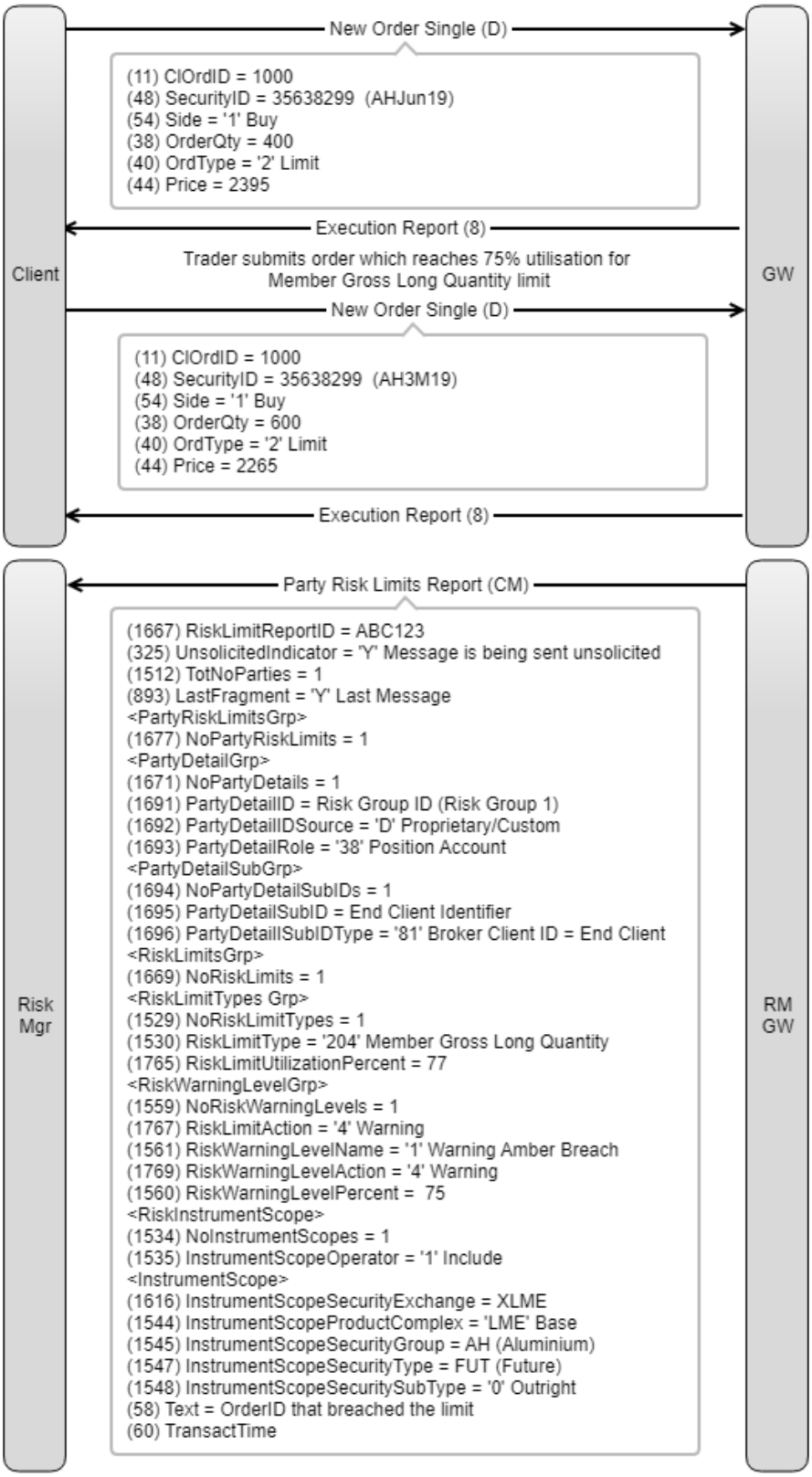
Snapshot of Risk Limit Definitions and Utilisations for a GCM at Member level



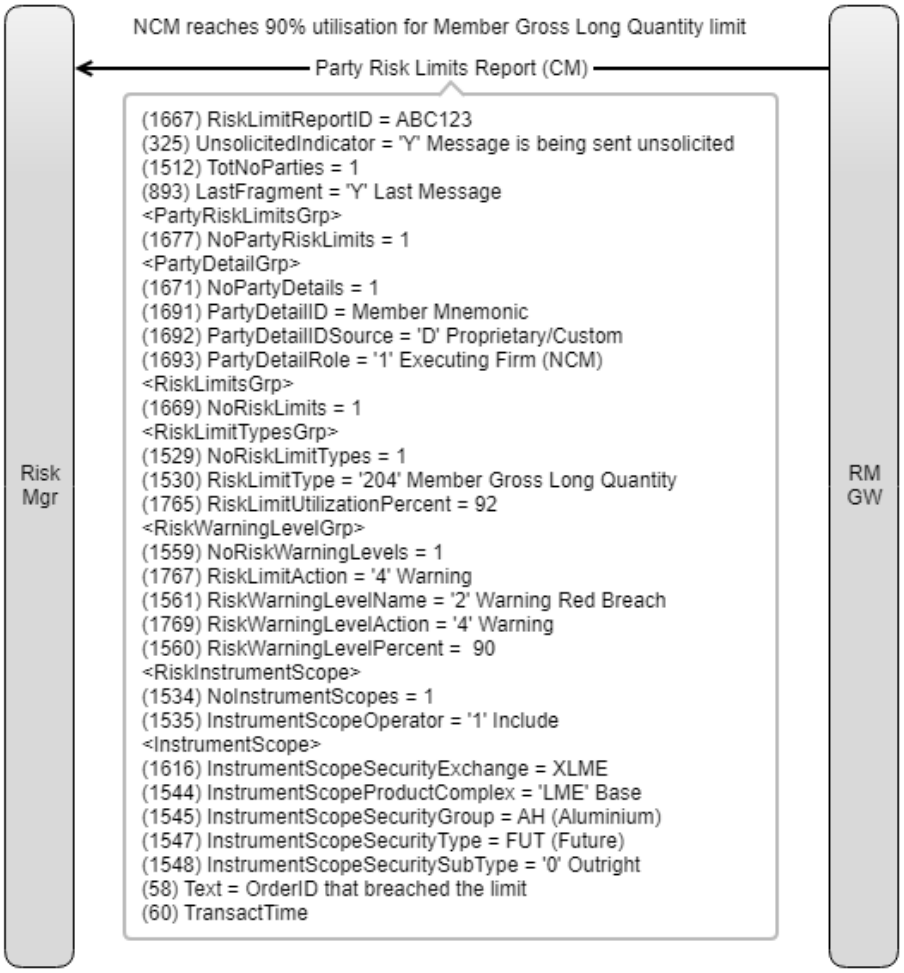
Snapshot of Risk Limit Definitions and Utilisations for a Risk Group



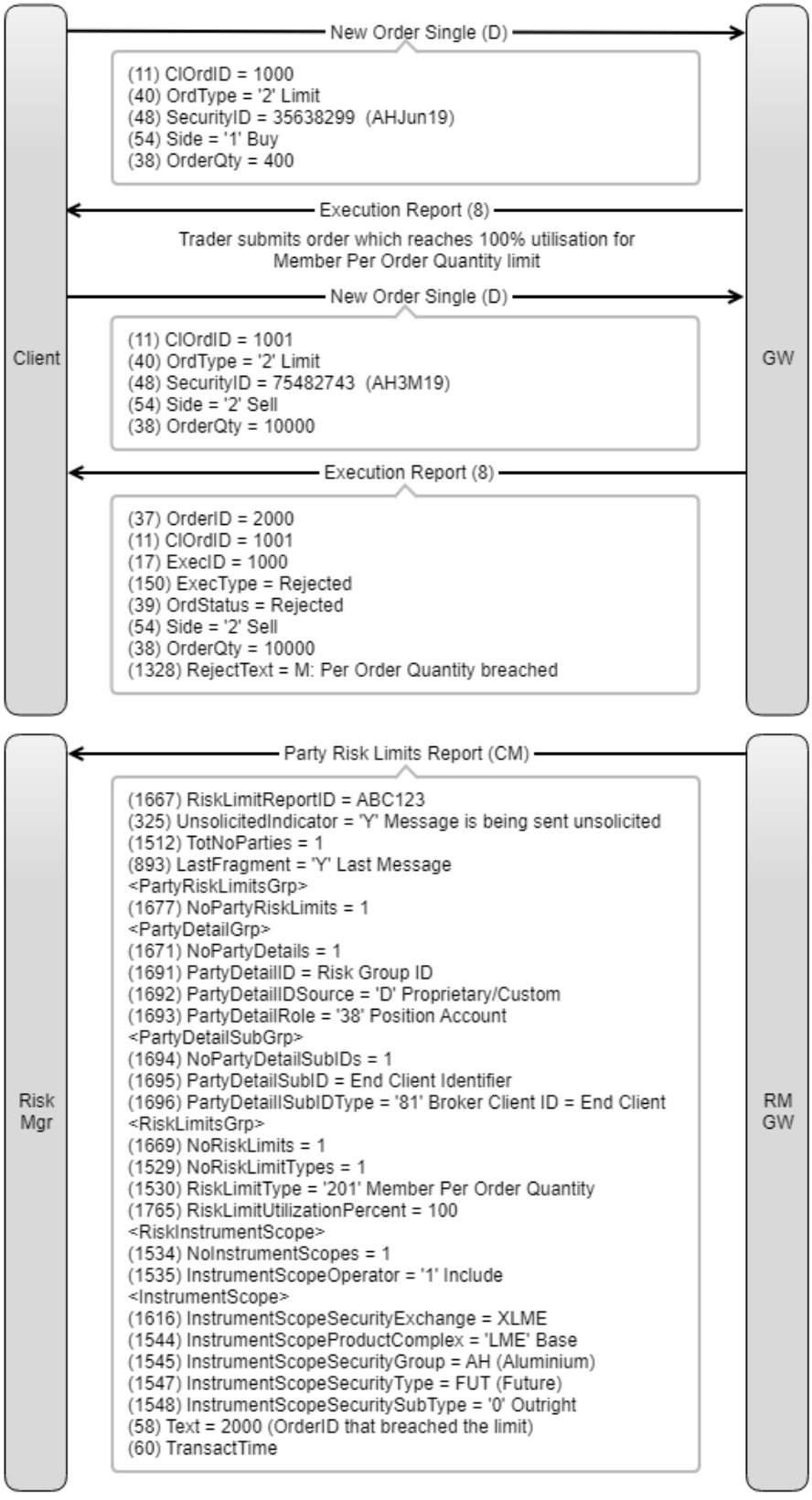
Risk Limit Breach Warning Notification at Risk Group level



Risk Limit Breach Warning Notification at Member level



Limit Breach Notification



4.7.11 Party Action Request (DH)

Party Action Request (35=DH) is used to suspend or halt the specified party from further trading activities i.e. it applies the 'kill switch'. Refer to the table in 3.5 Kill Switch for usage of the initiating PartyRole and resulting application target level. A Party Action Report (35=DI) message is sent in response.

Tag	Field Name	Req	Data Type	Description
2328	PartyActionRequestID	Y	String	Unique identifier for the Party Action Request.
2329	PartyActionType	Y	Int	Specifies the type of action to take or was taken for a given party. Valid values: 0 = Suspend (i.e. stop accepting further orders) 1 = Halt trading (i.e. kill switch - stop accepting further orders and pull all orders) 2 = Reinstate (i.e. re-enable trading)
Component Block <Parties>				
453	NoPartyIDs	Y	NumInGrp (1)	Number of parties specified. The value can only be 1.
>448	PartyID	Y	String (3)	Party identifier. Member mnemonic
>447	PartyIDSource	Y	Char	Source of PartyID value. Valid value: D = Proprietary/Custom
>452	PartyRole	Y	Int	Role of the specified PartyID. Valid values: 4 = Clearing Firm (GCM) 118 = Operator (party performing the action e.g. self)
End Component Block				
Component Block <RelatedPartyDetailGrp>				
1562	NoRelatedPartyDetailID	N	NumInGrp (1)	Number of related party detail identifiers. This value can only be 1.

Tag	Field Name	Req	Data Type	Description
				Optionally to specify the target of the instruction.
>1563	RelatedPartyDetailID	C	String (16)	Party identifier for the party related to the party specified. Conditionally required if NoRelatedPartyDetails (1562) > 0. Member mnemonic Risk Group identifier End Client identifier
>1564	RelatedPartyDetailIDSource	C	Char	Identifies the source of the RelatedPartyDetailID (1563). Conditionally required if NoRelatedPartyDetails (1562) > 0. Valid value: D = Proprietary/Custom
>1565	RelatedPartyDetailRole	C	Int	Identifies the type or role of the RelatedPartyDetailID (1563) specified. Conditionally required if NoRelatedPartyDetails (1562) > 0. Valid values: 1 = Executing Firm (NCM) 38 = Position account (for risk group) 81 = Broker Client ID (for end client) 118 = Operator (party performing the action e.g. self)
End Component Block				
Component Block <PartyRelationshipGrp>				
>1514	NoPartyRelationships	N	NumInGrp (1)	Number of party relationships. The value can only be 1. Only applicable if PartyActionType (2329) = ‘2’ Reinstate
>1515	PartyRelationship	C	Int	Identifies the type of party relationship requested. Conditionally required if NoPartyRelationships (1514) > 0. Valid value:

Tag	Field Name	Req	Data Type	Description
				4001 = Include lower levels
End Component Block				

4.7.12 Party Action Report (DI)

Party Action Report (35=DI) is used to respond to the Party Action Request (35=DH) message indicating whether the request has been received, accepted or rejected. This message can also be used in an unsolicited manner to report party actions initiated by Market Operations.

Tag	Field Name	Req	Data Type	Description
2328	PartyActionRequestID	C	String	Unique identifier of the Party Action Request. Conditionally required when responding to a Party Action Request.
2331	PartyActionReportID	Y	String	Unique identifier of the Party Action Report as assigned by the message sender.
2329	PartyActionType	Y	Int	Specifies the type of action to take or was taken for a given party. Valid values: 0 = Suspend (i.e. stop accepting further orders) 1 = Halt trading (i.e. kill switch - stop accepting further orders and pull all orders) 2 = Reinstate (i.e. re-enable trading)
2332	PartyActionResponse	Y	Int	Specifies the action taken as a result of the PartyActionType (2239) of the Party Action Request (DH) message. Valid values: 0 = Accepted 1 = Completed 2 = Rejected

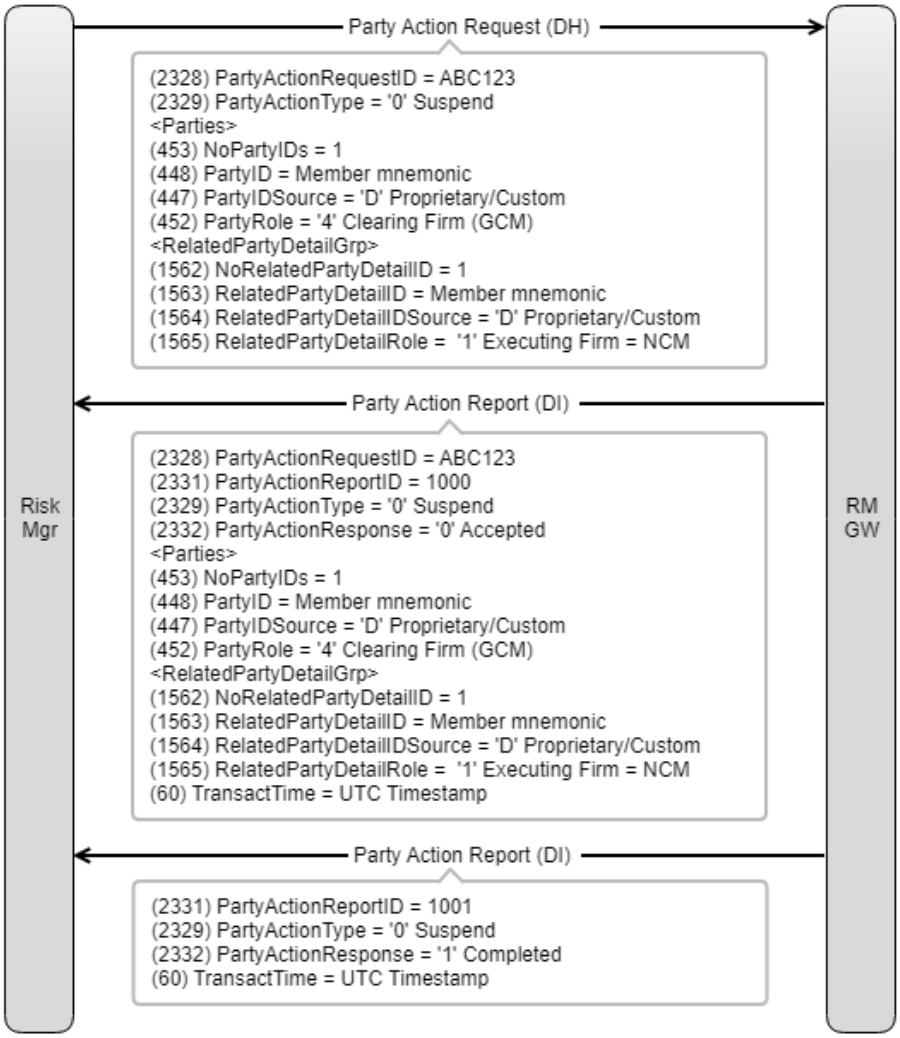
Tag	Field Name	Req	Data Type	Description
2333	PartyActionRejectReason	C	Int	Conditionally required when PartyActionResponse (2332) = '2' Rejected. Valid values: 0 = Invalid party or parties 98 = Not authorised 99 = Other
1328	RejectText	C*	String (75)	Identifies the reason for rejection. Conditionally required if PartyActionRejectReason (2333) = '99' Other.
Component Block <Parties>				
453	NoPartyIDs	Y	NumInGrp (1)	Number of parties specified. The value can only be 1.
>448	PartyID	Y	String (4)	Party identifier.
>447	PartyIDSource	Y	Char	Source of PartyID value. Valid value: D = Proprietary/Custom G = Market Identifier Code (MIC)
>452	PartyRole	Y	Int	Role of the specified PartyID (448). Valid values: 4 = Clearing Firm (GCM or ICM) 22 = Exchange (XLME) 118 = Operator (party performing the action e.g. self)
End Component Block				
Component Block <RelatedPartyDetailGrp>				
1562	NoRelatedPartyDetailID	N	NumInGrp (1)	Number of related party detail identifiers. This value can only be 1.

Tag	Field Name	Req	Data Type	Description
				Optionally to specify the target of the instruction.
>1563	RelatedPartyDetailID	C	String (16)	Party identifier for the party related to the party specified. Conditionally required if NoRelatedPartyDetails (1562) > 0.
>1564	RelatedPartyDetailIDSource	C	Char	Identifies the source of the RelatedPartyDetailID (1563). Conditionally required if NoRelatedPartyDetails (1562) > 0. Valid value: D = Proprietary/Custom
>1565	RelatedPartyDetailRole	C	Int	Identifies the type or role of the RelatedPartyDetailID (1563) specified. Conditionally required if NoRelatedPartyDetails (1562) > 0. Valid values: 1 = Executing Firm (NCM) 4 = Clearing Firm (GCM) 38 = Position account (for risk group) 81 = Broker Client ID (for end client) 118 = Operator (party performing the action e.g. self)
End Component Block				
Component Block <PartyRelationshipGrp>				
>1514	NoPartyRelationships	N	NumInGrp (1)	Number of party relationships. The value can only be 1. Only applicable if PartyActionType (2329) = '2' Reinstate
>1515	PartyRelationship	C	Int	Identifies the type of party relationship requested.

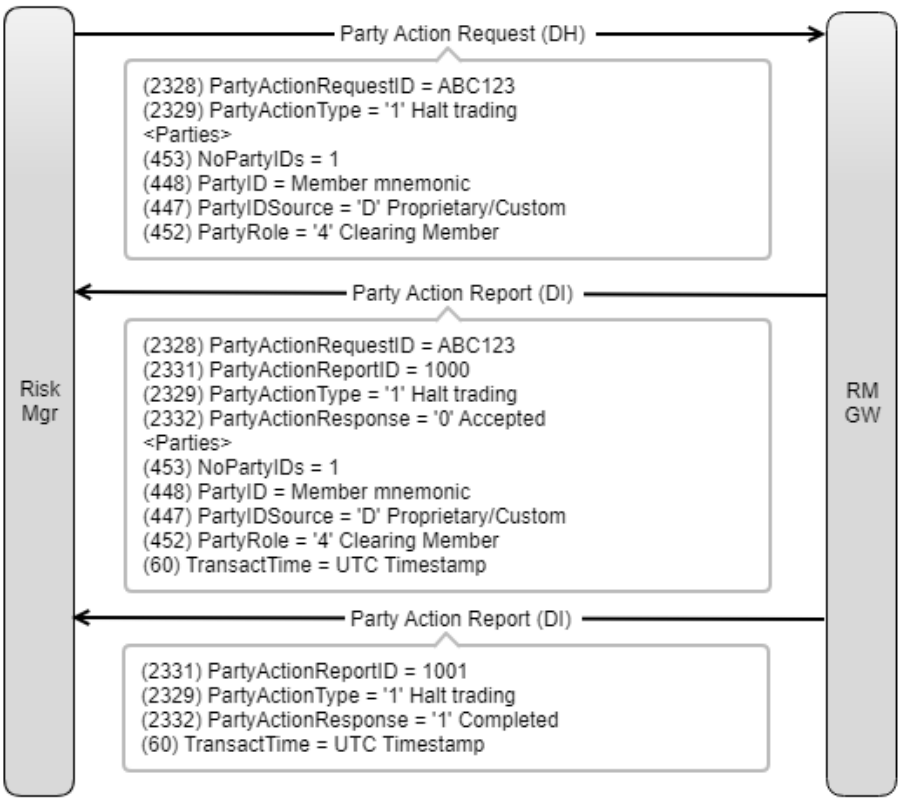
Tag	Field Name	Req	Data Type	Description
				Conditionally required if NoPartyRelationships (1514) > 0. Valid value: 4001 = Include lower levels
End Component Blocks				
60	TransactTime	Y*	UTCTimestamp	Timestamp when the message was generated.

Example Message Flows

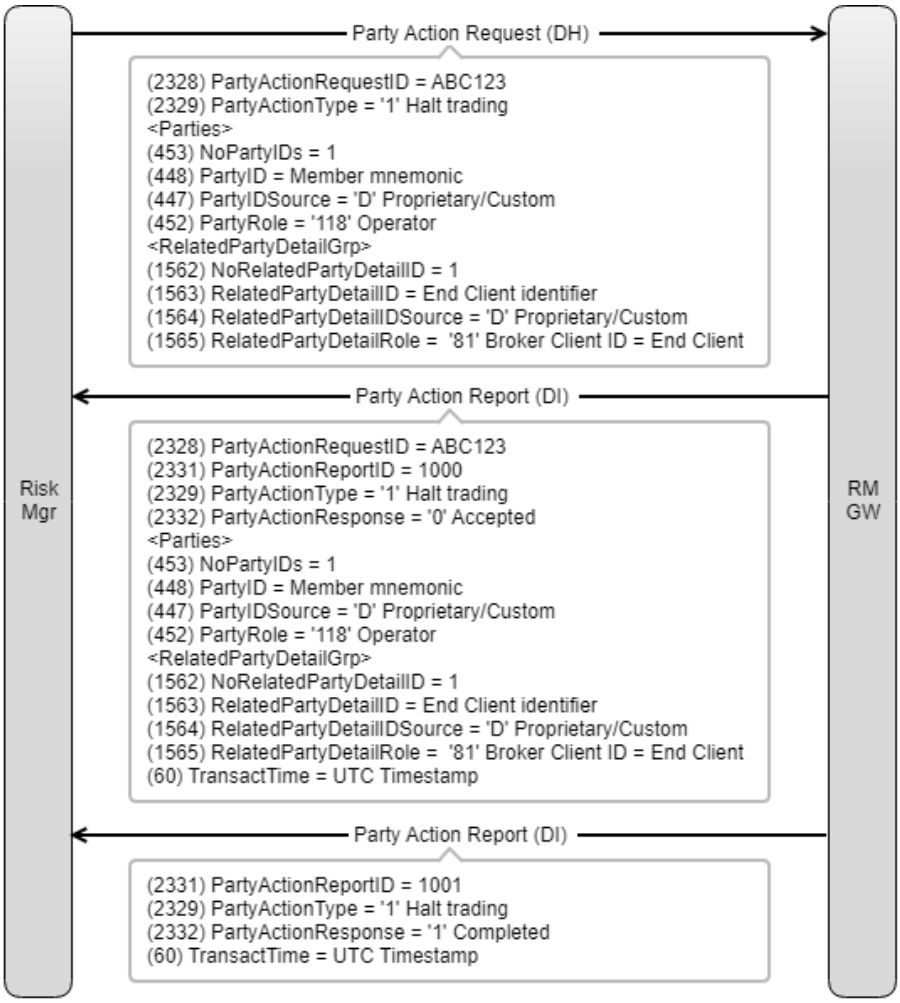
Kill Switch applied by GCM to suspend an NCM



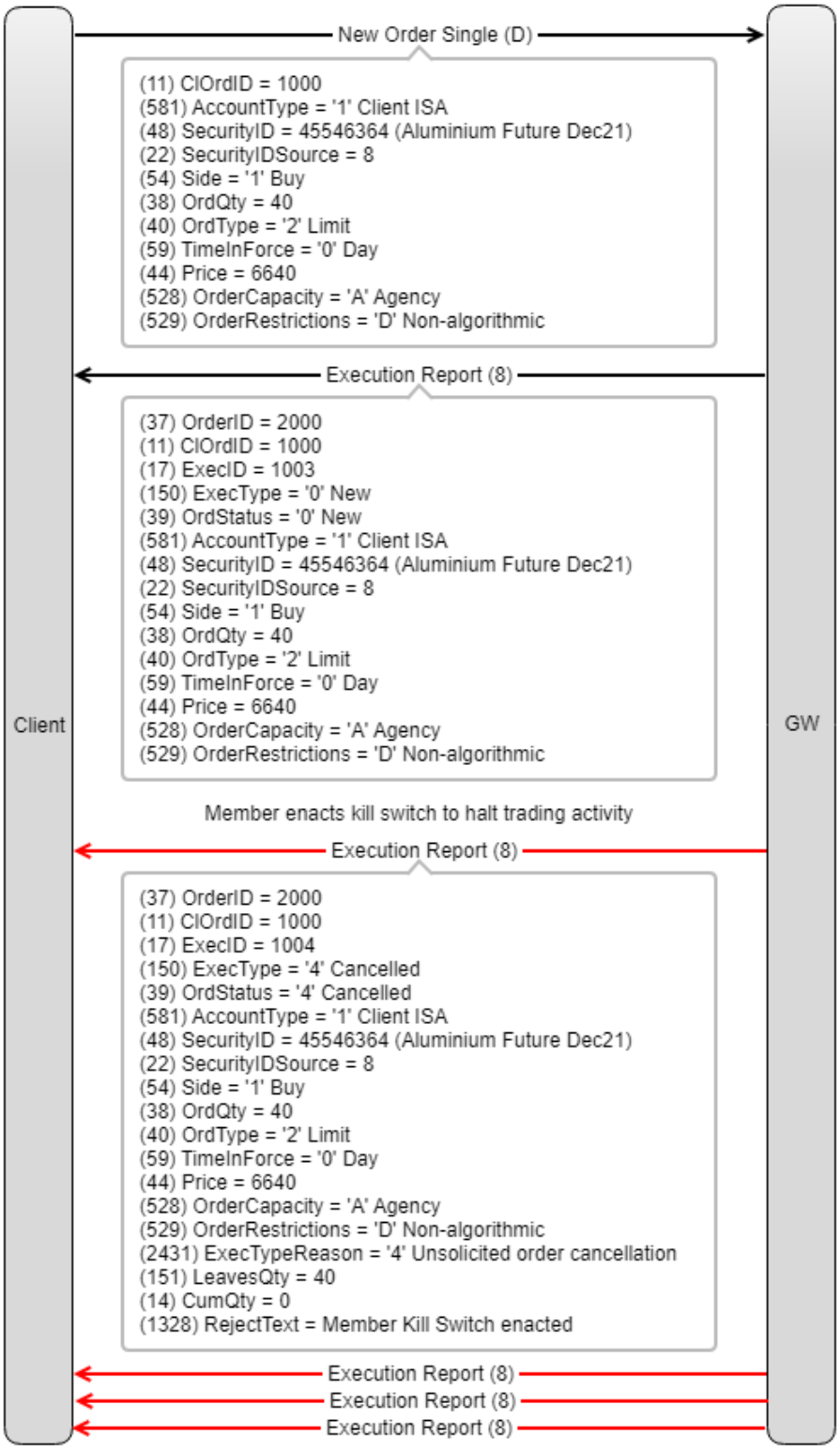
Kill Switch applied by GCM on themselves and their related entities



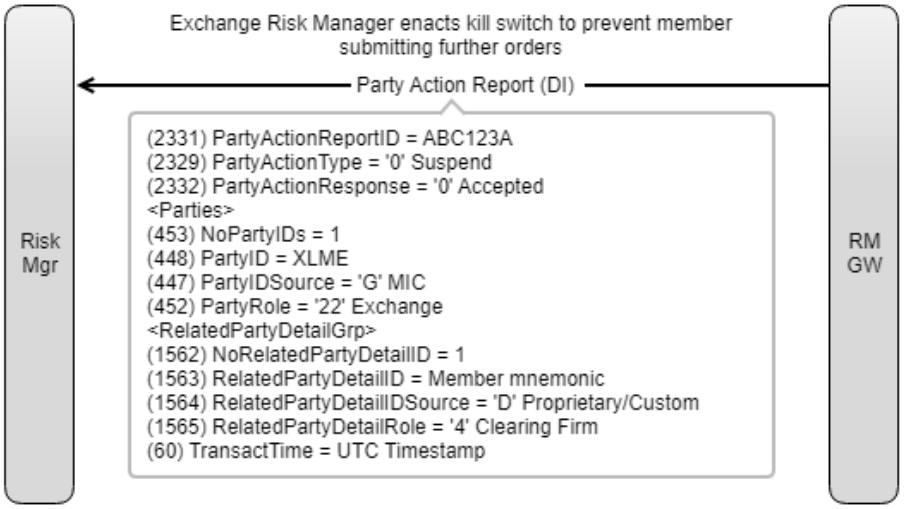
Kill Switch applied on End Client



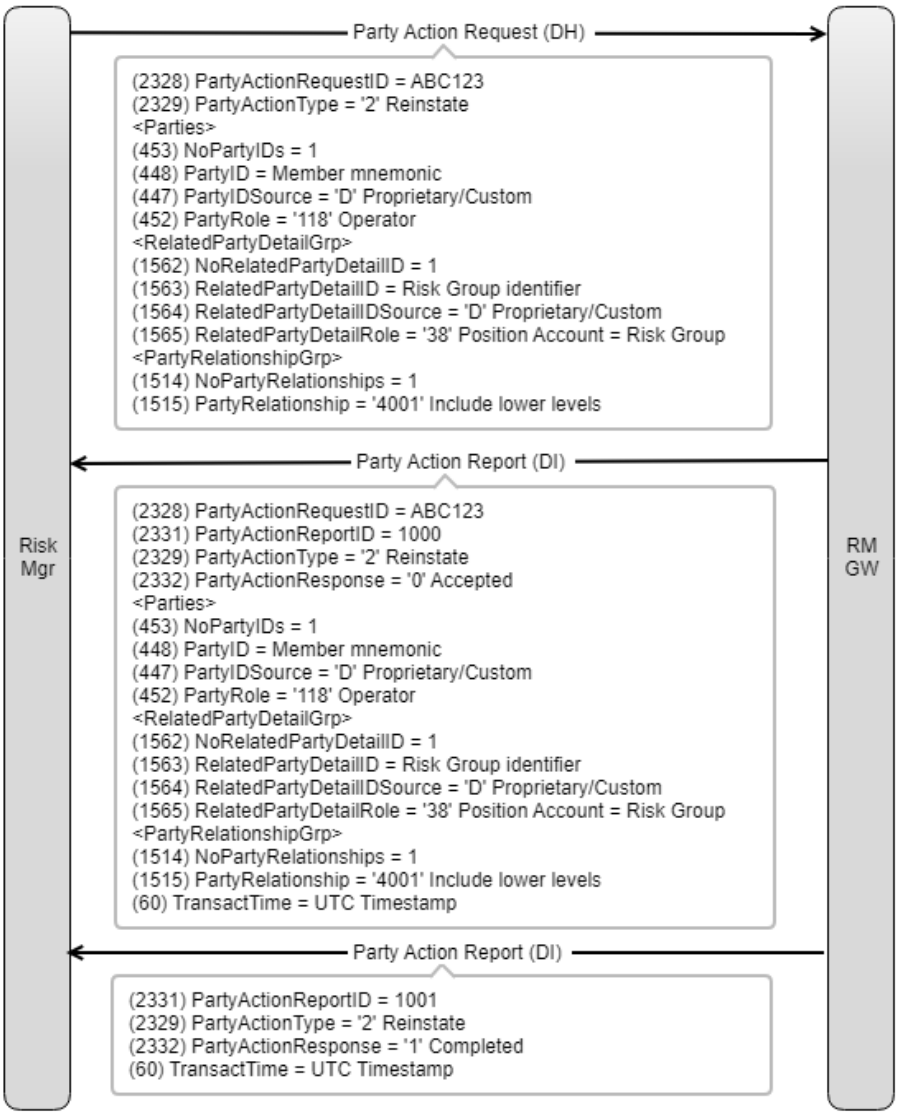
Orders cancelled as a result of Kill Switch applied by Member Risk Manager



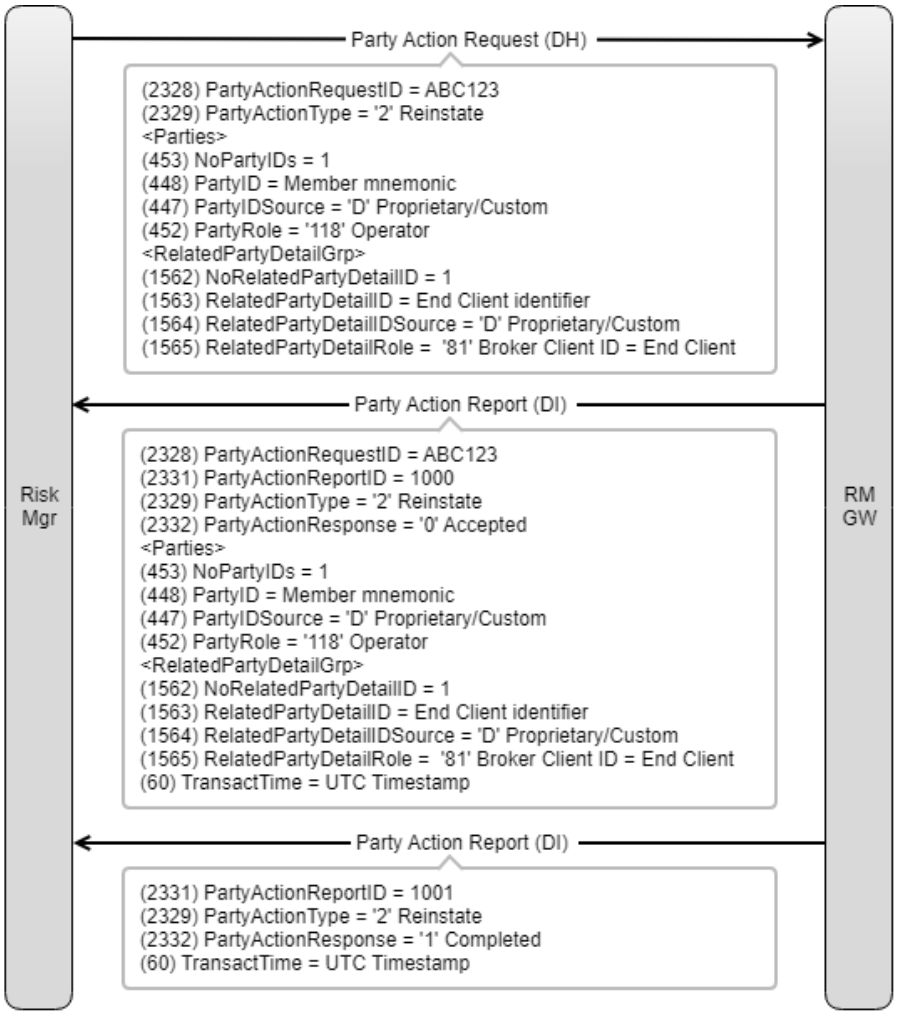
Notification to Member Risk Manager as a result of Kill Switch applied by Exchange



Reinstate Risk Group and End Clients



Reinstate End Client



4.7.13 Party Entitlements Definition Request (DA)

Party Entitlements Definition Request (35=DA) is used to enter the Self Execution Prevention identifier that the Member's traders will specify in SelfMatchPreventionID (2362) for orders and quotes. It also enables the Member to specify the Self Execution response when a SelfMatchPreventionID (2362) is matched.

Tag	Field Name	Req	Data Type	Description
1770	EntitlementRequestID	Y	String	Unique identifier for Party Entitlements Definition Request.
Component Block <PartyEntitlementUpdateGrp>				
1772	NoPartyEntitlements	Y*	NumInGrp (1)	Number of party entitlement values. Can only be 1.

Tag	Field Name	Req	Data Type	Description
1324	ListUpdateAction	Y	Char	Action to be performed. Valid values: A = Add M = Modify D = Delete
Component Block <EntitlementGrp>				
1773	NoEntitlements	Y*	NumInGrp (1)	Number of entitlement values.
>1774	EntitlementIndicator	Y	Boolean	Used to indicate that the entitlement includes attributes. Required by FIX in the message but will be ignored. Valid value: True
Component Block <EntitlementAttribGrp>				
>1777	NoEntitlementAttrib	Y*	NumInGrp (1)	Number of entitlement attributes. Must be 2 for Add or Modify. For Delete only EntitlementAttribType (1778) = '4001' SEP Match ID and EntitlementAttribValue (1780) = SEP Match ID value needs to be specified
>>1778	EntitlementAttribType	Y*	Int	Name of the entitlement attribute type. Must be in following order. 4001 = SEP Match ID 4002 = SEP Response
>>1779	EntitlementAttribDataType	Y*	Int	Datatype of the entitlement attribute. 1 = int
>>1780	EntitlementAttribValue	Y*	String (9)	Value of the entitlement attribute. Self Execution response when SelfMatchPreventionID (2362) is matched.

Tag	Field Name	Req	Data Type	Description
				SEP Match ID value when EntitlementAttribType (1778) = 4001 e.g. 123456789 SEP Response value when EntitlementAttribType (1778) = 4002 either: Valid values: 1 = Cancel incoming order 2 = Cancel resting order
End Component Blocks				

4.7.14 Party Entitlements Definition Request Ack (DB)

Party Entitlements Definition Request Ack (35=DB) is used as a response to the Party Entitlements Definition Request (35=DA) to accept or reject the definition of the party entitlement.

Tag	Field Name	Req	Data Type	Description
1770	EntitlementRequestID	Y	String	Unique identifier for Party Entitlements Definition Request (DA).
1882	EntitlementRequestStatus	Y	Int	Status of Party Entitlements Definition Request. Valid values: 0 = Successful 2 = Rejected
1881	EntitlementRequestResult	Y	Int	Result of the Party Entitlements Definition Request. Valid values: 0 = Successful 4 = Invalid entitlement ID 13 = Entitlement already defined for party 98 = Not Authorised 99 = Other
58	Text	C*	String (50)	Identifies the reason for rejection. Conditionally required if EntitlementRequestStatus (1882) = '99' Other

Example Message Flow

Add Self Execution Prevention Parameters