THOMSON REUTERS MATCHING API

THOMSON REUTERS MATCHING FIX INTERFACE USER GUIDE

Date of issue: 30 June 2015

THOMSON REUTERS

L

Important	This is an internal Thomson Reuters document and unless otherwise stated does not comprise any contractual obligation. It is subject to modification and all affected parties will be notified of any material changes.

Contents

About this document

Intended readership

All technical staff who are responsible for integrating and supporting their trading engines with the Thomson Reuters Matching FIX Interface.

In this document

This document defines the FIX Rules of Engagement (ROE) for the Thomson Reuters Matching FIX Interface.

What's new in this document?

- In version 1.5.5

Section 3.4.9	New order type of Iceberg added
Section 3.4.13	Locked order functionality added
Section 3.4.14	Jobbing mode now available on IOCs
Section 4.11.2	Bell notification added to remind party of unanswered proposals
Appendix E- section6	Timeout of incomplete FX Swap Workflows
Appendix H	Forward Bell Workflow

- In version 1.0.21

Section 2.1	FIX version number has been updated.
Section 2.2.2.1	New User type with credit administration permissions defined.
Section 3.6.2.7	Timed Orders has been updated
Section 3.6.8	Updated to reflect action which may be taken when throttle limits are breached.
Section 3.6.10	Maximum Order Size, updated to reflect MOS can be set for all Matching Site Ids (credit codes).
Section 3.6.11	New Section on Maximum Open Orders.
Section 3.6.13	Updated to reflect the new Forward Swap workflow support in NMH v1.4.
Section 3.6.8.4	Breaching Throttle Limits
Section 3.6.10	Maximum Order Size update
Section 3.6.11	Maximum Open Orders Per Instrument added
Section 3.6.13	Soft Match Updated for proposal negotiation
Section 4.1	TAG 1156 AppIExtID is Action changed to Ignored
Section 4.4	TAG 379 updated
Sections 4.5.1.1/2	TAG 1028 Note *This field is mandatory for the HTG and will become mandatory for production at a future time to be announced.*
Section 4.5.4.4	Tag 39 Valid values updated.
Section 4.5.5.1	Execution report updated to reflect final state of proposal negotiation.
Sections 4.5.6/7/8/9	New sections for FX Forwards Swap Proposal Negotiation
Section 4.6	New section for PBC Credit Admin Messages
Section 4.7	Updated to reflect TCR for FX Swaps.
Section 5.1.5	MOS and MOOPI workflow
Chapter 6	Updated trade capture report description to incorporate FX Swaps.

- In version 1.0.19

Section 1.1 and 2.1	Removed references to the old Auto Quote (AQ) Interface. AQ is now end of life and been deprecated from service

Sections 4.5.1.1, 4.5.1.2 and 4.5.1.2.1	Updated to include ManualOrderIndicator (1028) as an optional tag in order entry messages. This is new functionality introduced by the Matching v1.3.6 release for all Order Entry messages.
Section 4.5.4.1	due a documentation error in tags ListId (66) and Account (1) have been removed from the Order Cancel Reject message.
Section 4.5.4.6	Updated List Status Message table to reflect status text for ManualOrderIndicator (1028) rejection scenarios.
Appendix A	Updated to reflect ManualOrderIndicator (1028) order entry errors.

In version 1.0.18

Section 3.4.1	Updated to reduce the maximum number of FIX Trading users allocated to a FIX session from 20 to 12.

- In version 1.0.15/16
  - Updates for Minimum Quote Life (MQL) and Maximum Order Size (MOS)
  - The Following sections have been modified

    1.4, 3.6.2.2, 3.6.7, 3.9, 4.5.2.2, 4.5.4.1, 4.5.4.4.1, 4.5.4.5, 4.5.4.6

    Appendix A and Appendix C

    Sections 3.6.9, 3.6.10, 5.1.2 and 5.1.3 are new

    Corrected typos on Message tables for FIX vs Matching Mandatory fields

Feedback

If you have any comments on this document please contact your account manager, or Treasury Transactions Sales specialist. Alternatively, you may post a report in the Transactions support site at https://transactions.thomsonreuters.com/login.aspx.

Chapter 1 Introduction and Scope

1.1 Introduction

As part of our going commitment towards providing a world-leading FX Matching platform Thomson Reuters continues to enhance its Matching services. The Thomson Reuters Matching platform has recently been upgraded with a newly developed Matching engine, Matching API and graphical user interface. The new Matching service is scalable, highly performant, extensible and includes several new functional enhancements such as the Depth of Book (DOB).

The Matching API offers two separate interfaces, one dedicated for price discovery and another dedicated for order management. The Reuters Foundation API (RFA) is used for price discovery and an industry standard FIX interface has been introduced for order management and post trade processing. The Matching API solution discussed herein replaces the old Auto Quote API service.

1.2 Document Scope

This document defines the FIX Rules of Engagement (ROE) for the Matching Interface. The requirements and FIX message definitions provided herein only apply to the Thomson Reuters Matching service.

1.3 References

Reference #	Document Name
1	Thomson Reuters Matching Data Feed Direct User Guide
2	Thomson Reuters Matching Interface Error Messages Guide
3	Thomson Reuters Matching Rule Book

1.4 Terms and Acronyms

Acronym	Definition
AOR	Automatic Order Replacement
CA	Credit Administrator
DCS	Daily Confirmation Sheet
DOMCAD	Disable On MAPI Credit Admin Disconnect
Instrument	Used to denote a currency pair and tenor for FX Spot or FX Swaps trading
MAPI	Matching API

Acronym	Definition
MFG	Matching FIX Gateway
MDFD	Matching Data Feed Direct
MQL	Minimum Quote Life
MOOPI	Maximum Open Orders Per Instrument
MOS	Maximum Order Size
PB	Prime Broker
PBC	Prime Broker Client
RFA	Robust Foundation API
STU	Special Ticket User
TCID	Terminal Connection IDentifier
TCR	TradeCaptureReport (35=AE)
UPA	Ultra Performance API

Chapter 2 Introduction to the Matching API

The Matching API (MAPI) is designed to provide access to the Matching service using industry standard interfaces. This allows clients to implement their own applications on their own hardware without needing to use Thomson Reuters provided Keystation terminals and software.

MAPI is made of two key components:

- RFA/UPA: used to receive market data
- FIX: used to submit & mange orders and other requests

This document describes the FIX interface. For details of the market data provided by RFA/UPA please see the *Thomson Reuters Matching Data Feed Direct User Guide*.

2.1 Logical Architecture

The following diagram outlines the core components within the logical architecture.

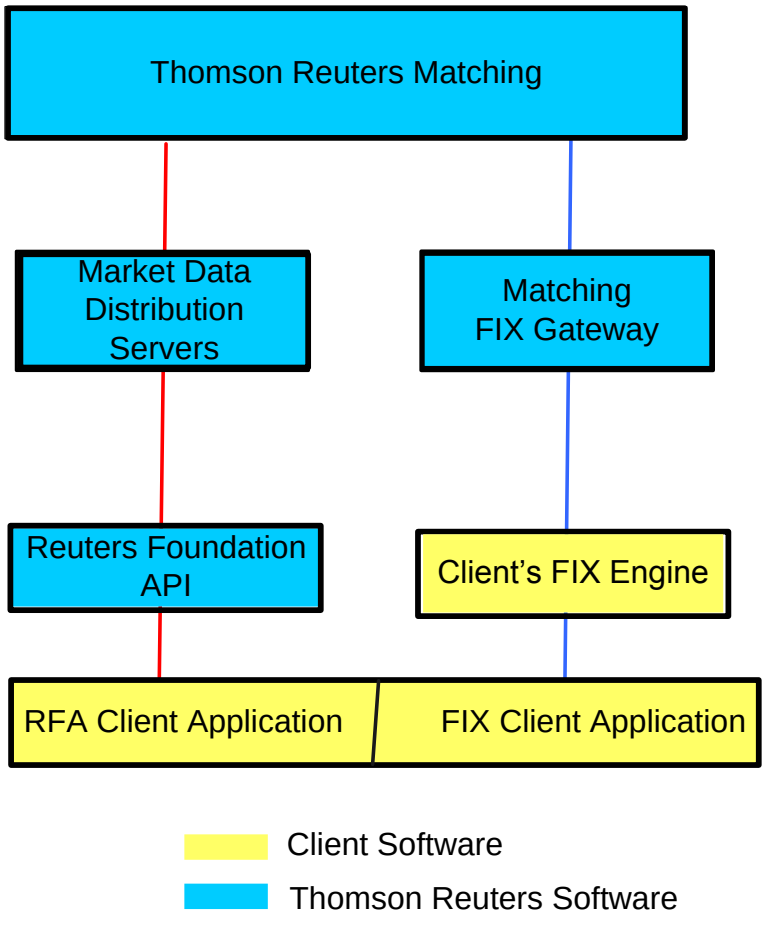


Figure 1 : High Level Logical Architecture

2.2 FIX Rules of Engagement for Matching

This document defines the specific set of FIX messages, constituent tags, and associated workflows that are supported by the Matching service. It should be used by clients as reference material during the development, certification and internal support of their FIX applications for use with Matching.

All FIX client applications accessing the Matching service must operate within the guidelines set out herein. Thomson Reuters reserves the right to disable any FIX client application that is violating these rules.

2.3 Functional Summary

MAPI provides access to the same common functions as available on a keystation.

2.3.1 Trading

FX Spot & FX Swap orders can be submitted via the API and match notifications received.

For FX Swap orders the negotiation phase that occurs after the soft match has occurred can also be handled via the API. Alternatively it is possible to transfer the proposal over to a keystation and complete the negotiation outside of the API.

2.3.2 Drop Copy

MAPI can provide FIX sessions for back office systems allowing them to receive a copy of the same settlement instructions that the MAPI traders received.

2.3.3 STU Management

Prime Brokers can login their Special Ticket Users (STUs) via MAPI. As well as enabling their clients to trade, the STU users will receive the match notifications & settlement instructions.

Whether a Prime Broker is using MAPI or not has no bearing on the client or vice-versa. A Prime Broker on MAPI can have clients who are using manual keystations, and clients using MAPI can work with Prime Brokers who manage the STU on a keystation.

2.3.4 PBC Session Control

Prime Brokers can login their Credit Admin users to inhibit, disable or enable specific PBCs.

2.3.5 MOS & MOOPI

Credit Administrators can set Maximum Order Size (MOS) & Maximum Open Orders per Instrument (MOOPI) limits.

2.3.6 Summary

Orders for FX Spot and FX Forward Swap instruments can be submitted

Function	Interbank	Prime Broker Client (PBC)	Prime Brokerage (PB)
FX Spot	Y	Y	
FX Swap	Y		
Drop Copy (STP)	Y	Y	Y
STU Management			Y
PBC Session Control			Y
MOS & MOOPI	Y		Y

Chapter 3 Matching Concepts

3.1 Trading Week

Matching runs for approximately 5.5 days each week. Its starts on Sunday and runs continuously until Friday at which point it goes offline during the weekend maintenance period.

The different stages in the week are:

Start Time	Mode	Description
Sunday 12:30 GMT	Non-trading	Earliest point at which a client can login. Trading is not allowed though. There will be no liquidity on any instruments. Any attempts to submit an order will be rejected.
Monday 04:45 Sydney local time	Trading	Trading is now allowed. Earliest point at which an order can be successfully submitted or liquidity is visible in the market.
Friday 18:00 New York local time	Non-trading	Trading is disabled. All open orders will be automatically cancelled and any attempts to submit new orders will be rejected. Users will not be logged out yet though.
Friday 19:00 New York local time	Offline	Service goes offline. Any users still connected will be logged out. Service will not come back online again until Sunday 12:30 GMT.

3.1.1 Daylight Saving Time

Not all of the stages above are defined in GMT; some refer to the local time in Sydney or New York. As such the GMT equivalent will vary during the year depending upon whether Daylight Saving Time is in effect in that city or not.

Both cities adjust their clocks on different dates so there are multiple dates when the GMT times change. Also as they are in different hemispheres they move in opposite directions (summer in one is usually winter in the other). As a result the trading week actually gets shorter or longer rather than being the same length all year round.

At the time of writing Sydney changes on the 1st Sunday in April & October, whilst New York changes 2nd Sunday in March & 1st Sunday in November.

The table below lists those four Sundays and shows what effect it has on the time difference generally, as well as highlighting the change it causes to the trading week.

Matching simply follows the official times in Sydney & New York though. If either location were to change when, or how, daylight saving time is implemented it would supersede the dates discussed in this section.

	2nd Sunday March	1st Sunday April	1st Sunday October	1st Sunday November
Sydney		Leaves Daylight Saving Time Now GMT+10	Enters Daylight Saving Time Now GMT+11	
New York	Enters Daylight Saving Time. Now GMT-4			Leaves Daylight Saving Time Now GMT-5
Service Online	Sun 12:30 GMT	Sun 12:30 GMT	Sun 12:30 GMT	Sun 12:30 GMT
Trading Enabled Mon 04:45 Sydney	Sun 17:45 GMT	Sun 18:45 GMT	Sun 17:45 GMT	Sun 17:45 GMT
Trading Disabled Fri 18:00 New York	Fri 22:00 GMT	Fri 22:00 GMT	Fri 22:00 GMT	Fri 23:00 GMT
Service Offline Fri 19:00 New York	Fri 23:00 GMT	Fri 23:00 GMT	Fri 23:00 GMT	Sat 00:00 GMT

Note: None of the times are specified in UK/London time. When the UK enters or leaves daylight saving time it has *no effect* on the trading week.

3.1.2 Trading Mode Notifications

In order to help applications determine when trading mode starts & stops notifications sent over the API to any logged in trading users. This allows the current state to be detected without having to track to times in Sydney and New York manually.

See Notification Messages section for further details.

3.1.3 Holidays

The same schedule is used every week regardless of whether there are major public holidays being observed in some locations. As a result the service does not shutdown or alter its timings even over periods such as Christmas or New Year.

However holidays may affect the settlement dates of certain instruments, and there may be less activity if participants in those locations choose not to trade during those holidays.

3.1.4 Maintenance Window

There is *no* daily restart or daily maintenance window.

Users can remain connected all week long if they wish and as a result orders could remain open all week long if no other event occurs to cause them to be cancelled or fully filled first (see Order Lifetime for further details).

All planned maintenance activities occur during weekend period when the service is already offline.

This means whenever new versions are released or instrument properties changed etc. they will take effect immediately when the system comes back online on Sunday, rather than coming into force part way through a trading week when users may already be connected and actively trading.

3.2 Soft & Hard Matches

There are two types of matches; soft & hard.

A hard match is absolute and non-negotiable. These occur with spot matches immediately and after the negotiation phase has successfully completed for forward swaps.

A soft match is where Matching has identified a possible match but the two parties involved need to complete a negotiation phase first in order to agree on the details. Soft matches are used in the initial stages of a forward swap trade.

The reason for the difference is due to the way Matching handles credit limits.

For spot, each party sets credit limits specifying the total daily limit they are willing to trade with every other party. Each time a match occurs the credit limit each party assigned to the other is adjusted accordingly. This allows Matching to make an immediate and fully automatic decision on whether the orders can match, and thus spot matches are hard matches.

Forward Swap trading however has more complex risks as the decision to trade would not only depend upon the party but also the currency pair, tenor and quantity involved in the trade. Instead of each party setting numerical limits for ever other party, they just set a Yes/No toggle.

As long as neither party has said no, Matching will provisionally allow the swap match by issuing a soft match and let the parties check the risk in more detail and negotiate the exact quantities they are willing to match in that specific instance. If the negotiation completes successfully a hard match then occurs.

3.3 TCIDs

TCIDs are the four letter identifiers that each party is known by. These are also commonly referred to as credit codes, dealing codes, branches or site IDs.

The TCID is the level at which certain functionality is rooted:

- Credit limits are assigned at this level
- Instrument permissions are assigned at this level
- Usernames are associated with exactly one TCID. If the same username exists in multiple TCIDs they are in fact unrelated users
- Matches can *never* occur between two users on the same TCID

In MAPI the TCID for a particular session can be determined from the first four letters of the SenderCompID. See the section on SenderCompID later in this document for further details.

3.4 Order Types & Entry Attributes

3.4.1 Summary of Order Entry Types

All order types in Matching are limit orders. The price specifies the worst price (lowest bid or highest offer) that is acceptable. Matches will occur at the best price (highest bid or a lowest offer) available and therefore depending upon the liquidity may be better than the limit price.

There are various types available that determine how long the order will remain open if it is not fully matched immediately upon receipt:

- Standard Order
- Immediate or Cancel (IOC)
- Good for Time (GFT)
- Good till Date (GTD)
- More Quantity
- I-Deal
- Iceberg

3.4.2 Order Lifetime

Unless the order type specifies further restrictions the order will remain open until the first one of the following events occurs:

- The quantity is fully filled, or reduced below the minimum quantity defined for the instrument
- A cancellation request is successful
- Logout of the trading user
- Network disconnection
- Non-trading mode at the end of the trading week (Friday 18:00 New York local time)

For PBCs these additional events will cancel open orders too:

- The Prime Broker's STU is logged out or disconnected
- The Prime Broker disables or inhibits the PBC
- The Prime Broker's CA is logged out or disconnected, and the Prime Broker has enabled DOMCAD.
- A match is attempted but fails because the credit limit assigned by the Prime Broker to the PBC has reached zero.

Most of these events are things that are either user initiated or the result of unexpected issues. It is possible that they could occur at any time and depending upon the circumstances might never occur at all. However the one event that is guaranteed to occur is the end of the trading week which means an order can never survive past Friday 18:00 New York time.

It is not possible for an order to remain open when the trading user is not logged in. This applies to all clients and it is not a configurable option that can be switched off.

Any open orders will always be cancelled automatically by Matching when either the user requests to logout or a disconnection is detected. As the FIX protocol uses a heartbeat mechanism the connection will always time out after a maximum of 12 seconds if the client application cannot or will not respond.

3.4.3 Standard Order

A Standard Order is the most basic order within Matching.

Once submitted, any remaining quantity that does not immediately match is positioned in the order book and will remain open until one of the scenarios in the previous sections occurs.

In MAPI a Standard Order is identified by the following FIX tags:

FIX Tag	Value
MsgType(35)	D [NewOrderSingle]
TimeInForce(59)	0 [Day] *This is also the default if TimeInForce(59) is not present at all*

3.4.4 I-Deal

An I-Deal is an order that includes both a BID & OFFER on the same instrument. Each portion of this order may have different quantities and different price.

Once an I-Deal has been entered each side of the order can be cancelled independently of the other. Alternatively the whole I-Deal can be cancelled.

They also have the optional feature One Cancels Other (OCO). If this is set on the I-Deal then once one side is completely filled the system will automatically cancel the other side.

In MAPI an I-Deal is identified by the following FIX tags:

FIX Tag	Value
MsgType(35)	E [NewOrderList]
ContingencyType(1385)	5 = BID and OFFER (normal I-Deal) 6 = BID and OFFER OCO (I-Deal with OCO enabled)

3.4.4.1 Differences allowed on each side

Order entry attributes such as Symbol, Currency, OrdType, QtyType, Account, ContractMultipler, SecutityType, TriggerInstructions (i.e. Action, Scope and Type) and ManualOrderIndicator are only required to be defined within the first order contained within the list.

Each subsequent order within the list will by default, inherit the attributes that are specified within the first order of the list. Users may optionally elect to specify the same attributes for each individual order in the list as long both orders contain the same values.

Conversely, traders may enter I-Deal Orders such that each order (i.e. Bid and Offer) can have the same or different values for their timed order entry attributes (i.e. TimeInForce, ExpireTime, ExposureDuration). Timed order entry attributes specified in the second order (Offer) of the list can either be inherited from the first order (Bid) or explicitly defined to the same or different values as the first order.

Note: I-Deal orders entered via the Matching Keystation enforce users to input the same quantity on each portion of the (buy/sell) order. When entered via MAPI they can include different amounts on the buy and sell side of the order.

3.4.5 Immediate or Cancel (IOC)

Unlike a Standard Order, an IOC order only exists for the duration of the order entry and match evaluation process. Any remaining quantity that cannot be filled immediately is cancelled by Matching automatically.

In MAPI an IOC is identified by the following FIX tags:

FIX Tag	Value
MsgType(35)	D [NewOrderSingle]
TimeInForce(59)	3 [Immediate or Cancel (IOC)]

Note: Unlike Fill or Kill orders (which Matching does not support), IOCs can partially match.

It is not possible to submit an order where the order is guaranteed to fully fill or not match at all, if the quantity is greater than the lot size.

3.4.6 Good Till Date (GTD)

Good till Date orders are similar to Standard Orders with the exception that they specify a timestamp when the order should automatically be cancelled.

Assuming none of the scenarios in section 3.4.2 occur first, then the order will be cancelled automatically by Matching when its clock reaches that timestamp.

When Matching receives the order the timestamp must be at least the minimum duration ahead of the current time, and at most the maximum duration ahead.

The minimum duration is specified in *T_ORD_MNS (FID -10068)* of the *Reference Data Model*, and the maximum in T_*ORD_MXS (FID -10069)*

FIX Tag	Value
MsgType(35)	D [NewOrderSingle]
TimeInForce(59)	6 [Good Till Date (GTD)]
ExpireTime(126)	*Timestamp at which point the order will be cancelled* *Must not include milliseconds.*

3.4.7 Good For Time (GFT)

Good till Time orders are similar to Good till Date orders except rather than specifying an absolute timestamp when they should be cancelled, a relative number of seconds is provided instead.

Assuming none of the scenarios in section 3.4.2 occur first, then the order will be cancelled automatically by Matching once it has remained positioned for that length of time.

Good For Time orders too must obey the minimum and maximum durations.

The minimum duration is specified in *T_ORD_MNS (FID -10068)* of the *Reference Data Model*, and the maximum in T_*ORD_MXS (FID -10069).*

FIX Tag	Value
MsgType(35)	D [NewOrderSingle]
TimeInForce(59)	A [Good For Time (GFT)]
ExposureDuration(1629)	*Number of seconds after which the order will be cancelled*

3.4.8 More Quantity

More Quantity orders split the quantity into two parts – primary and secondary. (All other order types discussed so far only have a primary quantity).

The primary quantity is visible and so if the order is positioned it can be included in the available prices shown in the market data. The secondary quantity is not visible.

Note: A More Quantity order is not the same as Iceberg order.

The primary and the secondary quantity are given different positions in the order book. The list below demonstrates the order of precedence that applies when there are multiple orders positioned with the same price:

- Oldest primary qty
- 2nd oldest primary qty
- ...
- Newest primary qty
- Oldest secondary qty
- 2nd oldest secondary qty
- ...
- Newest secondary qty

When submitting the primary quantity must be non-zero. It is not possible to have a More Quantity order which *only* has a secondary (non-visible) quantity.

The demarcation between primary and secondary quantity is permanent; at no point is more of the secondary converted into primary revealing visible quantity again.

Also an order cannot remain open when the primary quantity has been filled. Once the incoming order has been processed and all possible matches completed, then any More Quantity orders that have no primary quantity left but are not yet fully filled will automatically be cancelled by Matching.

In MAPI a More Quantity order is identified by the following tags:

FIX Tag	Value
MsgType(35)	D [NewOrderSingle]
OrderQty(38)	*Total quantity (i.e. primary + secondary)*
DisplayQty(1138)	*The primary quantity* *Must be > 0 and < OrderQty(38)*

3.4.9 Iceberg

An Iceberg also splits the order quantity into two parts – primary and secondary – just like More Quantity orders do.

The difference is that an Iceberg order is not automatically cancelled once the primary quantity has been filled or drops below the minimum quantity. Instead some of secondary quantity will be used to refresh the primary quantity, after an optional delay.

When the refresh occurs the primary quantity will be given a new position in the order book, and thus it will become the newest primary quantity in the list. The refresh will not change the position of the secondary quantity though.

For example consider an Iceberg order with ID *ICE1*. The table below shows an example the positions before, and how that would change after a refresh:

Before Refresh	After refresh
• Oldest primary qty • **ICE 1 primary qty** • ... • Newest primary qty • Oldest secondary qty • **ICE 1 secondary qty** • ... • Newest secondary qty	• Oldest primary qty • ... • Newest primary qty • **ICE1 primary qty (new position)** • Oldest secondary qty • **ICE1 secondary qty (same position)** • ... • Newest secondary qty

In addition to the general limits on the number of open orders (which includes Iceberg orders too), there are further restrictions specifically on the number of Iceberg orders open.

This limit is defined per instrument, per side and reported in *MNINO (FID-10780)* of the Reference Data Model. It is set by Thomson Reuters and applies to all participants; it is not user configurable.

In MAPI an Iceberg order is identified by the following tags:

FIX Tag	Value
MsgType(35)	D [NewOrderSingle]
TargetStrategy(847)	1001 [Matching Iceberg]
NoStrategyParameters(957)	*Repeating group specifying the tip sizes and chosen delay range*

3.4.9.1Tip Size

The Tip size specifies the size of the primary quantity. It can be specified in several ways:

- A list of 1 to 3 values (Q1, Q2 & Q3)
- An inclusive range (Qmin -> Qmax)

This is used when the order is first positioned and, if after a match occurs, the primary quantity reaches zero or drops below the minimum quantity for the given instrument.

If a list of more than one tip size is given then the system will rotate through them in a round-robin fashion. Q1 will be the primary quantity when the order is first positioned, on the next refresh the primary quantity will be set to Q2 and so on.

When the tip size is given as a range the system will randomly pick a value within the range [Qmin, QMax] each time a refresh is needed.

Once the order has been partially filled the limited amount of secondary quantity remaining could mean some or all the specified tip sizes are no longer achievable. For Q1, Q2 or Q3 any values that are too high value will be removed and subsequent refreshes to be picked from a smaller list. For Qmax the value will be set to the remaining secondary quantity and subsequent refreshes will use that new range to randomly select values from.

If *all* tip sizes are ignored then the next refresh will be equal to the entire remaining quantity thus making it the final refresh.

In addition to the normal min/max/lot size restrictions on quantities in Matching, the tip size(s) have further restrictions. It must be greater than or equal to the minimum permitted tip size and less than or equal to the maximum permitted tip size.

These are defined per instrument and reported in *MN_IBO_TP (FID -10782)* & *MX_IBO_TP (FID -10783)* of the Reference Data Model. These are set by Thomson Reuters and apply to all participants; they are not user configurable.

3.4.9.2Delay

When the primary quantity is fully filled there can be an optional delay before the primary quantity is refreshed. During this period the order will remain positioned but have only a secondary quantity and no primary.

The delay is specified by choosing one range from a pre-defined list of possible ranges. For example the initial list of choices when MAPI 1.5.5 was released were:

- Range 1: 250 – 1000ms
- Range 2: 1000 – 2000ms
- Range 3: 2000 – 3000ms

Each time a refresh is needed Matching will then randomly pick a delay within the specified range. If no range was specified when submitting the order then each refresh will occur immediately.

The current list can be obtained via the MDFD servers in *ITDM (FID -10775)* of the *Configuration Data Model*. These are set by Thomson Reuters and apply to all clients. They are not user configurable. See the Matching Data Feed Direct Guide, Appendix E for more details.

Matching will always ensure there is primary quantity at the same side & price from at least one participant (there will not be prices which are *only* secondary),

To achieve this if there is no primary quantity after a match and no orders have an immediate refresh defined then the system will force an immediate refresh on the order with the best position instead, ignoring the random delay in this instance.

3.4.9.3MQL

MQL on an Iceberg order is only measured from when the order is originally positioned in the order book the same as it is with other order types. When the Iceberg's tip is refreshed a new MQL period will not be started, nor will an existing period be cancelled.

3.4.10 Order Entry Attributes

In addition to the order types, when submitting an order the following order attributes may be specified to affect the order entry behaviour.

3.4.11 Jobbing Mode & Automatic Order Replacement (AOR)

If an order is received that has Jobbing Mode enabled, any existing, non-locked, orders for the same user, instrument & side will be cancelled before this order is positioned.

The new order will be placed at the end of the order book just like any other new order would be. Jobbing Mode does not allow the new order to keep the same temporal position of the previous order(s).

AOR is similar to Jobbing Mode except that more specific criteria are used and in some circumstances the temporal position of the order can differ.

In addition to the user, instrument and side, AOR also requires the **price** to be the same when selecting the existing orders it will cancel. The position the new order takes in the order book depends upon the asset class and quantity.

3.4.11.1 Jobbing Mode vs. AOR

The tables below show the differences between Jobbing Mode & AOR.

The properties that must be the same when selecting orders to be cancelled:

Property	Jobbing Mode	AOR
User	Y	Y
Instrument	Y	Y
Side	Y	Y
Price	**N**	**Y**

The position of the new order:

Asset Class	Qty	Jobbing Mode	AOR
Spot	Any	End of Book	End of Book
Swap	new >= old (i.e. same or increase)	End of Book	End of book
Swap	new < old (i.e. reduction)	End of Book	Same temporal position as oldest order it replaces

In MAPI Jobbing Mode & AOR are identified by the following tag:

FIX Tag	Value
TriggerScope(1628)	4 = All orders for the given security and side (jobbing mode) 5 = All orders for the given security, side and price (AOR)

3.4.11.2 Icebergs vs. other orders

Jobbing Mode and AOR treat Iceberg orders separately to the other order types.

If an Iceberg order is submitted with Jobbing Mode or AOR then it will *only* cancel existing Iceberg orders. Any other types of order will not be cancelled even if they otherwise match the criteria specified in the table above.

Likewise a non-Iceberg order with Jobbing Mode or AOR set will only cancel other non-iceberg orders.

3.4.12 I-Deal One-Cancels-Other (OCO)

An I Deal order which specifies that when one side is fully traded the linked order on the opposite side is automatically removed from the order book.

In MAPI OCO is identified by the following tag:

FIX Tag	Value
ContingencyType(1385)	6 = Bid and Offer (OCO)

3.4.13 Locked Orders

When submitting an FX Spot order it is possible to mark it as *locked* and result in being treated differently by Mass Action Requests, Jobbing Mode, or AOR orders.

Locked orders will be ignored by Mass Action Hold or Release requests (it is impossible to hold or release a locked order). They will also be ignored by Jobbing Mode or AOR orders (the new order will be placed without cancelling the locked ones).

Mass Action Cancel requests can specify whether it is to target:

- Non-locked orders only (the default)
- Locked orders only
- Locked & non-locked orders (i.e. all orders)

All other requests or events will not treat locked orders differently. For instance they can still be cancelled by explicitly sending an *OrderCancelRequest (35=F)* for that specific order and they will still be cancelled by the system automatically in the event of logouts or disconnections etc.

The locked flag can *only* be set when an order is originally submitted. It is not possible to change an order from locked to non-locked or vice-versa afterwards.

Locked orders are not allowed on:

- Orders for FX Forward Swap instruments
- I-Deal orders
- IOC orders

In MAPI a locked order is identified by the following tag:

FIX Tag	Value
LockedStatus(5007)	Y=Locked N=Not locked

3.4.14 Inter-compatibility

Certain order types or attributes have restrictions preventing them from being used on a particular asset class or in combination with one another. The table below lists which asset classes are *permitted*:

	None	GTD/GFT	Jobbing Mode	AOR	Locked Orders	One Cancels the Other
Standard Order	All	All	All	All	Spot	
IOC	All		All			
I-Deal	All	All	All	All		All
More Quantity	Spot	Spot	Spot	Spot	Spot	
Iceberg	Spot	Spot	Spot*	Spot*	Spot	
GTD/GFT						
Jobbing Mode	All	All			Spot	All
AOR	All	All			Spot	All

	None	GTD/GFT	Jobbing Mode	AOR	Locked Orders	One Cancels the Other
Locked Orders	Spot	Spot	Spot	Spot		
One-cancels-the other	All	All	All	All		

** Jobbing Mode & AOR on an Icebrg order will only cancel existing Iceberg orders not other order types & vice-versa.*

3.5 Minimum Quote Life (MQL)

MQL is the minimum amount of time an order must remain open before a cancel or hold request will be permitted.

It only affects the point at which the client may request the order is cancelled or held. Orders can still be matched or cancelled by the system at any time (e.g. IOCs, More Quantity orders, or disconnections) without waiting for MQL to pass.

The MQL is defined per instrument and is reported in MINQUL (FID -10535) of OMM 147. This is set by Thomson Reuters and applies to all participants; it is not user configurable.

The MQL is also included in the *Trading Rules Parameters* document on My Account.

3.5.1 Transactions Impacted by MQL

If a request is made that would violate MQL it will be rejected resulting in a response being sent back confirming it was due to MQL. The request is *not* queued and then processed later once the MQL has passed.

It is the client's responsibility to detect the rejection and to repeat the request later at the correct time if they still require the request to be carried out.

For requests that can act on multiple orders it is possible that there are both orders within their MQL and orders where the MQL has passed. The outcome depends on the particular request – some are designed to result in a complete rejection (all or nothing) whereas others are only a partial rejection.

The following table below lists the various requests that can be affected by MQL

Type of Request	Outcome if MQL not expired
Single Order Cancel	**Completely rejected:** Order not cancelled
Cancellation of an I-Deal Order	**Completely rejected:** Both sides of order not cancelled
Mass Action Cancel	**Partially rejected:** The order(s) failing MQL will not be cancelled. All other orders will be cancelled successfully.

Type of Request	Outcome if MQL not expired
Mass Action Hold	**Partially rejected:** The order(s) failing MQL *and* all orders for the same instrument(s) are not held. All other orders will be held.
Jobbing Mode order AOR order	**Completely rejected:** The order(s) failing MQL and all other orders will not be cancelled. New order will not be placed.

Clients must monitor the length of time their orders have been open and not try to cancel the orders prematurely. Any trading user that causes repetitive rejections due to MQL will be automatically logged out by the system and if the behaviour still continues further action may be taken.

3.6 Order Validation

An order must meet certain requirements before it will be accepted by the system. Orders that fail will be rejected giving an indication of the first identified problem.

It is important to note that various FIX messages may be used to convey the rejection and applications should ensure they can handle all of the following:

- ExecutionReport (35=8)
- ListStatus (35=N)
- BusinessMessageReject (35=j)
- Reject (35=3)

3.6.1 Throttling

There are maximum rates at which certain messages can be submitted and any requests above these limits will be rejected. Different limits apply to different categories of messages and at the time of writing the limits are:

Category	Rate
Orders	15 orders in the last 5 seconds
Mass Action Cancel (if they are not limited to a single instrument)	1 request in the last 60 seconds
Mass Action Hold	1 request in the last 60 seconds
Mass Action Release	1 request in the last 60 seconds

These limits are subject to change which will be communicated via PCN.

An I-Deal order is counted as one order not two.

All limits are applied per *trading* user.

Any trading user that causes repetitive rejections due to throttling will be automatically logged out by the system and if the behaviour still continues further action may be taken.

Other messages (including cancellations for single orders, or Mass Action Cancel requests for single instruments) are not subject to specific throttles.

However clients must still ensure their application uses the API correctly and does not cause the system to have to process unnecessary requests. For example care should be taken to avoid trying to cancel an order before its MQL has expired, or after being notified that it is no longer open.

3.6.1.1 Global Throttle Rate Reduction (GTRC)

The Matching service may in some rare cases (due to excessive market activity) be required to dynamically regulate the inbound order entry rate across all transactions and applied to all users

When invoked trading users are notified each time the global throttle rate is reduced or restored so that they are aware of when a reduced throttling rate is currently active.

3.6.2 MOS & MOOPI

For the following two limits, credit administrators have the ability to set even lower limits if they wish.

A Prime Broker's credit administrator can set the limits for their clients, whereas the administrator for an Interbank applies the limits to their own organisation.

(A Prime Broker Client cannot set these limits themselves and must adhere to the values set by their Prime Broker).

The currently enforced limits are notified to the trading user via the U3 message type.

3.6.2.1 Maximum Order Size (MOS)

The maximum quantity that can be specified single order is normally:

- 999 mio for FX Spot
- 50000 mio for FX Swap

If a lower limit is specified then it is in whole millions of USD and applies to the whole TCID.

When orders are placed in non-USD denominated instruments (e.g. EUR/GBP) it will be converted to their USD equivalent using the FX Spot rates sourced from Matching.

3.6.2.2 Maximum Open Orders per Instrument (MOOPI)

The maximum number of open orders per instrument is normally 200 orders.

A lower value can be set for a specific TCID.

3.7 Unacknowledged Matches

The ACF process is designed to minimise risk exposure associated with scenarios where one counterparty to a Match is unaware that the transaction completed.

Within the post trade Matching envelope any one the following scenarios could occur:

- Both counterparties have confirmed and know about the Trade
- Neither counterpart knows anything about the trade (Thomson Reuters support staff are alerted).
- One counterparty knows about the trade and they are informed that their counterparty does not (Thomson Reuters support staff are also alerted); In this scenario the match is marked as "unconfirmed". The Matching Rule Book states that any counterparty who has received an unconfirmed Match is still responsible for completing the deal with their counterparty.

The Matching ACF workflow is as follows:

- The Matching FIX Gateway is responsible for acknowledging match notifications sent by the Matching engine. The acknowledgement will be sent to the Matching engine immediately after the MFG has persisted the notification to its local message store. The MFG will not acknowledge the match notification if the FIX client (or target user) of the match has been logged out *
- Once the Matching engine successfully receives acknowledgements from both counterparties involved in a match both parties are sent trade tickets completing the deal (FIX clients are sent TCR messages (which is analogous to a trade ticket).
- The TCR will minimally include the Order Id, Client Order Id, instrument, base currency amount, contra currency amount, price and side, the counterparty details, payment instructions.

*(*The Keystation is also responsible for acknowledging match notifications sent by the Matching engine to manual users)*

If either side of the match is not acknowledged within a predefined period (global setting), the Matching engine will flag the trade as "unconfirmed". In this case, the Matching engine will generate and send a set of "unconfirmed" Trade Tickets to the MFG. When the MFG receives an unconfirmed trade ticket it will forward the trade ticket to the FIX client as per normal processing. Clients are required to settle both confirmed and unconfirmed trades with their counterparties.

Chapter 4 MAPI Concepts

4.1 FIX Protocol Versions

MAPI uses the latest version of the FIX protocol: 5.0 SP2. This in the only version supported and connections with older versions will not be accepted.

MAPI also requires use of one of the following extension packs:

- EP 100
- EP 171

EP 100 contains the extra tags and enumerated values in order to implement Matching functionality via FIX.

EP 171 contains the new message types used as part of the PBC session control functionality available to Prime Brokerages.

For full details on the FIX protocol, version 5.0 SP2 or the specific extension packs see the FIX Trading Community website (formerly FIXProtocol.org):

http://www.fixtradingcommunity.org/

The rest of this document details the specifics of the MAPI implementation. Familiarity with the FIX protocol specifications is assumed and applications also need to abide by those as well as this MAPI specification.

4.2 SenderCompID

The SenderCompID is the unique identifier for a particular FIX session. The following items are assigned at the SenderCompID level:

- Source IP permissioning
- FIX sequence numbers, message persistence/replay
- Drop Copy configuration

The format of the SenderCompID is:

- 4 letters, representing the TCID
- 2 digits, a number to make the SenderCompID unique when there are multiple sessions
- 4 or 5 further digits
- 4 letters, only for PBCs, represents the TCID of the prime broker

The last two items are always constant for given TCID. As such a useful shorthand when discussing MAPI sessions is to only refer to it by the first 6 characters (TCID and 2 digit number).

For example if the SenderCompID was ABCD037752WXYZ then it would be session 03 on TCID ABCD, and the prime broker is (or at least was) WXYZ.

ABCD 01 7752 WXYZ

ABCD 02 7752 WXYZ

ABCD 03 7752 WXYZ

...

ABCD 99 7752 WXYZ

The last two items are always constant for any given TCID and so useful shorthand when discussing MAPI sessions is to only refer to it by the first 6 characters (TCID and 2 digit number).

Note: The Prime Brokers TCID is only an indication used in the naming convention of sessions. It is *not* how the system determines the actual Prime Broker.

If the PBC moves Prime Broker then the SenderCompID may continue to reflect the old value for one or more weeks until it is brought back up-to-date again.

4.3 SenderLocationID

This tag is required as part of the FIX Logon (35=A) message and the required format is as follows:

- 5 static characters "TRFXM"
- 2 letter country code
- 5 digits
- up to 15 further characters

The first 3 values are mandatory and clients must use the values specified by Thomson Reuters in their credentials documents. The fourth item though is entirely optional.

Clients may append up to 15 further characters of their choosing to the SenderLocationID value. These extra characters are not used by Thomson Reuters in anyway but may be helpful for recording some indentifying information such as the source system or program that is performing the login.

In organisations with many systems and complex failover mechanisms this extra information can help support teams track a connection back to its source once they have been provided with a copy of the Logon (35=A) message by Thomson Reuters.

4.4 Identifiers

In MAPI there are several identifiers used in relation to orders.

4.4.1 OrderID(37)

This is the ID assigned to the order by Thomson Reuters. Therefore this ID number is not visible until the order is successfully accepted and a response sent back confirming the value. In scenarios where the order is rejected no ID will have been assigned.

The ID is an integer and is unique per instrument, per trading week.

4.4.2 ClOrdID(11)

This is the ID assigned to the order by the client. Therefore this ID number is already present on the incoming request.

The ID is a string and is unique per user, per instrument, per trading week.

The uniqueness of this value is NOT validated by Thomson Reuters.

If there are two or more open orders simultaneously with the same ClOrdID(11) then any request to cancel that ClOrdID(11) will be ambiguous and both orders will be cancelled as a result.

4.4.3 ExecID(15)

The ExecID(15) is assigned to the event by Thomson Reuters.

The ID is a string and is *not* unique; the same value will be used for all ExecutionReports (35=8) triggered by the same event.

For example if a user logs out whilst they have many open orders then all the ExecutionReports (35=8) sent notifying of the cancellation will have the same ExecID(15).

4.4.4 TradeID(1003)/SecondaryExecutionID(527)

This ID is the Trade ID assigned to the match.

TradeID(1003) is used on a TradeCaptureReports (35=AE) and SecondaryExecutionID(527) is used on an ExecutionReports (35=8). The two tags are used due to a lack of a common tag that is permissible in both message types. They represent the same Thomson Reuters Trade ID.

The ID is an integer and is unique per event. Matches occurring as the result of the same event (e.g. all matches occurring as the result of an aggressing order) will have the same trade ID.

4.4.5 TrdMatchID(880)

This ID is the value assigned by Thomson Reuters to the Match.

The ID is an integer and is unique per match.

Both counterparties will see the same match ID and the value is also given in Daily Confirmation Sheets (DCS). Therefore when cross-referencing details during settlement this is an ideal identifier to refer to.

4.4.6 Summary

Tag	Purpose	Unique per ...
OrderID(37)	Thomson Reuters ID for this order	Instrument and trading week.
ClOrdID(11)	Client's own ID for this order	Trading user, instrument and trading session.
ExecID(17)	Identifier for this event	Multiple execution reports generated by the same event will have the same ExecID(17). (e.g. unsolicited cancels due to logout, multi-way matches etc.) Otherwise the value will be unique.
TradeID(1003) SecondaryExecutionID(527)	Thomson Reuters Trade ID	Multiple matches occurring from the same event will have the same TradeID. Otherwise the trade will be unique.
MatchID(880)	Thomson Reuters Match ID	Match

4.5 Notification Messages

Notification messages are sent by MAPI to logged in users in order to broadcast certain information. The majority of these are delivered using a `UserNotification (35=CB)` message. The MOS, MOOPI & PBC Session controls though are delivered in a `U3 (35=U3)` message instead.

4.5.1 Matching Trading Mode (MTM)

The MTM notification confirms whether the system is current in trading or non-trading mode. It allows applications to determine the correct state without having to follow the local time in Sydney and New York (see Trading Week for more details).

The possible values are:

- MTM:0 – Non-trading mode
- MTM:1 – Trading mode

MTM:0 will occur during the few hours on a Sunday when the system first comes online and for an hour at the end of the week before the system goes offline. Trading is not allowed in this state and client applications must not attempt to send any orders.

If it occurs at any other time during the week then please contact the Thomson Reuters helpdesk for more information.

The notification is sent to all application users and to drop copy sessions. It is sent at the time the user logs in *and* immediately when the state changes (there is no need to logout and back in again to determine the current value).

4.5.2 Start/End of Trading Week (SOTW/EOTW)

These notifications provide a countdown to the MTM state changing.

The values are in the format:

- SOTW:<minutes>
- EOTW:<minutes>

The notification is sent to all application users and to drop copy sessions. It is sent at 1 minute intervals starting 10 minutes before the MTM state is due to change.

4.5.3 Special Ticket User Status (STU)

The STU notification allows PBCs to determine if their Prime Broker STU is online.

The possible values are:

- STU:0 – STU offline
- STU:1 - STU online

STU:0 will occur when MTM:0, and at any other time if the Prime Broker's STU logs out or is otherwise disconnected. Trading is not allowed in this state and client applications must not attempt to send any orders.

If STU:0 occurs with MTM:1 and it is during hours which the Prime Broker has agreed trading can take place then please contact the Prime Broker for more information. *Do not* raise this with the Thomson Reuters helpdesk.

The notification is sent to application users belonging to PBCs only. It is sent at the time the user logs in *and* immediately when the state changes (there is no need to logout and back in again to determine the current value).

4.5.4 Credit Threshold Alerts (CTA)

The CTA notification provides a warning that a credit limit is getting low.

The format of the notification is:

CTA:<TCID>:<whole million USD remaining>

The notification is sent to application users only. It is sent every time a match occurs *if* the remaining credit is below 25% of the limit set.

Note: Credit is maintained in USD. Therefore 1 mio of the instrument be traded could exceed the remaining credit and prevent even a partial match occurring.

4.5.4.1 PBCs

PBCs will see their own TCID in the alert as they only have a single limit which is managed by their Prime Broker.

Once the remaining credit reaches zero or is insufficient to allow a match to occur the PBC must cancel any orders that are still open and not submit any further orders.

This should then be raised with the Prime Broker and trading not resumed until more credit has been allocated. *Do not* raise this with the Thomson Reuters helpdesk.

4.5.4.2 Other Clients

Others will see the TCID of the party for which the limit is low. This should be escalated internally to the organisations credit administrator so they can consider whether to raise the limit or not.

Trading does not need to stop when the limit is too low as it is only for one specific party; the limits for other parties are unaffected.

However depending upon how much liquidity that party was providing it may cause a noticeable reduction in the screened prices and achievable matches until the limit is replenished.

4.5.5 Global Throttle Rate Change (GTRC)

The GTRC notification informs when a temporary new throttle rate is imposed or removed.

The format of the notification is:

- GTRC:<max orders>:<seconds> - new limit
- GTRC:0:0 – removal of temporary limit

The notification is sent to application users only. It is only sent when a temporary limit is set or removed. If normal throttle rates are currently in effect then no GTRC notification will be received.

4.5.6 MOS, MOOPI & PBC Session Control State

This notification provides information regarding several items:

- Current MOS limit
- Current MOOPI limit
- Current PBC Session state
- Current DOMCAD setting

- Whether the PB's Credit Administrator is currently logged in

The first two items would be provided regardless of the client type. However the last three are only given to Prime Brokers and Prime Broker Clients.

The notification is sent to application users only on sessions using FIX extension pack 171. It is sent at the time the user logs in *and* immediately when the state changes (there is no need to logout and back in again to determine the current value).

As the notification contains several pieces of information it is sent in a specific message type designed for the purpose – a `U3 (35=U3)`.

4.6 Authentication and Permission Management

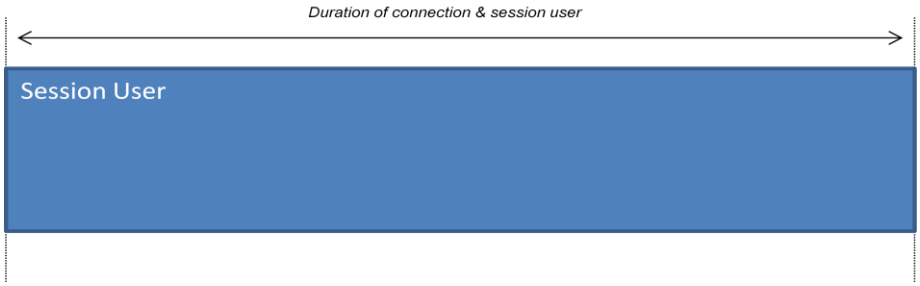
MAPI uses a two-level logon mechanism. These are referred to as the session user & application user logins respectively.

This allows separation of the MAPI connections and the application functionalities that are permitted.

4.6.1 Session User

The session level logon is performed with the FIX `Logon (35=A)` message type and the logoff with `Logout (35=5)`.

The session user is logged in for the whole duration of the connection; the `Logon (35=A)` is the very first message used to initiate the connection and when the session user logs out the connection terminates (or vice-versa).



This means there is always one session user per session.

A session user only provides access to basic FIX session level functionality such as message replay and heart-beating. In MAPI terms that means the session can be used to retrieve a RFA token (if the user has the appropriate permissions) or for Drop Copy.

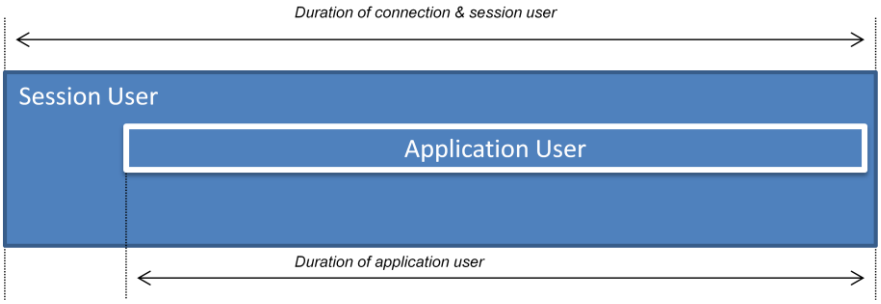
All other MAPI functionality such as trading, STU management & PBC session control would not be permitted without an additional application level login.

4.6.2 Application User

The Application User (trader, STU or Credit Admin user) logon is requested with the FIX `UserRequest (35=BE)` message. The logoff is requested with `UserRequest (35=BE)` & `UserRequestType(924) = 2 [log off user]`.

This message is sent on a FIX session that is currently connected and so there would already be a session user logged in too.

Typically the application user will login shortly after the session user and logout just before the session user at the end of the connection. However the application user can login or out at any time provided the session is already established.



Each session can currently support up to a maximum of 12 application users. The application users are independent of one another and each one can log in or out at any time without any affect on the rest.

4.6.3 SenderSubID

As multiple application users can be logged into the same session at any one time a mechanism is needed to identify which application user a message sent or receives the message.

Unfortunately the `Username(553)` tag is not a valid tag in the majority of FIX messages so another tag is needed to handle this association. MAPI uses `SenderSubID(50)` & `TargetSubID(57)` for this purpose.

When the application user logs in, the `UserRequest (35=BE)` message must provide the username in the `Username(553)` tag and also populate `SenderSubID(50)` with a chosen value. Every subsequent message sent from that application user must then also populate the `SenderSubID(50)` tag with the same value.

Similarly any messages sent by the MAPI FIX gateway to an application user will include the `TargetSubID(57)` tag with that same chosen value, allowing applications to identify the intended recipient.

Thomson Reuters do not specify what value must be used in the `SenderSubID(50)` tag. It is the client's choice what convention to use.

The simplest approach would be to merely duplicate the actual username so that both tags contain identical values. However that is not a requirement and the only restriction is that the `SenderSubID(50)` cannot duplicate a value already used by any other application users currently logged in on that session.

When deciding on the convention used though, consideration should be made for any humans or other systems that read the messages in addition to the actual application connecting to the session. MAPI sessions can remain connected for long durations (almost 5.5 days from start to end of the trading week) and there can be high volumes of messages involved. Whilst an application will be able to track the mapping between the SenderSubID(50) and Username(553), humans or other systems reading individual FIX messages in isolation will only see the tags that are present and might not be able to easily determine which user a given SenderSubID(50) represents.

4.6.4 Password Changes

Password changes in MAPI are achieved by supplying both the current password and the chosen new password in the login request simultaneously.

Effectively this makes the login and password change a single action. If the message is successful then the user has logged in *and* changed the password. If the message is not successful then the user has not logged in *and* the password has not been changed either.

The current password is provided in the `Password(554)` tag and the chosen new password in `NewPassword(925)`. To login without changing the password the NewPassword(925) tag must be omitted.

The principle is the same for both session and application users albeit that the former uses the `Logon (35=A)` and the later uses the `UserRequest (35=BE)` message.

If the tag is present but contains the same value as `Password(554)` then this will not be treated as a request to keep the current password. It will be an attempt to change the password that will always fail due to the password rules.

4.6.5 Password Rules

Passwords changes are mandatory in the following scenarios:

- The very first time a user logs in
- If the account has to be reset by Thomson Reuters (e.g. account is locked or existing password not known)

At any other time the password change is optional; clients may change the password if they wish but Thomson Reuters does not expire passwords after a given length of time.

When choosing a new password the following rules apply:

- Must be between 6 & 30 characters in length
- Must only consist of alphabetic characters or digits (A-Z and 0-9, no special characters)
- Case-insensitive
- Must contain between 1 & 5 digits
- Must have at least 3 characters that are different compared to current password

4.6.6 Logon failures

In line with security best practices, incorrect *session* login attempts will be ignored and not trigger a response. Instead the MAPI FIX gateway will merely close the TCP connection without sending back any message to indicate what was wrong.

The only exception to this in if there are errors with the FIX sequence numbers (MsgSeqNum(34) too low, or NextExpectedMsgSeqNum(789) too high). For these two scenarios the gateway will respond to confirm the current values of its incoming or outgoing sequence number is, allowing failover to backup/disaster recovery systems.

In all other cases though the only way to determine the cause of the login failure is to contact the Thomson Reuters helpdesk.

Application user login failures *will* report back the reason for the failure though. This is because application user logins can only occur when the session user is currently logged in and therefore any rejection reasons are being given back to an authorised application rather than an unknown source.

4.6.7 Logout

Logouts of a session or application user can be requested by the client via the Logout (35=5) or UserRequest (35=BE) message respectively.

When an application user logs off the following will be received by the client:

- Any messages currently in transit
- An ExecutionReport (35=8) for each open order confirming it has been cancelled due to the logout
- Finally a UserReponse (35=BF) confirming the logout is complete

The first two items in that list can occur in any sequence due to the asynchronous nature of the system. It is also possible that there are no messages in transit and no open orders, thus those two stages produce no responses at all.

The UserResponse (35=BF) will be the final message though confirming the logout is complete and no further messages for that user will be received.

When a session user logs off, any application users currently logged in that session will automatically be logged out and trigger the same responses as described above.

Once all application users are logged off the gateway will also send a Logout (35=5) confirming the session user is now logged out. At this point the MAPI FIX gateway will close the TCP connection and no further messages can be sent or received without reconnecting.

Because a client requested logout can result in other messages prior to the logout be completed it is important that applications do not simply send the request and then disconnect without listening for any responses.

Instead they should remain connected, process any responses received and continue to obey the FIX heartbeat interval in the meantime. The connection should only be terminated after the Logout (35=5) confirmation has been received.

In the event a client does disconnect prematurely before receiving all of the messages, the gateway will still produce them so they can be replayed once reconnecting next time as described in Connection Management and Recovery.

4.6.7.1 Forced Logouts

In addition to logouts requested by the client, Thomson Reuters can trigger a logout of application or session users. Such occurrences are not requests – they cannot be rejected or stopped by the client and so are referred to as *"forced logouts"*.

Although these occur without a request from the client the MAPI FIX Gateway still goes through the same process as described in section 4.6.7.

The various reasons why Thomson Reuters might trigger a forced logout are listed below:

Event	Type of user
Service goes offline, Fri 19:00 New York time	All session users
If system issues prevent the MAPI FIX gateway communicating successfully with the upstream Matching components	All session users
Manually requested by client via helpdesk (e.g. if unable to remotely access application to initiate shutdown)	Requested user

Event	Type of user
Repeated attempts to cancel orders before MQL has expired	Application User responsible for the activity
Repeated attempts to exceed the input throttle rate	Application User responsible for the activity
An unmatchable price remaining in the order book due to lack of credit	Application User that submitted the order
Any other incorrect usage of the API	Session or application user responsible for the activity
Any other event if needed to protect the Matching service	Session or application user responsible for the activity

Each FIX session will be logged out at the end of the trading session. If the logout reply includes a sequence gap, the FIX client application must send a Resend Request (to allow message synchronisation) followed by the logout message.

4.7 Connection Management and Recovery

The Matching FIX Gateway (MFG) will manage client connectivity onto the Matching service following FPL FIX standard. FIX sessions into the Matching service are capable of supporting one or more users simultaneously.

4.7.1 Heartbeat and Test Request

MAPI uses a 4 second Heartbeat Interval (HI) and a 2 second Grace Period (GP).

The heartbeat interval determines the maximum time that should elapse between the transmission of two consecutive messages on a single session. If no message otherwise needs to be sent a `Heartbeat (35=0)` will be transmitted in order to keep the session alive.

The grace period determines how long to wait before sending a `TestRequest (35=1)` message if the heartbeat interval expires and no message is received. As MAPI uses a grace period of 2 this means a `TestRequest (35=1)` will be sent after 6 seconds of silence from the other side.

MAPI will disconnect a FIX session if 2 x (HI + GP) pass without any message being received. This means a session can remain open for at most 12 seconds (2 x (4 + 2)).

4.7.2 Sequence Numbers

MAPI sessions maintain their sequence numbers for the whole trading week (Sunday -> Friday).

They are not reset daily and cannot be reset at the client's request. The MAPI FIX gateway will reject `Logon (35=A)` attempts that `ResetSeqNum(141)=Y`.

This approach is taken to ensure that sequence numbers are never reused during the week allowing any missing messages to be recovered.

As a result clients must also maintain their sequence numbers for the whole trading week.

In the event the client application loses track of its incoming and/or outgoing sequence numbers (e.g. unplanned move to DR facilities, loss/corruption of the persisted numbers etc) the following options are available:

- Set sequence numbers to 1/1 and try to login. The MAPI FIX Gateway will refuse the login but the Logout (35=5) response will confirm what number it was expecting instead.

- Contact the Thomson Reuters helpdesk, quoting the `SenderCompID(49)` in question and ask what the next sequence numbers are.

Note: Client applications are required to persist the sequence numbers. The above technique is intended for genuine, rare situations where the sequence numbers are unexpectedly lost. Applications are not permitted to take this approach on every start-up or connection attempt.

4.7.3 NextExpectedMsgSeqNum(789)

MAPI uses the `NextExpectedMsgSeqNum(789)` feature of the FIX Protocol in order to detect and recover gaps in sequence numbers when reconnecting.

This feature does more than just adding an additional tag to the Logon (35=A) message. It changes how a FIX engine should react and therefore clients need to ensure their engine supports this feature and it is enabled when used to connect to a MAPI session.

Briefly the principle is:

- Both sides include the `NextExpectedMsgSeqNum(789)` tag in their `Logon (35=A)` message
- The tag is populated with the next *incoming* sequence number expected from the other side
- Each side examines the tag in the received from the other side and compares it to next outgoing number they were going to use
- If a gap is found then it must be recovered automatically *before* sending any new messages
- The other side does *not* issue ResendRequest (35=2)

Further details on how the `NextExpectedMsgSeqNum(789)` changes the workflow are covered in the FIX Transport Protocol 1.1. (FIXT 1.1) specification which is available on the FIX Trading Community website:

http://www.fixtradingcommunity.org/pg/structure/tech-specs/session-level-specs

(Registration may be needed in order to access the specifications)

Additionally the Thomson Reuters document *A Note on Tag 789 – NextExpectedMsgSeqNum* provides further examples for those unfamiliar with the feature. This document is provided along with all the other MAPI documents and sample application during the on-boarding process.

4.7.4 Missing Message Recovery

Every outgoing FIX message is persisted by the gateway before sending to the client. In the event of the message never being received this will cause a gap in the FIX sequence numbers and in turn cause the FIX protocol to initiate recovery of those messages.

Usually the most common cases are caused by events that also result in a disconnection (e.g. network disconnects, application crashes) and so the recovery occurs upon reconnecting. It is possible though that a message is corrupted and ignored by the other side resulting in a gap and the recovery occurring whilst still connected.

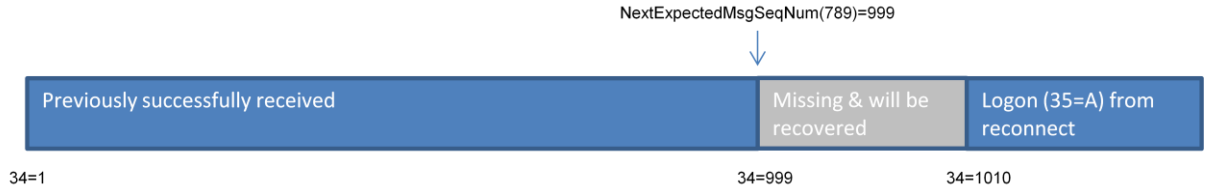
Either way the missing sequence numbers will be resent and flagged with the following as per the FIX protocol requirements:

- PossDupFlag(43)=Y
- OrigSendingTime(122)=<SendingTime(52) of original message>

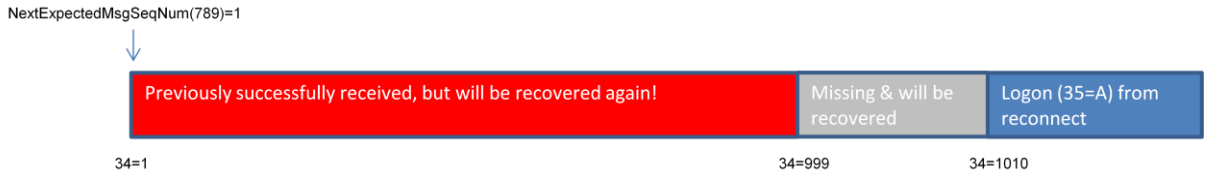
This recovery also ensures that clients receive any messages that were produced whilst disconnected – e.g.

- Unsolicited cancellations that were caused by the client disconnecting (as opposed to graceful logoff)
- TradeCaptureReport (35=AE) for trades occurring whilst a Drop Copy Session is not logged in

This is why it is important that sequence numbers are maintained and not reset arbitrarily. In order for the recovery to work reliably and efficiently the client must specify the next number they are truly expecting. This will allow the MAPI FIX Gateway to detect if there are any messages that the client is missing and resend just that block.



If no attempt is made to keep track of the sequence numbers and 789=1 is used instead it will result in the MAPI FIX gateway having to recover all messages since the start of the trading week. Most of these messages will have been received already and in turn this will waste system and network resources which is not permitted.



On the other hand if a deliberately high number is used in order to provoke a login failure and determine the expected value that will result in messages never being recovered. The login failure only reports what the gateway's last outbound number was. It does not attempt to track whether those were all successfully received and processed by the client already. As a result merely extracting the expected number and using that will result in a login with no recovery even though the client might not have received some of the earlier messages.

Message recovery when reconnecting occurs before any new application level message have been sent, which means it is not possible for any MAPI application users to be logged in at this point.

However application messages could still be recovered where the TargetSubID(57) refers to a user that is not presently logged in. Client applications should ensure that they can still correctly process such messages and do not inadvertently ignore them because the user is not yet logged in.

4.7.4.1Resend vs. Gap-fill

When recovering missing sequence numbers there are two possibilities: resend the original message (correctly adjusted so it is flagged as a resend) or issue a gap-fill to account for the sequence number without retransmitting the contents of that message.

The FIXT 1.1 protocol forbids certain message types from being resent and those *must* be gap-filled instead. Likewise MAPI forbids resending of orders or application user logon/logout requests.

The table below shows the combined effect:

MsgType(35)	MAPI -> Client	Client -> MAPI
Reject (35=3)	Resend	Resend
All other session level messages	Gap-fill	Gap-fill
NewOrderList (35=D) NewOrderList (35=E) UserRequest (35=BE)	N/A – types never sent in this direction	Gap-fill
OrderCancelRequest(35=9) ListCancelRequest(35=K)		Gap-fill if after a reconnect Can be resent if answering a ResendRequest(35=2) mid-connection though
All other application level messages	Resend	Resend

4.8 Upstream MAPI Disconnections

In the rare case that the MFG is disconnected from the matching service. The following applies:

- The Matching engine will detect the disconnect event, automatically logout and cancel all user orders associated with the FIX users that were managed by the failed MFG
- The IP address of the failed MFG will be dynamically assigned to another MFG instance
- The newly assigned MFG instance will process all FIX client logins using the same login and authentication criteria as the failed MFG. This is a seamless transition with no configuration changes required by the FIX client
- FIX client applications will be required to reconnect, login and re-authenticate to the Matching service
- In-flight messages:
    - Unsolicited cancels that were in flight from the Matching engine to the failed MFG at the time of the disconnect event will not be resent and therefore cannot be recovered by reconnected FIX clients. Thomson Reuters help desk will be available to answer any questions pertaining to order statuses.
    - Unsolicited match notifications and trade tickets that were in flight from the Matching engine to the failed MFG at the time of the disconnect event will not be resent and therefore cannot be recovered by reconnected FIX clients. All completed and unconfirmed trades will be available on Dealing Xtra. Clients must review their interim Daily Confirmation Statement (DCS) to determine of matches had occurred just prior to the disconnect event.

4.8.1 Recommended Actions after Disconnect events

Clients with open orders at the time of a disconnect event are required to reconnect to the Matching service. When reconnected the following business rules apply:

- Under normal conditions the reconnect workflow will result in the client receiving his set of pending unsolicited cancels, match and trade ticket notifications.
- If unsolicited Trade Tickets (TCRs) are received upon reconnection clients must be sure to inspect the TradeStatus for the unconfirmed state. If any given trade is marked as

unconfirmed clients are still required to settle the trade with their counterparty as per the Matching rule book.

- Clients are required to access Dealing Xtra (to review their DCS) in the event that they do not receive unsolicited messages after a given disconnect event. The DCS is used to determine if unconfirmed trades occurred prior to or during the disconnect event. If the DCS indicates that there are unconfirmed trades then the receiving client is still required to settle the trade with their counterparty as per normal operation.

See Matching Rule book (reference #3) for further details.

No action is required (other than reconnect) for those clients that did not have open orders at the time of a disconnect event.

4.9 Matching FIX Gateway (MFG) Failover

In the event of a MFG failure, all FIX message stores created by the primary server will be accessible to the recovered secondary MFG. FIX client applications are capable of seamlessly reconnecting to the service and resuming their processing of messages from the last valid sequence number. Messages in flight at the time of failure will not be recoverable and interim DCS will be required to determine if any matches were consummated prior to or during the failure event.

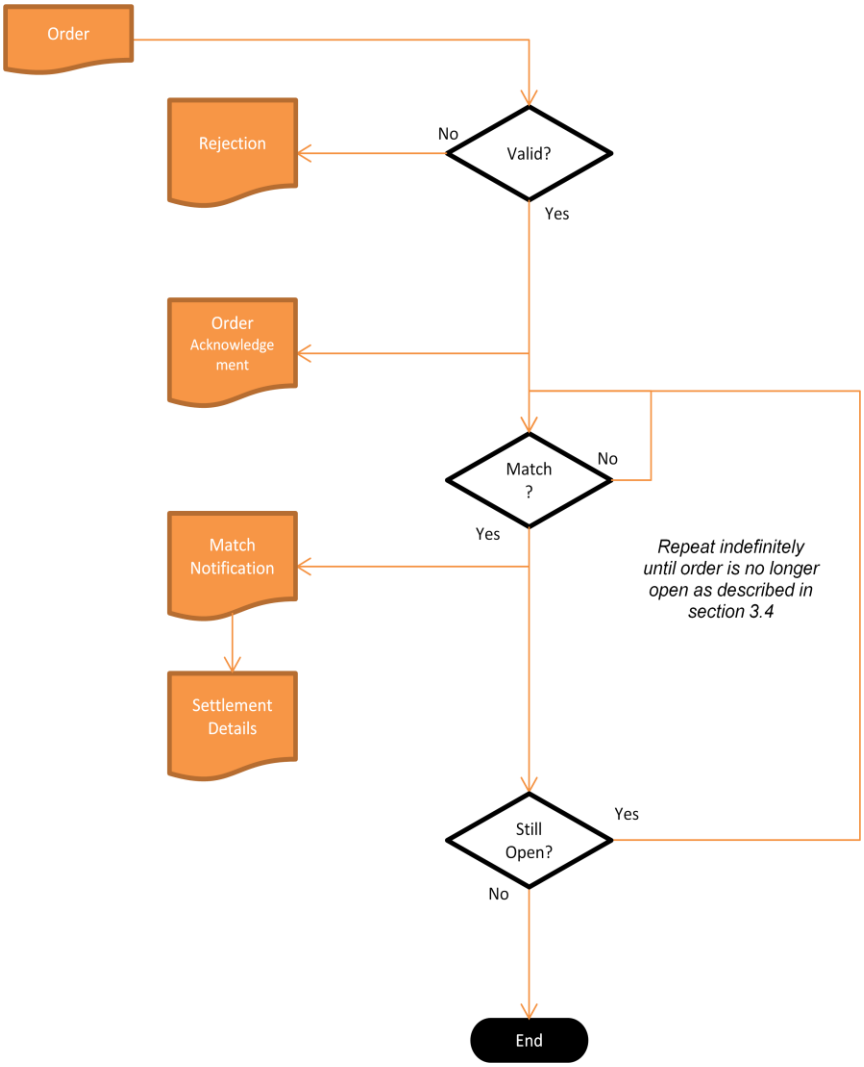
4.10 Disaster Recovery (DR)

In a true disaster scenario, when the Matching engine and or the FIX infrastructure are lost the Matching engine and associated FIX infrastructure will be started from the DR site. New FIX trading sessions will be established and sequence numbers will be reset as per every normal trading week. In this scenario, pending messages will likely be lost trade statues will need to be reconciled by clients from the Matching Database of Record. The state of all trades can always be obtained by via the retrieval of an interim Daily Confirmation Statement.

4.11 Workflows

4.11.1 FX Spot Orders

Below is the general workflow for an FX Spot order.



4.11.1.1 Order

A new order is submitted via a NewOrderSingle (35=D) or NewOrderList (35=E) FIX message.

4.11.1.2 Rejection

The order received by MAPI is validated to ensure it meets the requirements in this guide, that the user is correctly logged in and has appropriate permissions and that the service is in a state where it can accept new orders. This includes (but is not limited to) the following:

- Matching not in trading mode
- Prime Broker not online (for Prime Broker Clients only)
- Exceeds the maximum number of open orders
- Exceeds the order input throttle rate

- Symbol(55) tag containing an unknown non-permitted instrument
- Price (44) not being a multiple of the pip size
- OrderQty(38) not being within the minimum and maximum limits or a multiple of the lot size

Note: A rejection does not occur due to lack of liquidity. An attempt to submit an order at an unattractive price will still be accepted even if it does not result in an immediate match.

Any order that fails validation will be rejected. The exact FIX message type used to notify of the rejection will vary depending upon the specific error that triggered it. Any of the following FIX message types could be used:

- ExecutionReport (35=8)
- ListStatus (35=N)
- BusinessMessageReject (35=j)
- Reject (35=3)

For details of the specific errors raised in each scenario please see the *MAPI Error Codes and Reasons Guide*.

4.11.1.3 Order Acknowledgement

If the order is valid a new order acknowledgement is sent back confirming the order has been accepted. This will be using the message type `ExecutionReport (35=8)` with `OrdStatus(150)=0 [new]`.

The majority of tags within the message are merely echoing back the values present on the original order for verification of what was received by MAPI. However it will also include the `OrderID(37)` containing the ID number assigned by Thomson Reuters for this order

No order acknowledgement will be given for IOC orders though.

Non-IOC orders receive this acknowledgement in order to immediately confirm the order has been accepted and there will not be another message afterwards until the order either matches or is cancelled. Depending upon market conditions this means a long time could elapse before a message is sent.

IOC orders on the other hand always result in an immediate outcome and response; if they don't match immediately they will be cancelled automatically. This makes the order acknowledgment redundant for IOCs and so it is not sent for efficiency reasons.

4.11.1.4 Match Notification

When each match occurs a match notification will be immediately sent in the form of an `ExecutionReport (35=8)` with `ExecType(150)=F [Trade]`. This alerts that a match has taken place and confirms:

- `LastPx(31)` - The price matched
- `LastQty(32)` - The quantity matched
- `RootPartySubID(1121)` - The counterparty's TCID
- `TrdMatchID(880)` - The match ID

Each `ExecutionReport (35=8)` is only referring to a single match. When multiple matches occur, multiple `ExecutionReports (35=8)` will be generated. This means the details given in the report are exact – they are not aggregated/average values covering multiple matches.

All matches in Matching are final and the details will not be revised or undone.

Whilst a `TradeCaptureReport (35=AE)` may follow shortly with the settlement details, the price, quantity and counterparty will not change. This means an `ExecutionReport (35=8)` is sufficient notification to determine the match has occurred and that the quantity is no longer available to be cancelled or matched by others.

4.11.1.5 Settlement Details

As well as the match notification, each match will also result in the settlement details being sent in a `TradeCaptureReport (35=AE)`.

Unlike the `ExecutionReport (35=8)` though these message are not sent until Matching has been able to determine if the counterparty has also received their Match Notification. This ensures the `MatchStatus(573)` tag can be set appropriately.

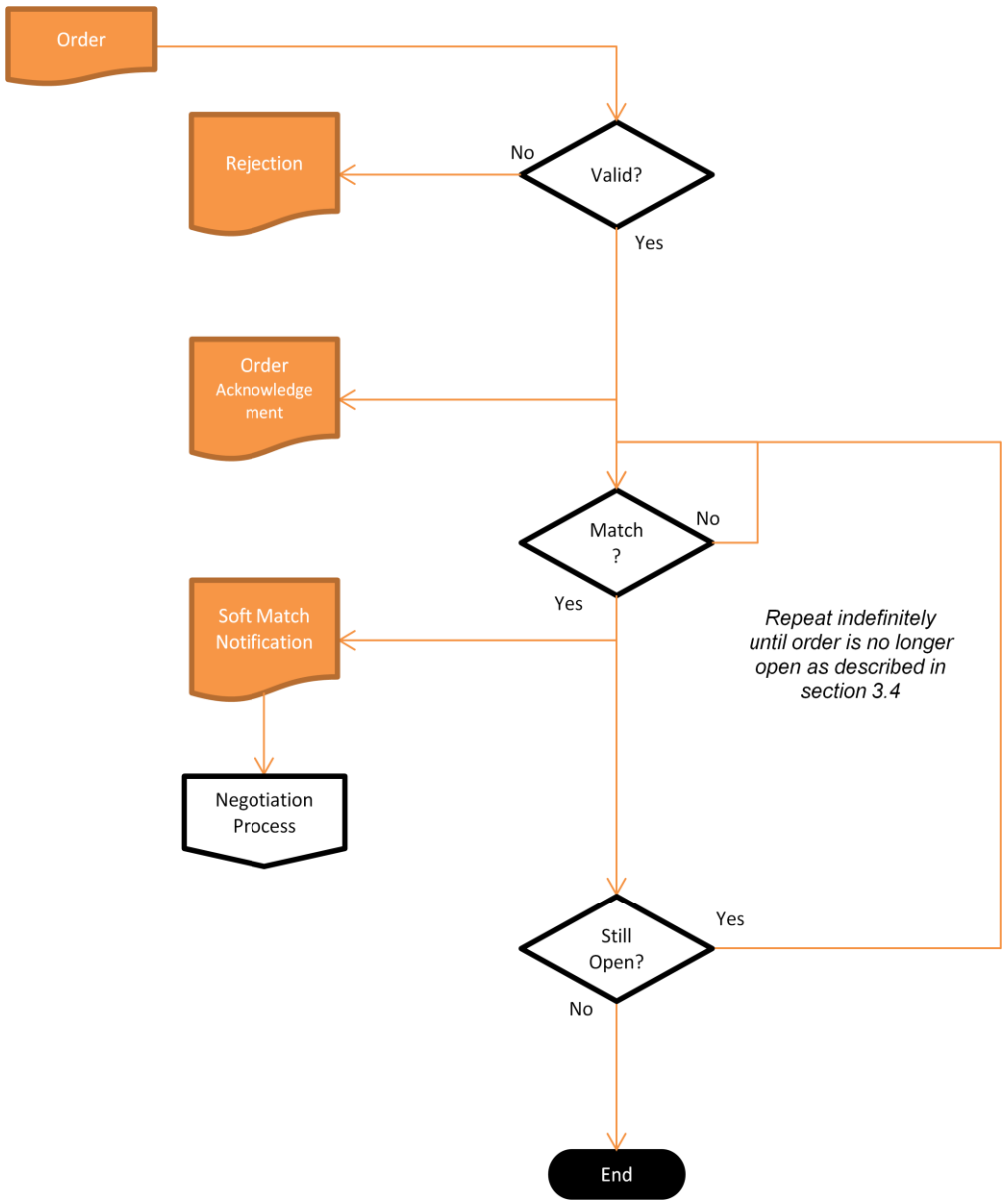
As a result it is possible for there to be a gap of up to 60 seconds between the `ExecutionReport (35=8)` and `TradeCaptureReport (35=AE)`. This in turn means that when there are many matches occurring in a short period of time it's possible for the `TradeCaptureReports (35=AE)` to be returned out of sequence or overlapping with subsequent `ExecutionReports (35=8)`.

Therefore applications should ensure they are not relying on the `ExecutionReports (35=8)` and `TradeCaptureReports (35=AE)` coming back in any particular sequence.

Note: Clients who also use a Drop Copy session can opt to have the TradeCaptureReports disabled on the trading session. In which case it will *only* be received on the Drop Copy session.

4.11.2 FX Forward Swap Orders

The initial stages of the FX Forward Swap workflow are very similar to the FX Spot workflow. Orders are submitted using the same message types and they are validated and acknowledged in the same manner.



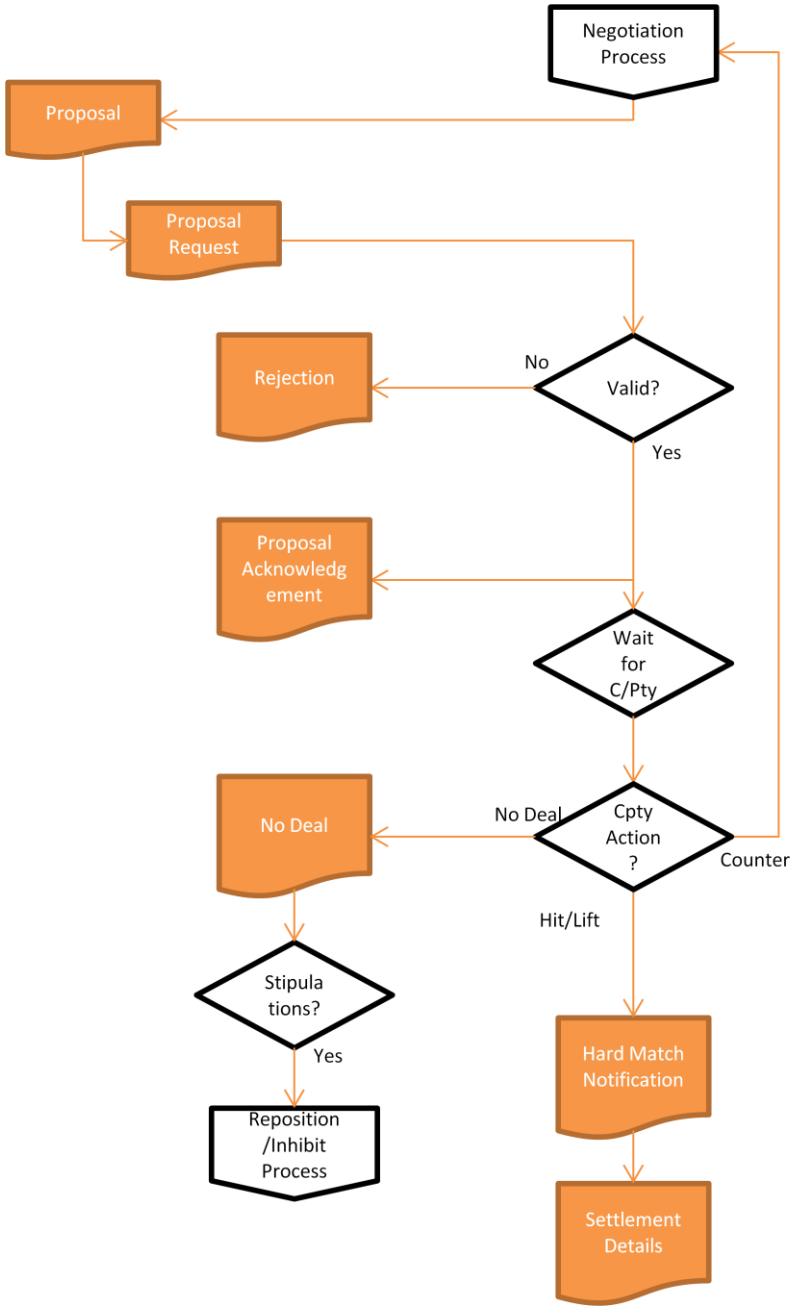
Note: FX Forward Swaps cannot be traded by Prime Broker Clients

4.11.2.1 Soft Match Notification

If the order matches a soft match rather than a hard match notification is sent (see Soft & Hard Matches for further details). This too is an `ExecutionReport (35=8)` confirming the price, quantity and counterparty but unlike the hard match the deal is not yet done.

A negotiation phase must then take place to accept, refine or reject the proposed soft match. Only the quantity may be altered during the negotiation. It is not possible to change the price or counterparty.

4.11.2.2 Negotiation Process



This negotiation can either be done via the API by the same trading user who submitted the order, or it can be sent to a nominated transfer user and the process completed on a manual keystation outside of MAPI.

The decision on which method to use must be made when submitting each individual order as it cannot be revised afterwards.

The remaining sections discuss the message workflow if the negotiation is completed via MAPI.

If a transfer user is nominated then the soft match notification is the final message received via MAPI. All of the negotiation messages and any hard matches that may result will then occur on the keystation and be invisible to MAPI.

4.11.2.3 Proposal

The Proposal is sent via a `Quote (35=S)` FIX message. It contains the proposed near and far leg quantities as the status of the current proposal:

- Indicative (the initial proposal from the soft match)
- Counter (an altered proposal from the counterparty)

4.11.2.4 Proposal Request

Once a proposal has been received the user needs to evaluate the risk and respond with a `QuoteResponse (35=AJ)` FIX message choosing whether to:

- Accept the proposed match
- Reject the match (No Deal)
- Propose new quantities

If proposing new quantities the values must still adhere to all the usual rules around quantities:

- Within the minimum and maximum limits for the instrument
- A multiple of the instrument's lot size
- Within the MOS set by the organisation

The quantity doesn't necessarily have to be negotiated downwards – it is possible to increase the quantities too. However in some scenarios there is an extra stipulation restricting the maximum permitted quantity. If this applies then the Soft Match notification will contain `StipulationType(233)=MAXORDQTY`, with the maximum limit given in `StipulationValue(234)`.

In order to ensure the proposal does not go on indefinitely there is a limit on the number of requests that can be made. If the proposal still hasn't been accepted by both parties after 50 requests, the system will automatically set it as No Deal.

4.11.2.5 Proposal Acknowledgement

The system will validate the proposal request and respond with a `QuoteStatusReport (35=AI)` containing the appropriate `QuoteStatus(297)`:

- Active
- Reject

An active response means the proposal is valid and the system is now waiting for the counterparty to respond.

A rejected response does *not* mean the counterparty has refused to accept the proposal. It simply means the proposal failed the validation. The reason should be examined, the mistake corrected and the proposal retried.

4.11.2.6 No Deal

In the event that the counterparty or the system ends the negotiation a `QuoteStatusReport (35=AI)` will be sent with `QuoteStatus(297)=17 [Cancelled]`.

This can be for any of the following reasons:

- Counterparty responded with "No Deal" in their `QuoteResponse (35=AJ)`

- User logged off
- End of trading week
- Maximum number of proposal requests were made and no agreement reached

When the negotiation results in no deal there are two other workflows that might need to be performed. This will be communicated by the stipulations given in `StipulationValue(234)` when `StipulationType(233)=TEXT`. The possible values are:

Stipulation	Requirement
Inhibit workflow required	An inhibit or do-not inhibit request *must* be sent The Reposition workflow is not allowed.
Reposition workflow required	A reposition or cancel request *must* be sent The Inhibit workflow may also optionally be followed
None	Neither the Inhibit nor the Reposition workflows are allowed

4.11.2.7 Hard Match Notification

If both sides accept the proposed quantities at any stage the soft match will become a hard match resulting in an `ExecutionReport (35=8)`.

4.11.2.8 Settlement Details

The settlement details will then follow the hard match notification in a `TradeCaptureReport (35=AE)`.

4.11.2.9 Inhibit Workflow

If a stipulation states that the inhibit workflow is required then the client must decide whether they would be willing to negotiate with the same party later if another match occurs.

The choice is either:

- Inhibit – No further matches would be allowed until next credit reset
- Do not inhibit – Further matches would be allowed

The decision is sent by the client using a `PartyActionRequest (35=DH)` message and acknowledged by MAPI with a `PartyActionReport (35=DI)`.

The inhibit workflow is stipulated in scenarios where the negotiation has failed but the client has choice on whether to reposition the soft match quantity.

4.11.2.10 Reposition Workflow

If a stipulation states that the reposition workflow is required then the client must decide what to do with the soft matched quantity now that the negotiation is over.

The choice is either:

- Do not reposition – the quantity will be cancelled
- Reposition – the quantity will be added back to the order allowing others to match it

To reposition an OrderCancelReplaceRequest (35=G) is sent by the client. If the decision is made to not reposition then an OrderCancelRequest (35=F) is sent instead. MAPI will respond with an OrderCancelReject (35=9) or an ExecutionReport (35=8) as appropriate.

The reposition workflow does not allow the client to choose how much of the quantity to reposition. The quantity involved determined by Matching automatically from the result of the negotiation.

4.11.2.11 Bell Notification

In the event a party doesn't respond to a proposal either way for some length of time it is possible to send a bell notification in order to remind them that the proposal is still active.

A bell is sent by the client using a QuoteStatusRequest (35=a) message. The recipient gets a QuoteStatusReport (35=AI) with QuoteStatus(297)=8 [Query] as a result.

4.11.3 Cancelling Orders

There are several ways in which to request cancellation of orders.

An OrderCancelRequest (35=F) message can be used to cancel a single order, or a single side of an I-Deal order by referring to the OrderID(37) or ClOrdID(11) of the original order.

Alternatively both sides of an I-Deal order can be cancelled with a single ListCancelRequest (35=K) message by referring to the ListID(66) of the original I-Deal.

Mass Action Requests (later in this document) can be used to cancel subsets of orders including by instrument, asset class or all orders.

Order cancellation takes priority over any other operation (such as order match, order book update, etc).

4.11.3.1 Pending Cancel

An ExecutionReport (35=8) will be sent with ExecType(150)=6 [Pending] to confirm the request has been received.

This does not necessarily mean the request will be successful though. The request may still subsequently fail due to MQL, incorrect details or because the order has already been filled.

4.11.3.2 Cancelled

Once the cancellation has successfully been performed a further ExecutionReport (35=8) will be sent with ExecType(150)=4 [Cancelled] to confirm the order is no longer open.

Unsolicited cancellations (triggered by the system rather than at the user's request) will also generate this message. In those scenarios the ExecRestatementReason(378) and/or Text(58) tags may provide details of the cause of the cancellation.

4.11.3.3 Cancel Reject

In the event the cancellation is unsuccessful an OrderCancelReject (35=9) will be sent to report it was unsuccessful and explain the reason why.

Note that MAPI only checks the given ID against currently open orders. If no open order exists with that ID it does use further resources to determine whether the ID refers to an unknown order or a known, but already closed order. Because of this the reason given in both scenarios would be CxlRejReason(102)=0 [Too Late to Cancel] regardless.

4.11.4 Mass Action Requests

It is possible to perform cancel, hold or release actions on multiple orders using an `OrderMassActionRequest (35=CA)` message.

The `OrderMassActionRequest (35=CA)` will be validated. If it is accepted an `OrderMassActionReport (35=BZ)` with `MassActionResponse(1375)=1 [Accepted]` then be sent confirming it has been received and is being processed.

`ExecutionReports (35=8)` will then be sent for each order matching the criteria (if no orders match the criteria, then no messages will be sent).

Finally another `OrderMassActionReport (35=BZ)` with `MassActionResponse(1375)=1 [Accepted]` will be sent marking the mass action request completed. This one will also include `TotalAffectedOrders(533)` confirming how many orders were affected.

Throttling limits apply to most mass action requests, and all are subject to MQL. See the relevant sections in chapter 3 for details.

4.11.4.1 Cancel

A mass action request for a cancelation can specify the additional restrictions:

- All orders
- All orders for a given instrument
- All orders for a given asset class
- All orders for a given asset class and side

(These are all *per trading user*. It is impossible to request cancellation of orders belonging to another user.)

4.11.4.2 Hold

Holding orders results in them being removed from the order book temporarily. Held orders can then later be “Released” back into the book.

A mass action request to hold an order cannot specify further restrictions limiting it to only a subset of orders. All orders belonging to the trading user will be processed.

However all orders for an instrument must be in the same state. It is not possible for a trading user to have a mixture of orders where some are held and some are not held on the same instrument. Therefore should any order fail to be held because its MQL has not yet expired, that order plus all other orders for the same instrument will not be held.

Once all orders on an instrument are held, any new orders (except for FX Spot IOCs) on the same instrument will automatically be held too.

4.11.4.3 Release

Releasing puts previously held orders back into the order book. Like the request to hold the orders this too cannot specify further restrictions limiting it only to a subset of orders.

The orders will get a new temporal position at the end of the book just like a newly placed order would do. However unlike cancelling and then resubmitting orders, the throttle rate for order entry does not apply when releasing held orders.

4.11.5 Prime Brokers

4.11.5.1 STU Management

Prime Brokers can use MAPI to login their Special Ticket Users (STUs) via FIX.

The STU logs in as an application user via the `UserRequest (35=BE)` message. Once the STU is logged on the PBCs attached to that STU will be allowed to trade provided:

- Matching is in trading mode (MTM:1)
- The PBC session controls are not otherwise preventing the client from trading

The STU will also receive the following messages:

- CTA alerts for both
  - PB -> PBC limits
  - PB -> other party limits
- Match Notifications corresponding to the Match Notification received by the PBC
- Settlement Details corresponding to the Settlement Details received by the PBC

Prime Brokers receive `ExecutionReports (35=8)` and `TradeCaptureReports (35=AE)` just like the PBCs do with almost identical contents. However a Prime Broker receives 2 of each message type:

- One for trade between PBC & PB
- One for trade between PB & counterparty

Whilst the Match ID, price & quantity of these trades will all be the same, the direction will be opposite to reflect it is being bought from the PBC and sold to the counterparty (or vice-versa).

Note: If the counterparty is actually another PBC belonging to the same TCID on the Prime Broker then there are still only 2 of each message type.

It does *not* cause 4 of each type to be generated – 2 for the one PBC and 2 for the other.

The `Account(1)` tag on the `ExecutionReports (35=8)` and `TradeCaptureReports (35=AE)` reflects the value used by the PBC when the order was submitted.

4.11.5.2 PBC Session Controls

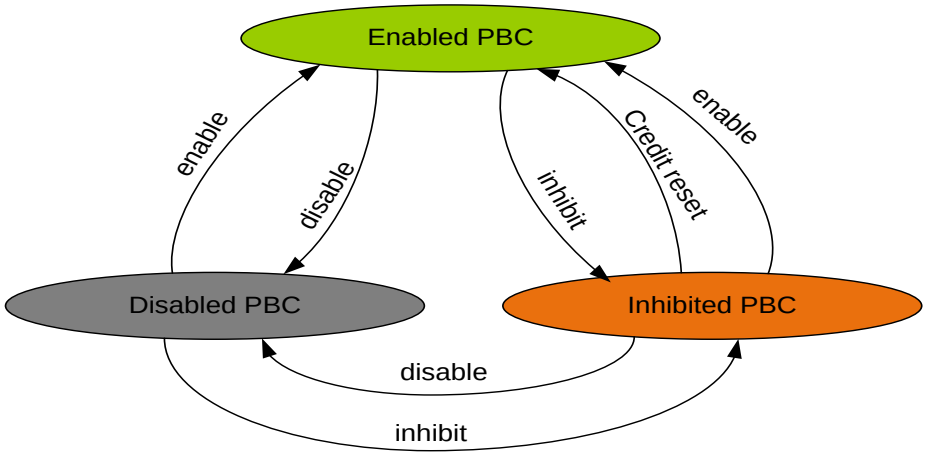
Prime Brokers can also utilise MAPI to:

- Inhibit a specific PBC
- Disable a specific PBC
- Enable a specific PBC

This provides an alternative way of halting a PBC from trading without needing to logout the STU user.

When inhibited a PBC is prevented from trading until the next credit reset.

When disabled a PBC is prevented from trading indefinitely and must be explicitly enabled for trading to recommence immediately, or moved back to inhibited where they will stay until the next credit reset.



Current state	Action	End state
Disabled PBC	User selects ‘inhibit’	Inhibited PBC
Disabled PBC	User selects ‘enable’	Enabled PBC
Enabled PBC	User selects ‘disable’	Disabled PBC
Enabled PBC	User selects ‘inhibit’	Inhibited PBC
Inhibited PBC	User selects ‘disable’	Disabled PBC
Inhibited PBC	User selects ‘enable’	Enabled PBC
Inhibited PBC	When credit is reset	Enabled PBC

The PBC session controls require the session `Logon (35=A)` to contain `DefaultApplExtID(1407)=171` and will need a credit administrator to login at the application level. This will result in the user receiving a `U3 (35=U3)` notification containing the current state for all the PBCs under that TCID.

To change the state a `PartyActionRequest (35=DH)` message is submitted. The response will be notified in a `PartyActionReport (35=DI)` message and assuming it is successful also result in `U3 (35=U3)` notifications being broadcast to inform the client of the change.

4.11.5.3 DOMCAD

Disable On MAPI Credit Admin Disconnect (DOMCAD) specifies what happens if there are no credit administrators logged in via MAPI.

If enabled then should all the credit administrators logout or get disconnected all the PBCs are automatically prevented from trading (analogous to the STU not being connected). If disabled then logouts or disconnects of the credit administrators have no effect.

Prime Brokers using PBC Session Controls via MAPI may want to consider enabling this setting as a safety measure. This would ensure that in the event there is no active connection over which the PBC session control messages can be sent trading will be halted automatically.

The DOMCAD setting cannot be changed via MAPI. It can only be enabled or disabled in the New Credit Admin GUI.

4.11.6 MOS & MOOPI

Interbank or Prime Brokers can change the MOS and MOOPI limits via MAPI. This will require the session `Logon (35=A)` to contain `DefaultApplExtID(1407)=171` and will need a credit administrator to login at the application level.

This will result in the user receiving a `U3 (35=U3)` notification including the current MOS & MOOPI limits for their organisation or PBCs.

To change the limits a `U1 (35=U1)` message is submitted. The response will be notified in a `U2 (35=U2)` message and assuming it is successful also result in `U3 (35=U3)` notifications being broadcast to inform the users of the change.

4.11.7 Drop Copy

The Drop Copy implementation can simply be thought of as a read-only FIX session. The client application performs the normal FIX session level functions (login, sequence number reconciliation etc.) but never sends any application level requests.

Instead once the session is successfully established the application just heartbeats at the appropriate intervals in order to keep the session alive.

Every time a match occurs (acknowledged and unacknowledged) on trading session though, MAPI will automatically send a copy of the `TradeCapttureReport (35=AE)` message on the Drop Copy session too. This message will be identical to the version received on the trading session except for the necessary amendments to make the FIX message legal in this session. That includes:

- BodyLength(9)
- MsgSeqNum(34)
- SenderCompID(49)
- SendingTime(52)
- CheckSum(10)

The only other application level messages a Drop Copy session will receive are notifications of:

- Trading mode state (MTM:0/MTM:1)
- Start/End of week countdown (SOTW:n/EOTW:n)

A Drop Copy session will not receive `ExecutionReports (35=8)` for the matches or any other messages relating to non-match activity. For instance it will not be informed when orders are placed, cancelled or rejected. Therefore whilst the Drop Copy session can be used for back office settlement due to the `TradeCaptureReports (35=AE)` it cannot be used to provide other functions such a remote audit trail.

4.11.7.1 Recovery

In the event that the drop copy session is not connected at the time the match occurs then the message will be persisted as the next outgoing sequence number. This will cause the message to be recovered as already described in the Missing Message Recovery section.

As this recovery mechanism is based on the FIX sequence numbers it means missing messages can only be recovered during the same trading week.

Note:	This mechanism is for genuine & unexpected recovery scenarios only (e.g. network disconnects, application crashes etc.). It is not acceptable to trigger a replay of already seen messages for other purposes.

The MAPI gateway remains online 1 hour after trading finishes. This provides the opportunity to reconnect and recovery missing messages even if a disconnection occurs in the last few seconds of trading time.

After this hour though, once the service goes offline it will no longer be possible to recover that week's trades via FIX anymore. The next time the service comes back online a new week will have started with sequence numbers back at 1/1. Any missing trades at this point will have to be indentified from a Daily Confirmation Sheet and recorded manually.

4.11.7.2 Drop Copy Deployment

Drop Copy sessions can be deployed in the many-to-one or one-to-many configurations.

Many-to-one:

Many trading FIX sessions target the same drop copy session. This is useful for consolidating the matches from many locations/applications into a single back office feed.

One-to-many:

The same trading session targets many drop copy sessions causing the match to be duplicated across each one. This is useful for redundancy or parallel running if the ability to duplicate the messages cannot be performed on the client end.

When a Drop Copy is used there is also the choice to have the TradeCaptureReports (35=AE) messages disabled on the original trading session. This means the trading application does not even have to see the TradeCaptureReport (35=AE) reducing the bandwidth and processing time.

4.11.7.3 FIX2TOF Adapter

Thomson Reuters Deal Tracker Server (DTS) provides the ability to connect to a drop copy session and will provide the matches as TOF instead. For further details please contact your Treasury Transactions Specialist.

Chapter 5 FIX Message Tables

This chapter contains tables listing the FIX message types and tags used by MAPI.

5.1 FIX Header and Routing Tags

The following table lists the FIX header tags defined in FIX 5.0 SP2.

The required column identifies those that are needed by MAPI. The *Action* column identifies whether MAPI will accept the message and process the tag, accept but ignore the tag, or reject the message.

Tag	Required?	Action	Description
BeginString(8)	Y	Processed	Always FIXT.1.1
BodyLength(9)	Y	Processed	
MsgType(35)	Y	Processed	
SenderCompID(49)	Y	Processed	Value provided by Thomson Reuters when environment credentials are issued
TargetCompID(56)	Y	Processed	Always “TR MATCHING”
OnBehalfOnCompID(115)	N	Rejected	
DeliverToCompID(128)	N	Rejected	
SecureDataLen(90)	N	Ignored	
SecureData(91)	N	Ignored	
MsgSeqNum(34)	Y	Processed	
SenderSubID(50)	N - Session level	Rejected	
	Y – Application level	Processed	Used to link back to application user that sent it.
SenderLocationID(142)	Logon (35=A) only	Processed	Value provided by Thomson Reuters when environment credentials are issued
TargetSubID(57)	Y	Processed	Always FXM
TargetLocationID(143)	N	Ignored	
OnBehalfOfSubID(116)	N	Ignored	
OnBehalfOfLocationID(144)	N	Ignored	
DeliverToSubID(129)	N	Ignored	

Tag	Required?	Action	Description
DeliverToLocationID(145)	N	Ignored	
PossDupFlag(43)	N	Processed	MAPI does not accept the following message types with this flag set: • NewOrderSingle (35=D) • NewOrderList (35=E) • UserRequest (35=BE)
PossResend(97)	N	Processed	MAPI does not accept the following message types with this flag set: • NewOrderSingle (35=D) • NewOrderList (35=E) • UserRequest (35=BE)
SendingTime(52)	Y	Processed	Maximum tolerance is +/- 120 seconds (i.e. 2 minutes) of the time on MAPI servers. Values exceeding this will cause the session to be logged out.
OrigSendingTime(122)	N	Processed	
XmlDataLen(212)	N	Ignored	
XmlData(213)	N	Ignored	
MessageEncoding(347)	N	Ignored	
LastMsgSeqNumProcessed(369)	N	Ignored	
NoHops(627)	N	Ignored	
HopCompID(628)	N	Ignored	
HopSendingTime(629)	N	Ignored	
HopRefID(630)	N	Ignored	
ApplVerID(1128)	N	Processed	9=FIX 5.0 SP2
CstmApplVerID(1129)	N	Ignored	
ApplExtID(1156)	N	Ignored	• 100 • 171

MAPI assumes that clients will connect directly and not via another hub. As such the tags OnBehalfOf* and DeliverTo* are not used.

5.2 Supported Messages

The table below lists all FIX message types supported by Thomson Reuters Matching: Any message received by the MFG that is not part of the list below will be rejected.

Type	Message Name	Message Description	FIX version
0	Heartbeat *	Used when data has not been sent for *HeartBtInt* seconds.	FIXT 1.1
1	Test Request *	Used to verify the connection between Reuters Matching and its counterparty is alive.	FIXT 1.1
2	Resend Request *	Used to initiate the retransmission of a message.	FIXT 1.1
3	Reject *	Response to messages that cannot be sent through to the application level (i.e.: incorrect message format, unknown message type, etc …)	FIXT 1.1
4	Sequence Reset *	Used to reset incoming sequence number on the opposing side.	FIXT 1.1
5	Logout *	Initiates or confirms the termination of a FIX session.	FIXT 1.1
8	Execution Report	Used to notify order receipts, order matches, order confirmations, order status.	FIX 5.0 SP2 EP100
9	Order Cancel Reject	Used to Reject an Order Cancel Request.	FIX 5.0 SP2
A	Logon *	Used to initiate and confirm a FIX session.	FIXT 1.1
D	New Order – Single	Used to place a SPOT order (Bid, Offer, Fill or Kill) onto Reuters Matching.	FIX 5.0 SP2 EP100
E	New Order List	Used to place an FX Spot or FX SWAP I-Deal order onto Reuters Matching.	FIX 5.0 SP2 EP100
F	Order Cancel Request	Used to cancel an existing individual or individual list entry order (i.e. one side of an I Deal).	FIX 5.0 SP2
j	Business Message Reject	Used as a response to an application message that will not be processed.	FIX 5.0 SP2
K	ListCancel	Used to cancel both sides of an I-Deal order.	FIX 5.0 SP2
AE	Trade Capture Report	Used to deliver Complete ticket confirmation to the end user.	FIX 5.0 SP2 EP100
CA	Order Mass Action Request	Used to Hold, Release or Cancel groups of orders.	FIX 5.0 SP2
BE	User Request	Used to provide client authentication	FIX 5.0 SP2
BF	User Response	Use to respond to a user request message and for Forced Logout Notification	FIX 5.0 SP2
BZ	Order Mass Action Report	Used to notify the result of Order Mass Action Request.	FIX 5.0 SP2

Type	Message Name	Message Description	FIX version
CB	User Notification	Used to provide user notifications. This message can be used to provide PBC notice of PB STU logout.	FIX 5.0 SP2
N	ListStatus	Used to provide acknowledgements and responses to ListCancelRequests and rejections for NewOrderList.	FIX 5.0 SP2
Q	DontKnowTrade	Used by clients to inform the MFG when they have received a trade execution report that they were not party to.	FIX 5.0 SP2
U1	U1	Used to request changes of Max Order Size (MOS) and/or Max Open Orders per Instrument (MOOPI) on a PBC branch.	FIXT 5.0 SP2 (custom message type)
U2	U2	Used to response the U1 message.	FIXT 5.0 SP2 (custom message type)
U3	U3	PBC session Control broadcast message	FIXT 5.0 SP2 (custom message type)
DH	PartyActionRequest	For FX swap, it is used to inhibit or to inform the Matching service not to inhibit a counterparty. For FX spot, it is used to enable or disable or inhibit a PBC branch.	FIXT 5.0 SP2 (EP171)
DI	PartyActionReport	Used to response the PartyActionRequest (DH) message.	FIXT 5.0 SP2 (EP171)
AJ	QuoteResponse	Used to negotiate the Leg quantities of a FX Swap deal with your counterparty	FIXT 5.0 SP2
S	Quote	Used to inform the MAPI user he is now in proposal with Matching counterparty	FIXT 5.0 SP2
AI	QuoteStatusReport	Used to report the status of a given FX Swap negotiation sequence	FIXT 5.0 SP2
G	Order Cancel/Replace Request	**FX Swap Reposition Workflow only** Used to reposition the remaining quantity from proposal back into the Matching Order Book. This message cannot be used more generally as Matching does not allow orders to be modified.	FIXT 5.0 SP2
a	Quote Status Request	Used to query the status of that quote with your counterparty	FIXT 5.0 SP2

Note: The messages flagged with * are session messages

5.2.1 Supported Tags

All tags listed within this chapter are supported by the MAPI. Any other valid tag it receives that is part of the FIX v5.0 SP2 standard but is not defined within this specification will be ignored.

The MFG will reject messages that do not contain FIX conditionally required or mandatory tags when they should.

The MFG will reject messages that do not contain conditionally required or mandatory tags required by Matching.

The MFG will reject messages that incorrectly specify conditionally required tags.

5.3 Session Level Messages

5.3.1 Heartbeat (35=0)

Tag	Required?	Description
TestReqID(112)	N	Echoed back when Heartbeat is a solicited response to TestRequest (35=1)

5.3.2 Logon (35=A), Client -> MAPI

Tag	Required?	Description
EncryptMethod(98)	Y	0 = None
HeartBtInt(108)	Y	4
ResetSeqNum(141)	N	Always N
NextExpectedMsqSeqNum(789)	Y	The next incoming sequence number expected by the client
Username(553)	Y	Username of session user
Password(554)	Y	Current password
NewPassword(925)	CR	**Only when changing password** The chosen new password, which conform to requirements listed in *Password Rules*
DefaultApplVerID(1137)	Y	9 = FIX 5.0 SP2
DefaultApplExtID(1407)	Y	• 100 • 171

5.3.3 Logon (35=A), MAPI -> Client

Tag	Required?	Description
EncryptMethod(98)	Y	0 = None
HeartBtInt(108)	Y	4
NextExpectedMsqSeqNum(789)	Y	The next incoming sequence number expected by the MAPI gateway
DefaultApplVerID(1137)	Y	9 = FIX 5.0 SP2
DefaultApplExtID(1407)	Y	• 100 • 171
SessionStatus(1409)	Y	0 = Session Active
Text(58)	Y	RFA:<*value*> The RFA token needed to authenticate with MDFD servers

5.3.4 Logout (35=5), Client -> MAPI

Requests logout for session user.

Tag	Required?	Description
Text(58)	N	Reason for request. Has no effect on the API but may be useful for logging purposes

5.3.5 Logout (35=5), MAPI -> Client

Successful response to Logout (35=5) request, or unsolicited logout initiated by MAPI FIX Gateway.

Tag	Required?	Description
Text(58)	CR	**For forced logouts initiated by the MAPI FIX Gateway only** Reason for the logout

5.3.6 Reject (35=3)

Tag	Required?	Description
RefSeqNum(45)	Y	MsgSeqNum(34) of the message being referred to
RefTagID(371)	N	The tag number of the field being referred to
RefMsgType(372)	N	The MsgType(35) of the message being referred to
SessionRejectReason(373)	N	0 = Invalid Tag Number 1 = Required Tag Missing 2 = Tag not defined for this message type 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 = SendingTime Accuracy Problem 11 = Invalid MsgType 12 = XML Validation Error 13 = Tag appears more than once 14 = Tag specified out of required order 15 = Repeating group fields out of order 16 = Incorrect NumInGroup count for repeating group 17 = Non "Data" value includes field delimiter (character) 18 = Invalid/Unsupported Application Version 99 = Other
Text(58)	N	Further explanation of the rejection

5.3.7 ResendRequest (35=2)

Tag	Required?	Description
BeginSeqNo(7)	Y	Start of the range to resend (inclusive)
EndSeqNo(16)	Y	End of the range to resend (inclusive) Use 0 to represent infinity

5.3.8 SequenceReset-GapFill (35=4)

Tag	Required?	Description
GapFillFlag(123)	Y	Always Y
NewSeqNo(36)	Y	The next MsqSeqNum(34) number to be used after this message

5.3.9 TestRequest (35=1)

Tag	Required?	Description
TestReqID(112)	Y	Identifier for this test request

5.4 Application Level Messages

5.4.1 BusinessMessageReject (35=j)

Tag	Required?	Description
RefSeqNum(45)	Y	MsgSeqNum(34) this is a response to
RefMsgType(372)	Y	MsgType(35) this is a response to
BusinessRejectRefID(379)	CR	The ID this message is a response to. Which particular ID is used depends upon the RefMsgType(372): D = ClOrdID(11) CA = ClOrdID(11) F = ClOrdID(11) K = ListID(66) E = ListID(66)
BusinessRejectReason(380)	Y	0 = Other 3 = Unknown Message Type 4 = Application Not available 6 = Not Authorized 7 = DeliverToFirm Not available at this time
Text(58)	N	Further details on the reason for the rejection

5.4.2 DontKnowTrade (35=Q), Client -> MAPI

Tag	Required?	Description
OrderID(37)	Y	OrderID(37) from the problem execution
ExecID(17)	Y	ExecID(17) from the problem execution
DKReason(127)	Y	A=Unknown Symbol B=Wrong Side C=Quantity Exceeds Order D=No Matching Order E=Price Exceeds Limit F=Calculation Difference Z=Other
Symbol(55)	Y	Symbol(55) from the problem execution
Side(54)	Y	1=Buy 2=Sell
OrderQty(38)	Y	OrderQty(38) from the problemexecution

5.4.3 ExecutionReport (35=8), MAPI -> Client

Tag	Required?	Description
ListID(66)	CR	**I-Deal Orders Only** Echoed back from the original order
OrderID(37)	Y	ID number assigned by Thomson Reuters to this order Can be “NONE”
ClOrdID(11)	CR	ID number assigned by the client.
OrigClOrdID(41)	CR	Echoed back from request
Side(54)	Y	Echoed back from original order
ExecID(17)	Y	ID number assigned to the event that caused this execution. Note this is not unique – single events that cause multiple executions will share the same ID
SecondaryExecID(527)	CR	**Only if ExecType(150)=F or I** The Trade ID assigned by Matching
ExecType(150)	Y	The reason for this execution report being sent: 0 = New 4 = Cancelled 5 = Replaced 6 = Pending Cancel 8 = Rejected 9 = Suspended [Hold] D = Restated E = Pending Replace F = Trade [hard match] I = Order Status [soft match] L = Triggered or Activated by System [Release]
SecurityType(167)	CR	**Only if ExecType(150) != 6, E** Echoed back from original order
Symbol(55)	Y	Echoed back from original order
Currency(15)	CR	**Only if ExecType(150) != 6, E** Echoed back from original order
LeavesQty(151)	Y	The quantity still open available for further execution
CumQty(14)	Y	The cumulative quantity matched so far for this order (this execution + previous executions)

Tag	Required?	Description
TradeDate(75)	CR	**Only if ExecType(150)=F or I** The GMT date the match occurred This is *not* rolled over into the next business day. It is simply the date portion of the TransactTime(60) tag.
AggressorIndicator(1057)	CR	**Only if ExecType(150)=F or I** Y = This order was the aggressor N = This order was passive
NoLegs(555)	CR	**Only for SecurityType(167)=FXSWAP** Repeating group containing the details for each leg
LegSymbol(600)	Y	Echoed back from Symbol(55) of the original order
LegCurrency(556)	Y	Volume currency for each leg
LegSide(624)	Y	1 = Buy 2 = Sell
LegOrderQty(685)	Y	Quantity of the base currency negotiated
LegRefID(654)	Y	• Near • Far
LegLastPx(637)	Y	All in price
LegCalculatedCcyLastQty(1074)	Y	The calculated quantity of the quoted currency. The value will be rounded *down* to a whole number. When PriceType = 20 this value can be derived from: [LegOrderQty * LegLastPx] * ContractMultiplier (1,000,000). When PriceType = 21 this value can be derived from: [LegOrderQty /LegLastPx]* ContractMultiplier (1,000,000).
TransactTime(60)	Y	The timestamp the event took place
OrdStatus(39)	Y	The status of the order: 0 = New 1 = Partially Filled 2 = Filled 4 = Cancelled 6 = Pending Cancel 8 = Rejected 9 = Suspended C = Expired [GTD/GFT] E = Pending Replace

Tag	Required?	Description
DisplayQty(1138)	CR	**More Quantity orders only and ExecType(150) != 6 [Pending Cancel]** Echoed back from original order
Price(44)	CR	Echoed back from original order
TargetStrategy(847)	CR	**Only if Iceberg order and ExecType(150)=0 [new], 4[cancelled], 8 [rejected], 9 [suspended],** Echoed back from original order
NoStrategyParameters(957)	CR	**Only if Iceberg order and ExecType(150)=0 [new], 8 [rejected]** Echoed back from original order
StrategyParameterName(958)	CR	**If and only if TargetStrategy(847)=1001** T = Tip Specification Type
StrategyParameterValue(960)	CR	**If and only if TargetStrategy(847)=1001** D = Defined R = Random
OrdRejReason(103)	N	Rejection reason: 1 = Unknown Symbol 11 = Unsupported Order Characteristic 13 = Incorrect Quantity 99 = Other
Account(1)	CR	**Only if ExecType(150) != 6** Echoed back from original order
TriggerType(1100)	CR	**Only if ExecType(150) != 6, E** Echoed back from original order
TriggerAction(1101)	CR	**Only if ExecType(150) != 6, E** Echoed back from original order
TriggerScope(1628)	CR	**Only if ExecType(150) != 6, E** Echoed back from original order
OrderQty(38)	Y	Echoed back from original order
ExecRestatementReason(378)	CR	**Only for unsolicited cancels** 5 = Partial decline of OrderQty (e.g. exchange initiated partial cancel) [remaining qty below minimum] 6= Cancel on Trading Halt 12 = Cancel on Connection Loss 13 = Cancel on Logout **Only for ExecType(150) = D** 99=Other

Tag	Required?	Description
ContactMultiplier(231)	CR	**Only if ExecType(150) != 6, E** Echoed back from original order
QtyType(854)	CR	**Only if ExecType(150) != 6, E** Echoed back from original order
OrdType(40)	CR	**Only if ExecType(150) != 6, E** Echoed back from original order
TimeInForce(59)	CR	**Only if ExecType(150) != 6, E** Echoed back from original order
ExpireTime(126)	CR	**Only if ExecType(150) != 6, E** Echoed back from original order
ExposureDuration(1629)	CR	**Only if ExecType(150) != 6, E** Echoed back from original order
ContingencyType(1385)	CR	**Only if ExecType(150) != 6, E** Echoed back from original order
NoPartyIDs(453)	CR	**Only if ExecType(150)=F or I** Repeating group containing details of this party & the counterparty Always 2
PartyID(448)	Y	Answerback of party
PartyIDSource(447)	Y	C = Generally accepted market participant identifier
PartyRole(452)	Y	13 = Order Origination Firm [this party] 56 = Acceptable Counterparty [counterparty]
NoPartySubIDs(802)	Y	Repeating group containing the TCIDs and usernames of trader and negotiation user For this party will provide 2 or 3 values: • TCID • Trader • FX Swap Negotiation user if specified and ExecType(150)=I If given, the negotiation user will be listed after the trader. For counterparty will provide only 1 value: • TCID
PartySubID(523)	Y	The TCID or user ID
PartySubIDType(524)	Y	2 = Person [Trader or negotiating users user ID] 25 = Location Desk [TCID]

Tag	Required?	Description
TrdMatchID(880)	CR	**Only if ExecType(150)=F, I or a response to the FX Swap reposition workflow** The Match ID assigned by Thomson Reuters
LastPx(31)	CR	**Only if ExecType(150)=F and SecurityType(167)=FXSPOT** The price of this match.
PriceType(423)	CR	**Only if ExecType(150)=F or I** 20 = Normal 21 = Inverse
LastSpotRate(194)	CR	**Only if ExecType(150)=F or I and SecurityType(167)=FXSWAP** Spot Reference Rate used by Matching for this match
NoStipulations(232)	CR	**Only if ExecType(150)=F and SecurityType(167)=FXSWAP** Repeating group specifying the finalisation workflows that are required now the negotiation is over
StipulationType(233)	Y	TEXT=Freeform text
StipulationValue(234)	Y	• Inhibit workflow required • Reposition workflow required • None
NoLimitAmt(1630)	CR	**Only if ExecType(150)=F and SecurityType(167)=FXSPOT** Repeating group containing credit limit information Always 1
LimitAmtType(1631)	Y	0 = Credit Limit
LastLimitAmt(1632)	Y	Credit consumed in the last match. Whole number of millions
LastLimitRemaining(1633)	Y	Credit now remaining in limit set by this party for the counterparty Whole number of millions
LimitAmtCurrency(1634)	Y	Currency credit is maintained in
Text(58)	CR	Further details for the rejection: MQL=Minimum Quote Life MQL:0= Minimum Quote Life on another order MQL:1=Minimum Quote Life on this order TRE:O:n:m=Throttle Rate Exceeded for orders STU:0=PB is offline MTM:0 =Non-trading mode

5.4.3.1 Usage of ExecRestatementReason(378), ExecType(150) and OrdStatus(39)

The following table identifies a set of scenarios (both solicited and unsolicited) in which cancels can occur along with the expected values for the subject FIX tags.

FIX Trading Scenario	Implementation
Matching enters into Non-Trade Mode	Tag 150 = 4 Tag 39 = 4 Tag 378 = 6
P2PS connection failure, User Logout, Forced User logout	Tag 150 = 4 Tag 39 = 4 Tag 378 = 13
FIX Client App Disconnects and or FIX session is lost	Tag 150 = 4 Tag 39 = 4 Tag 378 = 12
Standing Swaps orders that are cancelled as a result of a match when their associated transfer user is disconnected	Tag 150 = 4 Tag 39 = 4 Tag 378 = 13
A standing order that is linked as part of an cancel I-Deal order attribute is automatically cancelled when its opposite side fully trades away.	Tag 150 = 4 Tag 39 = 4 Tag 378 = 5
Automatic cancellation of PBC orders when their PB STU has been logged out or disconnected.	Tag 150 = 4 Tag 39 = 4 Tag 378 = 13
Order is cancelled after a partial match because the remaining qty is less than the minimum order qty (for a given market).	Tag 150 = 4 Tag 39 = 4 Tag 378 = 5
User requests a logout with open orders via FIX User Request message	Tag 150 = 4 Tag 39 = 4 Tag 378 = 13
FIX session manually closed via Thomson Reuters operations (user is forced to logout)	Tag 150 = 4 Tag 39 = 4 Tag 378 = 12
Hold Request results in potential MQL violation; order remains in Held state and the order state is “Restated”	Tag 150 = D – restated Tag 39 = 1 or 4 new or partially filled Tag 378 = 99 – Other
FIX Tag 378 is not used for the following scenarios	

New Order Positioned	Tag 150 = 0 Tag 39 = 0
New Order Rejected	Tag 150 = 8 Tag 39 = 8
Order Cancel Request Acknowledgment	Tag 150 = 6 Tag 39 = 6
Cancelled order requested by the user – (e.g. Cancel request is being confirmed as cancelled)	Tag 150 = 4 Tag 39 = 4
Timed Order Expires	Tag 150 = C Tag 39 = C
Suspended timed order expires	Tag 150 = C Tag 39 = C
Suspended Order that is Cancelled	Tag 150 = 4 Tag 39 = 4
Order(s) are placed on Hold	Tag 150 = 9 Tag 39 = 9
Order(s) releases from being suspended	Tag 150 = L Tag 39 = 0(New) or 1(Partial Fill) Whatever the state of the order prior to being suspended.

Table 1 : ExecRestatmentReason(378), EcecType(150) and OrdStatus(39)

5.4.4 ListStatus (35=K), MAPI -> Client

Tag	Required?	Description
ListID(66)	Y	Echoed back from the original request
ListStatusType(429)	Y	1 = Ack 2 = Response
NoRpts(82)	Y	Number of reports in this series
ListOrderStatus(431)	Y	4 = Cancelling 6 = All Done 7 = Reject
ContingencyType(1385)	CR	Echoed back from the original request
ListRejectReason(1386)	CR	5 = Unknown Order 11 = Unsupported order characteristic 99 = Other
RptSeq(83)	Y	Number of this report within the series
TransactTime(60)	CR	**Only when ListOrderStatus(431) != 7** Timestamp the request was processed
ListStatusText(444)	CR	**Only when ListOrderStatus(431)=7** Further details of the rejection • MQL • "List orders are unknown or may have already been filled or cancelled". • "List is incorrectly specified" • "List is incorrectly specified: ManualOrderIndicator(1028) must be the same for both orders" • "Order # 'N' is invalid".
TotNoOrders(68)	Y	Number of orders in the original request
NoOrders(73)	Y	Number of order statuses in this response
ClOrdID(11)	Y	Echoed back from the original request
OrderID(37)	CR	**Only if ListStatusType(429) = 2** ID number assigned by Thomson Reuters
CumQty(14)	Y	Total quantity field so far
OrdStatus(39)	Y	4 = Cancelled 6 = Pending Cancel 8 = Rejected
LeavesQty(151)	Y	Remaining quantity still open

Tag	Required?	Description
CxlQty(84)	Y	Total quantity cancelled
AvgPx(6)	Y	**If OrdStatus(39) = 4 or 6** Echoed back from Price(44) of original order. **If OrdStatus(39) = 8** 0
OrdRejReason(103)	CR	1 = Unknown Symbol 11 = Unsupported Order Characateristic 13 = Incorrect Quantity 99 = Other
Text(58)	CR	**Only if ListOrderStatus(431)= 7** Further details on the rejection

5.4.5 NewOrderSingle (35=D), Client -> MAPI

Tag	Required?	Description
ClOrdID(11)	Y	Order ID assigned by the client.
NoPartyIDs(453)	CR	**Swap only** Repeating group listing the negotiation users if transferring the negotiation to a keystation. Maximum 2. If not given for a Swap trade then negotiation will be carried out via the API by the same user that submitted this order.
PartyID(448)	Y	Username of the negotiation user
PartyIDSource(477)	Y	C = Generally accepted market participant identifier
PartyRole(452)	Y	53 = Trader mnemonic
Account(1)	N	Not interpreted by Matching. If provided will be echoed back on any ExecutionReport(35=8) or TradeCaptureReport(35=AE) messages.
TransactTime(60)	Y	Time the order request was initiated
TimeInForce(59)	N	0 = Day 3 = Immediate or Cancel (IOC) 6 = Good Till Date (GTD) A = Good For Time (GFT) If not specified then defaults to 0.

Tag	Required?	Description
ExpireTime(126)	CR	**If and only if TimeInForce(59)=6** Timestamp when the order will be automatically cancelled by Matching. Must not include milliseconds. Must not be sooner than the minimum duration set by the instrument. Must not be later than the end of the trading week.
ExposureDuration(1629)	CR	**If and only if TimeInForce(59)=A** Duration in seconds after which the order will be automatically cancelled by Matching. Must be within the min & max duration limits set by the instrument.
TriggerType(1100)	CR	**Required for Jobbing Mode / AOR orders** 5 = On Order Entry or Order Modification
TriggerAction(1101)	CR	**If and only if TriggerType(1100)=5** 3 = Cancel
TriggerScope(1628)	CR	**If and only if TriggerType(1100)=5** 4 = All orders for the given security and side (jobbing mode) 5 = All orders for the given security, side and price (AOR)
SecurityType(167)	Y	FXSPOT = FX Spot FXSWAP = FX Swap
Symbol(55)	Y	The instrument name. The correctly formatted value is given in DSPLY_NAME (FID 3) of the MARKET_BY_PRICE model in RFA.
Currency(15)	Y	Dealt currency
Side(54)	Y	1 = Buy 2 = Sell
OrdType(40)	Y	2 = Limit
OrderQty(38)	Y	Order Quantity Must be greater than the instruments minimum quantity , a multiple of the instruments lot size and less than the instruments MOS.
ContractMultiplier(231)	Y	Always 1000000 Unless otherwise specified all quantity related fields will be implicitly multiplied by this amount. e.g. OrderQty(38)=1 means 1,000,000 (1 mio) not just 1
QtyType(854)	Y	1 = Contracts
DisplayQty(1138)	N	The primary quantity in a More Quantity order. If zero, or equal to OrderQty(38) then not a More Quantity order.

Tag	Required?	Description
Price(44)	Y	The price For FX Swap orders it specifies the swap points not the all-in rate. Price must be a multiple of the pip size and not exceed the left or right decimal place limits.
TargetStrategy(847)	CR	**Required for Iceberg orders** 1001 = Matching Iceberg
NoStrategyParameters(957)	CR	**If and only if TargetStrategy(847)=1001** Repeating group containing the tip sizes and choice of delay range
StrategyParameterName(958)	CR	**If and only if TargetStrategy(847)=1001** T = Tip Specification Type
StrategyParameterValue(960)	CR	**If and only if TargetStrategy(847)=1001** D = Defined R = Random
StrategyParameterName(958)	CR	**If and only if StrategyParameterName(958)=T & StrategyParameterValue(960)=D** Q1 = 1st Tip (primary) quantity
StrategyParameterValue(960)	CR	**If and only if StrategyParameterName(958)=Q1** The primary quantity to be used for the 1st tip
StrategyParameterName(958)	N	Q2 = 2nd tip
StrategyParameterValue(960)	CR	**If and only if StrategyParameterName(958)=Q2** The primary quantity to be used for the 2nd tip
StrategyParameterName(958)	N	Q3 = 3rd tip
StrategyParameterValue(960)	CR	**If and only if StrategyParameterName(958)=Q3** The primary quantity to be used for the 3rd tip
StrategyParameterName(958)	CR	**If and only if StrategyParameterName(958)=T & StrategyParameterValue(960)=R** Qmin = Minimum random tip
StrategyParameterValue(960)	CR	**If and only if StrategyParameterName(958)=Qmin** The lower bound for the randomly chosen quantity
StrategyParameterName(958)	CR	**If and only if StrategyParameterName(958)=T & StrategyParameterValue(960)=R** Qmax = Maximum random tip
StrategyParameterValue(960)	CR	**If and only if StrategyParameterName(958)=Qmax** The upper bound for the randomly chosen quantity

Tag	Required?	Description
StrategyParameterName(958)	N	D = Delay If not specified then no delay is used and refreshes will be immediate
StrategyParameterValue(960)	CR	**If and only if StrategyParameterName(958)=D** The ID number of the chosen delay range
ManualOrderIndicator(1028)	Y	Y = Order entered by manual user N = Order entered by algorithm
LockedStatus(5007)	N	**Spot Only** Y = Yes N = No (default) Not permitted on I-Deal or IOC orders

5.4.6 NewOrderList (35= E), Client -> MAPI

Tag	Required?	Description
ListId(66)	Y	ID number for the i-Deal assigned by the client
BidType(394)	Y	1="Non Disclosed" style (e.g. US/European)
TotalNoOrders(68)	Y	Always 2
ContingencyType(1385)	Y	5=Bid and Offfer 6=Bid and Offer (OCO)
NoOrders(73)	Y	Repeating group listing the individual orders that the I-Deal is comprised of. Always 2
ClOrdID(11)	Y	Order ID assigned by the client.
ListSeqNo(67)	Y	Order number within the list. 1=1st order 2=2nd order
NoPartyIDs(453)	CR	**Swap only** Repeating group listing the negotiation users if transferring the negotiation to a keystation. Maximum 2. If not given for a Swap trade then negotiation will be carried out via the API by the same user that submitted this order.
PartyID(448)	Y	Username of the negotiation user
PartyIDSource(477)	Y	C = Generally accepted market participant identifier
PartyRole(452)	Y	53 = Trader mnemonic

Tag	Required?	Description
Account(1)	N	Not interpreted by Matching. If provided will be echoed back on any ExecutionReport(35=8) or TradeCaptureReport(35=AE) messages.
TransactTime(60)	Y	Time the order request was initiated
TimeInForce(59)	N	0 = Day 3 = Immediate or Cancel (IOC) 6 = Good Till Date (GTD) A = Good For Time (GFT) If not specified then defaults to 0.
ExpireTime(126)	CR	**If and only if TimeInForce(59)=6** Timestamp when the order will be automatically cancelled by Matching. Must not include milliseconds. Must not be sooner than the minimum duration set by the instrument. Must not be later than the end of the trading week.
ExposureDuration(1629)	CR	**If and only if TimeInForce(59)=A** Duration in seconds after which the order will be automatically cancelled by Matching. Must be within the min & max duration limits set by the instrument.
TriggerType(1100)	CR	**Required for Jobbing Mode / AOR orders** 5 = On Order Entry or Order Modification
TriggerAction(1101)	CR	**If and only if TriggerType(1100)=5** 3 = Cancel
TriggerScope(1628)	CR	**If and only if TriggerType(1100)=5** 4 = All orders for the given security and side (jobbing mode) 5 = All orders for the given security, side and price (AOR)
SecurityType(167)	Y	FXSPOT = FX Spot FXSWAP = FX Swap
Symbol(55)	Y	The instrument name. The correctly formatted value is given in DSPLY_NAME (FID 3) of the MARKET_BY_PRICE model in RFA.
Currency(15)	Y	Dealt currency
Side(54)	Y	1 = Buy 2 = Sell
OrdType(40)	Y	2 = Limit
OrderQty(38)	Y	Order Quantity

Tag	Required?	Description
ContractMultiplier(231)	Y	Always 1000000 Unless otherwise specified all quantity related fields will be implicitly multiplied by this amount. e.g. OrderQty(38)=1 means 1,000,000 (1 mio) not just 1
QtyType(854)	Y	1 = Contracts
DisplayQty(1138)	N	**Spot Only** The primary quantity in a More Quantity order. If zero, or equal to OrderQty(38) then not a More Quantity order.
Price(44)	Y	The price For FX Swap orders it specifies the swap points not the all-in rate. Price must be a multiple of the pip size and not exceed the left or right decimal place limits.
ManualOrderIndicator(1028)	Y	Y = Order entered by manual user N = Order entered by algorithm

5.4.7 OrderCancelReject (35=9), MAPI -> Client

Tag	Required?	Description
ClOrdID(11)	Y	Echoed back from the cancel request
SecondaryClOrdID(526)	CR	**Only for FX Swap Reposition Workflow** Thomson Reuters Match ID
OrderID(37)	Y	The Thomson Reuters assigned ID number for the order. If the value was not given in the cancel request and no open order exists with the given OrigClOrdID(41) then “NONE” will be returned.
OrigClOrdID(41)	CR	Echoed back from the original order
OrdStatus(39)	Y	0 = New 1 = Partially filled 8 = Rejected
ListID(66)	CR	**Only if I-Deal order** Echoed back from the original I-Deal
Account(1)	N	Echoed back from original order
TransactTime(60)	Y	Time this request was processed
CxlRejResponseTo(434)	Y	1 = Order cancel request 2 = Order cancel/replace request

Tag	Required?	Description
CxlRejReason(102)	Y	0 = Too late to cancel 99 = Other
Text(58)	CR	Further detail regarding the reason MQL=Minimum Quote Life

5.4.8 OrderCancelReplaceRequest (35=G), Client -> MAPI

Tag	Required?	Description
SecondaryClOrdID(526)	Y	The Match ID assigned by Thomson Reuters
ClOrdID(11)	Y	Unique ID assigned by the client for this request
OrigClOrdID(41)	Y	ClOrdID(11) from the original order
TransactTime(60)	Y	The time this request was initiated
Symbol(55)	Y	Symbol(55) from the original order
Side(54)	Y	Side(54) from the original order
OrderType(40)	Y	2 = Limit
Price(44)	Y	Price(44) from the original order
OrderQty(38)	Y	OrderQty(38) from the original order

5.4.9 OrderCancelRequest (35=F), Client -> MAPI

Tag	Required?	Description
SecondaryClOrdID(526)	CR	**Do not reposition requests only** The Match ID assigned by Thomson Reuters
OrigClOrdID(41)	CR	**If cancelling based on client assigned ID** ClOrdID(11) of the order
OrderID(37)	CR	**If cancelling based on Thomson Reuters assigned ID** OrderID(37) of the order
ClOrdID(11)	Y	Client assigned ID for *this* request Note this is not the ID of the order to cancel (see OrigClOrdID(11))
Symbol(55)	Y	Symbol(55) from original order
Side(54)	Y	Side(54) from original order
TransactTime(60)	Y	Time this request was initiated

Tag	Required?	Description
OrderQty(38)	Y	OrderQty(38) from original order

5.4.10 OrderMassActionReport (35=BZ), MAPI -> Client

Tag	Required?	Description
ClOrdID(11)	Y	Echoed back from the OrderMassActionRequest (35=CA)
MassActionReportID(37)	Y	ID assigned by Thomson Reuters for the request. Where two reports a generated from the same request this value will be the same in both reports.
MassActionType(1373)	Y	Echoed back from the OrderMassActionRequest (35=CA)
MassActionScope(1374)	Y	Echoed back from the OrderMassActionRequest (35=CA)
MassActionReponse(1375)	Y	0 = Rejected 1 = Accepted
MassActionRejectionReason(1376)	CR	**Only when MassActionReponse(1375)=0** 0 = Mass Action Not Supported 1 = Invalid or unknown security 5 = Invalid or unknown Security Type 99 = Other
TotalAffectedOrders(533)	CR	**Only on second instance of MassActionReponse(1375)=1** The number of orders successfully affected by the request
Symbol(55)	CR	Echoed back from the OrderMassActionRequest (35=CA)
SecurityType(167)	Y	Echoed back from the OrderMassActionRequest (35=CA)
Side(54)	CR	Echoed back from the OrderMassActionRequest (35=CA)
TransactTime(60)	Y	The timestamp the response was completed
Text(58)	CR	**Only when MassActionRejectionReason(1376)=99** Further details on the cause of the rejection
OrdersLockFilter(20020)	N	Echoed back from the OrderMassActionRequest (35=CA)

5.4.11 OrderMassActionRequest (35=CA), Client -> MAPI

Tag	Required?	Description
ClOrdID(11)	Y	Client assigned ID for this request
MassActionType(1373)	Y	1 = Hold orders 2 = Release orders 3 = Cancel orders
MassActionScope(1374)	Y	**MassActionType(1373)=3 only** 1 = All orders for an instrument 5 = All orders for an asset class **Any MassActionType(1373) value** 7 = All orders
Symbol(55)	CR	**MassActionScope(1374)=1 only** The instrument to cancel for
SecurityType(167)	CR	**MassActionScope(1374)=1 or 5 only** FXSPOT = FX Spot FXSWAP = FX Forward Swap
Side(54)	N	**MassActionType(1373)=3 only** Optionally restrict cancellations to a single side 1 = Buy 2 = Sell
TransactTime(60)	Y	The time this request was initiated
OrdersLockFilter(20020)	N	**MassActionType(1373)=3 only** Whether locked orders are cancelled 1 = Non-locked only [default] 2 = Locked Only 3 = Both locked and non-locked

5.4.12 PartyActionReport (35=DI), MAPI -> Client

Tag	Required?	Description
PartyActionRequestID(2328)	Y	Echoed back from the original request
PartyActionReportID(2331)	Y	Unique identifier of this report
PartyActionType(2329)	Y	0 = Suspend [inhibited] 2 = Reinstate [uninhibited]
PartyActionReponse(2332)	Y	0 = Accepted [pending] 1 = Completed 2 = Rejected
PartyActionRejectReason(2333)	CR	**Only if PartyActionReponse(2332)=2** 0 = Invalid Party 1 = Unknown Requesting Party 98 = Not Authorised 99 = Other
RejectText(1328)	CR	**Only when PartyActionRejectReason(2333)= 99** Further details about the rejection
NoPartyIDs(453)	Y	Repeating block containing the list of parties to perform this request on. Always 1.
PartyID(448)	Y	The answerback of the party
PartyIDSource(447)	Y	C = Generally Accepted Market Participant
PartyRole(452)	Y	56 = Acceptable counterparty
NoPartySubIDs(802)	Y	Repeating group containing the TCID for this party Always 1
PartySubID(523)	Y	Party's TCID
PartySubIDType(803)	Y	25 = Location Desk
TransactTime(60)	Y	Timestamp the request was processed

5.4.13 PartyActionRequest (35=DH), Client -> MAPI

Tag	Required?	Description
PartyActionRequestID(2328)	Y	Unique ID for this request
InstrumentScopeSecurity(1547)	CR	**Required for PBC Session Control** FXSPOT = PBC Session Control FXSWAP = Swap inhibit workflow [default]
PartyActionType(2329)	Y	**For InstrumentScopeSecurity(1547)=FXSPOT** 0 = Suspend [inhibit] 1 = Halt Trading [disable] 2 = Reinstate [enable] **For InstrumentScopeSecurity(1547)=FXSWAP** 0 = Suspend [inhibit] 2 = Reinstate [do not inhibit]
NoPartyIDs(453)	Y	Repeating block containing the list of parties to perform this request on. Always 1.
PartyID(448)	Y	The answerback of the party
PartyIDSource(447)	Y	C = Generally Accepted Market Participant
PartyRole(452)	Y	**For InstrumentScopeSecurity(1547)=FXSPOT** 3 = Client ID **For InstrumentScopeSecurity(1547)=FXSWAP** 56 = Acceptable counterparty
NoPartySubIDs(802)	Y	Repeating group containing the TCID for this party Always 1
PartySubID(523)	Y	Party's TCID
PartySubIDType(803)	Y	25 = Location Desk
OrderQty(38)	N	New limit to use for Maximum order size (MOS) Valid values: 0 – 999
MaximumOpenOrdersPerInstrument(20003)	N	New limit to use for Maximum Open Orders per Instrument (MOOPI) Valid values: 1 – 200

5.4.14 QuoteMessage (35=S), MAPI->Client

Tag	Required?	Description
QuoteID(117)	Y	Thomson Reuters Match ID from soft match Uniquely identifies the proposal
QuoteType(537)	Y	0 = Indicative 3 = Counter
BidSpotRate(188)	Y	The spot reference rate
OfferSpotRate(190)	Y	The spot reference rate
Symbol(55)	Y	Echoed back from original order
TransactTime(60)	Y	Timestamp the proposal was processed
OrderQty(38)	Y	Echoed back from original order
NoStipulations(232)	N	Repeating group listing any stipulations that must be followed during the negotiation
StipulationType(233)	Y	MAXORDQTY = Maximum Order Size
StipulationValue(234)	Y	Maximum quantity that can be negotiated
NoLegs(555)	CR	**Only if QuoteRespType(694) != 6** Repeating group containing proposed quantities for each leg Always 2
LegSymbol(600)	Y	Echoed back from original quote
LegCurrency(556)	Y	Echoed back from original quote
LegOrderQty(685)	Y	Proposed quantity for this leg
LegRefID(654)	Y	• Near • Far

5.4.15 QuoteResponseMessage (35=AJ), Client -> MAPI

Tag	Required?	Description
QuoteRespID(693)	Y	ID assigned by the client for the response
QuoteID(117)	Y	Echoed back from original quote
QuoteRespType(694)	Y	1 = Hit/Lift 2 = Counter 6 = Pass [No Deal]

Tag	Required?	Description
ClOrdID(11)	CR	**Only if QuoteRespType(694)=1 or 2** ClOrdID(11) of the original soft match
Symbol(55)	Y	Echoed back from original quote
NoLegs(555)	CR	**Only if QuoteRespType(694) != 6** Repeating group containing proposed quantities for each leg Always 2
LegSymbol(600)	Y	Echoed back from original quote
LegCurrency(556)	Y	Echoed back from original quote
LegOrderQty(685)	Y	Proposed quantity for this leg
LegRefID(654)	Y	• Near • Far
Text(58)	CR	**Only if QuoteRespType(694)= 6** Reason for no deal This text will be passed onto the counterparty
OrderQty(38)	Y	Echoed back from the original order
Side(54)	Y	Echoed back from the original order

5.4.16 QuoteStatusRequest (35=a), Client -> MAPI

The Quote Status Request is used to query the **status** of that **quote** with your counterparty. Please note the counterparty can be a manual UI trader or another MAPI client application.

Tag	Required?	Description
QuoteID(117)	Y	QuoteID(117) of the original proposal
QuoteStatusReqID(649)	Y	Unique identifier for this request

5.4.17 QuoteStatusReport (35=AI), MAPI -> Client

Tag	Required?	Description
QuoteRespID(693)	CR	**Only if a solicited response to QuoteResponseMessage (35=AJ)** ID of the request this is a response to
QuoteReqID(649)	CR	**Only if QuoteStatus(297)=8** Echoed from the original QuoteStatusRequest (35=a) message
QuoteID(117)	Y	Echoed back from original quote

Tag	Required?	Description
QuoteStatus(297)	Y	0 = Accepted 5 = Rejected 8 = Query [Bell notification to counterparty] 16 = Active 17 = Cancelled [No Deal]
TransactTime(60)	Y	Timestamp the request was processed
QuouteRejectReason(300)	CR	**Only if QuoteStatus(297)=5** 99 = Other
Text(58)	CR	**Only if QuoteStatus(297)=17 [No Deal]** String of the form <Sys|User>:<Further detail> Sys = No deal by the system • Sys:Comms fail • Sys:Forced shutdown • Sys:Forced logoff • Sys:Host entered non-trading mode • Sys:User logoff • Sys:MAX_NUM_PROPOSALS_EXCEEDED User = No deal by the user • User:<reason given by counterparty>
NoStipulations(232)	CR	**Only if QuoteStatus(297)=17 [No Deal]** Repeating group listing any finalisation workflows that must be followed after the no deal
StipulationType(233)	Y	TEXT = Freeform text
StipulationValue(234)	Y	• Inhibit workflow required • Reposition workflow required • None

5.4.18 TradeCaptureReport (35=AE)

Tag	Required?	Value
TradeID(1003)	Y	Matching's Trade ID for this match.
TradeReportTransType(487)	Y	0 = New
MatchStatus(573)	Y	0 = Acknowledged match 1 = Unacknowledged match
TradeReportType(856)	Y	2 = Accept

Tag	Required?	Value
UnsolicitedIndicator(325)	Y	Y = Message is being sent unsolicited
TrdType(828)	Y	Always: 54 = OTC Trade
TrdMatchID(880)	Y	Matching's Match ID for this match
TradeDate(75)	Y	The GMT date the match occurred. Note this does not take account of the rollover time for the instrument. It will always be just the date portion of TransactTime(60).
ExecID(17)	Y	ID for the execution event. This will *not* be unique when multiple matches occurs as the result of the same event.
ExecType(150)	Y	Always: F = Trade
SettlDate(64)	CR	**Spot only** The value/settlement date. This will take account of the instruments rollover time and holidays.
Symbol(55)	Y	The Matching instrument
SecurityType(167)	Y	FXSPOT = FX Spot FXSWAP = FX Forward Swap
Currency(15)	Y	Base currency of the instrument
SettlCurrency(120)	Y	Quoted currency of the instrument
ContractMultiplier(231)	Y	Except where otherwise stated, any quantity related fields should be multiplied by this amount in order to determine the correct amount.
QtyType(854)	Y	1 = Contracts
LastSwapPoints(1071)	CR	**Swap only** The swap points for this match
LastQty(32)	Y	Quantity for this match
LastPx(31)	Y	Price for this match. For Swaps this is the all-in price of the far leg
PriceType(423)	Y	20 = Normal 21 = Inverse
LastSpotRate(1056)	CR	**Swap only** Spot Reference Rate for this match

Tag	Required?	Value
CalculatedCcyLastQty(1056)	CR	**Spot only** The calculated quantity of the quoted currency This field has already been multiplied by the ContractMultiplier(231).
NoLegs(555)	CR	**Swap only** Repeating group always containing two instances – one for each leg of the swap.
LegSymbol(660)		Same as Symbol(55) from the original order
LegCurrency(556)		Volume currency
LegSide(624)		1 = Buy 2 = Sell
LegQty(687)		Hard matched quantity
LegRefID(654)		Possible values: Near Far
LegSettlDate(588)		Value/settlement date
LegLastPx(637)		All-in price
LegCalculatedCcyLastQty(1074)		The calculated quantity of the quoted currency This field has already been multiplied by the ContractMultiplier(231). Value will be rounded *down* to a whole number.
NoRootPartyIDs(1116)	Y	Repeating group where each instance represents a party in the trade. Always 2 instances, one for this organisation and one for the counterparty
RootPartyID(1117)	Y	The Answerback of the party
RootPartyIDSource(1118)	Y	Always: C = Generally accepted market participant identifier
RootPartyRole(1119)	Y	Possible values: 13 = This organisation 56 = The counterparty
NoRootPartySubIDs(1120)	Y	Repeating subgroup containing further details of the participant. This organisation will always have 2 instances, the counterparty will have only 1.
RootPartySubID(1121)	Y	Either the TCID or trader's username depending upon RootPartySubIDType(1122).

Tag	Required?	Value
RootPartySubIDType(1122)	Y	Possible values: 2 = Trader's username 25 = TCID
NoSides(552)	Y	Repeating group where each instance represents a leg in the trade. Spot trades will always have one instance, Swap trades will have 2.
Side(54)	Y	Possible values: 1 = Buy 2 = Sell
SideCurrency(1154)	CR	Swap only Volume currency
Account(1)	N	Echoed back from the original order
NoLimitAmts(1630)	CR	**Spot only** Repeating subgroup containing credit limit information Always 1 instance
LimitAmtType(1631)	Y	Always: 0 = Credit Limit
LastLimitAmt(1632)	Y	Credit consumed by this match
LimitAmtRemaining(1633)	Y	Credit remaining
LimitAmtCurrency(1634)	Y	Currency in which credit limit is managed
NoSettlDetails(1158)	Y	Repeating subgroup containing settlement details for each party. Always 2 instances, the first is for this party, the second is for the counterparty.
SettlObligSource(1164)	Y	4 = Buyer's settlement instructions 5 = Seller's settlement instructions
NoSettlPartyIDs(781)	Y	Repeating subgroup containing the party's settlement instructions. Always 1 instance.
SettlPartyID(782)	Y	The settlement instructions specified by the party. If no details are specified then the contents will be: "No Instructions Specified"
SettlPartyIDSource(783)	Y	C = Generally accepted market participant identifier
SettlPartyRole(784)	Y	27 = Buyer/Seller (Receiver/Deliverer) 86 = CLS Member Bank

Tag	Required?	Value
NoSettlPartySubIDs(801)	N	Repeating subgroup containing identifiers for settlement party. If present always only 1 instance
SettlPartySubID(785)		Party's Swift BIC
SettlPartySubIDType(786)		16 = BIC
AggressorIndicator(1057)	Y	Y = This order was the aggressor N = This order was passive
OrderID(37)	Y	Order ID assigned by Matching
ClOrdID(11)	Y	Order ID assigned by client
OrderQty(38)	Y	Echoed back from the original order
ExpireTime(126)	N	Echoed back from the original order

5.4.19 UserNotification (35=CB), MAPI -> Client

Tag	Required?	Description
UserStatus(926)	Y	6 = Other
Text(58)	Y	CTA:<TCID>:<mio USD remaining> = Low Credit Threshold Alert EOTW:<minutes> = Countdown to End of Trading Week GTRC:<n>:<m> = Global Throttle Rate Change override – new rate is *n* orders in *m* seconds GTRC:0:0 = Global Throttle Rate removed – normal rate restored MTM:0 = Non-trading mode MTM:1 = Trading mode SOTW:<minutes> = Countdown to Start of Trading Week STU:0 = PB STU Offline STU:1 = PB STU Online

5.4.20 UserRequest (35=BE), Client -> MAPI

Tag	Required?	Description
UserRequestID(923)	Y	Unique ID assigned by the requestor
UserRequestType(924)	Y	1 = Log On user 2 = Log Off User 3 = Change Password For User Note UserRequestType(924)=3 can only be used when the user is *already* logged in. In order to change the *initial* password the change must be done as part of the UserRequestType(924)=1 step instead.
Username(553)	Y	Username
Password(554)	CR	**Only if UserRequestType(924)=1, 3** The current password
NewPassword(925)	CR	**Only if changing the password** The new password

5.4.21 UserResponse (35= BF), MAPI -> Client

Tag	Required?	Description
UserRequestID(923)	Y	Echoed back from original request If an unsolicited forced logout then this will be generated by MAPI and refer to the event that triggered the logout.
Username(553)	Y	Username
UserStatus(926)	Y	**1** = Logged in 2 = Not Logged In 3 = User Not Recognized 4 = Password Incorrect 5 = Password Changed 6 = Other 7 = Forced user logout by Exchange
UserStatusText(927)	CR	Further details for the reason for failure

5.4.22 U1 (35=U1)

Tag	Required?	Description
UserRequestID(923)	Y	Unique ID for this request
NoPartyIDs(453)	Y	Repeating block containing the list of parties to perform this request on. Always 1.
PartyID(448)	Y	The answerback of the party
PartyIDSource(447)	Y	C = Generally Accepted Market Participant
PartyRole(452)	Y	3 = Client ID
NoPartySubIDs(802)	Y	Repeating group containing the TCID for this party Always 1
PartySubID(523)	Y	Party's TCID
PartySubIDType(803)	Y	25 = Location Desk
OrderQty(38)	N	New limit to use for Maximum order size (MOS) Valid values: 0 – 999
MaximumOpenOrdersPerInstrument(20003)	N	New limit to use for Maximum Open Orders per Instrument (MOOPI) Valid values: 1 - 200

5.4.23 U2 (35=U2)

Tag	Required?	Description
UserRequestID(923)	Y	Echoed back from original request
PbcUpdateResultStatus(20005)	Y	0 = Success” 1 = Fail Value Too Large 2 = Fail Value Too Small 3 = Credit entity not found for Prime Broker; 4 = Invalid prime broker; 5 = Invalid prime broker client 6 = PBC does not belong to PB 7 = Global permission required; 8 = Local permission required 9 = Credit entity not found for Prime Broker client 10 = Inconsistent updates for MAPI 11 = Invalid NCG state 12 = Invalid max open orders per instrument 13 = Invalid Request 14 = Permission denied 15 = Fail others 16 = unknown
Text(58)	N	Further details if PbcUpdateResultStatus(20005) is a negative reason

5.4.24 U3 (35=U3)

Tag	Required?	Description
DOMCAD(20006)	N	True if CreditAdminStatus(20011)=2 would prevent the party from trading
PartyStatusRptID(20013)	Y	Identifier for this report If report is fragmented over several messages this value will be non-unique
RptSeq(83)	Y	Sequence number within this PartyStatusRptID(20013)
NoRpts(82)	Y	Total number of messages within this PartyStatusRptID(20013)
LastFragment(893)	CR	Only when RptSeq(83) = NoRpts(82) True if this is the last message within this PartyStatusRptID(20013)
NoPBCs(20009)	N	Repeating group containing the PBCs covered by this message

Tag	Required?	Description
PBCID(20012)	Y	String in the format: <TCID>*<Answerback>
NoPartyIDs(453)	Y	Repeating block containing the list of parties to perform this request on. Always 1.
PartyID(448)	Y	The answerback of the party
PartyIDSource(447)	Y	C = Generally Accepted Market Participant
PartyRole(452)	Y	3 = Client ID
NoPartySubIDs(802)	Y	Repeating group containing the TCID for this party Always 1
PartySubID(523)	Y	Party's TCID
PartySubIDType(803)	Y	25 = Location Desk
BranchStatus(20010)	N	1 = Enabled 2 = Disabled 3 = Inhibited
CreditAdminStatus(20011)	N	1 = PB CA is logged in 2 = PB CA is not logged in
OrderQty(38)	N	Maximum Order Size (MOS)
MaxOpenOrdersPerInstrument(20003)	N	Maximum Open Orders Per Instrument (MOOPI)

Chapter 6 Example Workflows

6.1 FIX Interface Workflows

6.1.1 Standard FX Spot Limit Order Entry

The following examples show orders entered with a primary quantity and more quantity equal to zero.

6.1.1.1 Order Filled after order resting in the book:

Msg Received	Msg Sent	Exec Type	OrdStatus	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comments
NewOrderSingle				100				
	ExecutionReport	New (0)	New (0)	100	0	100	0	
	ExecutionReport	Trade (F)	Partially Filled (1)	100	20	80	20	Execution of 20
	ExecutionReport	Trade (F)	Partially Filled (1)	100	30	70	10	Execution of 10
	ExecutionReport	Trade (F)	Filled (2)	100	100	0	70	Execution of 70

Order Part-filled after resting in the book and results in an unsolicited cancel.

Cancel can be a result of a user disconnect, admin disconnect or of end of trading session cancels.

Msg Received	Msg Sent	Exec Type	OrdStatus	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comments
NewOrderSingle				100				
	ExecutionReport	New (0)	New (0)	100	0	100	0	
	ExecutionReport	Trade (F)	Partially Filled (1)	100	20	80	20	Execution of 20
	ExecutionReport	Trade (F)	Partially Filled (1)	100	30	70	10	Execution of 10
	ExecutionReport	Cancelled (4) **	Cancelled (4) **	100	30	0	0	0

** End of Trading session or TimeInForce expiration will result in an ExecType and OrdStatus of Expired (C).

Order immediately filled upon hitting the book

Msg Received	Msg Sent	Exec Type	OrdStatus	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comments
NewOrderSingle				100				
	ExecutionReport	New (0)	New (0)	100	0	100	0	
	ExecutionReport	Trade (F)	Filled (2)	100	100	0	100	Execution of 100

Order partially filled upon hitting the book

Msg Received	Msg Sent	Exec Type	OrdStatus	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comments
NewOrderSingle				100				
	ExecutionReport	New(0)	New(0)	100	0	100	0	
	ExecutionReport	Trade (F)	Partially Filled (1)	100	70	30	70	Execution of 70

Spot order rejected upon entry

Msg Received	Msg Sent	Exec Type	OrdStatus	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comments
NewOrderSingle				100				
	ExecutionReport	Rejected (8)	Rejected (8)	100	0	0	0	Order Rejected by MFG

Spot limit order entry with More Quantity = Order Quantity

This example also covers the cases where DisplayQty = 0 or DisplayQty not present.

OrderQty = 100

DisplayQty = 100 | DisplayQty = 0 | DisplayQty not present.

Translates into our Matching Message Spec as:

Primary Qty = 100

More Qty = 0

Msg Received	Msg Sent	Exec Type	OrdStatus	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comments
NewOrderSingle				100				
	ExecutionReport	New (0)	New (0)	100	0	100	0	
	ExecutionReport	Trade (F)	Partially Filled (1)	100	20	80	20	Execution of 20
	ExecutionReport	Trade (F)	Partially Filled (1)	100	30	70	10	Execution of 10
	ExecutionReport	Trade (F)	Filled (2)	100	100	0	70	Execution of 70

.

New Order Single Limit Order with Display Qty = 30 and Order Qty = 100

Example the order matches for DisplayQty only. More Qty is cancelled.

Original Order specifies:

- OrderQty = 100
- DisplayQty = 30

Translates into our Matching Message Spec as:

- Primary Qty = 30
- More Qty = 70

Msg Received	Msg Sent	Exec Type	OrdStatus	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comments
NewOrderSingle				100				
	ExecutionReport	New (0)	New (0)	100	0	100	0	
	ExecutionReport	Trade (F)	Partially Filled (1)	100	20	80	20	Execution of 20
	ExecutionReport	Trade (F)	Cancelled (4) *	100	30	0	10	Execution of 10

* Indicates that the remaining more qty (70 mio) was cancelled. Notice the cancelled status allows the LeavesQty to be zero rather than OrderQty – CumQty.

New Order with Display Qty = 30 and Order Qty = 100

Multi-way match (MWM) for a quantity greater than the DisplayQty, which in turn results in a portion of the More Qty being consumed across multiple counterparties.

Msg Received	Msg Sent	Exec Type	OrdStatus	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comments
NewOrderSingle				100				
	ExecutionReport	New (0)	New (0)	100	0	100	0	
	ExecutionReport	Trade (F)	Partially Filled (1)	100	20	80	20	Execution of 20
	ExecutionReport	Trade (F)	Partially Filled (1)	100	40	60	20	Execution of 20 (MWM)
	ExecutionReport	Trade (F)	Partially Filled (1)	100	60	40	20	Execution of 20 (MWM)
	ExecutionReport	Trade (F)	Cancelled (4)*	100	70	0	10	Execution of 10 (MWM)

* Multi-way matches do not consume full value of more qty. 30 mio of the OrderQty is cancelled.

New Order with Display Qty = 30 and Order Qty = 100

Multi-way match (MWM) for a quantity greater than the DisplayQty, which in turn results all of the More Qty being consumed across multiple counterparties.

Msg Received	Msg Sent	Exec Type	OrdStatus	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comments
NewOrderSingle				100				
	ExecutionReport	New (0)	New (0)	100	0	100	0	
	ExecutionReport	Trade (F)	Partially Filled (1)	100	20	80	20	Execution of 20
	ExecutionReport	Trade (F)	Partially Filled (1)	100	40	60	20	Execution of 20 (MWM)
	ExecutionReport	Trade (F)	Partially Filled (1)	100	60	40	20	Execution of 20 (MWM)
	ExecutionReport	Trade (F)	Filled (2) *	100	100	0	40	Execution of 40 (MWM)

* Multi-way matches consume the full value of the original FIX OrderQty (MFG Primary + More Qty). Order is Filled. See logic in example #1.

6.1.1.2Mine/Yours Spot Fill or Kill (FIX Immediate or Cancel)

Aggressing order filled completely upon order entry with one standing order:

Msg Received	Msg Sent	Exec Type	OrdStatus	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comments
NewOrderSingle				1000				
Match 1 for 1000								New not Message Required
	ExecutionReport	(F) Trade	Filled (2)	1000	1000	0	1000	Order Completely Fills

Aggressing order immediately filled with two standing orders upon order entry:

Msg Received	Msg Sent	Exec Type	OrdStatus	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comments
NewOrderSingle				1000				
Match 1 400								
	ExecutionReport	(F) Trade	Part Filled (1)	1000	400	600	400	
Match 2 600								
	ExecutionReport	Trade (F)	Filled (2)	1000	1000	0	600	Order Completely Fills

Notice that, in the first FIX Execution Report the order status is Part Filled and in the final one this has changed to Filled. If there are several matches each will have a status of Part Filled except the final Execution Report which will have a status of Filled.

Aggressing order is partially filled on order entry with one standing order:

Msg Received	Msg Sent	Exec Type	OrdStatus	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comments
NewOrderSingle				1000				
Match 1 600								
	ExecutionReport	Trade (F)	Cancelled (4)	1000	600	0	600	Residual Qty is cancelled

Notice that for IOC orders the ‘New’ ExecType and OrdStatus is optimized away.

Aggressing order is partially filled on order entry with two standing orders:

Msg Received	Msg Sent	Exec Type	OrdStatus	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comments
NewOrderSingle				1000				
Match 1 400								
	ExecutionReport	(F) Trade	Part Filled (1)	1000	400	600	400	
Match 2 250								
	ExecutionReport	Trade (F)	Cancelled (4)	1000	650	0	250	Order Partially Fills

Notice that, in the first FIX Execution Report the order status is Part Filled and in the final one this has changed to Cancelled. If there are several matches each will have a status of Part Filled and the final Execution Report will have a status of Cancelled.

Aggressing order does not match on order entry:

Msg Received	Msg Sent	Exec Type	OrdStatus	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comments
NewOrderSingle				1000				
Match 1 600								
	ExecutionReport	Cancelled (4)	Cancelled (4)	1000	0	0	0	

Notice that there is no ‘New’ state for IOC orders.

6.1.1.3Standard FX Swaps Limit Order Entry

Notice that for all FX Swaps orders an ExecType of (I = Status) is used for all trade execution reports that have an OrderStatus of Part Filled or Filled.

Order Soft Matched for full quantity immediately upon entering the book

Msg Received	Msg Sent	Exec Type	OrdStatus	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comments
NewOrderSingle				1000				
	ExecutionReport	New (0)	New (0)	1000	0	1000	0	
	ExecutionReport	(I) Status	Filled (2)	1000	1000	0	1000	Execution of 1000

Order Soft Matched for full quantity after resting in the book:

Msg Received	Msg Sent	Exec Type	OrdStatus	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comments
NewOrderSingle				10000				
	ExecutionReport	New (0)	New (0)	10000	0	10000	0	
	ExecutionReport	(I) Status	Partially Filled (1)	10000	2000	8000	2000	Execution of 2000
	ExecutionReport	(I) Status	Partially Filled (1)	10000	3000	7000	1000	Execution of 1000
	ExecutionReport	(I) Status	Filled (2)	10000	10000	0	7000	Execution of 7000

Order Soft Matched (partially filled) upon hitting the book

Msg Received	Msg Sent	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comments
NewOrderSingle				10000				
	ExecutionReport	New (0)	New (0)	10000	0	10000	0	
	ExecutionReport	(I) Status	Partially Filled (1)	10000	7000	3000	7000	Execution of 7000

Order Soft Matched for partial quantity after resting in the book and results in an unsolicited cancel.

Cancel can be a result of a user disconnect, transfer user disconnected, admin disconnect or from an end of trading session close.

Msg Received	Msg Sent	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comments
NewOrderSingle				10000				
	ExecutionReport	New (0)	New (0)	10000	0	10000	0	
	ExecutionReport	Status (I)	Partially Filled (1)	10000	2000	8000	2000	Execution of 2000
	ExecutionReport	Status (I)	Partially Filled (1)	10000	3000	7000	1000	Execution of 1000
	ExecutionReport	Cancelled (4)**	Cancelled (4) **	10000	3000	0	0	0

** ** End of Trading session or expired limit orders entered with TimeInForce='A' or '6' will result in an ExecType and OrdStatus of Expired (C).

Soft Matching Order Rejected Upon Order Entry

Msg Received	Msg Sent	ExecType	Ord Status	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comments
NewOrderSingle				10000				
	ExecutionReport	Rejected (8)	Rejected (8)	10000	0	0	0	Example Reason: Transfer User not logged in

6.1.1.4 Mine/Yours Swaps Fill or Kill (FIX Immediate or Cancel)

Aggressing order filled completely upon order entry with one standing order:

Msg Received	Msg Sent	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comments
NewOrderSingle				1000				
Match 1 for 1000								New not Message Required
	ExecutionReport	(I) Order Status	Filled (2)	1000	1000	0	1000	Order Completely Fills

Aggressing order filled completely upon order entry with two standing orders:

Msg Received	Msg Sent	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comments
NewOrderSingle				1000				
Match 1 400								

Msg Received	Msg Sent	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comments
	ExecutionReport	OrderStatus (I)	Part Filled (1)	1000	400	600	400	
Match 2 600								
	ExecutionReport	OrderStatus (I)	Filled (2)	1000	1000	0	600	Order Completely Fills

Notice that, in the first FIX Execution Report the order status is Part Filled and in the final one this has changed to Filled. If there were several matches each would have a status of Part Filled and the final FIX Execution Report would have a status of Filled.

Aggressing order is partially filled on order entry with one standing order:

Msg Received	Msg Sent	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comments
NewOrderSingle				1000				
Match 1 600								
	ExecutionReport	Cancelled (4)	Cancelled (4)	1000	600	0	600	Residual Qty is cancelled

Notice that the “New” and “Part Filled” ExecType and OrdStatus are optimized away.

This should be updated to (I) status --

Aggressing order is partially filled on order entry with two or more standing orders:

Msg Received	Msg Sent	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comments
NewOrderSingle				1000				
Match 1 400								
	ExecutionReport	(I) OrderStatus	Part Filled (1)	1000	400	600	400	
Match 2 250								
	ExecutionReport	Cancelled (4)	Cancelled (4)	1000	650	0	250	Order Partially Fills

Notice that, in the first FIX Execution Report the order status is Part Filled and in the final one this has changed to Cancelled. If there were several matches each would have a status of Part Filled and the final Execution Report would have a status of Cancelled.

This should be updated to (I) status --

Aggressing order does not match on order entry:

Msg Received	Msg Sent	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comments
NewOrderSingle				1000				
	ExecutionReport	Cancelled (4)	Cancelled (4)	1000	0	0	0	

6.1.1.5User Initiated Order Cancellation

Order Entry and Cancellation (FX Spot or FX Swaps) (No Filled Quantity)

Msg Received	Msg Sent	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comments
NewOrderSingle or NewOrderMultiLeg				1000				
	ExecutionReport	New (0)	New (0)	1000	0	0	0	
	ExecutionReport	Cancelled (4)	Cancelled (4)	1000	0	0	0	

Order Entry and Cancellation FX Spot (Part Filled Quantity)

Msg Received	Msg Sent	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comments
NewOrderSingle				1000				
	ExecutionReport	New (0)	New (0)	1000	0	0	0	
Match (600								
	ExecutionReport	Trade (F)	Part Filled (1)	1000	600	400	600	Match 600
OrderCancelRequest								
	ExecutionReport	(6) Pending Cancel	(6) Pending Cancel	1000	600	400	600	* See Notes
	ExecutionReport	Cancelled (4)	Cancelled (4)	1000	600	0	600	

* Notes: The quantities reported in the pending cancel are based on the order state within the MFG cache is at the time of receipt of the cancel receipt. It is possible (due to race conditions) that the reported quantities in the final cancel message are different than that which was sent in the pending cancel.

Order Entry and Cancellation FX Swaps (Part Filled Quantity)

Msg Received	Msg Sent	Exec Type	OrdStatus	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comments
NewOrderSingle				1000				
	ExecutionReport	New (0)	New (0)	1000	0	0	0	
Match (600								
	ExecutionReport	Status (I)	Part Filled (1)	1000	600	400	600	Match 600
OrderCancelRequest								
	ExecutionReport	Pending Cancel (6)	Pending Cancel (6)	1000	600	400	600	* See Notes
	ExecutionReport	Cancelled (4)	Cancelled (4)	1000	700	0	100	* See Notes

*Notes: The quantities reported in the pending cancel are based on the order state within the MFG cache is at the time of receipt of the cancel receipt. It is possible (due to race conditions) that the reported quantities in the final cancel message are different than that which was sent in the pending cancel. See the next scenario.

Order Entry and Cancellation (FX Spot or FX Swaps) (Too Late to cancel)

Msg Received	Msg Sent	Exec Type	OrdStatus	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comments
NewOrderSingle or NewOrderMiltiLeg				1000				
	ExecutionReport	New (0)	New (0)	1000	0	0	0	
Match 600								
	ExecutionReport	Spot =(F) Trade Swaps=(I) Status	PartialFill (1)	1000	600	400	600	
OrderCancelRequest								
	ExecutionReport	(6) Pending Cancel	(6) PendingCancel	1000	600	400	600	Async Match in progress
Match (400) ** or see Match 200 below**								
	ExecutionReport	Spot =(F) Trade Swaps=(I) Status	Filled (2)	1000	1000	0	400	
	Cancel Reject (35=9)	n/a	n/a	n/a	n/a	n/a	n/a	*See Notes Too Late to Cancel

* -- Cancel can be rejected after reporting pending cancel for a part filled or unfilled order. This hold true for both FX Spot and FX Swaps.

OR

Match (200)	This Match replaces the match for 400 above							
	ExecutionReport	Spot =(F) Trade Swaps=(I) Status	(1) PartFilled	1000	800	200	200	
	ExecutionReport	Cancelled (4)	Cancelled (4)	1000	800	0	200	* See Notes above

6.1.2 FX Swap Negotiation, Reposition, Hard Match and Trade ticket Workflows

6.1.2.1 The MFG FIX swap rules

FIX Field	Description
Order Qty (38) and Cum Qty (14) and Leaves Qty (151)	Always reflect the quantity that is “currently” in the Parent order – This in turn will indirectly drive the Market data Updates at any given point in time when the Original Parent Order Qty is updated i.e. At Soft Match or at time of reposition(if applicable)
Last Qty (32)	It is last quantity that you have soft matched against – used in the Execution Reports as we do today for the Execution Status type (I). The last qty will remain the same when reported in the hard matched Execution and will not reflect the results of the negotiation, or reposition. The Last quantity shall be reported only when ExecutionType (8) tag 150=F (Trade) or I(orderstatus) **and** Trade Ticket(AE) in the MFG negotiation flow.
LegOrderQty(685)	Always report the negotiated Quantity in all executions and quote messages.

The following workflow are supported in MFG and applied for both Initial and Half Duplex negotiation.

Usecase	Workflow Link
Negotiate with full quantity, single match	FIX A and FIX B negotiate with FULL quantity, single matched
Negotiate quantity up, single match	FIX A and FIX B negotiate the quantity UP, single matched
Negotiate quantity down, single match	FIX A and FIX B negotiate the quantity DOWN, single matched
Negotiate the far quantity up, single match	FIX A and FIX B negotiate the Far Quantity Up, single matched
Reposition after hard matched, single matched	FIX A and FIX B reposition after hard matched, single matched
Do not reposition after hard matched, single matched	FIX A and FIX B do not reposition after hard matched, single matched
Reposition after hard matched, multi-way matched	FIX A and FIX B reposition after FIX B and FIX C hard matched, multi-way match matched
Do not reposition after hard matched, multi-way matched	FIX A and FIX B do not reposition after FIX B and FIX C hard matched, multi-way match matched
No Deal, Reposition/do not reposition	No deal (initiated by user /system) FIX A reposition, FIX B do not reposition after No Deal.
Full Soft Matched, Parent Order is cancelled while negotiations are in Progress, Hard Matched	Fully soft matched, Parent order is cancelled while the Negotiations are in progress

Usecase	Workflow Link
Partially Soft Matched, Parent Order is cancelled while negotiations are in Progress, Hard Matched	Partially soft matched, Parent order is cancelled while the Negotiations are in progress
Full Soft Matched, Parent Order is held while negotiations are in Progress, Hard Matched, Reposition/do not reposition	Fully soft matched, Parent order is held while the Negotiations are in progress, Reposition
Partially Soft Matched, Parent Order is held while negotiations are in Progress, Hard Matched, Reposition/do not reposition	Partially soft matched, Parent order is held while the Negotiations are in progress, Reposition
Partially soft matched, user A log out while negotiations are in Progress.	Partially soft matched, user A logs out during negotiation
Fully soft matched, user A log out while negotiations are in Progress	Fully soft matched, user A logs out during negotiation
Partially soft matched Yours/Mine, No deal	Partially soft matched YOURS/MINE, NO Deal
Fully soft matched Yours/Mine, No Deal	Fully soft matched YOURS/MINE, NO Deal
Partially soft matched Yours/Mine, user A log out while negotiations are in Progress.	Partially soft matched YOURS/MINE, user logs out
Fully soft matched Yours/Mine, user A log out while negotiations are in Progress.	Fully soft matched YOURS/MINE, user logs out

6.1.2.2OrdStatus (tag 39) usages

According to FIX protocol, in an Execution Report (8) the OrdStatus (39) is used to convey the current state of the order. If an order simultaneously exists in more than one order state, the value with highest precedence is the value that is reported in the OrdStatus (39) field. The MFG supported order statuses are as follows (in highest to lowest precedence):

Precedence	OrdStatus (39)	Description
7	Pending Cancel	Order with an Order Cancel Request pending, used to confirm receipt of an Order Cancel Request (F) . DOES NOT INDICATE THAT THE ORDER HAS BEEN CANCELED.
6	Pending Replace	Order with an Order Cancel/Replace Request pending, used to confirm receipt of an Order Cancel/Replace Request (G) . DOES NOT INDICATE THAT THE ORDER HAS BEEN REPLACED.
5	Filled	Order completely filled, no remaining quantity
4	Suspended	Order has been placed in suspended state at the request of the client.
3	Cancelled	Cancelled order with or without executions
3	Expired	Order has been cancelled in broker's system due to time in force instructions.

Precedence	OrdStatus (39)	Description
2	Partially Filled	Outstanding order with executions and remaining quantity
1	New	Outstanding order with no executions
1	Rejected	Order has been rejected by sell-side (broker, exchange, ECN). NOTE: An order can be rejected subsequent to order acknowledgment, i.e. an order can pass from New to Rejected status.

6.1.2.3FIX A and FIX B negotiate with FULL quantity, single matched

[QTY.NEG.FULL.001] both counterparties shall be able to negotiate with full softmatch quantity at any stage in the negotiation.

Main flow - FIX A submits an original order of 120 and FIX B of 180, soft matched 120, and hard matched 120/120

1) **Softmatch notification.** Execution Report(8) soft match report for both FIX A and FIX B shall display as follows

MFG=>FC	FC=>MFG	LeavesQty	CumQty	OrderQty	LegOrderQty1	LegOrderQty2	LastQty	OrdStatus	ExecType
FIX A									
ExecutionReport(soft match report)	*Standard MFG ack*	0	120	120	n/a	n/a	120	2(filled)	I(Order Status)
FIX B									
ExecutionReport(soft match report)	*Standard MFG ack*	60	120	180	n/a	n/a	120	1(partially filled)	I(Order Status)

2) **Initial Negotiation flow**. Both counterparties negotiate with full soft match quantity of 120/120, the proposals match.

MFG=>FC	FC=>MFG	QuoteType/ QuoteRespType	LegOrderQty1	LegOrderQty2	OrderQty	QuoteStatus
FIX A						
Quote		Indicative	120	120	120	n/a
	QuoteResponse	counter	120	120	120	n/a
QuoteStatusReport		n/a	n/a	n/a	n/a	active
FIXB						
Quote		Indicative	120	120	180	n/a
	QuoteResponse	counter	120	120	180	n/a

MFG=>FC	FC=>MFG	QuoteType/ QuoteRespType	LegOrder Qty1	LegOrderQty2	OrderQty	QuoteStatus
QuoteStatusReport		n/a	n/a	n/a	n/a	active

3) **Hard Match Notification.** Both counterparties hard match of 120/120, the Execution Report (8) hard match report for both FIX A and FIX B shall display as follows.

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX A									
ExecutionReport(hard match report)	*Standard MFG ack*	0	120	120	120	120	120	2(filled)	F(trade)
FIX B									
ExecutionReport(hard match report)	*Standard MFG ack*	60	120	180	120	120	120	1(Partially filled)	F(trade)

4) As there is no remaining quantity that can be negotiated on the parent order for FIX A; the NMH will send MFG an OrderRemoveNotification (ORN) message. On receipt of ORN, the Execution report (cancelled) will be sent to FIX A.

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX A									
ExecutionReport	*Standard MFG ack*	0	120	120	n/a	n/a	n/a	2(filled)	cancelled

5) **Trade Ticket.** The Trade Capture Report (AE) for both FIX A and FIX B shall display as follows.

MFG=>FC	OrderQty	LegQty1	LegQty2	LastQty	ExecType
FIX A					
Trade Capture Report	120	120	120	120	F(trade)
FIX B					
Trade Capture Report	180	120	120	120	F(trade)

6.1.2.4FIX A and FIX B negotiate the quantity UP, single matched

[QTY.NEG.UP.001] both counterparties shall be able to negotiate quantity *up* at any stage in the negotiation provided that the newly proposed quantity is less than or equal to the original soft matched quantity.

FIX A submits an original order of 120 and FIX B of 180, soft matched 120, and hard matched 110/110

1) **Softmatch notification** – same as [QTY.NEG.FULL.001]
2) **Initial Negotiation flow**. FIX A negotiates quantity to 100/100, FIX B negotiates up to 110/110(but less than original Soft Match of 120), the proposals do not match.

MFG=>FC	FC=>MFG	QuoteType/ QuoteRespType	LegOrder Qty1	LegOrderQty2	OrderQty	QuoteStatus
FIX A						
Quote		Indicative	120	120	120	n/a
	QuoteResponse	counter	100	100	120	n/a
QuoteStatusReport		n/a	n/a	n/a	n/a	active
FIXB						
Quote		Indicative	120	120	180	n/a
	QuoteResponse	counter	110	110	180	n/a
QuoteStatusReport		n/a	n/a	n/a	n/a	active

3) **Half-duplex Negotiation flow.** The proposal of 100/100 is sent to FIX B while FIX A is in the waiting state. FIX B still wants to match at 110/110, therefore doesn't agree with FIX A' s proposal and send proposal of 110/110 to FIX A again. FIX A finally agrees with the proposal of 110/110, the proposals match.

MFG=>FC	**FC=>MFG**	**QuoteType/ QuoteRespType**	**LegOrderQty1**	**LegOrderQty2**	**OrderQty**	**QuoteStatus**
FIXB						
Quote		n/a	100	100	180	n/a
	QuoteResponse	counter	110	110	180	n/a
QuoteStatusReport		n/a	n/a	n/a	n/a	Active
FIXA						
Quote		n/a	110	110	120	n/a
	QuoteResponse	Hit/lift	110	110	120	n/a

MFG=>FC	FC=>MFG	QuoteType/ QuoteRespType	LegOrderQty1	LegOrderQty2	OrderQty	QuoteStatus
QuoteStatusReport		n/a	n/a	n/a	n/a	Active

4) **Hard Match Notification.** Both counterparties hard match of 110/110, the Execution Report (8) hard match report for both FIX A and FIX B shall display as follows.

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	Last Qty	OrdStatus	ExecType
FIX A									
ExecutionReport(hard match report)	*Standard MFG ack*	0	120	120	110	110	120	2(filled)	F(trade)
FIX B									
ExecutionReport(hard match report)	*Standard MFG ack*	60	120	180	110	110	120	1(Partially filled)	F(trade)

5) **Trade Ticket.** The Trade Capture Report (AE) for both FIX A and FIX B shall display as follows.

MFG=>FC	OrderQty	LegOrderQty1	LegOrderQty2	LastQty	ExecType
FIX A					
Trade Capture Report	120	110	110	120	F(trade)
FIX B					
Trade Capture Report	180	110	110	120	F(trade)

[QTY.NEG.UP.002] The value of the maximum order quantity which is populated in StipulationValue(234) shall be sent to FIX client only when SituplationType(233)=MAXORDQTY (when the client cannot negotiate quantity *up* above the original soft match amount)

Note: SituplationType(233)=MAXORDQTY is not present if and only if the aggressor order consumes all of the provider's remaining primary quantity.

For multiple orders with the same price, see sample below:

FIX A places 2 BUY orders with quantity of 100 each.

FIX B places 1 SELL order and fully matched with FIX A 's quantity on one order.

Even though FIXB has consumed all of FIXA's quantity on one order, FIXA still has quantity at the same price (in the 2nd order) on the book so it fails the condition that the aggressor must consume all of the provider's quantity.

[QTY.NEG.UP.003] Both counterparties shall be able to negotiate quantity *up* above the original soft matched amount if there is no SituplationType(233)=MAXORDQTY on the initial Proposal Notification messages to the counterparties.

FIX A submits an original order of 120 and FIX B of 180, soft matched 120, and hard matched 130/130

1) **Softmatch notification** – same as **[QTY.NEG.FULL.001]**
2) **Initial Negotiation flow**. Both counterparties negotiate the quantity up to 130/130, the proposals match.

MFG=>FC	**FC=>MFG**	**QuoteType/ QuoteRespType**	**LegOrderQty1**	**LegOrderQty2**	**OrderQty**	**QuoteStatus**
FIX A						
Quote		Indicative	120	120	120	n/a
	QuoteResponse	counter	130	130	120	n/a
QuoteStatusReport		n/a	n/a	n/a	n/a	active
FIXB						
Quote		Indicative	120	120	180	n/a
	QuoteResponse	counter	130	130	180	n/a
QuoteStatusReport		n/a	n/a	n/a	n/a	active

3) **Hard Match Notification.** Both counterparties hard match of 130/130, the Execution Report (8) hard match report for both FIX A and FIX B shall display as follows.

MFG=>FC	**FC=>MFG**	**Leaves Qty**	**Cum Qty**	**Order Qty**	**LegOrderQty1**	**LegOrderQty2**	**LastQty**	**OrdStatus**	**ExecType**
FIX A									
ExecutionReport(hard match report)	*Standard MFG ack*	0	120	120	130	130	120	2(filled)	F(trade)
FIX B									
ExecutionReport(hard match report)	*Standard MFG ack*	60	120	180	130	130	120	1(Partially filled)	F(trade)

4) As there is no remaining quantity that can be negotiated on the parent order for FIX A; the NMH will send MFG an OrderRemoveNotification (ORN) message. On receipt of ORN, the Execution report (cancelled) will be sent to FIX A.

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX A									
ExecutionReport	*Standard MFG ack*	0	120	120	n/a	n/a	n/a	2(filled)	cancelled

5) **Trade Ticket.** The Trade Capture Report (AE) for both FIX A and FIX B shall display as follows.

MFG=>FC	OrderQty	LegOrderQty1	LegOrderQty2	LastQty	ExecType
FIX A					
Trade Capture Report	120	130	130	120	F(trade)
FIX B					
Trade Capture Report	180	130	130	120	F(trade)

6.1.2.5FIX A and FIX B negotiate the quantity DOWN, single matched

[QTY.NEG.DOWN.001] both counterparties shall be able to negotiate quantity *down* at any stage in the negotiation provided that the newly proposed quantity is greater than or equal to the minimum lot size.

FIX A submits an original order of 120 and FIX B of 180, soft matched 120, and hard matched 100/100.

1) **Softmatch notification-** same as **[QTY.NEG.FULL.001]**
2) **Initial Negotiation flow**. Both counterparties negotiate the quantity down to 100/100, the proposals match.

MFG=>FC	**FC=>MFG**	**QuoteType/ QuoteRespType**	**LegOrderQty1**	**LegOrderQty2**	**OrderQty**	**QuoteStatus**
FIX A						
Quote		Indicative	120	120	120	n/a
	QuoteResponse	counter	100	100	120	n/a
QuoteStatusReport		n/a	n/a	n/a	n/a	active
FIXB						

MFG=>FC	FC=>MFG	QuoteType/ QuoteRespType	LegOrderQty1	LegOrderQty2	OrderQty	QuoteStatus
Quote		Indicative	120	120	180	n/a
	QuoteResponse	counter	100	100	180	n/a
QuoteStatusReport		n/a	n/a	n/a	n/a	active

3) **Hard Match Notification.** Both counterparties hard match of 100/100, the Execution Report (8) hard match report for both FIX A and FIX B shall display as follows.

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX A									
ExecutionReport(hard match report)	*Standard MFG ack*	0	120	120	100	100	120	2(filled)	F(trade)
FIX B									
ExecutionReport(hard match report)	*Standard MFG ack*	60	120	180	100	100	120	1(Partially filled)	F(trade)

4) **Trade Ticket.** The Trade Capture Report (AE) for both FIX A and FIX B shall display as follows.

MFG=>FC	OrderQty	LegOrderQty1	LegOrderQty2	LastQty	ExecType
FIX A					
Trade Capture Report	120	100	100	120	F(trade)
FIX B					
Trade Capture Report	180	100	100	120	F(trade)

6.1.2.6FIX A and FIX B negotiate the Far Quantity Up, single matched

[NEG.UNEVEN.001] both counterparties shall be able to negotiate the Far Quantity Up or down provided that it is greater than or equal to the negotiated (near leg) quantity.

FIX A submits an original order of 120 and FIX B of 180, soft matched 120, and hard matched 120/130.

1) **Softmatch notification. -** same as [QTY.NEG.FULL.001]
2) **Initial Negotiation flow**. Both counterparties negotiate the quantity down to 120/130, the proposals match.

MFG=>FC	FC=>MFG	QuoteType/ QuoteRespType	LegOrder Qty1	LegOrderQty 2	OrderQty	QuoteStatus
FIX A						
Quote		Indicative	120	120	120	n/a
	QuoteResponse	counter	120	130	120	n/a
QuoteStatusReport		n/a	n/a	n/a	n/a	active
FIXB						
Quote		Indicative	120	120	180	n/a
	QuoteResponse	counter	120	130	180	n/a
QuoteStatusReport		n/a	n/a	n/a	n/a	active

3) **Hard Match Notification.** Both counterparties hard match of 120/130, the Execution Report (8) hard match report for both FIX A and FIX B shall display as follows.

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX A									
ExecutionReport(hard match report)	*Standard MFG ack*	0	120	120	120	130	120	2(filled)	F(trade)
FIX B									
ExecutionReport(hard match report)	*Standard MFG ack*	60	120	180	120	130	120	1(Partially filled)	F(trade)

4) As there is no remaining quantity that can be negotiated on the parent order for FIX A; the NMH will send MFG an OrderRemoveNotification (ORN) message. On receipt of ORN, the Execution report (cancelled) will be sent to FIX A.

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX A									
ExecutionReport	*Standard MFG ack*	0	120	120	n/a	n/a	n/a	2(filled)	cancelled

5) **Trade Ticket.** The Trade Capture Report (AE) for both FIX A and FIX B shall display as follows.

MFG=>FC	OrderQty	LegOrderQty1	LegOrderQty2	LastQty	ExecType
FIX A					
Trade Capture Report	120	120	130	120	F(trade)
FIX B					
Trade Capture Report	180	120	130	120	F(trade)

6.1.2.7FIX A and FIX B reposition after hard matched, single matched

[REPO.NEG.QTY.001] Both Parties Reposition after a hard march. FIX A submits an original order of 120 and FIX B of 180, soft matched 120, and hard matched 80/80, then FIX A and B reposition after hard matched.

1) **Softmatch notification.-** same as [QTY.NEG.FULL.001]
2) **Half-duplex Negotiation flow.** FIX B agreed the quantity of 80/80, the proposals match.

MFG=>FC	**FC=>MFG**	**QuoteType/ QuoteRespType**	**LegOrderQty 1**	**LegOrderQty 2**	**OrderQty**	**QuoteStatus**
FIX A						
Quote		Indicative	120	120	120	n/a
	QuoteResponse	counter	80	80	120	n/a
QuoteStatusReport		n/a	n/a	n/a	n/a	active
FIXB						
Quote		Indicative	120	120	180	n/a
	QuoteResponse	counter	90	90	180	n/a
QuoteStatusReport		n/a	n/a	n/a	n/a	active
FIXB						
Quote		n/a	80	80	180	n/a
	QuoteResponse	Hit/lift	80	80	180	n/a
QuoteStatusReport		n/a	n/a	n/a	n/a	Active

3) **Hard Match Notification.** Both counterparties hard match of 80/80, the Execution Report (8) hard match report for both FIX A and FIX B shall display as follows.

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX A									
ExecutionReport(hard match report)	*Standard MFG ack*	0	120	120	80	80	120	2(filled)	F(trade)
FIX B									
ExecutionReport(hard match report)	*Standard MFG ack*	60	120	180	80	80	120	1(partially filled)	F(trade)

4) **Trade Ticket.** The Trade Capture Report (AE) for both FIX A and FIX B shall display as follows.

MFG=>FC	OrderQty	LegOrderQty1	LegOrderQty2	LastQty	ExecType
FIX A					
Trade Capture Report	120	80	80	120	F(trade)
FIX B					
Trade Capture Report	180	80	80	120	F(trade)

5) **Reposition Quantity Request Flow**

a) Both counterparties reposition the remaining quantity to the market, the *Order Cancel/Replace Request (G)* shall display as follows.

MFG=>FC	FC=>MFG	LegOrderQty1	LegOrderQty2	OrderQty
FIX A				
	Order Cancel/Replace Request	n/a	n/a	120
FIX B				
	Order Cancel/Replace Request	n/a	n/a	180

b) **Execution Reports as a response to reposition quantity request.** The Execution Reports (**pending replace** and **replaced**) as a response to the *Order Cancel/Replace Request (G)* shall display as follows.

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX A									

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
ExecutionReport	*Standard MFG ack*	0	120	120	n/a	n/a	n/a	E(Pending replace)	E(Pending replace)
FIX B									
ExecutionReport	*Standard MFG ack*	60	120	180	n/a	n/a	n/a	E(Pending replace)	E(Pending replace)

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX A									
ExecutionReport	*Standard MFG ack*	40	80	120	n/a	n/a	n/a	1(Partially filled)	5(Replaced)
FIX B									
ExecutionReport	*Standard MFG ack*	100	80	180	n/a	n/a	n/a	1(Partially filled)	5(Replaced)

6.1.2.8FIX A and FIX B do not reposition after hard matched, single matched

[REPO.NEG.QTY.002] FIX A submits an original order of 120 and FIX B of 180, soft matched 120, and hard matched 80/80, FIX A and B do not reposition after hard matched.

1) **Softmatch notification.-** same as [REPO.NEG.QTY.001]
2) **Half-duplex Negotiation flow. -**same as [REPO.NEG.QTY.001]
3) **Hard Match Notification. -**same as [REPO.NEG.QTY.001]
4) **Trade Ticket. -**same as [REPO.NEG.QTY.001]
5) **No Reposition Quantity Request Flow**

a) Both counterparties do not reposition the remaining quantity to the market, the *Order Cancel Request (F)* shall display as follows.

MFG=>FC	FC=>MFG	LegOrderQty1	LegOrderQty2	OrderQty
FIX A				
	Order Cancel Request	n/a	n/a	120
FIX B				
	Order Cancel Request	n/a	n/a	180

b) **Execution Reports as a response to <u>No</u> reposition quantity request.** The Execution Reports (**pending cancel** and **cancelled**) as a response to the *Order Cancel Request (F)* shall display as follows.

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX A									
ExecutionReport	*Standard MFG ack*	0	120	120	n/a	n/a	n/a	6(Pending cancel)	6(Pending cancel)
FIX B									
ExecutionReport	*Standard MFG ack*	60	120	180	n/a	n/a	n/a	6(Pending cancel)	6(Pending cancel)

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX A									
ExecutionReport	*Standard MFG ack*	0	120	120	n/a	n/a	n/a	2(filled)	4(Canceled)
FIX B									
ExecutionReport	*Standard MFG ack*	60	120	180	n/a	n/a	n/a	1(partially filled)	4(Canceled)

6) As there is no remaining quantity that can be negotiated on the parent order for FIX A; the NMH will send MFG an OrderRemoveNotification (ORN) message. On receipt of ORN, the Execution report (cancelled) will be sent to FIX A.

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX A									
ExecutionReport	*Standard MFG ack*	0	120	120	n/a	n/a	n/a	2(filled)	cancelled

6.1.2.9 <u>FIX A and FIX B reposition after FIX B and FIX C hard matched, multi-way matched</u>

[MULTI.WAY.MATCH.001] FIX A submits an original order of 120, FIX B of 180 and FIX C of 150. FIX A and B soft matched 120 and FIX B and C soft matched 60. FIX A and B reposition after FIX B and FIX C hard match.

1) **Softmatch notification A-B-C.** Execution Report(8) soft match report for both FIX A, FIX B and FIX C shall display as follows

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX A									
ExecutionReport(soft match report)	*Standard MFG ack*	0	120	120	n/a	n/a	120	2(filled)	I(Order Status)
FIX B(1)									
ExecutionReport(soft match report)	*Standard MFG ack*	60	120	180	n/a	n/a	120	1(partially filled)	I(Order Status)
FIX B(2)									
ExecutionReport(soft match report)	*Standard MFG ack*	0	180	180	n/a	n/a	60	2(filled)	I(Order Status)
FIX C									
ExecutionReport(soft match report)	*Standard MFG ack*	90	60	150	n/a	n/a	60	1(partially filled)	I(Order Status)

2) **Negotiation flow A-B**

a) **Initial Negotiation flow A-B**. FIX A negotiates quantity to 100/100 and FIX B proposes the same quantity, the proposals match.

MFG=>FC	**FC=>MFG**	**QuoteType/ QuoteRespType**	**LegOrderQty 1**	**LegOrderQty 2**	**OrderQty**	**QuoteStatus**
FIX A						
Quote		Indicative	120	120	120	n/a
	QuoteResponse	counter	100	100	120	n/a
QuoteStatusReport		n/a	n/a	n/a	n/a	active
FIXB						
Quote		Indicative	120	120	180	n/a
	QuoteResponse	counter	100	100	180	n/a
QuoteStatusReport		n/a	n/a	n/a	n/a	active

b) **Hard Match Notification A-B.** Both counterparties hard match of 100/100, the Execution Report (8) hard match report for both FIX A and FIX B shall display as follows.

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrderQty1	LegOrderQty2	LastQty	OrdStatus	ExecType
FIX A									
ExecutionReport(hard match report)	*Standard MFG ack*	0	120	120	100	100	120	2(filled)	F(trade)
FIX B									
ExecutionReport(hard match report)	*Standard MFG ack*	0	180	180	100	100	120	2(filled)	F(trade)

c) **Trade Ticket A-B.** The Trade Capture Report (AE) for both FIX A and FIX B shall display as follows.

MFG=>FC	OrderQty	LegOrderQty1	LegOrderQty2	LastQty	ExecType
FIX A					
Trade Capture Report	120	100	100	120	F(trade)
FIX B					
Trade Capture Report	180	100	100	120	F(trade)

3) **Negotiation Flow B-C**

a) **Initial Negotiation flow B-C**. FIX B negotiates quantity to 70/70 and FIX C agrees, the proposals match.

MFG=>FC	FC=>MFG	QuoteType/ QuoteRespType	LegOrderQty1	LegOrderQty2	OrderQty	QuoteStatus
FIX B						
Quote		Indicative	60	60	180	n/a
	QuoteResponse	counter	70	70	180	n/a
QuoteStatusReport		n/a	n/a	n/a	n/a	active
FIXC						
Quote		Indicative	60	60	150	n/a

MFG=>FC	FC=>MFG	QuoteType/ QuoteRespType	LegOrderQty1	LegOrderQty2	OrderQty	QuoteStatus
	QuoteResponse	counter	70	70	150	n/a
QuoteStatusReport		n/a	n/a	n/a	n/a	active

b) **Hard Match Notification.** Both counterparties hard match of 70/70, the Execution Report (8) hard match report for both FIX B and C shall display as follows.

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX B									
ExecutionReport(hard match report)	*Standard MFG ack*	0	180	180	70	70	60	2(filled)	F(trade)
FIX C									
ExecutionReport(hard match report)	*Standard MFG ack*	90	60	150	70	70	60	1(Partially filled)	F(trade)

c) **Trade Ticket.** The Trade Capture Report (AE) for both FIX A and FIX B shall display as follows.

MFG=>FC	OrderQty	LegOrderQty1	LegOrderQty2	LastQty	ExecType
FIX B					
Trade Capture Report	180	70	70	60	F(trade)
FIX C					
Trade Capture Report	150	70	70	60	F(trade)

4) **A-B Reposition AFTER B-C hard match**

a) **Reposition Quantity Request A-B.** Both counterparties reposition the remaining quantity to the market after FIX B and C hard matched, the *Order Cancel/Replace Request (G)* shall display as follows.

MFG=>FC	FC=>MFG	LegOrderQty1	LegOrderQty2	OrderQty
FIX A				
	Order Cancel/Replace Request	n/a	n/a	120

MFG=>FC	FC=>MFG	LegOrderQty1	LegOrderQty2	OrderQty
FIX B				
	Order Cancel/Replace Request	n/a	n/a	180

b) **Execution Reports as a response to reposition quantity request.** The Execution Reports (**pending replace** and **replaced**) as a response to the *Order Cancel/Replace Request (G)* shall display as follows.

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX A									
ExecutionReport	*Standard MFG ack*	0	120	120	n/a	n/a	n/a	E(Pending replace)	E(Pending replace)
FIX B									
ExecutionReport	*Standard MFG ack*	0	180	180	n/a	n/a	n/a	E(Pending replace)	E(Pending replace)

MFG=>FC	FC=>MFG	LeavesQty	CumQty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX A									
ExecutionReport	*Standard MFG ack*	20	100	120	n/a	n/a	n/a	1(Partially filled)	5(Replaced)
FIX B									
ExecutionReport	*Standard MFG ack*	20	160	180	n/a	n/a	n/a	1(Partially filled)	5(Replaced)

6.1.2.10 FIX A and FIX B do not reposition after FIX B and FIX C hard matched, multi-way matched

[MULTI.WAY.MATCH.002] FIX A submits an original order of 120, FIX B of 180 and FIX C of 150. FIX A and B soft matched 120 and FIX B and C soft matched 60. FIX A and B do not reposition after FIX B and FIX C hard match.

1) **Softmatch notification A-B-C.-** same as [MULTI.WAY.MATCH.001]
2) **Negotiation flow A-B** - same as [MULTI.WAY.MATCH.001]
3) **Negotiation Flow B-C -**- same as [MULTI.WAY.MATCH.001]
4) **A-B do not** reposition after B-C hard match

    a) **No Reposition Quantity Request.** Both counterparties do not reposition the remaining quantity to the market, the *Order Cancel Request (F)* shall display as follows.

MFG=>FC	FC=>MFG	LegOrderQty1	LegOrderQty2	OrderQty
FIX A				
	Order Cancel Request	n/a	n/a	120
FIX B				
	Order Cancel Request	n/a	n/a	180

b) **Execution Reports as a response to <u>No</u> reposition quantity request.** The Execution Reports (**pending cancel** and **cancelled**) as a response to the *Order Cancel Request (F)* shall display as follows

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX A									
ExecutionReport		0	120	120	n/a	n/a	n/a	6(pending Cancel)	6(pending Cancel)
FIX B									
ExecutionReport		0	180	180	n/a	n/a	n/a	6(pending Cancel)	6(pending Cancel)

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX A									
ExecutionReport	*Standard MFG ack*	0	120	120	n/a	n/a	n/a	2(filled)	4(Cancelled)
FIX B									
ExecutionReport	*Standard MFG ack*	0	180	180	n/a	n/a	n/a	2(filled)	4(Cancelled)

c) As there is no remaining quantity that can be negotiated on the parent order for both FIX A and B, the NMH will send MFG an OrderRemoveNotification (ORN) message. On receipt of ORN, the Execution report (cancelled) will be sent to both parties.

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX A									
ExecutionReport	*Standard MFG ack*	0	120	120	n/a	n/a	n/a	2(filled)	4(Cancelled)

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX B									
ExecutionReport	*Standard MFG ack*	0	180	180	n/a	n/a	n/a	2(filled)	4(Cancelled)

6.1.2.11 No deal (initiated by user /system) FIX A reposition, FIX B do not reposition after No Deal.

[NO.DEAL.REPO.001] FIX A submits an original order of 120 and FIX B of 180, soft matched 120, FIX A no deal before hard match.

1) **Softmatch notification-** same as [QTY.NEG.FULL.001]
2) No Deal occurs (initiated by user or system), then user A reposition the remaining qty but user B do not reposition.

   a. FIX A reposition the quantity.

MFG=>FC	FC=>MFG	LegOrderQty1	LegOrderQty2	OrderQty
FIX A				
	Order Cancel/Replace Request	n/a	n/a	120

   b. FIX B do not reposition the quantity

MFG=>FC	FC=>MFG	LegOrderQty1	LegOrderQty2	OrderQty
FIX B				
	Order Cancel Request	n/a	n/a	180

3) Execution report as response to RepositionQtyRequest (**reposition**)to FIX A

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX A									
ExecutionReport	*Standard MFG ack*	0	120	120	n/a	n/a	n/a	Pending replace	Pending replace

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX A									

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
ExecutionReport	*Standard MFG ack*	120	0	120	n/a	n/a	n/a	0(new)	replaced

4) Execution report as response to RepositionQtyRequest (**no reposition**) to FIX B

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX B									
ExecutionReport	*Standard MFG ack*	60	120	180	n/a	n/a	n/a	Pending cancel	Pending cancel

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX B									
ExecutionReport	*Standard MFG ack*	60	120	180	n/a	n/a	n/a	1(Partially filled)	Cancelled

6.1.2.12 Fully soft matched, Parent order is cancelled while the Negotiations are in progress

[PARENT.CANCEL.FULL.001] FIX A submits an original order of 120, soft matched 120 with FIXB.

1) Softmatch notification-

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX A									
ExecutionReport(soft match report)	*Standard MFG ack*	0	120	120	n/a	n/a	120	2(filled)	I(Order Status)

2) FIX A cancels the parent order (by sending the ***Order Cancel Request*** message) while the negotiations with FIX B are in progress.
3) **Execution Report** (8) as a successful response to Order cancel request (F) is sent to FIX A.

MFG=>FC	FC=>MFG	LeavesQty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX A									
ExecutionReport	*Standard MFG ack*	0	120	120	n/a	n/a	n/a	Pending cancel	Pending cancel

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX A									
ExecutionReport	*Standard MFG ack*	0	120	120	n/a	n/a	n/a	filled	Cancelled

4) FIX A hard matched with FIX B 80/80, the Execution Report(8) – hard match report of FIX A shall be as follows:

MFG=>FC	FC=>MFG	Leaves Qty	CumQty	Order Qty	LegOrderQty1	LegOrderQty2	Last Qty	OrdStatus	ExecType
FIX A									
ExecutionReport(hard match report)	*Standard MFG ack*	0	120 (because it is no reposition)	120	80	80	120	filled	F(trade)

5) **Trade Ticket.** The Trade Capture Report (AE) for FIX A shall display as follows.

MFG=>FC	OrderQty	LegOrderQty1	LegOrderQty2	LastQty	ExecType
FIX A					
Trade Capture Report	120	80	80	120	F(trade)

6) FIX A cannot reposition the remaining quantity due to the parent order is cancelled.

6.1.2.13 Partially soft matched, Parent order is cancelled while the Negotiations are in progress

[PARENT.CANCEL.PARTIAL.001] FIX A submits an original order of 120, soft matched 90.

1) **Softmatch notification**

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrderQty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX A									
ExecutionReport(soft match report)	*Standard MFG ack*	30	90	120	n/a	n/a	90	1(Partially filled)	I(Order Status)

2) FIX A cancels the parent order (by sending the ***Order Cancel Request*** message) while the negotiations with FIX B are in progress.
3) **Execution Report** (8) as a successful response to Order cancel request (F) is sent to FIX A.

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX A									
ExecutionReport	*Standard MFG ack*	30	90	120	n/a	n/a	n/a	Pending cancel	Pending cancel

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX A									
ExecutionReport	*Standard MFG ack*	0	90	120	n/a	n/a	n/a	Cancelled	Cancelled

4) FIX A hard matched with FIX B 80/80, the Execution Report(8) – hard match report of FIX A shall be as follows:

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX A									
ExecutionReport(hard match report)	*Standard MFG ack*	0	90	120	80	80	90	cancelled	F(trade)

5) **Trade Ticket.** The Trade Capture Report (AE) for FIX A shall display as follows.

MFG=>FC	OrderQty	LegOrderQty1	LegOrderQty2	LastQty	ExecType
FIX A					
Trade Capture Report	120	80	80	90	F(trade)

6) FIX A cannot reposition the remaining quantity due to the parent order is cancelled.

6.1.2.14 Fully soft matched, Parent order is held while the Negotiations are in progress, Reposition

[PARENT.HELD.FULL.001] FIX A submits an original order of 120, soft matched 120 with FIXB.

1) **Softmatch notification-** same as [PARENT.CANCEL.FULL.001]
2) Parent order of FIX A is held (by sending the ***Mass Action Held*** message) while the negotiations with FIX B are in progress.
3) **Execution Report** (8) as a successful response to Order Mass Action Request (CA) is sent to FIX A. (not report LegOrderQty1/ LegOrderQty2 in this ER)

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	OrderQty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX A									
ExecutionReport	*Standard MFG ack*	0	120	120	n/a	n/a	n/a	Filled	9(Suspended)

4) FIX A hard matched with FIX B 80/80, the Execution Report(8) – hard match report of FIX A shall be as follows:

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX A									
ExecutionReport(hard match report)	*Standard MFG ack*	0	120	120	80	80	120	Filled	F(trade)

5) **Trade Ticket.** The Trade Capture Report (AE) for FIX A shall display as follows.

MFG=>FC	OrderQty	LegOrderQty1	LegOrderQty2	LastQty	ExecType
FIX A					
Trade Capture Report	120	80	80	120	F(trade)

6) FIX A reposition the remaining quantity

a. FIX A submits the Order Cancel/Replace Request (G) message.

MFG=>FC	FC=>MFG	LegOrderQty1	LegOrderQty2	OrderQty
FIX A				
	Order Cancel/Replace Request	n/a	n/a	120

b. Execution report as response to RepositionQtyRequest (**reposition**)to FIX A

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX A									
ExecutionReport	*Standard MFG ack*	0	120	120	n/a	n/a	n/a	Pending replace	Pending replace

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX A									
ExecutionReport	*Standard MFG ack*	40	80	120	n/a	n/a	n/a	9(Suspended)	replaced

[PARENT.RELEASE.001] if the parent order is released, the Execution Report (8) as a response to Order Mass Action Request (CA- Release All) shall operate in the same way as FX Spot.

7) FIX A do not reposition the remaining quantity
   a. FIX A submits the Order Cancel Request (F) message.

MFG=>FC	FC=>MFG	LegOrderQty1	LegOrderQty2	OrderQty
FIX A				
	Order Cancel Request	n/a	n/a	120

   b. Execution report as response to RepositionQtyRequest (**no reposition**) to FIX A

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX A									
ExecutionReport	*Standard MFG ack*	0	120	120	n/a	n/a	n/a	Pending cancel	Pending cancel

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX A									
ExecutionReport	*Standard MFG ack*	0	120	120	n/a	n/a	n/a	2(filled)	Cancelled

   c. As there is no remaining quantity that can be negotiated on the parent order for FIX A; the NMH will send MFG an OrderRemoveNotification (ORN) message. On receipt of ORN, the Execution report (cancelled) will be sent to FIX A.

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX A									
ExecutionReport	*Standard MFG ack*	0	120	120	n/a	n/a	n/a	2(filled)	Cancelled

6.1.2.15 Partially soft matched, Parent order is held while the Negotiations are in progress, Reposition

[PARENT.HELD.PARTIAL.001] FIX A submits an original order of 120, soft matched 90 with FIXB.

1) **Softmatch notification-** same as [PARENT.CANCEL.PARTIAL.001]
2) Parent order of FIX A is held (by sending the ***Mass Action Held*** message) while the negotiations with FIX B are in progress.
3) **Execution Report** (8) as a successful response to Order Mass Action Request (CA) is sent to FIX A.

MFG=>FC	**FC=>MFG**	**Leaves Qty**	**Cum Qty**	**Order Qty**	**LegOrderQty1**	**LegOrderQty2**	**LastQty**	**OrdStatus**	**ExecType**
FIX A									
ExecutionReport	*Standard MFG ack*	30	90	120	n/a	n/a	n/a	9(Suspended)	9(Suspended)

4) FIX A hard matched with FIX B 80/80, the Execution Report(8) – hard match report of FIX A shall be as follows:

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrderQty1	LegOrderQty2	LastQty	OrdStatus	ExecType
FIX A									
ExecutionReport(hard match report)	*Standard MFG ack*	30	90	120	80	80	90	9(Suspended)	F(trade)

5) **Trade Ticket.** The Trade Capture Report (AE) for FIX A shall display as follows.

MFG=>FC	OrderQty	LegOrderQty1	LegOrderQty2	LastQty	ExecType
FIX A					
Trade Capture Report	120	80	80	90	F(trade)

6) FIX A reposition the remaining quantity
    a. FIX A submits the Order Cancel/Replace Request (G) message.

MFG=>FC	FC=>MFG	LegOrderQty1	LegOrderQty2	OrderQty
FIX A				
	Order Cancel/Replace Request	n/a	n/a	120

    b. Execution report as response to RepositionQtyRequest (**reposition**)to FIX A

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX A									
ExecutionReport	*Standard MFG ack*	30	90	120	n/a	n/a	n/a	Pending replace	Pending replace

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX A									
ExecutionReport	*Standard MFG ack*	40	80	120	n/a	n/a	n/a	9(Suspended)	replaced

7) FIX A do not reposition the remaining quantity
   a. FIX A submits the Order Cancel Request (F) message.

MFG=>FC	FC=>MFG	LegOrderQty1	LegOrderQty2	OrderQty
FIX A				
	Order Cancel Request	n/a	n/a	120

   b. Execution report as response to RepositionQtyRequest (**no reposition**) to FIX A

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX A									
ExecutionReport	*Standard MFG ack*	30	90	120	n/a	n/a	n/a	Pending cancel	Pending cancel

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX A									
ExecutionReport	*Standard MFG ack*	30	90	120	n/a	n/a	n/a	9(Suspended)	Cancelled

6.1.2.16 Partially soft matched, user A logs out during negotiation (abnormal flow)

[USER.LOGOUT.PARTIAL.001] FIX A submits an original order of 120, soft matched 90 with FIXB. FIX B original qty 90

1) Softmatch notification

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX A									
ExecutionReport(soft match report)	*Standard MFG ack*	30	90	120	n/a	n/a	90	1(Partially filled)	I(Order Status)

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX B									
ExecutionReport(soft match report)	*Standard MFG ack*	0	90	90	n/a	n/a	90	filled	I(Order Status)

2) FIX A logs out while the negotiations are in progress,
    a. FIX A receives Execution report (cancelled) of the order. Note -FIX A also receives ER(cancelled) for all of his open orders(same as today)

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX A									
ExecutionReport(cancelled)	*Standard MFG ack*	0	90	120	n/a	n/a	n/a	cancelled	cancelled

    b. FIX A receives *QuoteStatusReport* (cancelled) to indicate that the negotiation has ended.

MFG=>FC	FC=>MFG	QuoteType/ QuoteRespType	LegOrderQty1	LegOrderQty2	OrderQty	QuoteStatus
FIX A						
QuoteStatusReport		n/a	n/a	n/a	n/a	cancelled

3) FIX B(counterparty) receives :
    a. As there is no remaining quantity that can be negotiated on the parent order for FIX B; the NMH will send MFG an OrderRemoveNotification (ORN) message. On receipt of ORN, the Execution report (cancelled) will be sent to FIX B.

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX B									

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
ExecutionReport	*Standard MFG ack*	0	90	90	n/a	n/a	n/a	filled	Cancelled

b. *QuoteStatusReport* (cancelled) to indicate that the negotiation has ended**.**

MFG=>FC	FC=>MFG	QuoteType/ QuoteRespType	LegOrderQty1	LegOrderQty2	OrderQty	QuoteStatus
FIX B						
QuoteStatusReport		n/a	n/a	n/a	n/a	cancelled

6.1.2.17 Fully soft matched, user A logs out during negotiation

[USER.LOGOUT.FULL.001] FIX A submits an original order of 120; FIX B submits original quantity of 180. Soft matched 120.

1) **Softmatch notification**

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX A									
ExecutionReport(soft match report)	*Standard MFG ack*	0	120	120	n/a	n/a	120	2(filled)	I(Order Status)

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX B									
ExecutionReport(soft match report)	*Standard MFG ack*	60	120	180	n/a	n/a	120	1(Partially filled)	I(Order Status)

2) FIX A logout while the negotiations are in progress,

a. FIX A receives Execution report (cancelled) of the order. Note -FIX A also receives ER(cancelled) for all of his open orders(same as today)

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX A									

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
ExecutionReport(cancelled)	*Standard MFG ack*	0	120	120	n/a	n/a	n/a	filled	cancelled

b. FIX A receives *QuoteStatusReport* (cancelled) to indicate that the negotiation has ended.

MFG=>FC	FC=>MFG	QuoteType/ QuoteRespType	LegOrderQty1	LegOrderQty2	OrderQty	QuoteStatus
FIX A						
QuoteStatusReport		n/a	n/a	n/a	n/a	cancelled

3) FIX B(counterparty) receives :

a. *QuoteStatusReport* (cancelled) to indicate that the negotiation has ended**.**

MFG=>FC	FC=>MFG	QuoteType/ QuoteRespType	LegOrderQty1	LegOrderQty2	OrderQty	QuoteStatus
FIX B						
QuoteStatusReport		n/a	n/a	n/a	n/a	cancelled

6.1.2.18 Partially soft matched YOURS/MINE, NO Deal

[PARTIAL.NODEAL.IOC.001] FIX A submits an original order of 120. FIX B submits YOURS/MINE order with original quantity of 180.Soft Matched 120.

1) Softmatch notification –FIX A

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX A									
ExecutionReport(soft match report)	*Standard MFG ack*	0	120	120	n/a	n/a	120	2(filled)	I(Order Status)

2) FIX B receives Softmatch notification.

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX B									
ExecutionReport	*Standard MFG ack*	0	120	180	n/a	n/a	120	cancelled	I(Order Status)

3) FIX A no deal, FIX A will receive ‘Quote Status Report’ (cancelled)

MFG=>FC	FC=>MFG	QuoteType/ QuoteRespType	LegOrderQty1	LegOrderQty2	OrderQty	QuoteStatus
FIX A						
QuoteStatusReport		n/a	n/a	n/a	n/a	cancelled

4) FIX B will receive ‘Quote Status Report’ (cancelled)

MFG=>FC	FC=>MFG	QuoteType/ QuoteRespType	LegOrderQty1	LegOrderQty2	OrderQty	QuoteStatus
FIX B						
QuoteStatusReport		n/a	n/a	n/a	n/a	cancelled

6.1.2.19 Fully soft matched YOURS/MINE, NO Deal

[FULL.NODEAL.IOC.001] FIX A submits an original order of 150, FIX B submits YOURS/MINE order with original quantity of 120.Soft Matched 120.

1) Softmatch notification for FIXA and FIX B

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX A									
ExecutionReport(soft match report)	*Standard MFG ack*	30	120	150	n/a	n/a	120	Partially filled	I(Order Status)

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX B									
ExecutionReport(soft match report)	*Standard MFG ack*	0	120	120	n/a	n/a	120	filled	I(Order Status)

2) FIX A no deal, FIX A will receive only ‘Quote Status Report’ (cancelled)

MFG=>FC	FC=>MFG	QuoteType/ QuoteRespType	LegOrderQty1	LegOrderQty2	OrderQty	QuoteStatus
FIX A						

MFG=>FC	FC=>MFG	QuoteType/ QuoteRespType	LegOrderQty1	LegOrderQty2	OrderQty	QuoteStatus
QuoteStatusReport		n/a	n/a	n/a	n/a	cancelled

3) FIX B will receive 'Quote Status Report' (cancelled)

MFG=>FC	FC=>MFG	QuoteType/ QuoteRespType	LegOrderQty1	LegOrderQty2	OrderQty	QuoteStatus
FIX B						
QuoteStatusReport		n/a	n/a	n/a	n/a	cancelled

6.1.2.20 Partially soft matched YOURS/MINE, user logs out

[PARTIAL.LOGOUT.IOC.001] FIX A submits an original order of 120. FIX B submits YOURS/MINE order with original quantity of 180.Soft Matched 120.

1) Softmatch notification (FIX A and FIX B) – same as [PARTIAL.NODEAL.IOC.001]
2) FIX A logs out, FIX A will receive Execution Report(cancelled) followed by 'Quote Status report' (cancelled)

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX A									
ExecutionReport(cancelled)	*Standard MFG ack*	0	120	120	n/a	n/a	n/a	filled	Cancelled

MFG=>FC	FC=>MFG	QuoteType/ QuoteRespType	LegOrderQty1	LegOrderQty2	OrderQty	QuoteStatus
FIX A						
QuoteStatusReport		n/a	n/a	n/a	n/a	cancelled

3) FIX B (counterparty) will receive only **'**Quote Status report'

MFG=>FC	FC=>MFG	QuoteType/ QuoteRespType	LegOrderQty1	LegOrderQty2	OrderQty	QuoteStatus
FIX B						
QuoteStatusReport		n/a	n/a	n/a	n/a	cancelled

6.1.2.21 Fully soft matched YOURS/MINE, user logs out

[FULL.LOGOUT.IOC.001] FIX A submits an original order of 150. FIX B submits YOURS/MINE order with original quantity of 120.Soft Matched 120.

1) Softmatch notification (FIX A and FIX B) – same as [FULL.NODEAL.IOC.001]
2) FIX A logs out, FIX A will receive Execution Report (cancelled) followed by 'Quote Status report' (cancelled).

MFG=>FC	FC=>MFG	Leaves Qty	Cum Qty	Order Qty	LegOrder Qty1	LegOrder Qty2	LastQty	OrdStatus	ExecType
FIX A									
ExecutionReport(cancelled)	*Standard MFG ack*	0	120	150	n/a	n/a	n/a	cancelled	4(cancelled)

MFG=>FC	FC=>MFG	QuoteType/ QuoteRespType	LegOrderQty1	LegOrderQty2	OrderQty	QuoteStatus
FIX A						
QuoteStatusReport		n/a	n/a	n/a	n/a	cancelled

3) FIX B (counterparty) will receive only 'Quote Status report'

MFG=>FC	FC=>MFG	QuoteType/ QuoteRespType	LegOrderQty1	LegOrderQty2	OrderQty	QuoteStatus
FIX B						
QuoteStatusReport		n/a	n/a	n/a	n/a	cancelled

6.1.3 Minimum Quote Life Workflows

The following sections summarize the reject messaging workflows and relevant tags as a result of MQL violations.

6.1.3.1 Single Order Cancel

Client to MFG	MFG to Client
OrderCancelRequest (F)	**ExecutionReport (8)** Tag 39/OrderStatus = 6 (Pending Cancel)

	OrderCancelReject (9) Tag102/CxlRejReason = 99 (Other) Tag58/Text = “MQL” Tag 39/OrdStatus =0 (New) or 1 (Partially Filled) Tag 41/ OrigClOrdID = ClOrdID (tag11) of the original order.

6.1.3.2Cancellation of an I-Deal Order

Client to MFG	MFG to Client
ListCancelRequest (K) Tag66/ListID	**ListStatus** (N) Tag 429 /ListStatusType=1
	ListStatus (N) (one only, representing both sides) Tag 429 /ListStatusType=2 Tag1386/ListRejectReason = 99 (Other) Tag444/ListStatusText = “MQL” For each side represented in the OrdListStatGrp: Tag103/OrdRejReason = 99 (Other) Tag58/Text = “MQL:0” if the order was not cancelled because *the other side* would have violated MQL = “MQL:1” if the order was not cancelled because this side would have violated MQL

6.1.3.3Cancel All

Client to MFG	MFG to Client
OrderMassActionRequest(CA) Tag11/ClOrdID Tag1373/MassActionType = 3 (Cancel Orders)	**OrderMassActionReport (BZ)** Tag11/ClOrdID Tag1373/MassActionType = 3 (Cancel Orders) Tag1373/MassActionScope = 1 All orders for the security = 5 All orders for the security type = 7 All orders Tag1375/MassActionResponse = 1 (Accepted)

	ExecutionReport (8) for each order successfully cancelled Tag 39/OrdStatus =4 (cancelled) Tag150/ExecType =4 (cancelled) **OrderCancelReject** (9) for each order not cancelled due to MQL Tag102/CxlRejReason = 99 (Other) Tag58/Text = “MQL” Tag434/CxlRejResponseTo = 1 (Order Cancel Request) Tag 39/OrdStatus =0 (New) or 1 (Partially Filled) Tag 41/ OrigClOrdID =ClOrdID (tag11) of the original order Tag11/ClOrdID = same as tag 11 from CA message.
	OrderMassActionReport (BZ) Tag11/ClOrdID Tag533/TotalAffectedOrders = total number of orders successfully cancelled

6.1.3.4 Hold All Request

Client to MFG	MFG to Client
OrderMassActionRequest (CA) Tag11/ClOrdID Tag1373/MassActionType = 1 (Suspend Orders) Tag1374/MassActionScope = 7 (All orders)	OrderMassActionReport (BZ) Tag11/ClOrdID Tag1373/MassActionType = 1 (Suspend Orders) Tag1374/MassActionScope = 7 (All orders) Tag1375/MassActionResponse = 1 (Accepted)

	ExecutionReport (8) for each order successfully held, Tag 39/OrdStatus = 9 (suspended) Tag150/ExecType =9 (suspended) **ExecutionReport (8)** for each order not held due to MQL, as above except: Tag150/ExecType = D (Restated) – Newly introduced for MQL violations Tag39/OrdStatus = 0 (New) or 1 (Partially Filled) Tag378/ExecRestatementReason = 99 (Other) Tag151/LeavesQty= values that are associated with the active order which remains in the market Tag58/Text (see note1 below) = "MQL:0" if the order was not held because *another* order would have violated MQL = "MQL:1" if the order was not held because *it* would have violated MQL
	OrderMassActionReport (BZ) Tag11/ClOrdID Tag533/TotalAffectedOrders = total number of orders successfully held.

6.1.3.5 Single Order Entry with AOR or Jobbing Mode Enabled

Client to MFG	MFG to Client
NewOrderSingle (D)	**ExecutionReport** (8) Tag103/OrdRejReason = 99 (Other) Tag58/Text = "MQL"

6.1.3.6 I-Deal Order Entry with AOR or Jobbing Mode Enabled

Client to MFG	MFG to Client
NewOrderList (E)	2 x **ExecutionReport (8)** each report contains: Tag 39/ OrdStatus =8 (rejected) Tag150/ExecType =8 (rejected) Tag103/OrdRejReason = 99 (Other) Tag58/Text = "MQL"

6.1.3.7 Summary of Tag 58 (Text) for MQL Violations

- 'MQL' -- denotes that the transaction could not complete due an MQL violation associated with this order.
- 'MQL:0 -- represents that a different order (not this order) within the transaction caused an MQL violation for the given instrument
- 'MQL:1' -- represents that the order specified within 'this' ExecutionReport caused the MQL violation for the given instrument.
- The syntax 'MQL' is also used in tag 444 as a response message to a ListCancelRequest(K)

6.1.3.8 MQL Impact to MAPI Client Applications

The introduction of MQL only introduces two new enumerated values does and does not introduce any new messages to the existing Matching FIX Interface. The new values are both used within ExecutionReports relating to failed Hold All requests and are outlined below:

- ExecRestatementReason(378) = 99 (Other)
- ExecType(150) = D (Restated) – indicates that the order has not been held and is still active

6.1.4 Maximum Order Size Workflows

The following sections summarize the reject messaging workflows and relevant tags as a result of MOS violations.

6.1.4.1 Order Entry MOS Rejections

Client to MFG	MFG to Client
NewOrderSingle (D)	**ExecutionReport** (8) Tag103/OrdRejReason = 99 (Other) Tag58/Text = "MOS"

6.1.4.2 I-Deal Order Entry MOS Rejections

Client to MFG	MFG to Client
NewOrderList (E)	**2 x ExecutionReport** (8) each report contains: Tag 39/ OrdStatus =8 (rejected) Tag150/ExecType =8 (rejected) Tag103/OrdRejReason = 99 (Other) Tag58/Text = "MOS"

6.1.5 Maximum Open Orders per Instrument Workflows

The following sections summarize the reject messaging workflows and relevant tags as a result of MOOPI violations.

6.1.5.1Order Entry MOOPI Rejections

Client to MFG	MFG to Client
NewOrderSingle (D)	**FIX BusinessMessageReject (35=j)** Tag 45/RefSeqNum = <Sequence Number of the Rejected Message> Tag 58/Text = > OL:<limit> e.g. OL:200 Tag 372/RefMsgType = The FIX message type of the message rejected Tag 380/BusinessRejectReason = 6 (Not Authorised)

6.1.5.2I-Deal Order Entry MOOPI Rejections

Client to MFG	MFG to Client
NewOrderList (E)	**2 x FIX BusinessMessageReject (35=j)** each report containing: Tag 45/RefSeqNum = <Sequence Number of the Rejected Message> Tag 58/Text = > OL:<limit> e.g. OL:200 Tag 372/RefMsgType = The FIX message type of the message rejected Tag 380/BusinessRejectReason = 6 (Not Authorised)

Chapter 7 Trade Capture Report Description

7.1 FX Spot/Swap Matching Trade Capture Report

.[1] This chapter provides an overview of the Trade Capture Report for FX Spot and SwapTrades to assure that the post trade processing is clearly defined.

7.2 Components of the Trade Capture Report

Fields within the Matching Trade Capture Report are categorized into the following logical components:

- Main section of the Message
- Instrument Specific Tags
- Trade Details
- Report Side
- Root Parties Block
- Settlement Details Block

7.2.1 Main Section of the TCR Message

The main block of the Trade Capture Report (TCR) the TCR identifiers and specific execution information such as price, quantity, side, match status, trade type, and execution time value date etc.

The following descriptions apply to the TCR identifiers:

- **MatchStatus** indicates the confirmation status of the trade. Trades consummated on Matching can be marked as confirmed or unconfirmed. An unconfirmed trade indicates that the matching service has booked the trade however the recipient's counterparty may not have received their match notification and or trade capture report. The trade is still valid and is requires settlement with the counterparty.
- **TradeId** represents the trade identifier allocated by the Matching service. This value is available on the users Daily Confirmation Statement (DCS).
- **TrdMatchId** represents the unique match identifier associated with the trade. For multi-way trades one or more match identifiers can be associated with a given trade Id. This identifier is also provided in the match execution report (match notification) message. This value is available on the user's Daily Confirmation Statement.
- **ExecId** represents the Matching transaction Id which was allocated to the match execution report (match notification) message. This value is available on the Daily Confirmation Statement.
- **ExecType** describes the specific execution, for Spot and Swap Matching a value of 'F' - (Trade) will be used.

[1] FX Swaps deals can still be delivered directly to manual users who negotiate FX Swap deals on the specified transfer Keystation.

- **TradeReportType** represents the type of report that has been issued by the matching service. In Matching this will always be set to 2 = 'Accept'.
- **TradeReportTransType** will always be set to 0 = 'New'. Trade cancellations are not supported by the Matching service.
- **UnsolicitiedIndicator** indicates whether or not message is being sent as a result of a subscription request or not. This value is always set to Y.
- **TrdTyp** represents the type of trade that has been published within the trade capture report. Spot and Swap Matching on Reuters trades are listed as (54) OTC trades.
- **TrdMatchID – Unique Match identifier**

7.2.2 Instrument Specific Tags

The instrument detail provides information pertaining to the traded currency pair.

- **TradeDate** indicates the date of the trade.
- **TransactTime** indicates the time (UTC) of the trade.
- **SettleDate** indicates the value date for the Spot trade. FX Spot will be T+1 (CAD) or T+2. Not used for an FX Swap trade
- **Symbol** represents currency pair traded.
- **SecurityType:**
  - **FXSPOT**
  - **FXSWAP**
- **ContractMultiplier** describes the ratio or multiply factor to convert from "nominal" units (e.g. lots of 1 million) to total units (e.g. millions of the base currency). In Matching this value is always set to 1,000,000 which represents 1 million of base currency. This also currently represents the minimum order size on the Matching service.
- **Currency** describes the dealt (base) currency of the trade. Follows the ISO standard.

SettleCurrency Represents the Contra currency of the deal. This Applies to FX Spot and both legs of the FX Swaps Deal and follows the ISO standard.

7.2.3 Trade Detail Tags

- **OrderQty** specifies the original order quantity of the order which this trade relates to.
- **LastQty** specifies the traded quantity (in millions) of the deal. This value is always a positive integer
- **LastPx** specifies the price of the quoted currency against the base currency for this trade. For an FX Swap trade it is the All-in price for Far leg.
- **CalculatedCCYLastQty** for Spot this specifies the calculated quantity of the quoted currency (i.e. CCY 2) for this trade. The quoted quantity is derived from (LastQty x LastPx). Not needed for FX Swaps refer to the leg level for FX Swap
- **PriceType** is used to define an enumerated value which denotes the way in which the currencies involved with the trade are quoted (i.e. Normal or Inverse).
  - Normal implies that the rate in the deal represents the number of units of currency 2 needed to purchase one unit of currency 1. Thus, the volume of currency 2 is equal to the volume of currency 1 multiplied by the exchange rate.

- Inverse implies that the rate in the deal represents the number of units of currency 1 needed to purchase one unit of currency 2. Thus, the volume of currency 2 is equal to the volume of currency 1 divided by the exchange rate.

- **LastSpot Rate.** For an FX Swap trade this is the Spot Reference Rate which in most cases is the price associated with the near leg of the transaction.**LastSwapPoints.** For FX Swap, this is used to express the last market event for the differential between the far leg's bid/offer and the near leg's bid/offer in a fill or partial fill. Value can be negative. Expressed in decimal form. For example, 61.99 points is expressed and sent as 0.006199

7.2.4 Traded Instrument Leg Group

This is repeating group for a FX Swap trade

- **NoLegs.** For an FX Swap trade this is always 2
- **LegSymbol** For a FX Swap trade this is the same as Tag 55
- **LegSide** The trade direction of the leg 1=Buy, 2=Sell
- **LegQty** The trade quantity for the leg
- **LegRefID** Unique indicator for a specific leg for the FX Swap trade. Supported value:1st leg =Near, 2nd leg=Far
- **LegSettlDate** The settlement date for the leg, for an FX swap this will be either the near leg date or the far leg date.
- **LegLastPx** Near or far leg all-in price. Typically, the Near Leg will be Spot Ref Rate and Far Leg will be the All In Rate.
- **LegCalculatedCcy LastQty**. The calculated quantity of the quoted currency for this leg of the FX Swap trade.

7.2.5 Report Side

- **NoSides** defines the number of sides that are being reported within this message. Matching trades are always reported from the perspective of the receiving user as a single side. For FX Spot tag 552 is always = 1, for FX Swaps tag 552 is always = 2
- **Side** describes the side of the deal (1 = BUY, 2 = SELL).
- **SideCurrency Used defines** the trading currency on the Trade Capture Report Side for an FX Swap Trader. This is the volume currency of the deal.
- **AggressorIndicator** a flag which denotes whether or not this side of the trade was standing in the book at the time of the match or if it was the aggressing order that triggered the match event. Y- Order initiator is aggressor , N - Order initiator is passive
- **OrderId** represents the unique integer value allocated by the Matching service. This value is only unique per instrument per trading session and is sometimes referred to as the Matching Order Id.
- **ClOrdId** represents a unique integer value allocated by the FIX client application. This order id must be unique per user, per instrument per trading session. Duplicate ClOrdIds from the same user will not be rejected by the Matching service however the submission of duplicate orders could result in incorrect order cancelation results.
- **Account** identifies the trading account associated with the originating order. Account tags are useful for position keeping and a prime broker post trade management workflows.
- **LimitRemainingAmt** defines an integer value which denotes the bi-lateral credit which remains between the two counterparties after the subject trade has been executed. Within Matching the credit remaining value is always denominated in whole millions and is only required for FX Spot trades.

- **LastLimitAmt** specifies the quantity which was drawn down (or consumed) between the two counterparties as a result of the trade. Within Matching the credit drawn down value is always denominated in whole millions and is only required for FX Spot trades.

7.2.6 RootPartyId Block

Matching always provides two RootPartyId blocks within each Trade Capture Report (TCR). The receiver's RootPartyId block is always listed first followed by the counterparties RootPartyId Block. Each RootPartyId block contains a party identifier (1117), the Party's Role (1119) and Party Source (1118). Within a given party Id block RootPartySubIds are listed to further define the name of the parties Matching Site Id (credit code). In the case where the block represents the originator of the order an additional RootPartySubId is provided which denote the entering trader's name. Descriptions of the root party identification block are provided below.

Tag	Tag name	Valid Values
1116	NoRootPartyIDs	Defines the number of party identification blocks that are being reported within this TCR (e.g. Number of PartyID (1117), PartyIDSource (1118), and PartyRole (1119) entries). Always = 2
1117	RootPartyId	This represents the name of the originating firm's branch.
1118	RootPartyIDSource	C – Generally Accepted Market Participant is used to denote the name (Answerback) of the of the 'parties' branch.
1119	RootPartyRole	13 – Order Origination Firm is used when 'this party' is the owner of the TCR. 56 – Acceptable counterparty is used when 'this party' is the counterparty of the deal.
1120	NoRootPartySubIDs	1 – When the Party Role = 56 Acceptable Counterparty 2 – When the side of the deal is for 'this party'. The second identifier is used to specify trader name of 'this party'. Trader name is always listed after the trading desk identifier. Traders names are not disclosed to the counterparty.
1121	RootPartySubID	Matching Site Id of the Trader's branch.
1122	RootPartySubIDType	25 -- Location Desk Represents the originating users master dealing code (TCID).
1121	RootPartySubID	Login name of the user that that entered the original order.
1122	RootPartySubIDType	2 – Person. Only provided when receiving user is the owner of 'this' TCR.

Table 2: RootPartyId Block Description

7.2.7 Settlement Details Block

Tag	Tag name	Valid Values
1158	NoSettleDetails	Used to group each settlement party and always set to 2. **Important Note:** The first set of settle details provided within each TCR will be always represent the settlement details which have been defined by the receiver of this TCR. The second set of settle details provided within each TCR will always represent the settlement details which been defined by the receiver's counterparty.
1164	SettlObligSource	4 – Buyer 5 – Seller
781	NoSettlPartyIds	Repeating group to define the settlement parties. In Matching this will always be set to 1 for each set of SettlDetails.
782	SettlPartyID	This represents the Buyer or Seller's instructions for payment. If no details are provided for this party then this field contains "No Instructions Specified" For FX Swap Deals this tag contains the payment instructions for the given recipients LegSide of the Deal.
783	SettlPartyIDSource	C – Generally Accepted Market Participant is used to denote the name of the settle party.
784	SettlPartyRole	86 – 'CLS Member Bank'. Used to denote when the counterparty of the trade has been identified as being CLS eligible. 27 - Buyer/Seller (Receiver/Deliverer) for all non-CLS trades. The receiver's SettleRole will always be provided within the first of the two SettleDetails blocks provided. As such, receivers of this message should ignore the value of 86 when set within the first SettleDetails block. To determine the CLS status of your counterparty receivers of this message should inspect the second SettlPartyRole within the second block SettlDetails block.
801	NoSettlPartySubIDs	Will be = 1 when SWIFT BIC is specified by this party. Otherwise will be omitted. 1 -- if the BIC code has been specified
785	SettlPartySubID	Settlement Parties BIC or SWIFT code.
786	SettlPartySubIDType	16 – Bank Identifier Code (BIC) or SWIFT Code.

Table 3: Settlement Details Party Block

7.2.7.1 Sample Settlement Details structure for an FX Swap

MAPI (rate/volume) quotation

- The 1st currency displayed in FIX tag 55(symbol) is a base currency and the 2nd currency is a quoted currency
- Tag 15(currency) represents the **volume currency.**

  For normal quotation => base currency = volume currency
  For inverse quotation=>base currency = volume currency

- Trade direction is based on the 'Volume currency' perspective.

  For normal quotation
  BID =>sell volume ccy near leg and buy volume far leg (SELL/BUY)
  OFFER=> buy volume near leg and sell volume far leg. (BUY/SELL)

  For inverse quotation
  BID =>buy volume ccy near leg and sell volume far leg. (BUY/SELL)
  OFFER=> sell volume near leg and buy volume far leg. (SELL/BUY)

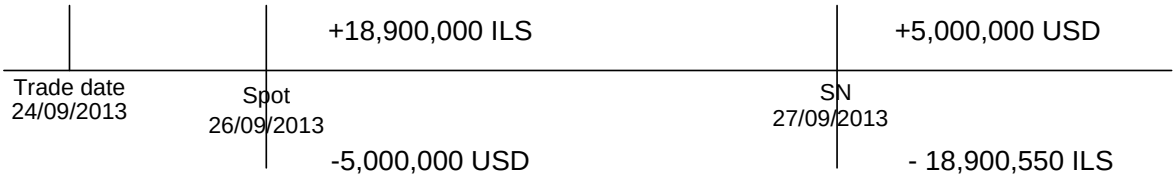
- In Matching, there are 2 types of quotation methods:
  - Based on the market convention for Price (rate quotation). **All** Matching instruments follow the normal market convention for Price (Rate) Quotation i.e. the exchange rate do not change no matter which volume currency is a quoted or a based currency.
  - Based on volume currency convention.
    - If Matching instruments have the volume currency designated as the base currency, then tag 423(PriceType) =20 (normal quotation) It means 1 unit of ccy1 can be exchanged for 'X' units of ccy2.
    - If Matching instruments have the volume currency designated as the quoted currency, then tag 423 (PriceType) =21 (Inverse quotation). It means1 unit of ccy 2 can be exchanged for 'X' units of ccy1

The Market convention for Price is always normal; the samples of Settlement Details in this section will be based on the volume currency convention.

Sample 1 - Normal Quotation

- Party A enters a (BID) Buy order for **5** Million **USD/ILS SN** and consummates an FX swap deal with Party B. Party A is the Buyer, Party B is the Seller of the Swap
- Spot rate =3.78 and swap point =1.1 (un-scaled). Near leg rate =3.78 and far leg rate=3.78011
- Tenor: near leg is Spot and far leg is 1 D (after Spot)
- Volume currency (15) and side currency (1154) =>USD.
- Payment instruction = receiving account (where money will be paid to)
- Trade direction :
  - **Party A** buy (volume) USD far leg => tag 54 =1(in order entry messages), 624[leg1] =2 and 624[leg2] =1. This is a **SELL/BUY** swap i.e. sell volume ccy near leg and buy volume ccy far leg.

    At Spot, Party A will give 5 million USD and receive 18,900,000 mio ILS. At 1 day after spot, Party A will give 18,900,550 Mio ILS and receive 5 million USD.



- **Party B** sell (volume) USD far leg => tag 54 =2(in order entry messages), 624[leg1] =1 and 624[leg2]=2. This is a **BUY/SELL** swap i.e. i.e. buy volume ccy near leg and sell volume ccy far leg.

  At Spot, Party B will give 18,900,000 mio ILS and receive 5mio USD. At 1 day after Spot, Party B will give 5 Mio USD and receive 18,900,550 mio ILS.

TCR for Party A (sell /buy swap) sell USD near leg buy USD far leg.

Note: The details for the receiver of this Trade Ticket will always be listed first. This holds true for Side and NoSelltDetails

Tag 15 =USD

Tag120=ILS

Tag624 (1st leg =2 sell (sell USD) 2nd leg =1 (buy USD)

Tag423=20(normal)

Tag1154 =USD

<TrdCapRptSideGrp> repeating group for period1 (near leg)		
NoSides	552	2
→Side	54	2(Sell)
->SideCurrency	1154	USD
→Account	1	Receiver's Account Number Specified during order entry (Party A's Account)
→OrderID	37	Unique Matching Order I
→ClOrdID	11	Assigned ID by the FIX client at Order Entry (Party A)
➔ **<SettlDetails> repeating group**		
→NoSettlDetails	1158	2
Settlement Detail Block1 - TCR receiver (party A) Period 1		
→→SettlObligSource	1164	5 (seller) - Party A sells USD (base) near leg.
➔ **<SettlParties> repeating group**		
→→NoSettlPartyIDs	781	1
→→→SettlPartyID	782	Party A's ILS account (Party A receive ILS)

→→→SettlPartyIDSource	783	C – Generally Accepted Market
→→→SettlPartyRole	784	86 – 'CLS Member Bank'. Used when the trade is marked for CLS settlement. 27 - Buyer/Seller (Receiver/Deliverer) for all non-CLS trades.
→→→NoSettlPartySubIDs	801	1 - When SWIFT BIC is specified by this party. Otherwise this repeating block will be omitted
→→→→SettlPartySubID	785	Carries the BIC or SWIFT code for the Parties A - ILS Instruction. Conditionally required if tag 801 = 1.
→→→→SettlPartySubIDType	786	16(BIC) Identifies SubId type e.g. Always =16 (BIC). Conditionally required if tag 801 = 1.
Settlement Detail Block2 - counterparty (party B)of the TCR period1		
→→SettlObligSource	1164	4 (buyer) - Party B buys USD near leg
➔ **<SettlParties> repeating group - see sample below**		
→→NoSettlPartyIDs	781	1
→→→SettlPartyID	782	Party B's USD account (Party B receive USD)
→→→SettlPartyIDSource	783	C – Generally Accepted Market
→→→SettlPartyRole	784	86 – 'CLS Member Bank'. Used when the trade is marked for CLS settlement. 27 - Buyer/Seller (Receiver/Deliverer) for all non-CLS trades.
→→→NoSettlPartySubIDs	801	1 (when SWIFT BIC is specified by this party. Otherwise will be omitted.)
→→→→SettlPartySubID	785	Carries the BIC or SWIFT code for the Parties B - USD Instruction. .Conditionally required if tag 801 = 1.
→→→→SettlPartySubIDType	786	16(BIC) Identifies SubId type e.g. Always =16 (BIC). Conditionally required if tag 801 = 1.
<TrdCapRptSideGrp> repeating group for period 2(far leg)		
→Side	54	1(buy)
->SideCurrency	1154	USD
→Account	1	Receiver's Account Number Specified in the Order Entry (In this case Party A's Account)
→OrderID	37	Unique Matching Order ID
→ClOrdID	11	Assigned ID by the FIX client at Order Entry (In This case Party A)
➔ **<SettlDetails> repeating group**		
→NoSettlDetails	1158	2
Settlement Detail Block1 - TCR receiver (party A) period2		
→→SettlObligSource	1164	4(buyer) - Party A buys USD far leg.
➔ **<SettlParties> repeating group**		
→→NoSettlPartyIDs	781	1
→→→SettlPartyID	782	Party A's USD account (Party A receive USD)
→→→SettlPartyIDSource	783	C – Generally Accepted Market
→→→SettlPartyRole	784	86 – 'CLS Member Bank'. Used when the trade is marked for CLS settlement. 27 - Buyer/Seller (Receiver/Deliverer) for all non-CLS trades.

→→→NoSettlPartySubIDs	801	1 (when SWIFT BIC is specified by this party. Otherwise will be omitted.)
→→→→SettlPartySubID	785	Carries the BIC or SWIFT code for Parties A - USD Instruction for far leg. Conditionally required if tag 801 = 1.
→→→→SettlPartySubIDType	786	16(BIC) Identifies SubId type e.g. Always =16 (BIC). Conditionally required if tag 801 = 1.
Settlement Detail Block2 - counterparty (party B)of the TCR period2		
→→SettlObligSource	1164	5(seller) - Party B sells USD far leg.
➔ **<SettlParties> repeating group**		
→→NoSettlPartyIDs	781	1
→→→SettlPartyID	782	Party B's ILS account (Party B receive ILS)
→→→SettlPartyIDSource	783	C – Generally Accepted Market
→→→SettlPartyRole	784	86 – 'CLS Member Bank'. Used when the trade is marked for CLS settlement. 27 - Buyer/Seller (Receiver/Deliverer) for all non-CLS trades.
→→→NoSettlPartySubIDs	801	1 (when SWIFT BIC is specified by this party. Otherwise will be omitted.)
→→→→SettlPartySubID	785	Carries the BIC or SWIFT code for Parties B - ILS Instruction for far leg. Conditionally required if tag 801 = 1.
→→→→SettlPartySubIDType	786	16(BIC) Identifies SubId type e.g. Always =16 (BIC). Conditionally required if tag 801 = 1.

TCR for Party B (buy/sell swap) – buy USD near leg sell USD far leg.

Note: The details for the receiver of this Trade Ticket will always be listed first. This holds true for NoSides and NoSelltDetails repeating groups

Tag 15 =USD

Tag120= ILS

Tag624 (1st leg =1 buy (buy USD) 2nd leg =2 (sell USD)

Tag423=20(normal)

Tag1154 =USD

<TrdCapRptSideGrp> repeating group for period1 (near leg)		
NoSides	552	2
→Side	54	1(buy)
->SideCurrency	1154	USD
→Account	1	Receiver's Account Number Specified during order entry (Party B 's Account)
→OrderID	37	Unique Matching Order I
→ClOrdID	11	Assigned ID by the FIX client at Order Entry (Party B)
➔ **<SettlDetails> repeating group**		
→NoSettlDetails	1158	2
Settlement Detail Block1 - TCR receiver (party B) Period 1		

→→SettlObligSource	1164	4(buyer) - Party B buys USD (base) near leg.
➔ **<SettlParties> repeating group**		
→→NoSettlPartyIDs	781	1
→→→SettlPartyID	782	Party B's USD account (Party B receive USD)
→→→SettlPartyIDSource	783	C – Generally Accepted Market
→→→SettlPartyRole	784	86 – 'CLS Member Bank'. Used when the trade is marked for CLS settlement. 27 - Buyer/Seller (Receiver/Deliverer) for all non-CLS trades.
→→→NoSettlPartySubIDs	801	1 - When SWIFT BIC is specified by this party. Otherwise this repeating block will be omitted
→→→→SettlPartySubID	785	Carries the BIC or SWIFT code for the Parties B -USD Instruction. Conditionally required if tag 801 = 1.
→→→→SettlPartySubIDType	786	16(BIC) Identifies SubId type e.g. Always =16 (BIC). Conditionally required if tag 801 = 1.
Settlement Detail Block2 - counterparty (party A)of the TCR period1		
→→SettlObligSource	1164	5 (Seller) - Party A sells USD near leg
➔ **<SettlParties> repeating group - see sample below**		
→→NoSettlPartyIDs	781	1
→→→SettlPartyID	782	Party A's ILS account (Party A receive ILS)
→→→SettlPartyIDSource	783	C – Generally Accepted Market
→→→SettlPartyRole	784	86 – 'CLS Member Bank'. Used when the trade is marked for CLS settlement. 27 - Buyer/Seller (Receiver/Deliverer) for all non-CLS trades.
→→→NoSettlPartySubIDs	801	1 (when SWIFT BIC is specified by this party. Otherwise will be omitted.)
→→→→SettlPartySubID	785	Carries the BIC or SWIFT code for the Parties A – ILS Instruction. Conditionally required if tag 801 = 1.
→→→→SettlPartySubIDType	786	16(BIC) Identifies SubId type e.g. Always =16 (BIC). Conditionally required if tag 801 = 1.
<TrdCapRptSideGrp> repeating group for period 2(far leg)		
→Side	54	2 (sell)
->SideCurrency	1154	USD
→Account	1	Receiver's Account Number Specified in the Order Entry (In this case Party B's Account)
→OrderID	37	Unique Matching Order ID
→ClOrdID	11	Assigned ID by the FIX client at Order Entry (In This case Party B)
➔ **<SettlDetails> repeating group**		
→NoSettlDetails	1158	2
Settlement Detail Block1 - TCR receiver (party B) period2		
→→SettlObligSource	1164	5(seller) - Party B sells USD far leg.
➔ **<SettlParties> repeating group**		
→→NoSettlPartyIDs	781	1
→→→SettlPartyID	782	Party B's ILS account (Party B receive ILS)

→→→SettlPartyIDSource	783	C – Generally Accepted Market
→→→SettlPartyRole	784	86 – 'CLS Member Bank'. Used when the trade is marked for CLS settlement. 27 - Buyer/Seller (Receiver/Deliverer) for all non-CLS trades.
→→→NoSettlPartySubIDs	801	1 (when SWIFT BIC is specified by this party. Otherwise will be omitted.)
→→→→SettlPartySubID	785	Carries the BIC or SWIFT code for Parties B- ILS Instruction for far leg. Conditionally required if tag 801 = 1.
→→→→SettlPartySubIDType	786	16(BIC) Identifies SubId type e.g. Always =16 (BIC). Conditionally required if tag 801 = 1.
Settlement Detail Block2 - counterparty (party A)of the TCR period2		
→→SettlObligSource	1164	4(buyer) - Party A buys USD far leg.
➔ **<SettlParties> repeating group**		
→→NoSettlPartyIDs	781	1
→→→SettlPartyID	782	Party A's USD account (Party A receive USD)
→→→SettlPartyIDSource	783	C – Generally Accepted Market
→→→SettlPartyRole	784	86 – 'CLS Member Bank'. Used when the trade is marked for CLS settlement. 27 - Buyer/Seller (Receiver/Deliverer) for all non-CLS trades.
→→→NoSettlPartySubIDs	801	1 (when SWIFT BIC is specified by this party. Otherwise will be omitted.)
→→→→SettlPartySubID	785	Carries the BIC or SWIFT code for Parties A – USD Instruction for far leg. Conditionally required if tag 801 = 1.
→→→→SettlPartySubIDType	786	16(BIC) Identifies SubId type e.g. Always =16 (BIC). Conditionally required if tag 801 = 1.

Sample 2 – Inverse Quotation

- Party A enters a (BID) Buy order for 5 Million **USD/ILS ON** and consummates an FX swap deal (transaction) with Party B. Party A is the Buyer, Party B is the Seller of the Swap
- In Matching Tonor = ON (overnight), the near leg value date is tomorrow (after trade date) and far leg value date is Spot (1 or 2 Business day after trade date depends on trade currency pair)
- Volume currency (15) and side currency (1154) =>ILS.
- Trade direction :
    - **Party A** buys ILS near leg => tag 54 =1 and tag 15 =ILS (in order entry messages).

      It is a **BUY/SELL** swap. Buy volume ccy (ILS) near leg and sell volume ccy far leg => tag 624[leg1] =1 and 624[leg2] =2.

      At tomorrow after trade date, Party A will receive 5 mio ILS and give 1,322,998 USD
      At Spot, Party A will receive 1,322,751 Mio USD and give 5 mio ILS to party B.

- **Party B** sell ILS near leg => tag 54 =2 and tag 15 =ILS (in order entry messages) Base currency (15) and side currency =>ILS.

  It is a **SELL/BUY** swap. Sell volume ccy (ILS) near leg and buy volume ccy far leg => (624[leg1] =2 and 624[leg2] =1.

  At tomorrow after trade date, Party B will receive 1,322,998 USD and give 5 mio ILS. At Spot, Party B will receive 5 Mio ILS and give 1,322,751 USD.

TCR for Party A (Buy/Sell swap) – Buy ILS near leg sell ILS far leg.

Note: The details for the receiver of this Trade Ticket will always be listed first. This holds true for NoSides and NoSelltDetails repeating groups

Tag 15 =ILS

Tag120=USD

Tag624 (1st leg =1 (buy ILS) 2nd leg =2 (sell ILS)

Tag423=21(inverse)

Tag1154 = ILS

<TrdCapRptSideGrp> repeating group for period1 (near leg)		
NoSides	552	2
→Side	54	1(buy)
->SideCurrency	1154	ILS
→Account	1	Receiver's Account Number Specified during order entry (Party A's Account)
→OrderID	37	Unique Matching Order I
→ClOrdID	11	Assigned ID by the FIX client at Order Entry (Party A)
➔ **<SettlDetails> repeating group**		
→NoSettlDetails	1158	2
Settlement Detail Block1 - TCR receiver (party A) Period 1		

→→SettlObligSource	1164	4 (buyer) - Party A buys ILS near leg
➔ **<SettlParties> repeating group**		
→→NoSettlPartyIDs	781	1
→→→SettlPartyID	782	Party A's ILS account (Party A receive ILS)
→→→SettlPartyIDSource	783	C – Generally Accepted Market
→→→SettlPartyRole	784	86 – 'CLS Member Bank'. Used when the trade is marked for CLS settlement. 27 - Buyer/Seller (Receiver/Deliverer) for all non-CLS trades.
→→→NoSettlPartySubIDs	801	1 - When SWIFT BIC is specified by this party. Otherwise this repeating block will be omitted
→→→→SettlPartySubID	785	Carries the BIC or SWIFT code for the Parties A -ILS Instruction. Conditionally required if tag 801 = 1.
→→→→SettlPartySubIDType	786	16(BIC) Identifies SubId type e.g. Always =16 (BIC). Conditionally required if tag 801 = 1.
Settlement Detail Block2 - counterparty (party B)of the TCR period1		
→→SettlObligSource	1164	5 (seller) - Party B sells ILS near leg
➔ **<SettlParties> repeating group - see sample below**		
→→NoSettlPartyIDs	781	1
→→→SettlPartyID	782	Party B's USD account (Party B receive USD)
→→→SettlPartyIDSource	783	C – Generally Accepted Market
→→→SettlPartyRole	784	86 – 'CLS Member Bank'. Used when the trade is marked for CLS settlement. 27 - Buyer/Seller (Receiver/Deliverer) for all non-CLS trades.
→→→NoSettlPartySubIDs	801	1 (when SWIFT BIC is specified by this party. Otherwise will be omitted.)
→→→→SettlPartySubID	785	Carries the BIC or SWIFT code for the Parties B - USD Instruction. .Conditionally required if tag 801 = 1.
→→→→SettlPartySubIDType	786	16(BIC) Identifies SubId type e.g. Always =16 (BIC). Conditionally required if tag 801 = 1.
<TrdCapRptSideGrp> repeating group for period 2(far leg)		
→Side	54	2(sell)
->SideCurrency	1154	ILS
→Account	1	Receiver's Account Number Specified in the Order Entry (In this case Party A's Account)
→OrderID	37	Unique Matching Order ID
→ClOrdID	11	Assigned ID by the FIX client at Order Entry (In This case Party A)
➔ **<SettlDetails> repeating group**		
→NoSettlDetails	1158	2
Settlement Detail Block1 - TCR receiver (party A) period2		
→→SettlObligSource	1164	5(seller) – Party A sells ILS far leg.
➔ **<SettlParties> repeating group**		

→→NoSettlPartyIDs	781	1
→→→SettlPartyID	782	Party A's USD account (Party A receive USD)
→→→SettlPartyIDSource	783	C – Generally Accepted Market
→→→SettlPartyRole	784	86 – 'CLS Member Bank'. Used when the trade is marked for CLS settlement. 27 - Buyer/Seller (Receiver/Deliverer) for all non-CLS trades.
→→→NoSettlPartySubIDs	801	1 (when SWIFT BIC is specified by this party. Otherwise will be omitted.)
→→→→SettlPartySubID	785	Carries the BIC or SWIFT code for Parties A –USD Instruction for far leg. Conditionally required if tag 801 = 1.
→→→→SettlPartySubIDType	786	16(BIC) Identifies SubId type e.g. Always =16 (BIC). Conditionally required if tag 801 = 1.
Settlement Detail Block2 - counterparty (party B)of the TCR period2		
→→SettlObligSource	1164	4(buyer) - Party B buys ILS far leg.
➔ **<SettlParties> repeating group**		
→→NoSettlPartyIDs	781	1
→→→SettlPartyID	782	Party B's ILS account (Party B receive ILS)
→→→SettlPartyIDSource	783	C – Generally Accepted Market
→→→SettlPartyRole	784	86 – 'CLS Member Bank'. Used when the trade is marked for CLS settlement. 27 - Buyer/Seller (Receiver/Deliverer) for all non-CLS trades.
→→→NoSettlPartySubIDs	801	1 (when SWIFT BIC is specified by this party. Otherwise will be omitted.)
→→→→SettlPartySubID	785	Carries the BIC or SWIFT code for Parties B – ILS Instruction for far leg. Conditionally required if tag 801 = 1.
→→→→SettlPartySubIDType	786	16(BIC) Identifies SubId type e.g. Always =16 (BIC). Conditionally required if tag 801 = 1.

TCR for Party B (sell/buy swap) – sell ILS near leg buy ILS far leg.

Note: The details for the receiver of this Trade Ticket will always be listed first. This holds true for NoSides and NoSelltDetails repeating groups

Tag 15 =ILS

Tag120=USD

Tag624 (1st leg =2 (sell ILS) 2nd leg =1 (buy ILS)

Tag423=21(inverse)

Tag1154 =ILS

<TrdCapRptSideGrp> repeating group for period1 (near leg)		
NoSides	552	2
→Side	54	2(sell)
->SideCurrency	1154	ILS

→Account	1	Receiver's Account Number Specified during order entry (Party B 's Account)
→OrderID	37	Unique Matching Order I
→ClOrdID	11	Assigned ID by the FIX client at Order Entry (Party B)
➔ **<SettlDetails> repeating group**		
→NoSettlDetails	1158	2
Settlement Detail Block1 - TCR receiver (party B) Period 1		
→→SettlObligSource	1164	5(seller) - Party B sells ILS near leg.
➔ **<SettlParties> repeating group**		
→→NoSettlPartyIDs	781	1
→→→SettlPartyID	782	Party B's USD account (Party B receives USD)
→→→SettlPartyIDSource	783	C – Generally Accepted Market
→→→SettlPartyRole	784	86 – 'CLS Member Bank'. Used when the trade is marked for CLS settlement. 27 - Buyer/Seller (Receiver/Deliverer) for all non-CLS trades.
→→→NoSettlPartySubIDs	801	1 - When SWIFT BIC is specified by this party. Otherwise this repeating block will be omitted
→→→→SettlPartySubID	785	Carries the BIC or SWIFT code for the Parties B- USD Instruction. Conditionally required if tag 801 = 1.
→→→→SettlPartySubIDType	786	16(BIC) Identifies SubId type e.g. Always =16 (BIC). Conditionally required if tag 801 = 1.
Settlement Detail Block2 - counterparty (party A)of the TCR period1		
→→SettlObligSource	1164	4 (Buyer) - Party A buys ILS near leg
➔ **<SettlParties> repeating group - see sample below**		
→→NoSettlPartyIDs	781	1
→→→SettlPartyID	782	Party A's ILS account (Party A receive ILS)
→→→SettlPartyIDSource	783	C – Generally Accepted Market
→→→SettlPartyRole	784	86 – 'CLS Member Bank'. Used when the trade is marked for CLS settlement. 27 - Buyer/Seller (Receiver/Deliverer) for all non-CLS trades.
→→→NoSettlPartySubIDs	801	1 (when SWIFT BIC is specified by this party. Otherwise will be omitted.)
→→→→SettlPartySubID	785	Carries the BIC or SWIFT code for the Parties A – ILS Instruction. .Conditionally required if tag 801 = 1.
→→→→SettlPartySubIDType	786	16(BIC) Identifies SubId type e.g. Always =16 (BIC). Conditionally required if tag 801 = 1.
<TrdCapRptSideGrp> repeating group for period 2(far leg)		
→Side	54	1 (buy)
->SideCurrency	1154	ILS
→Account	1	Receiver's Account Number Specified in the Order Entry (In this case Party B's Account)
→OrderID	37	Unique Matching Order ID

→ClOrdID	11	Assigned ID by the FIX client at Order Entry (In This case Party B)
➔ **<SettlDetails> repeating group**		
→NoSettlDetails	1158	2
Settlement Detail Block1 - TCR receiver (party B) period2		
→→SettlObligSource	1164	4(buyer) - Party B buys ILS far leg.
➔ **<SettlParties> repeating group**		
→→NoSettlPartyIDs	781	1
→→→SettlPartyID	782	Party B's ILS account (Party B receive ILS)
→→→SettlPartyIDSource	783	C – Generally Accepted Market
→→→SettlPartyRole	784	86 – 'CLS Member Bank'. Used when the trade is marked for CLS settlement. 27 - Buyer/Seller (Receiver/Deliverer) for all non-CLS trades.
→→→NoSettlPartySubIDs	801	1 (when SWIFT BIC is specified by this party. Otherwise will be omitted.)
→→→→SettlPartySubID	785	Carries the BIC or SWIFT code for Parties B- ILS Instruction for far leg. Conditionally required if tag 801 = 1.
→→→→SettlPartySubIDType	786	16(BIC) Identifies SubId type e.g. Always =16 (BIC). Conditionally required if tag 801 = 1.
Settlement Detail Block2 - counterparty (party A)of the TCR period2		
→→SettlObligSource	1164	5(seller) - Party A sells ILS far leg.
➔ **<SettlParties> repeating group**		
→→NoSettlPartyIDs	781	1
→→→SettlPartyID	782	Party A's USD account (Party A receive USD)
→→→SettlPartyIDSource	783	C – Generally Accepted Market
→→→SettlPartyRole	784	86 – 'CLS Member Bank'. Used when the trade is marked for CLS settlement. 27 - Buyer/Seller (Receiver/Deliverer) for all non-CLS trades.
→→→NoSettlPartySubIDs	801	1 (when SWIFT BIC is specified by this party. Otherwise will be omitted.)
→→→→SettlPartySubID	785	Carries the BIC or SWIFT code for Parties A – USD Instruction for far leg. Conditionally required if tag 801 = 1.
→→→→SettlPartySubIDType	786	16(BIC) Identifies SubId type e.g. Always =16 (BIC). Conditionally required if tag 801 = 1.

Appendix A Order Reject Workflow Summary

For a comprehensive list of the of the various reject scenarios which could be encountered when entering orders into Matching please refer to the **FIX Interface – Error Codes and Reasons** document.

Appendix B Tags Defined as part of EP100

1. EP100 Data Dictionary

The tags listed in **Error! Reference source not found.** have been identified for usage in order entry, order execution and trade capture reports.

Tag	Field Name	Action	Data type	Description	Applicable Messages
1385	ContingencyType	Add new enums	Int	5 = Bid and Offer – defines a contingency order with two or more sides (i.e. Bid and Offer). 6 = Bid and Offer OCO -- defines a contingency order with two or more sides (i.e. Bid and Offer) however one side is automatically cancelled when the other side is fully traded away. NoOrders tag in the NewOrderList is always an even number for these ContingencyTypes.	NewOrderList(E) ExecutionReport(8).
1100	TriggerType	Add new enum	Int	(5) – 'On Order Entry or Order Modification' This enumeration value can be used in conjunction with TriggerScope to indicate that the TriggerAction should be executed immediately upon order or order modification entry into the market.	Add new enumeration to TriggerType.
1628	TriggerScope	New Field	Int	Enumeration which defines the scope of TriggerAction = 3 (Cancel): Values: 0 – This order (default) 1 – Other order (use RefId) 2 – All other orders for the given security 3 – All other orders for the given security and price 4 – All other orders for the given security and side – Jobbing Mode 5 – All other orders for the given security, price and side – AOR	Add new enumerated value to the <TriggerInstruction> component. This new enumeration value is modelled after MassActionScope. TriggerScope will only be used with TriggerAction = 3 (Cancel).
59	TimeInforce	Add new enum	Int	New Enumeration Value 'A'– Good For Time – GFT	Add new enumerated value to the TimeInForce field.
1629	ExposureDuration	New Field	Int	For orders, this is the time in seconds of a "Good for Time' (GFT) TimeInForce. Positive integer value which represents the time is seconds in which the new order remains active in the market before it is automatically cancelled (e.g. expired).	Add the ExposureDuration field to ExecutionReport NewOrderSingle NewOrderList

Tag	Field Name	Action	Data type	Description	Applicable Messages
378	ExecRestatementReason	Add new enum		12 -- “Cancel On Connection Loss” This value is used only when unsolicited cancels are generated as a result of a network disconnect. 13 – “Cancel on Logout” – used to denote the case where unsolicited cancels are sent as a result of a user logout request while he has open orders.	Add new enumerated value to the ExecRestatementReason field.
1630	NoLimitAmts	New Field	Int	Indicates the number of Limits specified within the repeating group	Add the new field to: TradeCaptureReport (AE) ExecutionReport(8).
1631	LimitAmtType	New Field	Int	Enumeration with values of: 0 = Credit Limit 1= Gross Position Limit 2 - Net Position Limit 3 - Risk Exposure Limit 4 - Long Position Limit 5 - Short Position Limit Or any value conforming to the data typeReserved100Plus. Value must be provided if 1630 is greater than 1.	Add the new field to: TradeCaptureReport (AE) ExecutionReport(8).
1632	LastLimitAmt	New Field	Amt	Indicates the amount of the specified ‘LimitType’ that has been drawn down against the counterparty for the given trade.	Add the new field to: TradeCaptureReport (AE) ExecutionReport(8).
1633	LimitAmtRemaining	New Field	Amt	Indicates the remaining amount of the specified ‘LimitType’ between the receiver of the TCR (or ExectionReport) and the specified counterparty..	Add the new field to: TradeCaptureReport (AE) ExecutionReport(8).
1634	LimitAmtCurrency	New Field	Currency	Same values as Currency(15).. Indicates the currency that the limit is specified in.	Add the new field to: TradeCaptureReport (AE) ExecutionReport(8).
423	PriceType	Add new enumeration values	Int	Used in OTC FX trades to further qualify LastSpotRate(194) Price(44) and LastPx(31). 20 - (tbc) = Normal -- represents the number of units of currency 2 that is required to purchase one unit of currency 1. 21 - (tbc) = Inverse -- represents the number of units of 1 that is needed to purchase one unit of currency 2.	Add new enumeration values to the PriceType field.

Tag	Field Name	Action	Data type	Description	Applicable Messages
1164	SettlObligSource	Add new enums	Int	4 - (tbc) – Buyer -- Represent the Buyers settlement instructions. 5 - (tbc) -- Seller – Represents the Sellers settlement instructions.	Add new enumeration values to SettleInstrSource field.
784	SettlePartyRole	Add new enum	Int	82 - (tbc) – 'CLS Member Bank'	Add new enumeration value to SettlePartyRole field.

Table 4 : Tag Defined as part of FIX v5.0 SP2 EP100

Appendix C Message types and Tags Defined as post EP100.

1. Message Type in EP171

There are two new messages introduced into the v5.0 SP2 FIX Standard and has recently approved by the FPL.Org to EP 171 for Thomson Reuters. Details of these message types can be found in this document – see link in the table below:

Message type	Link to Message tables
PartyActionRequest (35=DH)	Party Action Request Message (MsgType = DH) – Matching Counterparty Inhibit Request PartyActionRequest (MsgType =DH) - Matching Enable/Disable/Inhibit PBC Request message
PartyActionReport (35=DI)	PartyActionReport Message (MsgType =DI) PartyActionReport (MsgType =DI) - Matching Enable/Disable/Inhibit PBC Respond.

2. Data Dictionary

Thomson Reuters have requested that ManualOrderIndicator (1028) is added as part of the NewListOrder (35=E) message. That decision is still pending.

3. Custom message type

There are three new custom messages introduced into Matching FIX gateway to support PBC session control functionality. These messages are not officially approved by FPL.Org.

However client must logon to FIX EP171 to be able to receive U3 notification

Details of these message types can be found in this document - see link in the table below:

Message type	Link to Message tables
U1 (35=U1)	U1 (MsgType =U1) - Matching MOS/MOOPI Request message
U2 (35=U2)	U2 (MsgType =U2) - Matching MOS/MOOPI Respond message
U3(35=U3)	U3 (MsgType =U3) Matching PBC session control Broadcast message

4. User Defined Fields

The following tags are not defined in any service pack or extension pack.

They use the tag numbers in the range 5000 – 9999 which the FIX protocol reserves for inter-firm communication or the range 10000+ reserved for internal use.

Tag Name	MsgType(35)	Data Type

Tag Name	MsgType(35)	Data Type
LockedStatus(5007)	NewOrderSingle (35=D) NewOrderList (35=E)	Boolean
OrdersLockFilter(20020)	OrderMassActionRequest (35=CA) OrderMassActionReport (35=BZ)	Int, enumerated: 1 = Non-locked only [default] 2 = Locked Only 3 = Both locked and non-locked

5. User Defined Values

The following tag values are not defined in any service pack or extension pack.

They use values in the 1000+ range which the FIX protocol reserves for bi-laterally agreed user defined values

Tag Name	User Defined Enumerated Value
TargetStrategy(847)	1001 = Matching Iceberg

Appendix D Enumerated Tags and Values Supported within the Matching FIX Interface

The tags listed in the table below contain enumerated values which are defined in FIX version 5.0 SP2 EP100. The Matching FIX Interface only uses a subset of the full list of enumerated values defined within the FIX 5.0 SP2 EP100 schema These have been listed below and cross referenced to the FIX message types that are supported by the Matching FIX Interface and therefore sent and from the Matching FIX Gateway.

Tag	Tag Name	Used in	Enumerated values used by Matching Interface	FIX Message Type
39	OrdStatus		0 = New 1 = Partially Filled 2 = Filled 4 = Cancelled 6 = Cancel Pending 7 = Stopped 8 = Rejected 9 = Suspended C = Expired	ExecutionReport [type '8'] OrderCancelReject [type '9'] ListStatus [type 'N']
40	OrdType		2 = Limit	ExecutionReport [type '8'] NewOrderSingle [type 'D'] NewOrderList [type 'E']
54	Side		1 = Buy 2 = Sell	ExecutionReport [type '8'] TradeCaptureReport [type 'AE'] OrderMassActionRequest [type 'CA'] NewOrderSingle [type 'D'] NewOrderList [type 'E'] OrderCancelRequest [type 'F']

Tag	Tag Name	Used in	Enumerated values used by Matching Interface	FIX Message Type
59	TimeInForce	■ ■ ■ ■ ■ ■ ■ ■	0 = Day order 3 = Immediate or Cancel 6 = Good Till Date A = Good For Time	■ ExecutionReport [type '8'] ■ NewOrderSingle [type 'D']
98	EncryptMethod	■	0 = None / Other	■ Logon [type 'A']
102	CxlRejReason	■ ■	0 = Too Late to cancel 99 = Other	■ OrderCancelReject [type '9']
103	OrderRejReason	■ ■ ■ ■	1(Unknown Symbol) 11(Unsupported Order Characteristic) 13(Incorrect quantity) 99(Other)	■ ListStatus [type 'N']
150	ExecType	■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■	0 = New 4 = Cancelled 6 = Cancel Pending 8 = Rejected 9 = Suspended (held) C = Expired D = Restated F = Trade I = Order Status L = Released	■ ExecutionReport [type '8'] ■ TradeCaptureReport [type 'AE']
378	ExecRestatementReason	■ ■ ■ ■ ■	5 = Partial decline of order quantity 6 = Cancel on Trading Halt 12 = Cancel on Connection Loss 13 = Cancel on Logout 99 = other	■ ExecutionReport [type '8']

Tag	Tag Name	Used in	Enumerated values used by Matching Interface	FIX Message Type
380	BusinessRejectReason		0 = Other 3 = Unsupported message type 4 = Application Not Available 5 = Conditionally required tag missing 6 = Not Authorised 7 = Deliver to Firm not available at this time	BusinessMessageReject [type 'j']
394	BidType		1 = "Non Disclosed" style (e.g. US/European)	NewOrderList [type 'E']
423	PriceType		20 = Normal 21 = Inverse	ExecutionReport [type '8'] TradeCaptureReport [type 'AE']
429	ListStatusType		1 = (Ack) 2 = (Response)	ListStatus [type 'N']
431	ListOrderStatus		4 = (Cancelling) 6 = (All Done) 7 = (Reject)	ListStatus [type 'N']
434	CxlRejResponseTo		1 = (Order Cancel Request)	OrderCancelReject [type '9']
447	PartyIDSource		C = Generally accepted market participant identifier (e.g. NASD mnemonic)	ExecutionReport [type '8'] NewOrderSingle [type 'D'] NewOrderList [type 'E']
452	PartyRole		13 = Owner origination firm 53 = Trader mnemonic 56 = Acceptable Counterparty	ExecutionReport [type '8'] NewOrderSingle [type 'D'] NewOrderList [type 'E']

Tag	Tag Name	Used in	Enumerated values used by Matching Interface	FIX Message Type
487	TradeReportTransType		0 = New	TradeCaptureReport [type 'AE']
552	NoSides		1 = One Side	TradeCaptureReport [type 'AE']

Tag	Tag Name	Used in	Enumerated values used by Matching Interface	FIX Message Type
573	MatchStatus		0 = Compared, matched or affirmed 1 = Uncompared, unmatched, or unaffirmed	TradeCaptureReport [type 'AE']
783	SettlPartyIDSource		C = Generally accepted market participant identifier <e.g. NASD mnemonic>	TradeCaptureReport [type 'AE']
784	SettlPartyRole		27 = Buyer/Seller 86 = Acceptable Settling Counterparty	TradeCaptureReport [type 'AE']
786	SettlPartySubIDType		16 = BIC	TradeCaptureReport [type 'AE']
803	PartySubIDType		2 = Person 25 = Desk	ExecutionReport [type '8']
828	TrdType		54 = OTC	TradeCaptureReport [type 'AE']
854	QtyType		1 = Contracts	ExecutionReport [type '8'] TradeCaptureReport [type 'AE'] NewOrderSingle [type 'D'] NewOrderList [type 'E']
856	TradeReportType		2 = Accept	TradeCaptureReport [type 'AE']
924	UserRequestType		1 = Logon 2 = Log Off 3 = Change Password	UserRequest [type 'BE']

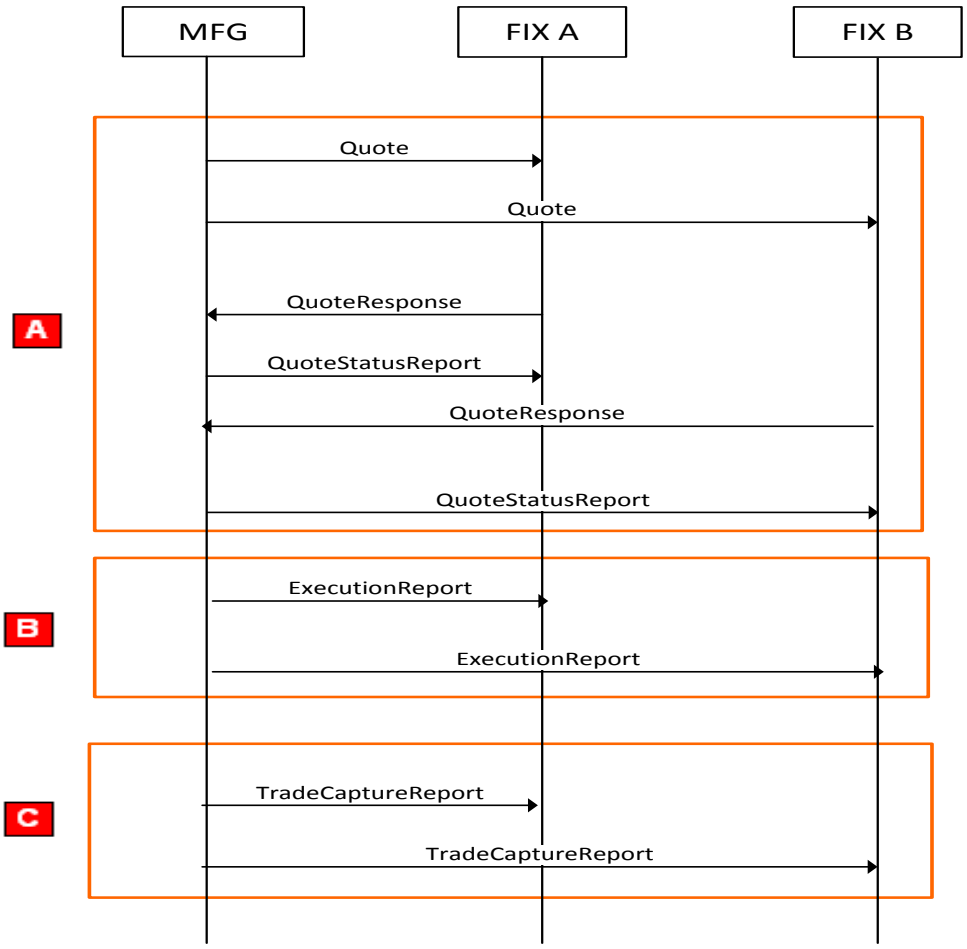
Tag	Tag Name	Used in	Enumerated values used by Matching Interface	FIX Message Type
926	UserStatus		1 = Logged In 2 = Not Logged In 3 = User Not Recognised 4 = Password Incorrect 5 = Password Changed 6 = Other 7 = Forced user logout by exchange	UserResponse [type 'BF']
1100	TriggerType		5 = On Order Entry or Modification	NewOrderSingle [type 'D'] NewOrderList [type 'E']
1101	TriggerAction		3 = Cancel	NewOrderSingle [type 'D'] NewOrderList [type 'E']
1118	RootPartyIDSource		C = Generally accepted market participant identifier <e.g. NASD mnemonic>	TradeCaptureReport [type 'AE']
1119	RootPartyRole		13 = Order Origination Firm 56 = Acceptable Counterparty	TradeCaptureReport [type 'AE']
1122	RootPartySubIDType		2 = Person 25 = Location Desk	TradeCaptureReport [type 'AE']
1128	ApplVerID		9 = FIX50SP2	ALL Message Types
1137	DefaultApplVerId		9 = FIX50SP2	Logon [type 'A']
1164	SettlObligSource		4 (Buyers settlement instructions) 5 (Sellers settlement instructions)	TradeCaptureReport [type 'AE']
1373	MassActionType		1 = Suspend/Hold 2 = Release 3 = Cancel	OrderMassActionReport [type 'BZ'] OrderMassActionRequest [type 'CA']

Tag	Tag Name	Used in	Enumerated values used by Matching Interface	FIX Message Type
1374	MassActionScope		1 = All orders for a security 5 = All orders for a SecurityType 7 = All orders	OrderMassActionReport [type 'BZ'] OrderMassActionRequest [type 'CA']
1375	MassActionResponse		1 = Accepted 0 = Rejected	OrderMassActionReport [type 'BZ']
1376	MassActionRejectReason		0 = Mass Action Not Supported 99 = Other	OrderMassActionReport [type 'BZ']

Tag	Tag Name	Used in	Enumerated values used by Matching Interface	FIX Message Type
1385	ContingencyType		5 = Bid and Offer 6 = Bid and Offer OCO	NewOrderList [type 'E'] ListStatus [type 'N']
1386	ListRejectReason		5 = Unknown Order 11 = Unsupported Order Characteristic 99 = Other	ListStatus [type 'N']
1409	SessionStatus		0 = Session active	Logon [type 'A']
1628	TriggerScope		4 = Jobbing Mode (All other orders for the given security and side) 5 = AutomaticOrderReplacement (All other orders for the given security, price and side)	NewOrderSingle [type 'D'] NewOrderList [type 'E']
1631	LimitAmtType		0 = Credit limit	ExecutionReport [type '8'] TradeCaptureReport [type 'AE']

Appendix E Forward Swap Proposal Negotiation Via FIX Session

1. Initial Proposal Request Matched



The negotiation shall be able to complete at the initial negotiation phase if the near and far quantities proposed by party 1 are the same as those proposed by party2. The supported flow in MFG is as follows:

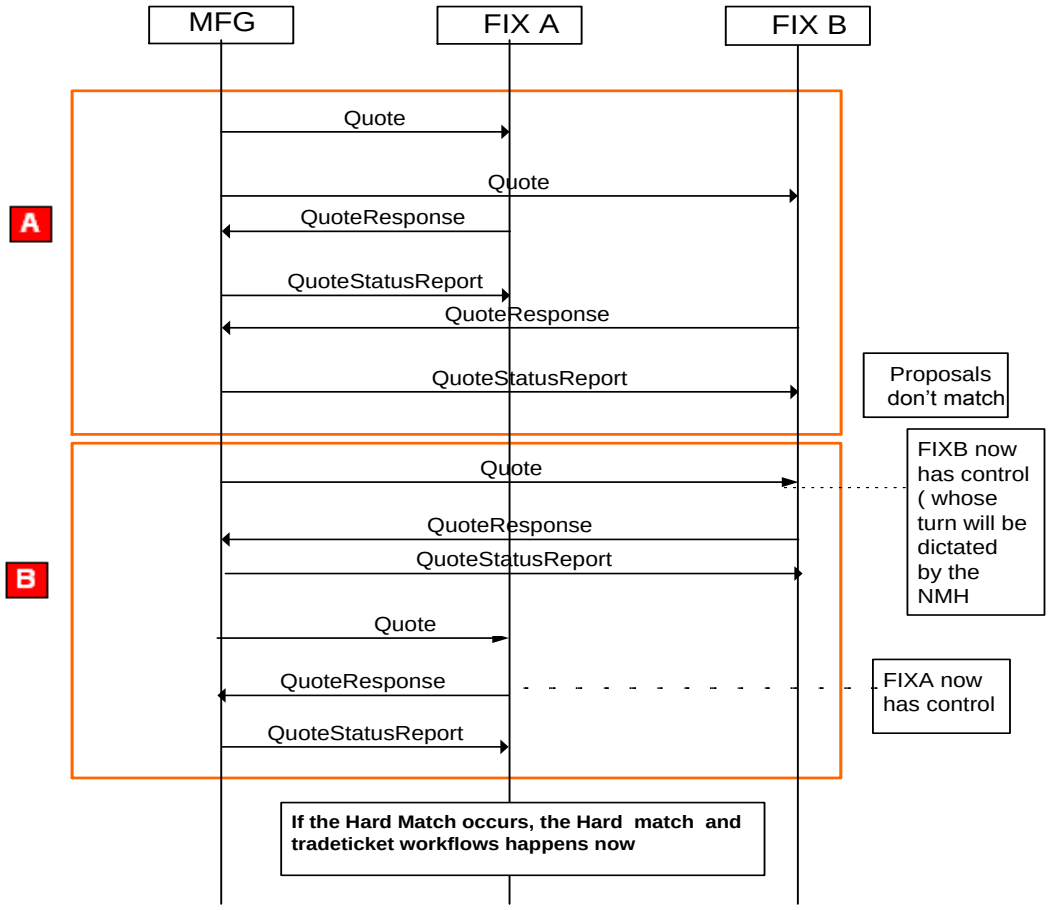
Message Flow	Description
A. Initial negotiation phase	On receipt of the Quote (35=S) message, each FIX negotiation user may submit (using the FIX Quote Response (35=AJ) message) the default quantity/far quantity to the host as a proposal or may amend quantity and final quantity up or down before re-submitting. Both counterparties must submit their proposal to the host before the host progresses to the next stage in the negotiation.
B.Hard Match	Once both proposals have been received, if both proposals match, MFG generate the Hard Match execution report (8) to both parties. If an order is agreed in a smaller quantity than the original soft matched quantity, client may reposition the remaining quantity (see workflow - Partial Match, Inhibit, and Reposition Qty.)

Message Flow	Description
C.Ticket	On receipt of the Match acknowledgement from the clients (within 60 seconds after match occurs), TradeCaptureReport will be generated to both parties.

he value of **'InhibitRequired'** and **'RepositionQuantityRequired'** shall be included in the ExecutionReport (8) hard match report.

FIX field	Value
NoStipulations	• Number of stipulation entries.
=>StipulationType	Supported value • Text=Free Form text
=>StipulationValue	The value of *'InhibitRequired' and 'RepositionQuantityRequired'.* Supported values: • Inhibit workflow required • Reposition workflow required • None

2. No Initial Match, FIX B Control first (Half –duplex)

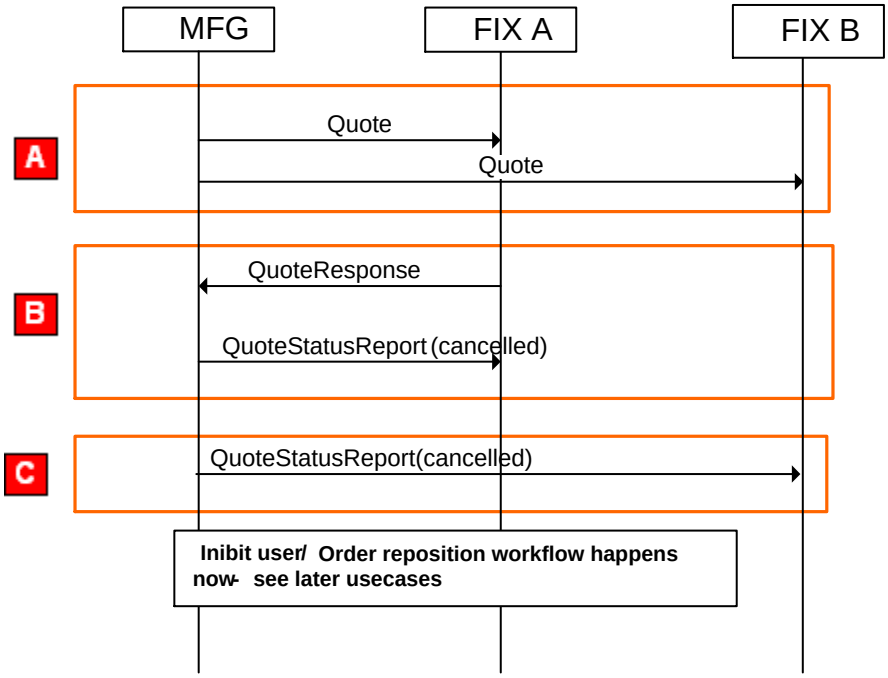


If the near and far quantities of the initial proposal do not match, the negotiation will enter a 'half-duplex' phase in which one Counterparty at a time agrees the other's proposal or submits a new proposal. The supported flow in MFG is as follows:

Message Flow	Description
A. Initial negotiation phase	Pre Half duplex phase see A. Initial negotiation phase
B. Half Duplex negotiation phase	The proposals do not match, the negotiation will enter a half duplex phase in which only one counterparty may modify the proposal at any time. The initial respondee in the half duplex phrase will be determined by the New Matching Host. The MFG send Quote (35=5) Message to the party indicating it is their turn to send the proposal (QuoteResponse 35=AJ) message. Once a counterparty has submitted a new proposal, they will be considered to be in 'waiting' state and the host will reject further proposals from them until a further proposal has been received from the other counterparty i.e. If the party sends the QuoteResponse (35=AJ) message before receiving *Quote(35=S) message, the request will be rejected.* During the half-duplex phase of negotiation, counterparties alternately send QuoteResponse (35=AJ) message to the Matching Gateway (but up to 50 messages in the course of any single negotiation) The negotiation continues until both sides agree on near and far quantities last entered by the other side after which the negotiations are complete.

When the Half Duplex negotiation phase is complete, the Execution report (hard match) and Trade Capture report will be generated to both parties.

3. Reject Quote (No Deal - initiated by user)



Either counterparty can *No Deal* a negotiation at any point up until the negotiation is agreed. The supported flow in MFG is as follows:

Message Flow	Description
A. Initial/Half Duplex negotiation phase	See Initial negotiation phase
B. NoDeal Response	A client can NoDeal during Initial or Half Duplex negotiation phase by submitting QuoteResponse (AJ) to MFG but before match occurs. Party A submits a Nodeal request using QuoteResponse (35=AJ) Message.The request is successful, the QuoteStatusReport (AI) is sent to Party A.
C.No Deal Notification	QuoteStatusReport (AI) will be generated to the counterparty.

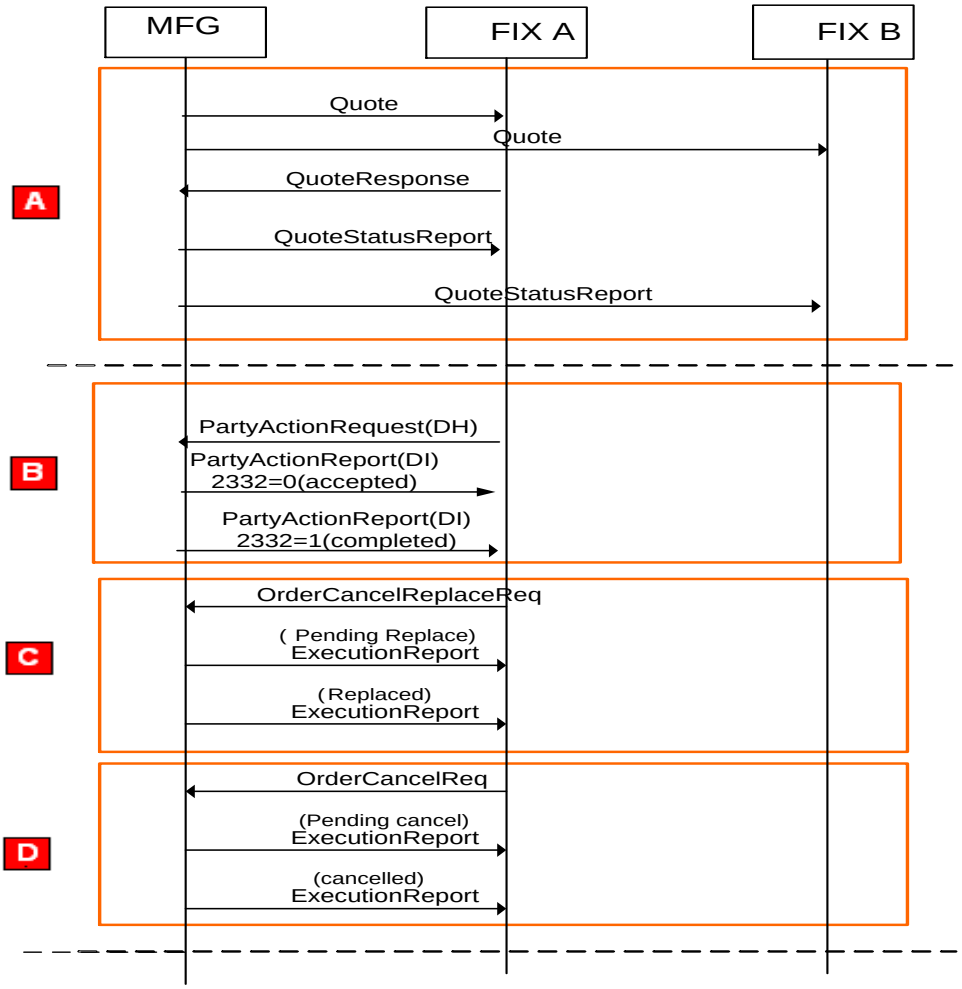
The following data shall be provided by clients in the QuoteResponse message (35=AJ) if they wish to no deal.

FIX field	Value
QuoteRespType (694)	6(Pass)
Text(58)	No Deal reasons provided by the client.

The corresponding *QuoteStatusReport* (AI) message shall provide an appropriate result and a specific self-explanatory reason code.

FIX field	Value
Quote Status(237)	17(Cancelled)
Text(58)	No Deal reason. The value will be taken from the Quote Response (AJ) message. Apply only for QuoteStatusReport as a response to C.No Deal Notification flow.
NoStipulations	Number of stipulation entries.
=>StipulationType	Supported value • Text=Free Form text
=>StipulationValue	The value of *'InhibitRequired' and 'RepositionQuantityRequired'.* Supported values: • Inhibit workflow required • Reposition workflow required • None

4. No Deal, Inhibit Counterparty, Reposition



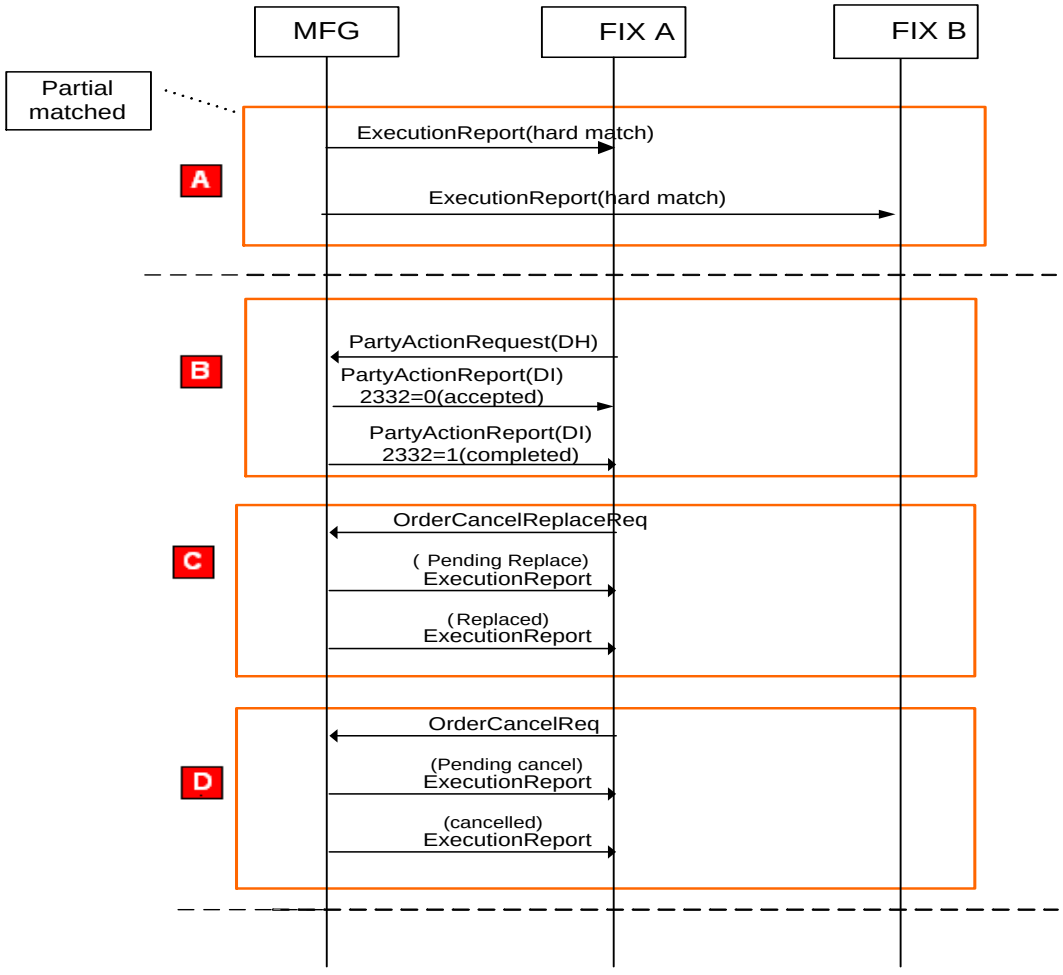
MAPI user shall be able to inhibit counterparty before they reposition the quantity as part of NoDeal workflow. The supported flow in MFG is as follows:

Message Flow	Description
A. NoDeal Flow	See No Deal workflow above.
B. Inhibit	The *PartyActionRequest* message shall be sent from the client to MFG after client 'No Deal' for following scenarios: **S1:** After client 'No Deal' and want to reposition the quantity but do not wish the repositioned quantity matches with the counterparty. **S2:** After client 'No Deal' and cannot reposition the quantity i.e. the original order is Yours/Mine order, NMH still requires MFG to send the Inhibit Request (InhibitBehaviour= DO_NOT_INHIBIT) in order to prevent Stuck Negotiation on PTC (Post Trade Controller). *In that case, as indicated by the *InhibitRequired* flag on the NoDealNotification2 message received. On receipt a successful inhibit response from the NMH, MFG generate the PartyActionReport message and pass on to the FIX client.
C.Reposition Quantity	The *Order Cancel Replace Request* (G) message shall be sent from the client to MFG After client 'No Deal'('Inhibit cpty') for following scenarios: **S1:** After client 'No Deal' and wish to reposition the quantity. On receipt a successful Reposition quantity response from the NMH, MFG generate the

Message Flow	Description
	Execution Report (8) and pass on to the FIX client.
D.Do-not reposition Quantity	The *Order Cancel Request* (F) message shall be sent from the client to MFG After client 'NO Deal' for following scenarios: **S1:** After client 'No Deal' but does not wish to reposition or the parent order is cancelled, NMH still requires MFG to send the *RepositionQtyRequest* 2(RepositionBehaviour=DO_NOT_REPOSITION) even client do not wish to reposition in order to prevent Stuck Negotiation on PTC (Post Trade Controller). *In that case, as indicated by the *RepositonRequired* flag on the NoDealNotification (35=AI (cancelled)) message received. On receipt a successful Do-not reposition quantity response, MFG generate the Execution Report (8) and pass on to the FIX client.

Once a Negotiation has entered *No Deal* state both counterparties shall be able to reposition the quantity if the order is *Bid/Offer* but not if *Yours/Mine*.

5. Partial Match, Inhibit, Reposition Qty



Client can reposition only for the near leg.

User can only reposition the entire quantity or nothing

MAPI user shall be able to reposition if either the remaining unmatched quantity (available parent order quantity) for a *Bid/Offer* order (not *Yours/Mine*) or the repositioned quantity is equal to or greater than the market minimum quantity for that instrument.

Samples below displayed the reposition behaviour in a market minimum quantity of 10

Sample1

FIX A original order quantity =70 soft match =60

FIX A negotiates the quantity down and hard match with FIX B @ 56

FIX A remaining quantity = 4 and the quantity on the book =10. FIX A can reposition the quantity even the remaining quantity =4 but the quantity on the book is 10 (equal to market minimum quantity). Quantity returned to the book =14.

Sample2

FIX A original order quantity =70 soft match =61

FIX A negotiates the quantity down and hard match with FIX B @ 52

FIX A remaining quantity = 9 and the quantity on the book =9. FIX A cannot reposition the quantity because both the remaining quantity and the quantity on the book is less than Market minimum quantity.

The supported flow in MFG is as follows:

Message Flow	Description
A. Initial/Half Duplex negotiation phase	[NO.INITIAL.MATCH.001.001], [NO.INITIAL.MATCH.001.002]
B. Inhibit	The *PartyActionRequest* message (35=DH) shall be sent from the client to MFG after client 'Partial Match' for following scenarios: **S1:** After Client 'Partial match' (after agreeing a smaller quantity than the original soft matched quantity) and want to reposition the remaining quantity but do not wish the repositioned quantity matches with previous counterparty. **S2:** After Client 'Partial match' and cannot reposition the quantity i.e. the original order is Yours/Mine order, NMH still requires MFG to send the Inhibit Request (InhibitBehaviour= DO_NOT_INHIBIT) in order to prevent Stuck Negotiation on PTC (Post Trade Controller). *In that case, as indicated by the *InhibitRequired* flag on the ProposalMatched3 message received. On receipt a successful inhibit response from the NMH, MFG generate the PartyActionReport message and pass on to the FIX client.
C.Reposition Quantity	The *Order Cancel Replace Request* (G) message shall be sent from the client to MFG After client 'Partial Match, ('Inhibit cpty') for following scenario: **S1:** After Client 'Partial match' and want to reposition the remaining quantity. On receipt a successful Reposition quantity response from the NMH, MFG generate the Execution Report (8) and pass on to the FIX client.
D.Do-not reposition Quantity	The *Order Cancel Request* (F) message shall be sent from the client to MFG After client 'Partial Match, for following scenario: **S1:** After Client 'Partial match' and does not want to reposition or the parent order is cancelled. NMH still requires MFG to send the *RepositionQtyRequest* 2(RepositionBehaviour=DO_NOT_REPOSITION) even client do not wish to reposition in order to prevent Stuck Negotiation on PTC (Post Trade Controller). *In that case, as indicated by the *RepositonRequired* flag on the ExecutionReport hard match (35=8) message received.

Message Flow	Description
	On receipt a successful Do-not reposition quantity response, MFG generate the Execution Report (8) and pass on to the FIX client.

If the user chooses not to reposition the quantity, the order will be automatically cancelled if the remaining quantity is below the minimum allowed for the instrument.

6. Summary of terminating message supported in MAPI for Proposal Negotiation

The workflows require sending of 'do-not Reposition' or 'do-not Inhibit' messages to terminate an FX Swap proposal in order to remove the negotiation from the Post trade control (PTC) are detailed as follows:

scenarios	Proposal State	Order Type	Terminating Message	Recommended messages for client to send
BID/Offer order is submitted and fully filled.	Fully Filled	Bid/Offer	No message required	NMH will automatically terminate an order in this state. Clients do not need to send do-not Reposition' or 'do-not Inhibit'
Yours/Mine order is submitted and fully filled.	Fully Filled	Yours/Mine	No message required	NMH will automatically terminate an order in this state. Clients do not need to send do-not Reposition' or 'do-not Inhibit'
A user does not want to reposition the remaining quantity after Partial filled.	Part Filled	Bid/Offer	Reposition or Do not reposition	FC sends do-not reposition request message.
Yours/Mine order is submitted and partially filled	Part Filled	Yours/Mine	Inhibit or Do not inhibit	FC sends do-not inhibit or inhibit request message. Note Client can send either Inhibit or Do - not inhibit to terminate the proposal (equally valid). It depends on client's decision. If clients don't want their orders to match with previous cpty, they need to send inhibit cpty request (effect till next credit reset). Otherwise, do-not inhibit request.
Bid/offer order is partially filled and the parent order is cancelled.	Part Filled, Parent Order Cancelled	Bid/Offer	Reposition or Do not reposition	FC sends do-not reposition request message. Note. Client can send either Reposition or Do-not reposition to terminate the proposal but the actual reposition will fail as the parent order is cancelled. Client shall send do-not reposition request message in order to prevent the rejection response from the MFG.

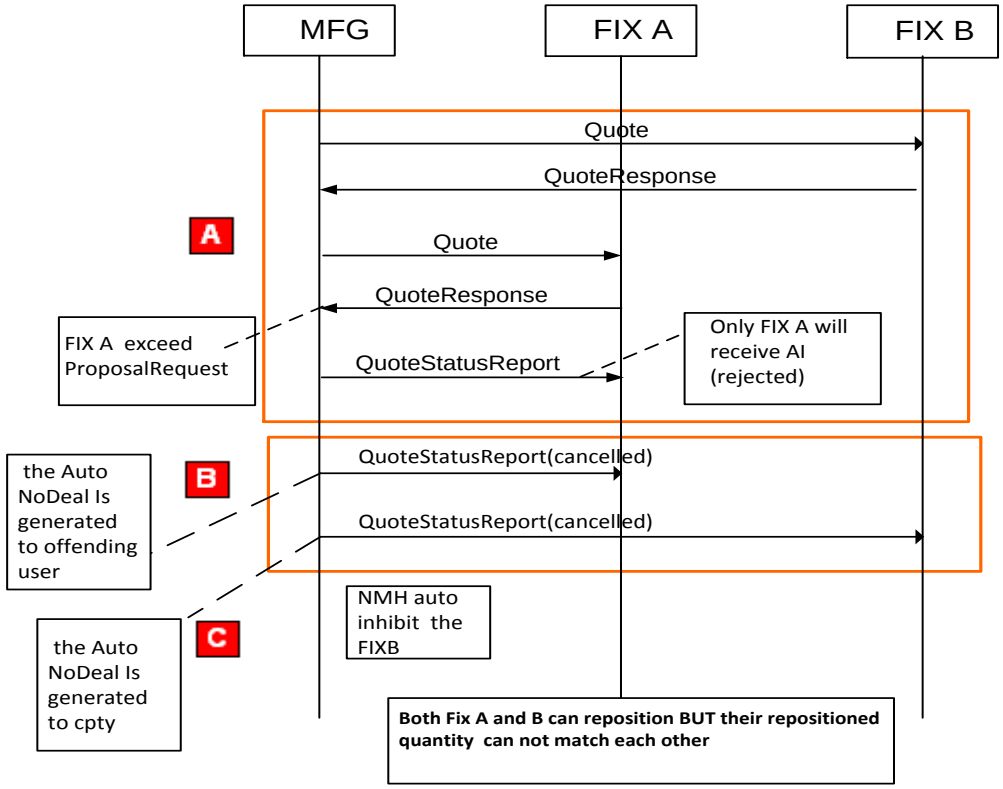
scenarios	Proposal State	Order Type	Terminating Message	Recommended messages for client to send
A user does not want to reposition the remaining quantity after no deal.	No Deal	Bid/Offer	Reposition or Do not reposition	FC sends do-not reposition request message.
A user no deal and parent order is cancelled.	No Deal	Bid/Offer	Reposition or Do not reposition	FC sends do-not reposition request message. Note. Client can send either Reposition or Do-not reposition to terminate the proposal but the actual reposition will fail as the parent order is cancelled. Client shall send do-not reposition request message in order to prevent the rejection response from the MFG.
A users no deal Yours/Mine order.	No Deal	Yours/Mine	Inhibit or Do not inhibit	FC sends do-not inhibit or inhibit request message. Note Client can send either Inhibit or Do - not inhibit to terminate the proposal (equally valid). It depends on client's decision. If clients don't want their orders to match with previous cpty, they need to send inhibit cpty request (effect till next credit reset). Otherwise, do-not inhibit request.

Proposals which have not been terminated by sending the appropriate *Reposition* or *Inhibit* request will be assumed to be 'dead' and will be terminated automatically after a configurable time interval. Initially, the interval shall be set to 8 *hours*

The time interval will be measured from the point at which the Host sends out Hard Match Notification (for partially Match) and No Deal Response messages.

If the appropriate *Reposition* or *Inhibit* request is submitted after the 'Dead proposal start time' but before the timeout, the workflow shall behave in the same way as today. If they are sent after the timeout, the request will be rejected by the MFG. (more detail see - Matching API error Guide)

7. Max MAPI proposal request exceed (Auto No Deal - initiated by the system)



If a MAPI user submits a proposal in any single negotiation (including any in the pre-half-duplex phase) that exceed the Maximum number of QuoteResponse messages.

The value of MAX_MAPI_PROPOSAL_REQUESTS is a configuration parameter of the system, stored in the CONFIGURATION table of the Host DB. The initial value of MAX_MAPI_PROPOSAL_REQUESTS is 50.

Message Flow	Description
A. Half Duplex negotiation phase, Max Proposal Request exceed	Half Duplex negotiation phase is as described by [NO.INITIAL.MATCH.001.002]. Then, the retry limit was exceeded by FIX A. **Only** FIX A (a party who exceed the max Proposal request) will receive the QuoteStatusReport(rejected)
B. System generated No Deal Notification	When the retry limit was exceeded, MFG generate QuoteStatus Report (cancelled) and pass on to the offending user and the counterparty.
C.Auto Inhibit the CP of the offending user	The NMH has auto inhibited FIX B (the counterparty of the offending user) in order to prevent the 2 counterparties immediately re-entering a similar loop if they both reposition the quantity.

The *QuoteStatusReport* (rejected) due to Maximum number of Proposal Request messages exceed will be as follows.

FIX field	Value
Quote Status(297)	5(rejected)
QuoteRejectReason(300)	99(others)
Text(58)	“Max MAPI proposal requests” (no ‘Sys’ prefix. Sys’ prefix will be populated only when Quote Status(297)=17)

The *QuoteStatusReport (cancelled)* as an auto nodeal notification due to Maximum *number* of Proposal Request messages exceed will be as follows:

FIX field	Value
Quote Status(297)	17(Cancelled)
Text(58)	“Sys:MAX_NUM_PROPOSALS_EXCEEDED”

8. Parent order is cancelled mid negotiation workflow

During the negotiation flow, the parent order can be cancelled as the result of the following scenarios:

- Client sends a Mass action cancel
- Client sends a Single Order cancel
- Another order is placed in Jobbing mode or AOR(the second order has cancelled the first)
- A Timed order expires

Note: If the parent is cancelled as the result of user logout, user disconnects/comms failures or NMH move to non-trading mode, once user logout they cannot resume negotiations.- see Abnormal workflow

The negotiation is allowed to complete when the parent order is cancelled.

Reposition is not allowed after No deal or partially match flow when the parent order is cancelled.

Cancelling the parent order where the leave Quantity of softmatch equal to zero is allowed if it is an FX Swap Order.

9. Parent order is held in the middle of the negotiation workflow

The negotiation is allowed to complete when the parent order is held.

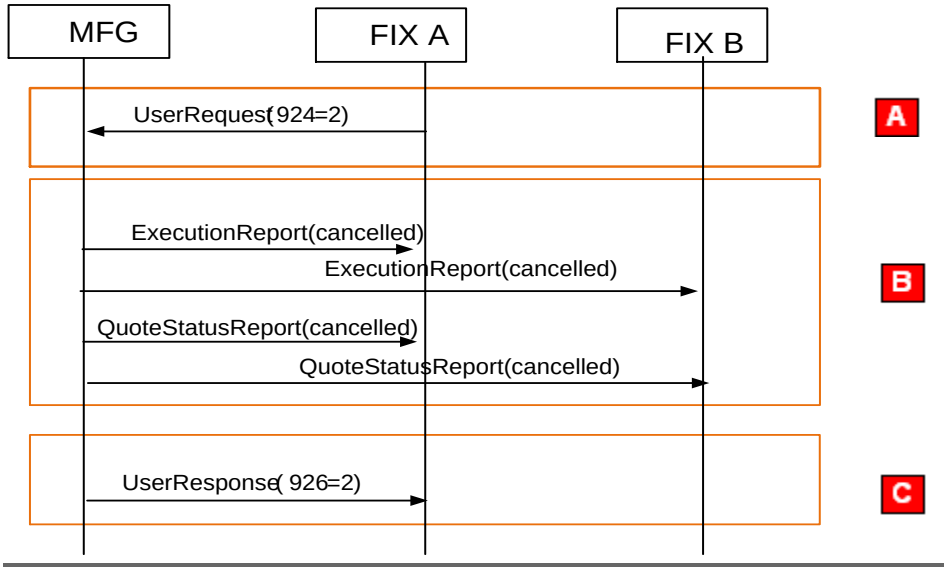
Reposition is allowed after No deal or partially match flow when the parent is held. (See Note below)

Note: The hold will not be released as a result of replacing quantity, but the replaced quantity will become visible to the market when the order is released

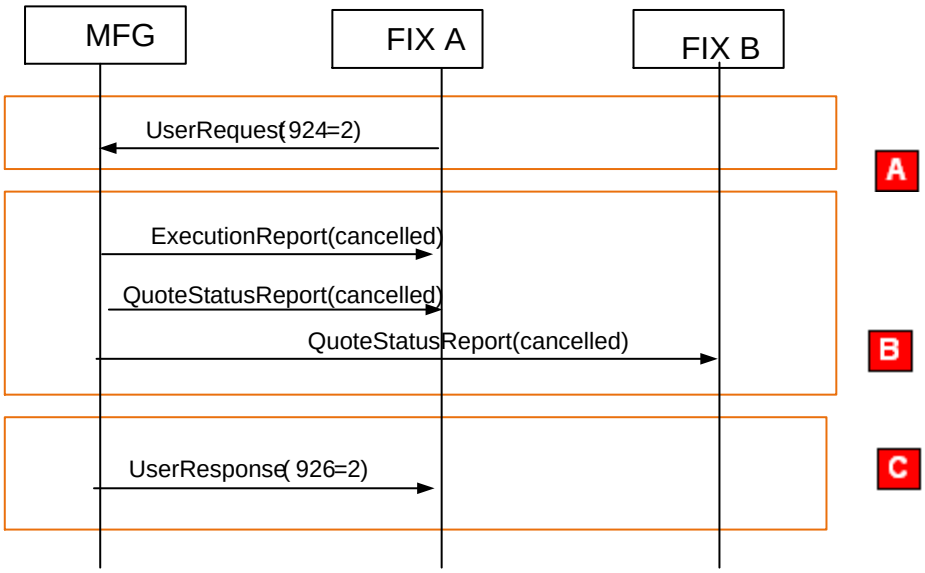
10. Abnormal workflow

User logs out while the negotiation is in progress

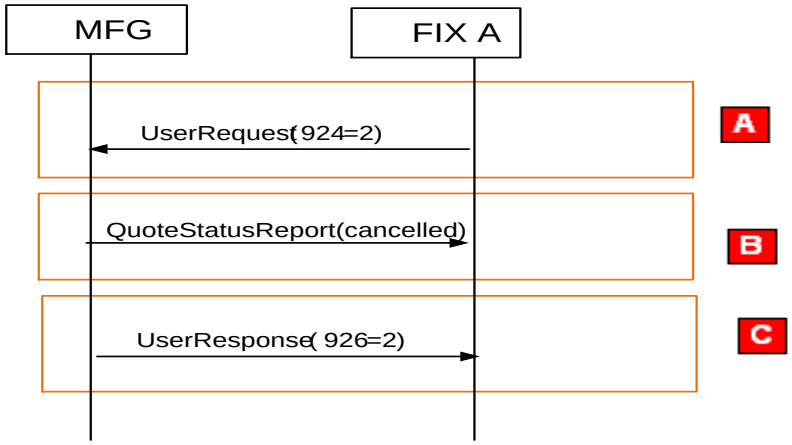
S1 FIX A logs out during negotiation and FIX B do not have remaining quantity that can be negotiated on the parent order.



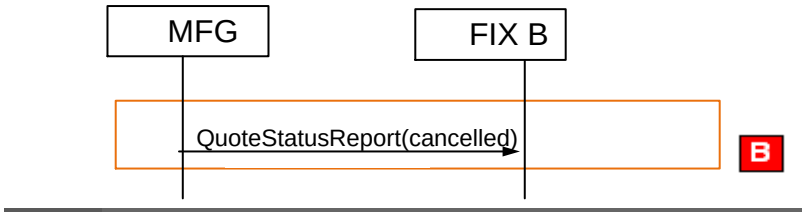
S2 FIX A logs out during negotiation and FIX B has remaining quantity that can be negotiated on the parent order.



S3 FIX A logs out during negotiation FIX A submits Yours/Mine order



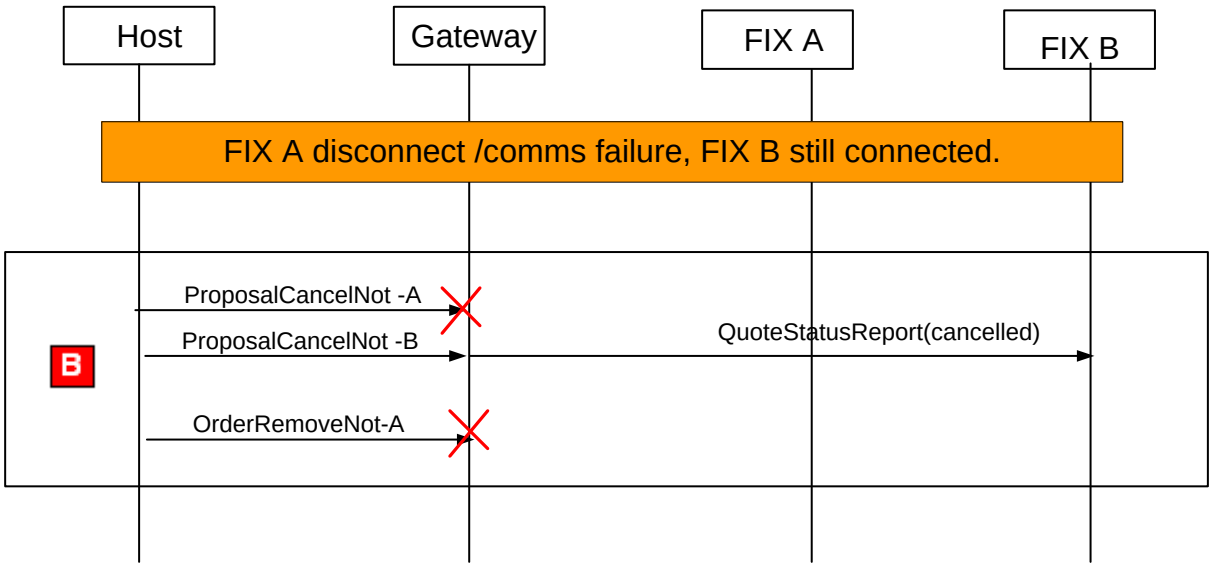
S4 FIX A logs out during negotiation FIX B submits Yours/Mine order



The following workflow is supported in MFG If a client logs out from the system while the negotiation is in progress,

Message Flow	Description
A. Logout Request	FIX A sends FIX User Request (BE) with User Request Type=2(log off user) to MFG.
B. Proposal cancel notification and Order Remove notification	**For S1** MFG generate the Execution report (cancelled) to both parties. MFG generate the *QuoteStatusReport* (cancelled) to indicate that they cannot resume negotiations (a proposal has been cancelled) to both parties. **Note –** MFG always generate a Cancelled ER following OrderRemovedNotification even when OrderRemovedReason=NO_OUTSTANDING_PROPOSAL. **For S2** MFG generate the Execution report (cancelled) to FIX A. Only FIX A will receive ER (cancelled) as the parent order of FIX B has not been cancelled. MFG generate the *QuoteStatusReport* (cancelled) to indicate that they cannot resume negotiations to both FIX A and FIX B. **ForS3/S4** MFG generate the *QuoteStatusReport* (cancelled) to indicate that they cannot resume negotiations *)* to the user who submitted the Yours/Mine order.(no matter what they logged out or not) For their counterparty – see S1 or S2
C.Logout Response	MFG response an successful log out in form of the UserResponse (BF) with UserStatus (926) =2 (not logged in) to the logged out user.(FIX A)

User disconnects/comms failure while the negotiation is in progress



The following workflow is supported in MFG If a client disconnects from the system or there is a comms failure while the negotiation is in progress,

Message Flow	Description
A. Proposal cancel notification and Order Remove notification	The NMH generate the PCN *and OrderRemoveNotifications to* the disconnected user (FIX A) but they will be discarded by the router because it can no longer forward them. The counterparty (assuming they are still connected) will receive same messages as described in [USER.LOG.OUT.001.002]

NMH change to Non-Trading mode while the negotiation is in progress

The NMH can change to non-trading mode for various reasons; the current NMH supported work flow is as follows:

- If it's not due to a component failure) then both parties shall receive both PCN *and OrderRemoveNotifications* messages.
- If it's due to the Matching Engine failing then the users will first receive a PCN from the Post Trade Controller followed later by an *OrderRemoveNotifications* when the Matching Engine restarts.
- If it's due to the Post Trade Controller failing then the users will first receive an *OrderRemoveNotifications* from the Matching followed by a PCN from the PTC when it restarts.

Error text – abnormal workflow

The QuoteStatusReport (AI) message following the *ProposalCancelNotification* shall include the data below the Text (58) field.

Scenario	Text(58)
Comms failure	"Sys:Comms fail"
Forced user logoff by the	"Sys:Forced logoff"

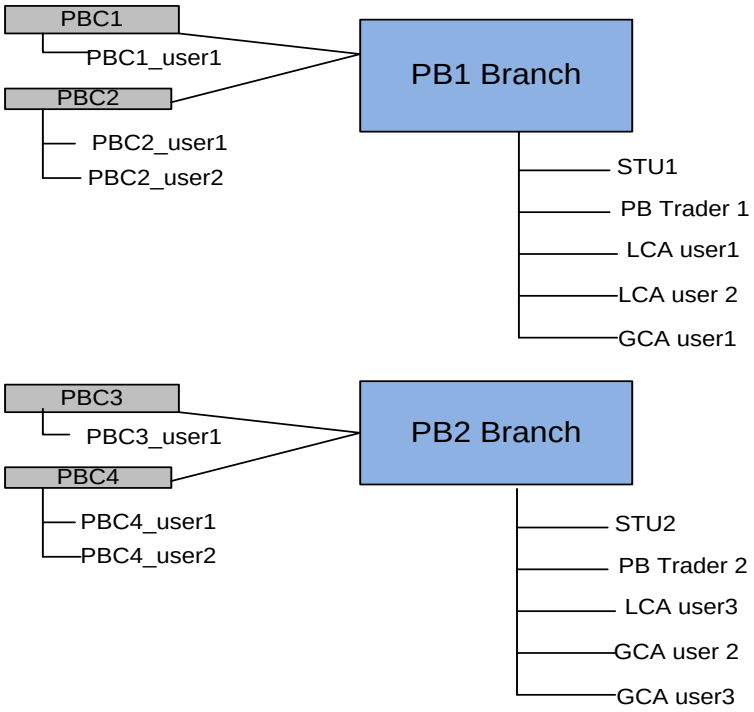
Scenario	Text(58)
system	
Forced shutdown by the system	"Sys:Forced shutdown"
NMH changes to Non- trading mode	"Sys:Host entered non-trading mode"
User log out from the system	"Sys:User logoff"

OrderRemovedNotifications with regards to negotiations will be sent from the NMH as the result of the following scenarios:

- If the negotiations agree for less than the match quantity (negotiated down) and either a party send (DO_NOT) Reposition or the party is not capable of sending Reposition (is transfer user) and if the whole order quantity matched.
- The negotiations agree for the match quantity (or more if upping the quantity is permitted) and the whole order quantity matched
- The negotiation ends in a No-Deal and either a party sends (DO_NOT) Reposition and if the whole order quantity matched
- The party is not capable of sending Reposition (is transfer user) and if the whole order quantity matched
- if only part of the original order matched then even if the parties agree to negotiate for the full match quantity, the non-matched part of the original order will remain on the books (subject to minimum market quantity / lot size) in which case an OrderRemovedNotification will not be sent for that order.

Appendix F U3 notification Use case

1. Global Credit Organisation (GCO) Adminstration



Within the GCO structure, only the Global credit administrator (GCA) can change the PBC state of Enable/Inhibit/disable, Max Open Orders per Instrument, Max Order Size i.e. Local credit administrator (LCA) is a view only user.

Login Use case

[GCO.LOGIN.U3.001] – GCA login

1. **GCA user1** successfully login
2. The U3 notification(s) containing the following data will be sent to GCA user1:
   - A Domcad of PB1 branch
   - PBC state of Enable/Inhibit/disable, Max Open Orders per Instrument, Max Order Size of all PBCs within the GCO (in this case are PBC1, PBC2, PBC3 and PBC4).

Note. See ***sample message1.*** The U3 notification shall not include the Credit admin state.

Sample message1

<Header>		
Domcad	20006	Y (PB CA MFG connection Required) – in this case it is DOMCAD for PB Branch1
PartyStatusRptID	20013	Abc1234
RptSeq	83	1

<Header>		
NoRpts	82	1
LastFragment	893	Y
NoPBCs	20009	4
<Parties>repeating group		
1st PBC Block – PBC1		
->PBCID	20012	PBC1* PBC1 bank
->NoPartyIDs	453	1
->->PartyId	448	PBC1 bank
->->PartyIDSource	447	C – Generally Accepted Market Participant
->->PartyRole	452	3– Client ID
->->NoPartySubIDs	802	1
->->->PartySubID	523	PBC1
->->->PartySubIDType	803	25 - Location Desk
->BranchStatus	20010	1(Enable PBC Branch)
-> OrderQty	38	100 (MOS=100 mio USD)
-> MaxOpenOrderPerInstrument	20003	8
2nd PBC block –PBC2		
->PBCID	20012	PBC2* PBC2 bank
->NoPartyIDs	453	1
->->PartyId	448	PBC2 bank
->->PartyIDSource	447	C – Generally Accepted Market Participant
->->PartyRole	452	3– Client ID
->->NoPartySubIDs	802	1
->->->PartySubID	523	PBC2
->->->PartySubIDType	803	25 - Location Desk
->BranchStatus	20010	1(Enable PBC Branch)
-> OrderQty	38	110
-> MaxOpenOrderPerInstrument	20003	14
3rd PBC block –PBC3		
->PBCID	20012	PBC3* PBC3 bank
->NoPartyIDs	453	1
->->PartyId	448	PBC**3** bank
->->PartyIDSource	447	C – Generally Accepted Market Participant
->->PartyRole	452	3– Client ID
->->NoPartySubIDs	802	1
->->->PartySubID	523	PBC**3**

<Header>		
->->->PartySubIDType	803	25 - Location Desk
->BranchStatus	20010	1(Enable PBC Branch)
-> OrderQty	38	50
-> MaxOpenOrderPerInstrument	20003	15
4th PBC block –PBC4		
->PBCID	20012	PBC4* PBC4 bank
->NoPartyIDs	453	1
->->PartyId	448	PBC**4** bank
->->PartyIDSource	447	C – Generally Accepted Market Participant
->->PartyRole	452	3– Client ID
->->NoPartySubIDs	802	1
->->->PartySubID	523	PBC**4**
->->->PartySubIDType	803	25 - Location Desk
->BranchStatus	20010	1(Enable PBC Branch)
-> OrderQty	38	80
-> MaxOpenOrderPerInstrument	20003	4
<Trailer >		

[GCO.LOGIN.U3.002] – LCA login

1. **LCA user3** successfully login
2. A same U3 notification which is sent to GCA user in [GCO.LOGIN.U3.001] use case (see sample message1) **except for** Domcad value (in this case will be DOMCAD for PB Branch2) will be sent to LCA user3.

[GCO.LOGIN.U3.003] – PBC login

1. **PBC4_user2** successfully login
2. The U3 notification(s) containing the following data will be sent to PBC4_user2:
   - PBC state of Enable/Inhibit/disable, Max Open Orders per Instrument, Max Order Size of his PBC branch (in this case is PBC4)
   - A Domcad of PB2 branch
   - Credit admin state of PB2 branch

   Note. See ***sample message2.***

Sample message2 –PBC login

<Header>		
Domcad	20006	Y (PB CA MFG connection Required) – in this case it is DOMCAD for PB Branch2
PartyStatusRptID	20013	BBB345
RptSeq	83	1
NoRpts	82	1
LastFragment	893	Y

<Header>		
NoPBCs	20009	1
<Parties>repeating group		
1st PBC block –PBC4		
->PBCID	20012	PBC4* PBC4 bank
->NoPartyIDs	453	1
->->PartyId	448	PBC4 bank
->->PartyIDSource	447	C – Generally Accepted Market Participant
->->PartyRole	452	3– Client ID
->->NoPartySubIDs	802	1
->->->PartySubID	523	PBC4
->->->PartySubIDType	803	25 - Location Desk
->BranchStatus	20010	1(Enable PBC Branch)
->CreditAdminStatus	20011	2(PB CA Connected) - Credit admin state of PB2 branch
-> OrderQty	38	80
-> MaxOpenOrderPerInstrument	20003	4
<Trailer >		

[GCO.LOGIN.U3.004] – PB STU login

Main flow

1. **STU1** successfully login, there will be no U3 notification message sent out from the MFG.

[GCO.LOGIN.U3.005] – PB trader login

Main flow

1. **PB trader 2** successfully login, there will be no U3 notification message sent out from the MFG.
2. PBC control updates Use case

[GCO.UPDATES.U3.001] – update MOS/MOOPI

Main flow

1. **GCA user1** change the MOS value of PBC2 from 110 million to 80millions AND change branch status of PBC3 from enable to disable.
2. There will be **2** updates within the U3 message(s)
   a) First update containing New MOS value of PBC2 will be sent to all logged in GCA user(s), all logged in LCA and PBC2 (all logged in users).see sample message3 below.

Sample message 3 – update PBC MOS

<Header>		
~~Domcad~~	~~20006~~	~~Y (PB CA MFG connection Required) – in this case it is DOMCAD for PB Branch1~~ not populated as it is

<Header>		
		unchanged.
PartyStatusRptID	20013	CCC345
RptSeq	83	1
NoRpts	82	1
LastFragment	893	Y
NoPBCs	20009	1
<Parties>repeating group		
1st PBC block –PBC2		
->PBCID	20012	PBC2* PBC2 bank
->NoPartyIDs	453	1
->->PartyId	448	PBC2 bank
->->PartyIDSource	447	C – Generally Accepted Market Participant
->->PartyRole	452	3– Client ID
->->NoPartySubIDs	802	1
->->->PartySubID	523	PBC2
->->->PartySubIDType	803	25 - Location Desk
~~->BranchStatus~~	~~20010~~	~~1(Enable PBC Branch)~~ not populated as it is unchanged.
~~->CreditAdminStatus~~	~~20011~~	~~2(PB CA Connected)~~ not populated as it is unchanged.
-> OrderQty	38	80
~~-> MaxOpenOrderPerInstrument~~	~~20003~~	~~14~~ not populated as it is unchanged.

3. Second update containing the new branch status of PBC3 (from enable to disable) will be sent to all logged in GCA user(s), all logged in LCA and PBC3 (all logged in users). See sample message4 below.

Sample message 4 – update PBC branch status

<Header>		
Domcad	20006	~~Y (PB CA MFG connection Required) – in this case it is DOMCAD for PB Branch1~~ not populated as it is unchanged.
PartyStatusRptID	20013	CCC123
RptSeq	83	1
NoRpts	82	1
LastFragment	893	Y
NoPBCs	20009	1
<Parties>repeating group		
1st PBC block –PBC3		

<Header>		
->PBCID	20012	PBC3* PBC3 bank
->NoPartyIDs	453	1
->->PartyId	448	PBC**3** bank
->->PartyIDSource	447	C – Generally Accepted Market Participant
->->PartyRole	452	3– Client ID
->->NoPartySubIDs	802	1
->->->PartySubID	523	PBC**3**
->->->PartySubIDType	803	25 - Location Desk
->BranchStatus	20010	2(Disable PBC Branch)
~~->CreditAdminStatus~~	~~20011~~	~~2(PB CA Connected)~~ not populated as it is unchanged.
~~-> OrderQty~~	~~38~~	~~**50**~~ not populated as it is unchanged.
~~-> MaxOpenOrderPerInstrument~~	~~20003~~	~~**15**~~ not populated as it is unchanged.
<Trailer >		

Logout Use case

[GCO.LOGOUT.U3.001] – Last GCA within the GCO logout

1. Last LCA within the GCO successfully logout, the 'Credit Admin state' will change false. U3 containing the Credit admin state (false) will be sent to PBCs that belong to the PB which the last GCA belonged to.

[GCO.LOGOUT.U3.002] – Last LCA within the GCO logout

1. Last LCA within the GCO successfully logout, there will be no U3 notification message sent out from the MFG.

[GCO.LOGOUT.U3.003] – Last PB Trader within the GCO logout

1. Last PB Trader within the GCO successfully logout, there will be no U3 notification message sent out from the MFG.

[GCO.LOGOUT.U3.004] – Last STU within the GCO logout

1. Last PB STU within the GCO successfully logout, there will be no U3 notification message sent out from the MFG.

[GCO.LOGOUT.U3.005] – Last PBC within the GCO logout

1. Last PBC within the GCO successfully logout, there will be no U3 notification message sent out from the MFG.

Change Domcad value in the new credit GUI (NCG)

[DOMCAD.CHANGE] - NCG credit admin user change Domcad value

1. A NCG credit admin user changes the Domcad value of PB1 branch from 'Y' to 'N'
2. The U3 notification(s) containing the following data will be sent to all logged in GCAs, LCAs and PBCs within the PB branch1

Sample message5 –update Domcad

<Header>		
Domcad	20006	Y (PB CA MFG connection not Required) – in this case it is DOMCAD for PB Branch1
PartyStatusRptID	20013	Abc1234
RptSeq	83	1
NoRpts	82	1
LastFragment	893	Y
<Trailer >		

MAPI credit administrator (either GCA or LCA) cannot change the Domcad value.

PBC details shall not be included in the U3 message if only Domcad value is changed.

Change Credit admin status

[ADMIN.STATUS.CHANGE] – PB credit admin status is changed

1. A credit admin (GCA) status of PB branch2 is changed from 'PB CA disconnected' to 'PB CA connected'
2. The U3 notification(s) containing the following data will be sent to all logged in PBCs of PB Branch2(in this case are PBC3 and PBC4). See sample message6 below.

Sample message 6 – update PBC credit admin status

<Header>		
Domcad	20006	Not populated as it is unchanged.
PartyStatusRptID	20013	Abc1234
RptSeq	83	1
NoRpts	82	1
LastFragment	893	Y
NoPBCs	20009	2
<Parties>repeating group		
1st PBC block –PBC3		
->PBCID	20012	PBC3* PBC3 bank
->NoPartyIDs	453	1
->->PartyId	448	PBC**3** bank
->->PartyIDSource	447	C – Generally Accepted Market Participant
->->PartyRole	452	3– Client ID
->->NoPartySubIDs	802	1
->->->PartySubID	523	PBC**3**
->->->PartySubIDType	803	25 - Location Desk
->CreditAdminStatus	20011	2(PB CA Connected) Credit admin state of PB2 branch
2nd PBC block –PBC4		

<Header>		
->PBCID	20012	PBC4* PBC4 bank
->NoPartyIDs	453	1
->->PartyId	448	PBC4 bank
->->PartyIDSource	447	C – Generally Accepted Market Participant
->->PartyRole	452	3– Client ID
->->NoPartySubIDs	802	1
->->->PartySubID	523	PBC4
->->->PartySubIDType	803	25 - Location Desk
->CreditAdminStatus	20011	2(PB CA Connected) - Credit admin state of PB2 branch
<Trailer >		

Auto Credit reset

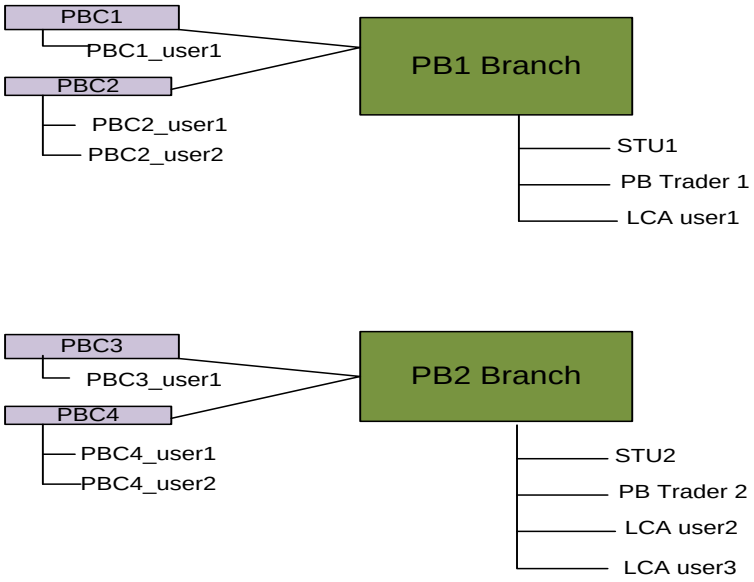
[GCO.CREDIT.RESET] Auto credit reset

1. NMH auto credit reset of PBC 3 (from inhibit to enable)
2. The U3 notification(s) containing the new branch status of PBC3 (from inhibit to enable) will be sent to all logged in GCA user(s), all logged in LCA and PBC3 (all logged in users). See sample message7 below.

Sample message 7 – auto credit reset update

<Header>		
Domcad	20006	Not populated as it is unchanged.
PartyStatusRptID	20013	CCC123
RptSeq	83	1
NoRpts	82	1
LastFragment	893	Y
NoPBCs	20009	1
<Parties>repeating group		
1st PBC block –PBC3		
->PBCID	20012	PBC3* PBC3 bank
->NoPartyIDs	453	1
->->PartyId	448	PBC**3** bank
->->PartyIDSource	447	C – Generally Accepted Market Participant
->->PartyRole	452	3– Client ID
->->NoPartySubIDs	802	1
->->->PartySubID	523	PBC**3**
->->->PartySubIDType	803	25 - Location Desk
->BranchStatus	20010	1(Enable PBC Branch)
<Trailer >		

2. Non-Global Credit admin Organisation



- Global credit administrator (GCA) does not exist in the non GCO structure.
- LCA (Local credit administrator) can change the PBC state of Enable/Inhibit/disable; Max Open Orders per Instrument, Max Order Size but only on their PB Branch.

[NON.GCO.U3.001] the U3 notification shall be supported for Non- Global admin organisation as follows:

scenario	U3 Notification
Scenario1 -LCA user 3 successful login	LCA user3 shall receive U3 notification(s) containing the following data: • A Domcad of PB2 branch • PBC state of Enable/Inhibit/disable, Max Open Orders per Instrument, Max Order Size of all PBCs within PB2 branch. Note. The U3 notification shall not include the Credit admin state.
Scenario2 –PBC4_user2 successful login	The U3 notification(s) containing the following data will be sent to PBC4_user2: • PBC state of Enable/Inhibit/disable, Max Open Orders per Instrument, Max Order Size of his PBC branch (in this case is PBC4) • A Domcad of PB2 branch • Credit admin state of PB2 branch Same as *[GCO.LOGIN.U3.003] – PBC login*
Scenario3 –STU1 successful login	No U3 notification message sent out from the MFG. Same as *[GCO.LOGIN.U3.004] – PB STU login*
Scenario4 –PB Trader 2 successful login	No U3 notification message sent out from the MFG. Same as *[GCO.LOGIN.U3.005] – PB trader login*
Scenario5 – LCA user1 updates MOS of PBC2	U3 notification(s) containing the following data shall be sent to LCA user1 and PBC2 (all logged in users).

scenario	U3 Notification
	• New MOS value of PBC2
Scenario6 – LCA user2 updates BranchStatus of PBC4 (from disable =>enable)	U3 notification(s) containing the following data shall be sent to LCA user2 and LCA user3 (if logged in) and PBC4 (all logged in users). • PBC state of Enable of PBC4
Scenario7 – Last LCA within PB branch1 logout	Credit Admin state will change false. The Credit admin state (false) will be sent to • All Logged in PBC user(s) within the PB Branch1.
Scenario8 – Last PB trader within PB branch logout	No U3 notification message sent out from the MFG.
Scenario9 – Last STU within PB branch logout	No U3 notification message sent out from the MFG.
Scenario10- PBC_2 user1 logout	No U3 notification message sent out from the MFG.
Scenario 11- NCG credit admin user change the Domcad value of PB1 branch	U3 notification(s) containing the following data shall be sent to LCA user1 and PBC1/PBC2 (all logged in users). • Updated DomCad value of PB Branch1
Scenario 12- Credit admin (LCA) status of PB branch2 is changed	U3 notification(s) containing the following data shall be sent to PBC3 and PBC4 (all logged in users). • Updated credit admin status of PB branch 2
Scenario13 – auto credit reset of PBC2	U3 notification(s) containing the following data shall be sent to LCA user1 and PBC2 (all logged in users). • Updated BranchStatus of PBC2(from inhibit to enable)

U3 -Notification to PB CA (GCA/LCA)

- Within the GCO (Global credit organization), there are 2 types of Prime broker credit Administrator (PB CA) i.e. GCA (Global credit administrator) and LCA (local credit Administrator)
- Within the non GCO, there is only one type of Prime broker credit Administrator (PB CA) i.e. LCA (local credit Administrator)
- On successful login the MFG must notify PB CA of current state of trading status (enable or Inhibit or disable), Max Open Orders per Instrument, Max Order Size each administrable PBC branch and Domcad.

Detail	Message Definition Cross Reference
If a PBC site is enabled then the MAPI CA will be notified	PrimeBrokerCreditNotification(U3) message containing BranchStatus(20010)=**1**(Enable PBC Branch)
If a PBC site is inhibited then the MAPI CA will be notified	PrimeBrokerCreditNotification(U3) message containing BranchStatus(20010)=**3** (Inhibit PBC Branch)
If a PBC site is disabled then the MAPI CA will be notified	PrimeBrokerCreditNotification(U3) message containing BranchStatus(20010)=**2** (Disable PBC Branch)

Detail	Message Definition Cross Reference
MAPI CA will be notified of the Max Open Orders Per Instrument applicable to each PBC	PrimeBrokerCreditNotification(U3) message containing value of the Maximum Open Orders Per Instrument in MaxOpenOrderPerInstrument(20003) if the MOOPI has never had set for a PBC; the default value of 200 will be generated by MFG in the U3 message
MAPI CA will be notified of the Max Order Size applicable to each PBC	PrimeBrokerCreditNotification (U3) message containing value of the Max Order size in OrderQty (38). If the MOS has never had set for a PBC; the default value of 999 will be generated by MFG in the U3 message.
MAPI CA will be notified if host setting requiring at least one MAPI PB CA to be connected to MFG is enabled Domcad value is specified only for the PB branch which MAPI CA belonging to.	PrimeBrokerCreditNotification(U3) message containing DOMCAD=**Y** (PB CA MFG connection Required)
MAPI CA will be notified if host setting requiring at least on MAPI PB CA to be connected to MFG is disabled Domcad value is specified only for the PB branch which MAPI CA belonging to.	PrimeBrokerCreditNotification(U3) message containing DOMCAD=**N**(PB CA MFG connection Not Required)

- Only the change of trading status, Max Open Orders per Instrument, Max Order Size for each administrable PBC branch and Domcad will be notified to PB CA. An unchanged data will not be populated in the U3 message.
- On successful login or on change of values current state of trading status, Max Open Orders per Instrument and Max Order Size for each administrable PBC branch, PB CA will not be notified the value of the creditadminState.

Note: Note: Un-inhibit is not valid use case, an inhibited PBC can only be enabled or Disabled

- MAPI must inform client to upgrade to EP 171 version if they need to support U3 messages.
- MFG will only send U3 notification to clients who log on to MFG with EP 171 or higher version.
- In the case whereby a client logs on to MFG with EP 171, but they do not support U3 message in their FIX data dictionary, MFG must not cease sending them U3 messages and shall handle their rejections (in the form of BMR or session level reject (3)) in the same way as today.
- If it is an empty update i.e. there are no changes of the PBC trading status (enable or Inhibit or disable), Max Open Orders per Instrument, Max Order Size each administrable PBC branch and Domcad specified in the U1 or Party action request (DH) messages, the U3 message shall not be generated to PB CA.

Note: PBC is either “Enabled”; “Disabled” or “Inhibited” so only the current state will be communicated

U3 -Notification to PBC

- On successful login the MFG must notify a MAPI PBC (all logged in users) with current state of their trading status, Max Open Orders Per Instrument, Max Order Size, Domcad and if trading is suspended due to no PB CA logged in via FIX.

Detail	Message Definition	Note
If a PBC site is enabled then the PBC will be notified	PrimeBrokerCreditNotification(U3) message containing BranchStatus(20010)=**1**(Enable PBC Branch)	order entry may be possible because the PBC site has been enabled by a PB CA
If a PBC site is inhibited then the PBC will be notified	PrimeBrokerCreditNotification(U3) message containing BranchStatus(20010)=**3** (Inhibit PBC Branch)	their site has been disabled by a PB CA
If a PBC site is disabled then the PBC will be notified	PrimeBrokerCreditNotification(U3) message containing BranchStatus(20010)=**2** (Disable PBC Branch)	
PBC will be notified of the Max Open Orders Per Instrument applicable to their branch	PrimeBrokerCreditNotification(U3) message containing value of the Maximum Open Orders Per Instrument in MaxOpenOrderPerInstrument(20003)	Notifying PBC with MOOPI for its site. if the MOOPI has never had set for a PBC; the default value of 200 will be generated by MFG in the U3 message
PBC will be notified of the Max Order Size applicable to their branch	PrimeBrokerCreditNotification(U3) message containing value of the Max Order size in OrderQty(38)	notifying PBC with MOS for its site If the MOS has never had set for a PBC; the default value of 999 will be generated by MFG in the U3 message.
PBC will be notified if host setting requiring at least one PB CA to be connected to MFG is enabled Domcad value is specified only for the PB branch which MAPI CA belonging to.	PrimeBrokerCreditNotification(U3) message containing DOMCAD=**Y** (PB CA MFG connection Required)	
PBC will be notified if host setting requiring at least on PB CA to be connected to MFG is disabled. Domcad value is specified only for the PB branch which MAPI CA belonging to.	PrimeBrokerCreditNotification(U3) message containing DOMCAD=**N** (PB CA MFG connection Not Required)	
PBC will be notified if no FIX PB CA logged in	PrimeBrokerCreditNotification(U3) message containing CreditAdminStatus(20011)=1 (PB CA Disconnected)	
PBC will be notified if FIX PB CA logged in.	PrimeBrokerCreditNotification(U3) message containing CreditAdminStatus(20011)=2 (PB CA Connected)	order entry may no longer be disabled because there is a PB CA connected to the MFG

- Only the change of their PBC trading status, Max Open Orders per Instrument, Domcad, Credit admin status and Max Order Size will be notified to impact PBC. An unchanged data will not be populated in the U3 message.
- For backward compatibility there will be no change to existing CB message, but it is used for notifying MOS will be marked as deprecated to be removed in a later release.
- If it is an empty update i.e. there are no changes of their PBC trading status (enable or Inhibit or disable), credit admin status, Max Open Orders per Instrument, Max Order Size each administrable PBC branch and Domcad specified in the U1 or Party action request (DH) messages, the U3 message shall not be generated to PBC.

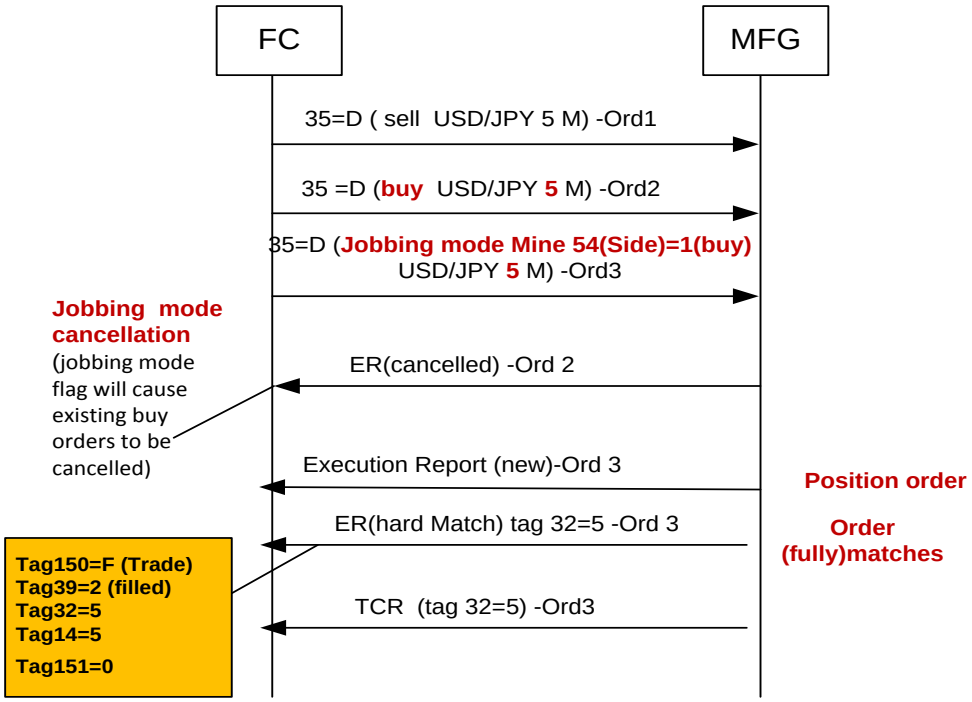
U3- out of scope

- There shall be no U3 notification message sent out from MFG to PB trader user.
- There shall be no U3 notification message sent from MFG to PB Special ticket user (STU).

Appendix G MFG support for Jobbing Mode on Yours/Mine orders

MAPI client	Client => MFG
A MAPI client submits a Bid order	Tag 54(side)=1(Buy)
A MAPI client submits an Offer order	Tag 54(side)=2(Sell)
A MAPI client submits a Bid order with jobbing mode enable	Tag 54(side)=1(Buy) Tag 1628(TriggerScope)=4(enables Jobbing Mode) Tag1100(TriggertYPE)=5 (on order entry or order modification) Tag1101(TriggerAction)=3 (cancel) 'Jobbing mode enabled = the previous BUY order of same instruments will be cancelled.
A MAPI client submits an Offer order with jobbing mode enable	Tag 54(side)=2(Sell) Tag 1628(TriggerScope)=4(enables Jobbing Mode) Tag1100(TriggertYPE)=5 (on order entry or order modification) Tag1101(TriggerAction)=3 (cancel) Jobbing mode enabled = the previous SELL order of same instruments will be cancelled.
A MAPI client submits a Mine order.	Tag 54(side)=1(Buy) Tag59(TimeInForce)=3 (immediate or Cancel-IOC)
A MAPI client submits a Yours order	Tag 54(side)=2(Sell) Tag59(TimeInForce)=3 (immediate or Cancel-IOC)
A MAPI client submits a Mine order. jobbing mode enable	Tag 54(side)=1(Buy) Tag59(TimeInForce)=3 (immediate or Cancel-IOC) Tag 1628(TriggerScope)=4(enables Jobbing Mode) Tag1100(TriggertYPE)=5 (on order entry or order modification) Tag1101(TriggerAction)=3 (cancel) Jobbing mode enabled+ IOC = the previous BUY order of same instruments will be cancelled.
A MAPI client submits a Yours order jobbing mode enable	Tag 54(side)=2(Sell) Tag59(TimeInForce)=3 (immediate or Cancel-IOC) Tag 1628(TriggerScope)=4(enables Jobbing Mode) Tag1100(TriggertYPE)=5 (on order entry or order modification) Tag1101(TriggerAction)=3 (cancel) Jobbing mode enabled+ IOC = the previous SELL order of same instruments will be cancelled.

7.2.8 S1- FX Spot Mine/Yours jobbing mode set and is filled completely with a single match



1) In this scenario a MAPI client submits **3** New Order Single messages to the FXFG for an FX Spot:
   a. Order 1 is an standing Offer (Sell) order USD/JPY with quantity =5 millions
   b. Order 2 is a standing Bid(Buy) order USD/JPY with quantity = 5 millions
   c. Order 3 is a Mine order (IOC tag 59= 3 and tag 54=1) order USD/JPY with Jobbing mode enabled set (tag 1628=4) Quantity= 5 millions
2) When an Order3 request is submitted to the FXFG, the jobbing mode flag will cause existing buy orders to be cancelled. In this flow is Order2. An Execution Report (cancel) of Order2 will be sent to the client. This message will contain:

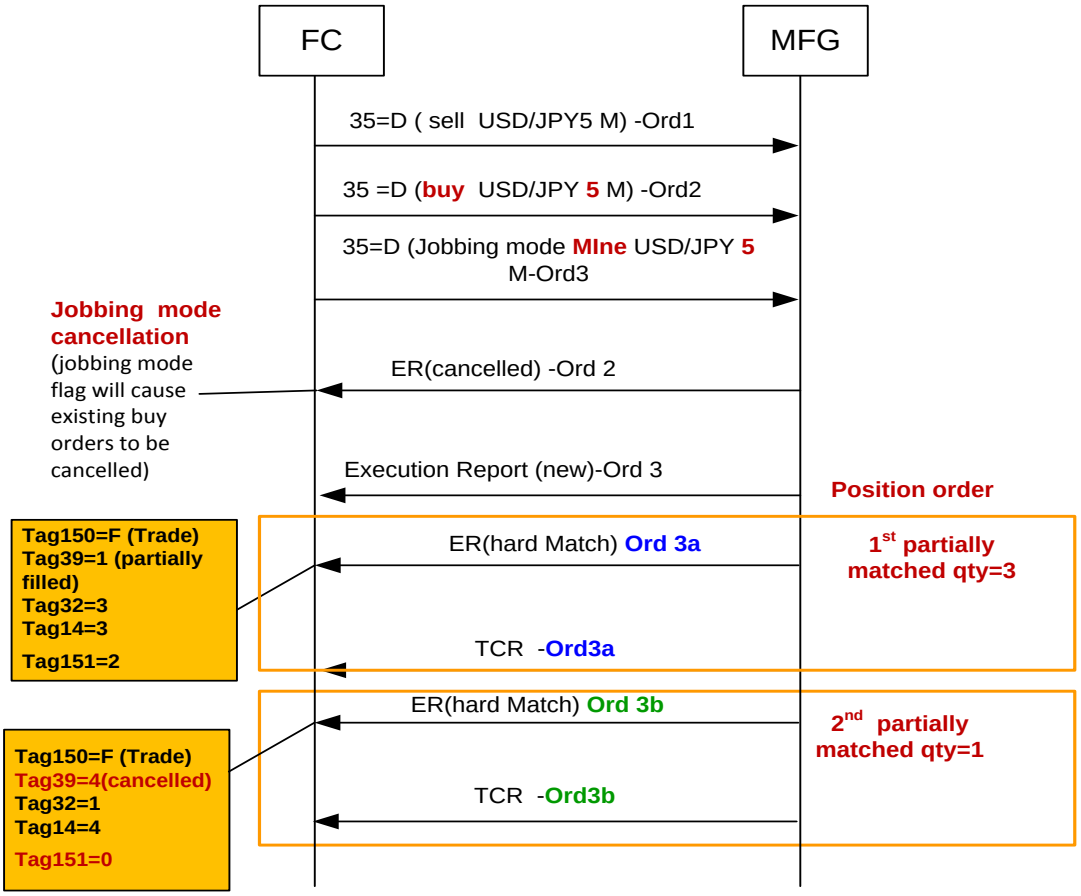
FIX Field	FIX Tag	Value	Note
OrdStatus	39	4 (Cancelled)	
ExecType	150	4 (Cancelled)	

3) An Order3 is fully matched with some existing market order, Execution report Hard Match (35=8) will be sent to client. This message will contain:

FIX Field	FIX Tag	Value	Note
OrdStatus	39	2 (filled)	
ExecType	150	F (Trade)	
LastQty	32	5	
CumQty	14	5	
LeaveQty	151	0	

4) When a match notification acknowledgment has been received from both counterparties, the Trade Capture Report of Order 3 will be generated to client.

7.2.9 S2- FX Spot Mine/Yours jobbing mode set and 2 partially matched (total order qty =5, first partially matched qty=3 and second qty =1)



1) Same as Items 1 to 2 in **S1.**
2) There are 2 partially match (of the Order3) with some existing market order.
    a. The quantity of the first partially matched (Ord3a) is 3 million. The Execution Report (hard match) will be sent to client. This message will contain:

FIX Field	FIX Tag	Value	Note
OrdStatus	39	1 (partially filled)	
ExecType	150	F (Trade)	
LastQty	32	3	
CumQty	14	3	
LeaveQty	151	2	

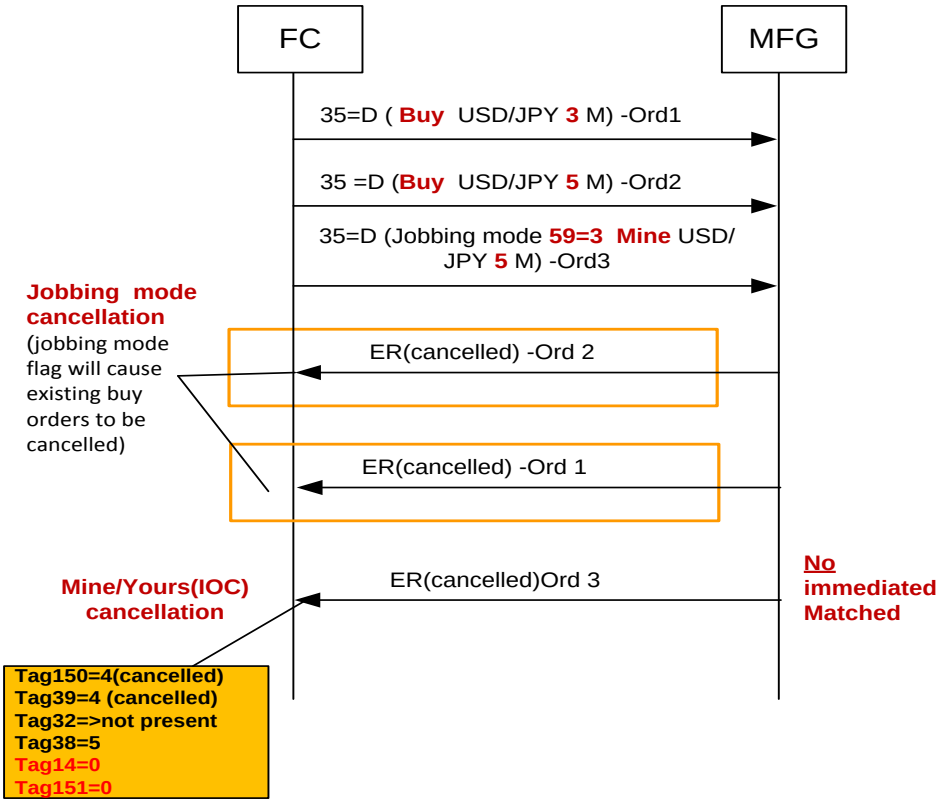
    b. The quantity of the Second partially matched (Ord3b) is 1 million. The Execution Report (hard match) will be sent to client. This message will contain:

FIX Field	FIX Tag	Value	Note
OrdStatus	39	4 (Cancelled)	Represent the cancellation status of a Mine/Yours order.
ExecType	150	F (Trade)	
LastQty	32	1	

FIX Field	FIX Tag	Value	Note
CumQty	14	4	
LeaveQty	151	0	Even there should be a quantity of 1million left, but this is a Mine/Yours order, the remaining quantity will be cancelled if it is not immediate matches.

3) When a match notification acknowledgment has been received from both counterparties, the Trade Capture Report of Order 3a and Order3b will be generated to client.

7.2.10 S3 - FX Spot Mine/Yours jobbing mode set and not immediate matched



1) In this scenario a MAPI client submits **3** New Order Single messages to the FXFG for an FX Spot:
   a. Order 1 is an standing Bid(Buy) order USD/JPY with quantity =3 millions
   b. Order 2 is a standing Bid(Buy) order USD/JPY with quantity = 5 millions
   c. Order 3 is a Mine (IOC tag 59= 3) order USD/JPY with jobbing mode enabled set (tag 1628=4). Quantity=5 millions
2) When an Order3 request is submitted to the FXFG, the jobbing mode flag will cause existing buy orders to be cancelled. In this flow is Order1and 2. Execution report (cancelled) will be generated in the same way as **Step2 in S1.**
3) Order3 has not been matched, an Execution report (cancelled) will be sent to client. This will contain:

FIX Field	FIX Tag	Value	Note
OrdStatus	39	4 (Cancelled)	
ExecType	150	4 (Cancelled)	

FIX Field	FIX Tag	Value	Note
LastQty	32	Not populated	Tag 32 is not populated in the ER (cancel) - existing functionality.
CumQty	14	0	
LeaveQty	151	0	

7.2.11 S4 - PBC places FX Spot Mine/Yours jobbing mode set is filled completely with a single match

The sequence of events for PBC is identical to FIX Client in S1.

PB STU will receive PB Match Notifications and PB Trade Ticket.

7.2.12 S5 - PBC places FX Spot Mine/Yours jobbing mode set, partially matched

The sequence of events for PBC is identical to FIX client in S2.

PB STU will receive PB Match Notifications and PB Trade Ticket.

7.2.13 S6- PBC places FX spot Mine/Yours jobbing mode set and not immediate matched

The sequence of events for PBC is identical to FIX client in S3.

PB STU will receive nothing.

7.2.14 S7- FX Swap Mine/Yours jobbing mode set and fully soft matched

The sequence of events is identical to S1 but without the Trade Ticket flow.

7.2.15 S8- FX Swap Mine/Yours jobbing mode set and partially Soft matched

The sequence of events is identical to S2 but without the Trade Ticket flow.

7.2.16 S9- FX Swap Mine/Yours jobbing mode set and not immediate matched

The sequence of events is identical to S3 but without the Trade Ticket flow.

Appendix H Forward swap Bell workflow

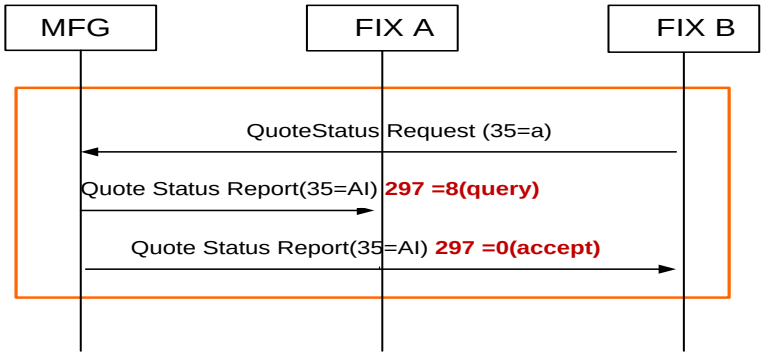


Diagram 1 - Successful Bell request

Scenario	workflow
Successful Bell request	A Quote Status Report (35=AI) containing tag 297=0 (accept) will be sent to the user who submit the request. A Quote Status Report (35=AI) containing tag 297=8 (query) will be sent to the counterparty.

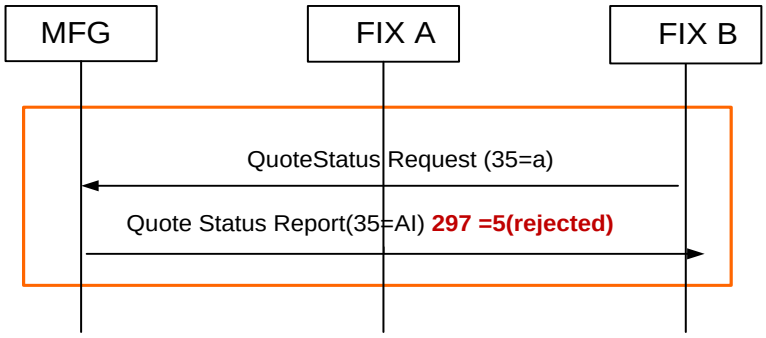


Diagram 2- Unsuccessful Bell request

Scenario	workflow
Unsuccessful Bell Request (diagram 2)	A Quote Status Report (35=AI) containing tag 297=7 (rejected) will be sent to the user who submit the request. The counterparty will receive nothing.

For more information

Send us a sales enquiry at
financial.thomsonreuters.com/sales

Read more about our products at
financial.thomsonreuters.com

Find out how to contact your local office
financial.thomsonreuters.com/locations

Date of issue: 30 June 2015

THOMSON REUTERS