FINANCIAL INFORMATION EXCHANGE PROTOCOL (FIX)

Version 5.0

VOLUME 5 – FIX APPLICATION MESSAGES: POST-TRADE

December 2006

© Copyright, 2006, FIX Protocol, Limited

DISCLAIMER

THE INFORMATION CONTAINED HEREIN AND THE FINANCIAL INFORMATION EXCHANGE PROTOCOL (COLLECTIVELY, THE "FIX PROTOCOL") ARE PROVIDED "AS IS" AND NO PERSON OR ENTITY ASSOCIATED WITH THE FIX PROTOCOL MAKES ANY REPRESENTATION OR WARRANTY, EXPRESS OR IMPLIED, AS TO THE FIX PROTOCOL (OR THE RESULTS TO BE OBTAINED BY THE USE THEREOF) OR ANY OTHER MATTER AND EACH SUCH PERSON AND ENTITY SPECIFICALLY DISCLAIMS ANY WARRANTY OF ORIGINALITY, ACCURACY, COMPLETENESS, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. SUCH PERSONS AND ENTITIES DO NOT WARRANT THAT THE FIX PROTOCOL WILL CONFORM TO ANY DESCRIPTION THEREOF OR BE FREE OF ERRORS. THE ENTIRE RISK OF ANY USE OF THE FIX PROTOCOL IS ASSUMED BY THE USER.

NO PERSON OR ENTITY ASSOCIATED WITH THE FIX PROTOCOL SHALL HAVE ANY LIABILITY FOR DAMAGES OF ANY KIND ARISING IN ANY MANNER OUT OF OR IN CONNECTION WITH ANY USER'S USE OF (OR ANY INABILITY TO USE) THE FIX PROTOCOL, WHETHER DIRECT, INDIRECT, INCIDENTAL, SPECIAL OR CONSEQUENTIAL (INCLUDING, WITHOUT LIMITATION, LOSS OF DATA, LOSS OF USE, CLAIMS OF THIRD PARTIES OR LOST PROFITS OR REVENUES OR OTHER ECONOMIC LOSS), WHETHER IN TORT (INCLUDING NEGLIGENCE AND STRICT LIABILITY), CONTRACT OR OTHERWISE, WHETHER OR NOT ANY SUCH PERSON OR ENTITY HAS BEEN ADVISED OF, OR OTHERWISE MIGHT HAVE ANTICIPATED THE POSSIBILITY OF, SUCH DAMAGES.

No proprietary or ownership interest of any kind is granted with respect to the FIX Protocol (or any rights therein), except as expressly set out in FIX Protocol Limited's Copyright and Acceptable Use Policy.

© Copyright 2003-2006 FIX Protocol Limited, all rights reserved

REPRODUCTION

FIX Protocol Limited grants permission to print in hard copy form or reproduce the FIX Protocol specification in its entirety provided that the duplicated pages retain the “Copyright FIX Protocol Limited” statement at the bottom of the page.

Portions of the FIX Protocol specification may be extracted or cited in other documents (such as a document which describes one’s implementation of the FIX Protocol) provided that one reference the origin of the FIX Protocol specification (***http://www.fixprotocol.org***) and that the specification itself is “Copyright FIX Protocol Limited”.

FIX Protocol Limited claims no intellectual property over one’s implementation (programming code) of an application which implements the behavior and details from the FIX Protocol specification.

Contents – Volume 5

© Copyright, 2006, FIX Protocol, Limited

APPENDIX A – TRADE AMENDMENT AND TRADE CANCEL WORK FLOW DIAGRAMS ... 154

© Copyright, 2006, FIX Protocol, Limited

© Copyright, 2006, FIX Protocol, Limited

FIX APPLICATION MESSAGES: POST-TRADE

Post-trade messaging is characterized as messages which are typically communicated after the placement and successful execution of an order and prior to settlement.

The specific FIX post-trade messaging categories are:

1. ALLOCATION
2. CONFIRMATION
3. SETTLEMENT INSTRUCTIONS
4. TRADE CAPTURE
5. REGISTRATION INSTRUCTIONS
6. POSITION MAINTENANCE
7. COLLATERAL MANAGEMENT

Descriptions and formats of the specific FIX post-trade application messages follow.

© Copyright, 2006, FIX Protocol, Limited

CATEGORY: ALLOCATION

See Volume 7 – PRODUCT: FIXED INCOME for specific usage guidance in using the allocation message set for Fixed Income.

See Volume 7 – PRODUCT: EQUITIES for specific usage guidance in using the allocation message set for Equities.

Overview

This section provides a overview on how the FIX protocol can be used to support the process of providing an allocation instruction together with the appropriate responses.

Note in all of the following, the term 'Initiator' is taken to mean the initiator of an Allocation Instruction and the 'Respondent' to mean the receiver of that instruction. In typical bi-party scenarios involving a buyside and a sellside firm, the buyside firm is the Initiator and the sellside firm the Respondent. A similar overview is also provided at start of the Catergory on FIX Confirmations. These two overviews provide a summary on how FIX messaging can be used for booking, allocation and confirmation up to the the start of settlement processing.

Further detail and additional optional flows for Allocations are included in "Example Usage" at the end of this Category section.

Allocation instructions can be communicated by the Initiator via three different options:

1. **Pre-allocated order** – in this option the Initiator would communicate the allocation instructions within the New Order message when the order is placed with the Respondent.
2. **Pre-trade allocation** – in this option the Initiator would communicate the allocation instructions to the Respondent in a separate message using the Allocation Instruction message. The Allocation message is sent after the order is placed with the Respondent but *before the trade is completed by the Respondent.*
3. **Post-trade allocation** – in this option the Initiator would communicate the allocation instructions to the Respondent in a separate message using the Allocation Instruction message *after the trade has been completed by the Respondent.*

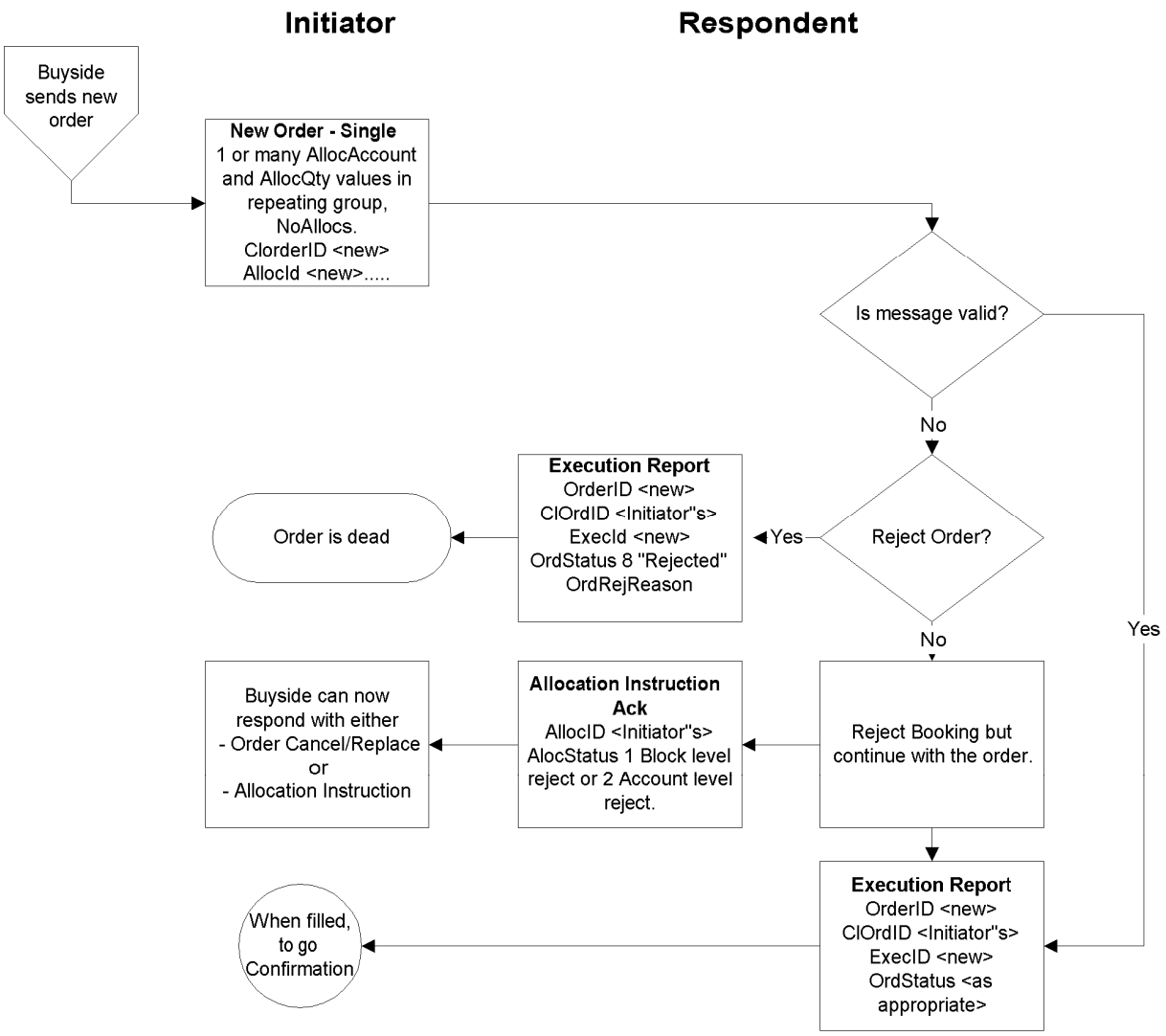
Note the use of options 1 and 2 lends itself best to scenarios where the average price can be agreed up front (e.g. principal trades) or where the allocation account details need to be communicated prior to execution in certain markets.

For the Initiator, options 2 and 3 represents the same message flow. The main difference is when the Allocation Instruction message is sent – in option 2 it is sent prior to the trade being completed and in option 3 it is sent after the trade has been completed. For the purposes of diagramming, options 2 and 3 will be represented as the same message flow diagram.

© Copyright, 2006, FIX Protocol, Limited

Pre-allocated order

Option 1 – Pre-allocated order: uses details on the New Order - single message



Click ***here*** to go to "Confirmation"

In the Pre-allocated order scenario the Initiator would send a New Order message that includes the allocation information needed by the Respondent to allocate the trade once the trade is completed. This scenario consists of the following steps:

- Initiator sends a New Order request message specifying one or more AllocAccount and AllocQty values within the repeating group designated by NoAllocs. This message will contain an AllocID which can be referenced in subsequent messages.
- Respondent sends Execution Report messages for the "New" and resulting fills.
- Respondent may optionally send an Allocation Instruction Ack of status 'received'.
- If there are errors in the allocation information it is possible to either:
    - reject the order

© Copyright, 2006, FIX Protocol, Limited

or

- to accept the order and reject the allocation details via the use of the Allocation Instruction Ack message (see Pre-trade allocation for detail of Block Level and Account Level reject. Either is possible here).

For example - one account cannot be identified, or the quantity of one allocation instance does not meet minimum quantity/minimum increment rules for the instrument, or the sum of allocated quantities does not equal the block trade quantity.

- Respondent may optionally send an Allocation Instruction Ack of status 'accepted'.
- The next step is "Confirmation", see Confirmation section.

Note where the average price or allocation quantity cannot be agreed up front but the allocation account details do need to be communicated prior to execution (e.g. for regulatory reasons), the Allocation Instruction can optionally be used post execution in 'Ready to Book' mode to communicate the booking instruction (including average price) to the sell side. As well as providing confirmation of the average price, this also supports the combination of orders for booking and allocation. If this is done, the Respondent should respond with Allocation Instruction ACKs of status 'received', then 'accepted'.

Cancel/Replace Processing for Pre-Allocated Orders

The AllocID on the New Order message is used to define uniquely the set of allocations contained within that order. If the order is replaced, the Cancel/Replace message should be formatted as follows:

- If the order details are changing but the allocation details are not (e.g. change in limit price), the NoAllocs group should **not** be populated.
- If the allocation details are changing, the NoAllocs group should be populated with the new complete set of allocation details with a **new** AllocID. This is regardless of whether the rest of the order details are changing or not. Examples of this are:

  a) the order is being re-allocated into different accounts

  or

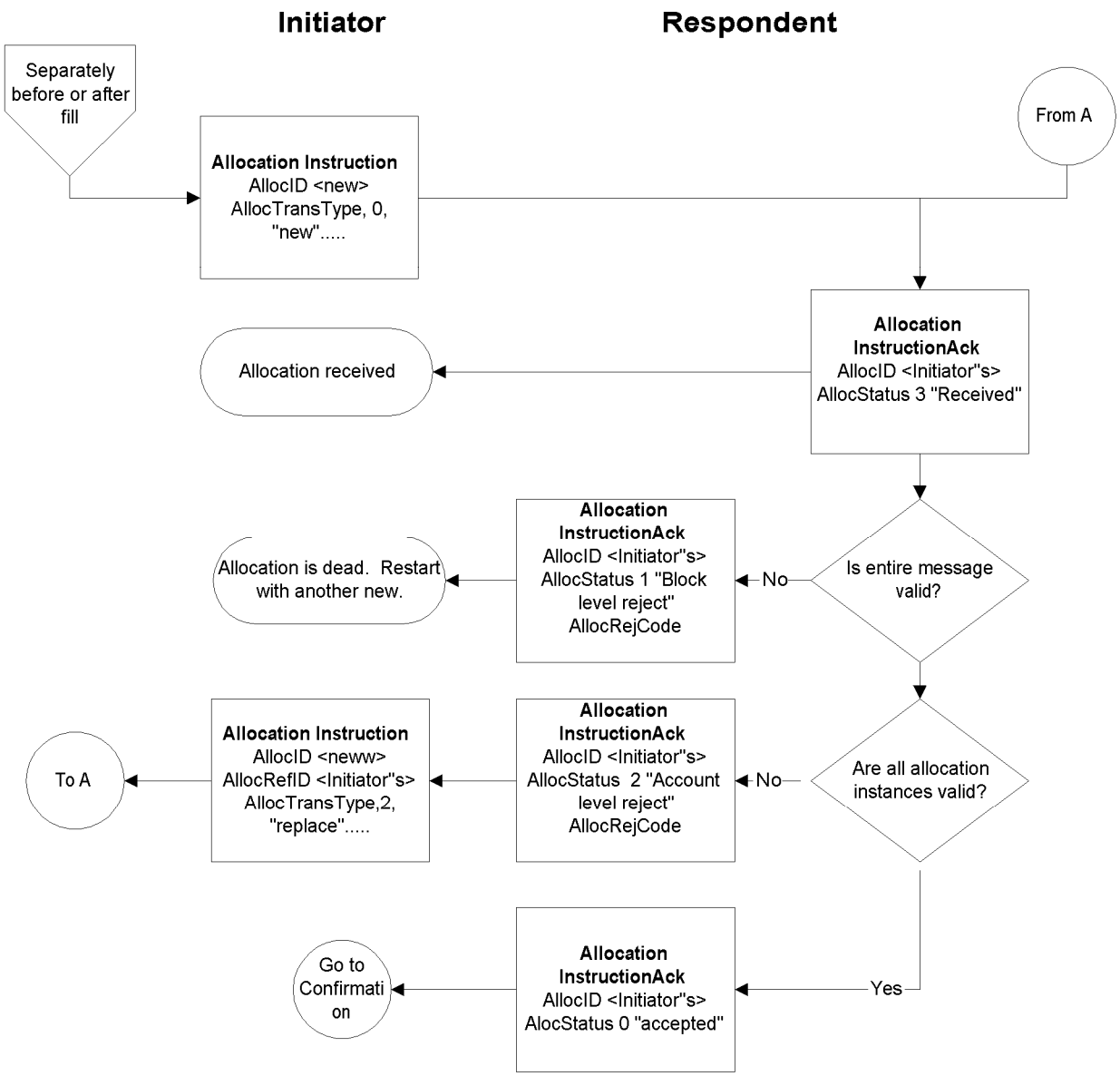
  b) the order quantity is changing (in which case the AllocShares allocated to each account will also need to change).

This ensures that AllocID is always unique on messages and therefore avoids any potential ambiguity arising from sharing different versions of allocation details for the same AllocID.

© Copyright, 2006, FIX Protocol, Limited

Pre-trade allocation

Option 2 & 3 – Pre-trade allocation and Post-trade allocation



Click ***here*** to go to "Confirmation"

In the Pre-trade allocation scenario, the Initiator would send the allocation instructions after placing the order but before the the order had been completed. This scenario consists of the following steps:

- Initiator sends a New Order request message (containing no allocation details)
- Initiator sends an Allocation Instruction message. If the average price has been agreed up front, this should be present on the message.
- Respondent sends Execution Report messages for the "New" and resulting fills.
- Respondent sends Allocation Instruction Ack of status 'received'.

© Copyright, 2006, FIX Protocol, Limited

- Before accepting the instruction, the Respondent should determine that all accounts are known, the quantity of each allocation instance meets minimum quantity/minimum increment rules for the instrument and the sum of allocated quantities equals the block trade quantity. If any error is found the Respondent must either:
  - reject the entire allocation using the Allocation Instruction Ack message with the appropriate reject reason code "Block Level reject"

    or
  - reject the accounts that are in error using the Allocation Instruction Ack message reject reason code "Account level reject".

  In this latter event, the Initiator can send another Allocation Instruction message with the correct instructions and information to the Respondent. This cycle can be repeated until the allocation is accepted by the Respondent.
- If the Respondent accepts the allocation, an Allocation Instruction Ack message is sent to the Initiator with an AllocStatus of "accepted".
- The next step is "Confirmation", see later section.

In the Pre-trade allocation scenario, the Allocation Instruction can be used for a number of purposes using the AllocType field to indicate the type or purpose of the message:

- Calculated (includes MiscFees and NetMoney), i.e. the flow commonly used for "US domestic equities booking and allocation model".
- Preliminary (without MiscFees and NetMoney), i.e. the flow commonly used for non-US domestic booking and allocation (the 'international equities model').
- Ready-To-Book, used to indicate to the Respondent firm that one or a combined (aggregated) set of orders are "Ready-To-Book" without specifying individual account breakdowns. This can be used to trigger post-trade allocation, matching, and settlement processing via other channels (e.g. post-trade industry utilities).
- Warehouse instruction, ***See Volume 7 – PRODUCT: EQUITIES*** for specific usage guidance on this topic.

Post-trade allocation

The Post-trade allocation scenario is very similar to that given above for Pre-trade allocation. In this scenario, the Initiator would send the allocation instructions to the Respondent after receiving the Execution Report message indicated that the trade is completed.

The Allocation Instruction can be used for a number of purposes using the AllocType field to indicate the type or purpose of the message:

- Calculated (includes MiscFees and NetMoney)
- Preliminary (without MiscFees and NetMoney)
- Ready-To-Book
- Warehouse instruction.

Post-Trade Allocation can be computed via one of two methods:

1. Using Average Price: Each AllocAccount has a single AllocAvgPx
2. Using Executed Price: Combination of each AllocAccount **and** AllocPrice (unique LastPx) (e.g. Japan)

© Copyright, 2006, FIX Protocol, Limited

Ready-To-Book Processing:

The Ready-To-Book capability of the Allocation Instruction message is designed to provide a clean interface between the "trading" and "booking" spaces. This allows buyside firms to both trigger and provide suitable references which can be passed down to assist in the matching process within industry utilities (e.g. Virtual Matching Utilities) or bilaterally with their sellside counterparts. Bookable units can be single fills,combinations of fills, single orders, or groups of orders for the same security, side, settlement date, etc. Automated booking instructions can be communicated either pre-trade or post-trade.

Booking instructions can be communicated **Pre-Trade** (at the time the order is being placed) to convey that as soon as the order is filled it can be considered by the Respondent as ready for booking (in particular when there is no additional quantity behind).

Booking instructions can also be communicated **Post-Trade** (after fills have been received and processed) to signal that a particular order is now ready for booking or to signal that a set of orders for the same security, side, settlement date, etc. are to be aggregated as single booking unit which is now ready for booking.

Central Counterparty-centric Allocation Model

An Overview of Three-party Allocations

The central counterparty-centric model allows the sell-side and buy-side to use a central clearing entity to manage allocation activity and remove all counterparty risk associated with the allocation transaction. This model is also known as the three-party allocation model. The model is distinguished from the two-party model described above by the fact that a central counterparty stands between two actors in the form of a clearing organization. This allocation model is commonly used in the world of listed derivatives.

Allocations are essential to the exchange-based business model in that they facilitate the movement of trades between parties after the trade has been made, in effect creating a second market between participants. They serve to improve liquidity and increase the overall viability of a market. The formal definition of an allocation is a transaction in which some portion of a trade or group of trades is offered to a second party at a specific price. The second party then has the option to accept or reject the offer. If accepted, the allocation is finalized and becomes a financial contract between the parties with the clearing house acting as the intermediary. The second party now has liability for the allocation. If rejected, the offering party may rescind the offer or modify it so that it is acceptable to the claiming party.

Three-party allocations allow brokers and clearing firms to specialize in what they do best. Firms who have a floor and electronic presence can concentrate on trade execution without having to worry about maintaining customer accounts. Firms that specialize in managing customer accounts can do so, relying on firms that specialize in trading for best execution and a pipeline of trades. The trading specialists are referred to as “give-up” firms. The firms that emphasize account management are referred to as “claim” firms.

Three-party Allocation Work Flow

The term “allocation” describes the method by which trades are routed to their ultimate firm and account held at the clearing organization. Trades can be designated for allocation simply by submitting them with an allocation indicator. Additional specific allocation information such as claim firm, account and origin can also be appended to trades marked for allocation. These specific allocation instructions can also be submitted as stand alone instructions referencing a group of trades already marked for allocation. Once these specific instructions have been submitted, whether with the initial trade information or later with stand alone allocation instructions, the Clearing House sends an Allocation Report message to the designated claim (repsondant) firm alerting them of the allocation. The claim firm can respond to this alert by either claiming or rejecting the allocation via an Allocation Report Acknowledgement message. The claim firm’s response would then be communicated back to the allocating firm by the Clearing House.

© Copyright, 2006, FIX Protocol, Limited

Regular Three-party Allocations

Basic Allocations are those that are done at a fixed trade price determined by the execution price of the trade. No price averaging is involved. Trades designated for basic allocation are aggregated according to pre-defined criteria into "allocation groups". The convention currently used for assigning trades to an allocation group uses the following criteria:

- Firm and Trader
- Trade Date
- Instrument definition (Symbol)
- Side of Market
- Trade Price
- Customer Account
- Trade Type
- Client Order ID

Designating Trades for Basic Three-party Allocation

Trades designated for basic allocation are first added to an allocation group and then, if the appropriate pre-allocation instruction details have been provided, becomes part of an allocation proper that is routed to the claim firm. The executing firm has the option of providing both the Group ID and Allocation ID, although under current listed derivatives convention the ID's are automatically assigned by the clearing house. At the time of execution, a firm has 2 options with respect to designating the allocation.

An executing firm may direct the trade into an allocation group pending the receipt of allocation instruction details. At this point the allocation is considered to be in "pending" status awaiting the executing firm to provide the details necessary to complete the allocation. The allocation Group ID is automatically assigned by the Clearing House and the information is sent to the claim firm in the form of an Allocation Group Notice.

An executing firm may also provide all requisite allocation details at the time of the trade. In this case, the trade will be assigned to an allocation group as well as create an actual allocation that will be sent to the claim firm. At this point, the allocation is considered to be in "preliminary" status.

Average Price Allocations

Average price allocations are different from Basic allocations in several significant ways. Trade Prices within a Group are averaged. The specifics of how they are averaged are discussed in the FIX Three-party Allocation Users Guide. For average price allocations, grouping is a two stage process; (1) Generic grouping of trades are averaged together according to a broad set of criteria. (2) For purposes of allocating, each Generic group will yield one or more underlying "Specific" groups whose criteria are defined in the same way as Basic allocation groups. Allocation instructions submitted for average price allocation groups are not released until the allocating firm has "completed" the average price group. Completion indicates that no more trades will be averaged in the designated group. Once the firm has completed the average price allocation group all existing allocation instructions are released to the designated claim firms.

Three-party Allocation Message Usage

There are three main message types used in the allocation process.

Allocation Instruction Alert

© Copyright, 2006, FIX Protocol, Limited

- Used to alert an allocating firm of the existence of pending Basic, Generic and Specific allocation groups. Sent by Clearing House or Exchange.

Allocation Instruction

- Sent by the allocating firm to submit an allocation instructions
- Sent by the allocating firm to mark an average price allocation group as complete or incomplete
- Passed through from the Clearing House or Exchange to the claim firm as a notice that an allocation instruction has been submitted
- Sent by the allocating firm to update previously submitted allocation instructions
- Sent by the claim firm to initiate a reversal of a previously accepted allocation

AllocationInstructionAck

- Used by the Clearing House to confirm receipt of Allocation Instructions
- Used by the claim firm to claim or reject allocations
- Sent to the allocating firm to notify it that an allocation has been claimed or rejected

Allocation Report

- Used by the Claim Firm to take further action on an allocation after it has been claimed, usually to modify the allocation account or perform sub-allocations

Allocation Report Ack

- Sent by the Clearing House to confirm receipt and acceptance of an Allocation Report

Fragmentation of Allocation Messages

FIX Allocation messages support fragmentation in a way similar to MassQuote and the List Order messages. If there are too many entries within a repeating group to fit into one physical message, the entries can be continued in subsequent messages by repeating the principal message reference and other required fields, then continuing with the repeating group. This is achieved by using an optional **TotNoAllocs** field (giving the total number of AllocAccount details across the entire allocation) that supplements the **NoAllocs** field (giving the number of AllocAccount details in a particular message fragment). The **TotNoAllocs** field is repeated with the same value in all fragments of the batch. For example, an Allocation Instruction with 200 allocation account instances could be fragmented across three messages - the first two containing TotNoAllocs=200, NoAllocs=80 and the third TotNoAllocs=200, NoAllocs=40. To help the receiver reconstitute the batch the Boolean field **LastFragment** is sent with a "Y" value in the last fragment.

For fragmented allocation events the receiving application must persist state between messages to determine whether all instances of the repeating group have been received before acting on the instruction or processing the report.

For this to work some key rules must be enforced:

1) The sender must supply a consistent value for TotNoAllocs in all related fragments and must use the same primary message reference in all fragments of the batch, e.g. AllocID in AllocationInstruction.
2) The sender must ensure that fragments are transmitted in order without intervening traffic.
3) The NoAllocs group must reach capacity only in the last fragment, and that message must contain LastFragment=Y.

© Copyright, 2006, FIX Protocol, Limited

4) The receiver must acknowledge every fragment received (AllocationInstructionAck with AllocStatus="received") and never reject a non-last fragment; acknowledgment of the final fragment accepts or rejects the entire set.

There are a number of design suggestions for implementing fragmentation:

1) Optional block-level fields supplied in early fragments need not be repeated in subsequent fragments. If they are repeated and the values are different, the receiver may choose to reject (on receiving the last fragment) or to apply the last received value to the event.
2) If a message supports multiple "Number of" groups, e.g. NoOrders, NoExecs, and NoAllocs in AllocationInstruction, the sender may distribute the array instances over any and all fragments, as long as the NoAllocs group is not filled before the last fragment.
3) The receiver must be able to abort collecting an incomplete array – either on expiration of a timer or the receipt of an unrelated message from the same counterparty.

FIX Message	<**Total number of**> field	related <**Number of**> field	Prinicipal message reference
AllocationInstruction	TotNoAllocs	NoAllocs (78)	AllocID (70)
AllocationReport	TotNoAllocs	NoAllocs (78)	AllocReportID (755)

Maximum message size for fragmentation purposes can be determined by using the optional MaxMessageSize field in the Logon message or by mutual agreement between counterparties.

© Copyright, 2006, FIX Protocol, Limited

Allocation Instruction

The Allocation Instruction message provides the ability to specify how an order or set of orders should be subdivided amongst **one or more** accounts. In versions of FIX prior to version 4.4, this same message was known as the Allocation message. Note in versions of FIX prior to version 4.4, the allocation message was also used to communicate fee and expense details from the Sellside to the Buyside. This role has now been removed from the Allocation Instruction and is now performed by the new (to version 4.4) Allocation Report and Confirmation messages.,The Allocation Report message should be used for the Sell-side Initiated Allocation role as defined in previous versions of the protocol.

Note the response to the Allocation Instruction message is the Allocation Instruction Ack message. In versions of FIX prior to version 4.4, the Allocation Instruction Ack message was known as the Allocation ACK message.

Allocation is typically communicated **Post-Trade** (after fills have been received and processed). It can, however, also be communicated **Pre-Trade** (at the time the order is being placed) to specify the account(s) and their respective order quantities which make up the order. This is a regulatory requirement in certain markets and for certain types of securities.

In the context of bilateral (buyside to sellside) communication, the buyside firm should be the "Initiator" of an Allocation Instruction message and a Sellside firm would be the "Respondent". An Allocation Instruction message can be submitted with AllocTransType of new, cancel or replace. The AllocType field indicates the type or purpose of the message:

- Calculated (includes MiscFees and NetMoney)
- Preliminary (without MiscFees and NetMoney)
- Ready-To-Book
- Warehouse instruction

It is possible either to specify, in the AllocSettlInstType field, full settlement instruction details on the Allocation Instruction message, to provide a reference to a settlement instruction held on a database of such instructions or to instruct the receiving party to perform one of the following actions:

- Use default instructions
- Derive the instructions from the parameters of the trade
- Phone for instructions

General guidelines applicable to this message:

- AllocID should be unique for all Allocation messages with AllocTransType=New.
- When submitting replace or cancel AllocTransType messages, the RefAllocID and AllocCancReplaceReason fields are required.
- To reject an Allocation Instruction message, an Allocation Instruction Ack with AllocStatus 'Block level reject' or 'Account level reject' should be used. Use of 'Block level reject' means the entire message has been rejected (e.g. due to one or more of the orders not matching, average price mismatch). 'Account level reject' is used when the block level matches successfully but one or more (or all) of the constituent account level details failed validation (e.g. account not found, incorrect MiscFees). In the latter case, the rejecting party can (optionally) notify the instructing party of those allocation details that are being rejected by listing the offending account IDs in the Allocation Instruction Ack message (a new NoAllocs repeating group has been introduced for this purpose).
- The correct response to an Allocation Instruction Ack of status 'Block level reject' is a new Allocation Instruction with AllocTransType 'New' (as the previous message has been rejected in entirety). In the case of an 'Account level reject', either the original Allocation Instruction should be cancelled (a new Allocation Instruction message referencing the original in RefAllocID, with AllocTransType 'Cancel') and reinstated (a second new Allocation Instruction message with AllocTransType 'New'), or fully replaced (a new Allocation Instruction, referencing the original in RefAllocID, with AllocTransType 'Replace'). Note a

© Copyright, 2006, FIX Protocol, Limited

replacement allocation message (AllocTransType=Replace) must contain **all** data for the replacement allocation message. It is the responsibility of the recipient of the Replace message to identify which items have been changed.

- It is permissible (though not mandatory) for the Respondent to reject an Allocation Instruction with AllocTransType = Cancel or Replace if the Allocation Instruction ACK of status 'Accepted' has already been sent. Manual communication would then be required to effect the required changes. This approach would generally be required where the Respondent is using the generation of the 'Accepted' Allocation Instruction ACK to move the allocation details into downstream processing (e.g. confirmation generation), in which case a subsequent cancellation of or amendment to the allocation details may require the details to be retrieved from the downstream process.
- Where amendment or cancellation of an allocation instruction has taken place out of band (i.e. manually or via some other means outside FIX), an Allocation Report message can be sent from the recipient of the allocation/cancellation to confirm back to the initiator that the relevant action has taken place.
- Where settling in markets where multiple alternative settlement locations exist, it is recommended that the settlement location (equivalent to ISO15022 'PSET' field) be identified on each allocation detail within the NoAllocs repeating group. A nested parties component block is provided which can be used for this purpose.

The allocation message contains repeating fields for each order, sub-account and individual execution. The repeating fields are shown in the message definition below in typeface ***Bold-Italic*** and indented with the → symbol. The field's relative position within the repeating group in the message is important. For example, each instance of allocation must be in the order as shown in the message definition below.

- The total quantity allocated must equal the Quantity value*. If present, the total quantity in the execution section must also be equal to this value. *Note that the total quantity of the allocation does not necessarily have to equal the total quantity of the orders being allocated. Good examples of where this does not necessarily take place are GT orders, especially where multi-day average pricing is taking place (refer to the 'Equities' section of Volume 7 for more details on these flows). The quantity of each order being booked must also be specified on the message. This will be equal to the order quantity if the entire order is being booked, though can be less if only part of the order is being booked. The sum of the order booking quantities must equal the Quantity value.
- The number of sub-account instances is indicated in NoAllocs.
- Multiple orders can be combined for allocation or for AllocType=" Ready-To-Book" or for AllocType = "Warehouse instruction". Note that combined orders must refer to the same instrument and have the same trade date, settlement date and side. The identification of the orders to be combined can be achieved in one of two ways:
    - By identifying the number of orders in the NoOrders field and each individual order in the OrderID fields. The AllocNoOrdersType field is used to denote that this is happening and takes value "1=Explicit list provided". If any orders were handled outside FIX, the ClOrdID must be set to 'MANUAL'. Regardless of whether the orders were handled within or outside FIX, the order quantity and average price must also be specified for each order. This is to assist in validating the message and, for manual orders, to help identify the correct orders to book.
    - By stating that an unspecified group of orders is to be combined. The NoOrders field in this case is left blank. The AllocNoOrdersType field is set to "0=Not specified" to specify that this is happening. Note use of this approach is only recommended where either the number of orders being booked is extremely large or some kind of aggregation rule is being used.
- Multiple executions can be combined for allocation by identifying the number of executions in the NoExecs field and each individual execution in the ExecID fields. Combined executions must refer to the same instrument, trade date, settlement date and side.

© Copyright, 2006, FIX Protocol, Limited

- Except where AllocTransType = 'Cancel' or where AllocNoOrdersType = "Not specified", the list of orders being booked or allocated must be specified by using their ClOrdID. If any orders were handled outside FIX, the ClOrdID must be set to 'MANUAL'. Regardless of whether the orders were handled within or outside FIX, and where the orders are specified, the order quantity and average price must also be specified for each order. This is to assist in validating the message and, for manual orders, to help identify the correct orders to book.

See "***Example Usage of Allocations and Ready-to-Book***" for more examples and details.

Allocation Instruction

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = J
70	AllocID	Y	Unique identifier for this allocation instruction message
71	AllocTransType	Y	i.e. New, Cancel, Replace
626	AllocType	Y	Specifies the purpose or type of Allocation message
793	SecondaryAllocID	N	Optional second identifier for this allocation instruction (need not be unique)
72	RefAllocID	N	Required for AllocTransType = Replace or Cancel
796	AllocCancReplaceReason	N	Required for AllocTransType = Replace or Cancel Gives the reason for replacing or cancelling the allocation instruction
808	AllocIntermedReqType	N	Required if AllocType = 8 (Request to Intermediary) Indicates status that is requested to be transmitted to counterparty by the intermediary (i.e. clearing house)
196	AllocLinkID	N	Can be used to link two different Allocation messages (each with unique AllocID) together, i.e. for F/X "Netting" or "Swaps"
197	AllocLinkType	N	Can be used to link two different Allocation messages and identifies the type of link. Required if AllocLinkID is specified.
466	BookingRefID	N	Can be used with AllocType=" Ready-To-Book "
857	AllocNoOrdersType	N	Indicates how the orders being booked and allocated by this message are identified, i.e. by explicit definition in the NoOrders group or not.
73	NoOrders	N	Indicates number of orders to be combined for allocation. If order(s) were manually delivered set to 1 (one).Required when AllocNoOrdersType = 1
→	**11** **ClOrdID**	N	Order ID assigned by client if order(s) were electronically delivered and executed. If order(s) were manually delivered this field should contain string "MANUAL".Note where an order has undergone one or more cancel/replaces, this should be the ClOrdID of the most recent version of the order

© Copyright, 2006, FIX Protocol, Limited

				Required when NoOrders > 0 and must be the first repeating field in the group.
→	**37**	**OrderID**	N	
→	**198**	**SecondaryOrderID**	N	Can be used to provide order id used by exchange or executing system.
→	**526**	**SecondaryClOrdID**	N	
→	**66**	**ListID**	N	Required for List Orders.
→	**component block <NestedParties2>**		N	Insert here the set of "NestedParties2" fields defined in "Common Components of Application Messages" This is used to identify the executing broker for step in/give in trades
→	**38**	**OrderQty**	N	
→	**799**	**OrderAvgPx**	N	Average price for this order
→	**800**	**OrderBookingQty**	N	Quantity of this order that is being booked out by this message (will be equal to or less than this order's OrderQty) Note that the sum of the OrderBookingQty values in this repeating group must equal the total quantity being allocated (in Quantity (53) field)
124	NoExecs		N	Indicates number of individual execution repeating group entries to follow. Absence of this field indicates that no individual execution entries are included. Primarily used to support step-outs.
→	**32**	**LastQty**	N	Amount of quantity (e.g. number of shares) in individual execution. Required if NoExecs > 0
→	**17**	**ExecID**	N	
→	**527**	**SecondaryExecID**	N	
→	**31**	**LastPx**	N	Price of individual execution. Required if NoExecs > 0
→	**669**	**LastParPx**	N	Last price expressed in percent-of-par. Conditionally required for Fixed Income trades when LastPx is expressed in Yield, Spread, Discount or any other price type
→	**29**	**LastCapacity**	N	Used to identify whether the trade was executed on an agency or principal basis.
→	**1003**	**TradeID**	N	
→	**1041**	**FirmTradeID**	N	
570	PreviouslyReported		N	
700	ReversalIndicator		N	
574	MatchType		N	
54	Side		Y	

© Copyright, 2006, FIX Protocol, Limited

component block <Instrument>		Y	Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages"
component block <InstrumentExtension>		N	Insert here the set of "InstrumentExtension" fields defined in "Common Components of Application Messages"
component block <FinancingDetails>		N	Insert here the set of "FinancingDetails" fields defined in "Common Components of Application Messages"
711	NoUnderlyings	N	Number of underlyings
→	**component block <UnderlyingInstrument>**	N	Must be provided if Number of underlyings > 0
555	NoLegs	N	Number of legs Identifies a Multi-leg Execution if present and non-zero.
→	**component block <InstrumentLeg>**	N	Must be provided if Number of legs > 0
53	Quantity	Y	Total quantity (e.g. number of shares) allocated to all accounts, or that is Ready-To-Book
854	QtyType	N	
30	LastMkt	N	Market of the executions.
229	TradeOriginationDate	N	
336	TradingSessionID	N	
625	TradingSessionSubID	N	
423	PriceType	N	
6	AvgPx	N	For F/X orders, should be the "all-in" rate (spot rate adjusted for forward points). For 3rd party allocations used to convey either basic price or averaged price Optional for average price allocations in the listed derivatives markets where the central counterparty calculates and manages average price across an allocation group.
860	AvgParPx	N	
component block <SpreadOrBenchmarkCurveData>		N	Insert here the set of "SpreadOrBenchmarkCurveData" fields defined in "Common Components of Application Messages"
15	Currency	N	Currency of AvgPx. Should be the currency of the local market or exchange where the trade was conducted.
74	AvgPxPrecision	N	Absence of this field indicates that default precision arranged by the broker/institution is to be used
component block <Parties>		N	Insert here the set of "Parties" (firm identification) fields defined in "Common Components of Application

© Copyright, 2006, FIX Protocol, Limited

			Messages"
75	TradeDate	Y	
60	TransactTime	N	Date/time when allocation is generated
63	SettlType	N	
64	SettlDate	N	Takes precedence over SettlType value and conditionally required/omitted for specific SettlType values.
775	BookingType	N	Method for booking. Used to provide notification that this is to be booked out as an OTC derivative (e.g. CFD or similar). Absence of this field implies regular booking.
381	GrossTradeAmt	N	Expressed in same currency as AvgPx. Sum of (AllocQty * AllocAvgPx or AllocPrice).
238	Concession	N	
237	TotalTakedown	N	
118	NetMoney	N	Expressed in same currency as AvgPx. Sum of AllocNetMoney.
77	PositionEffect	N	
754	AutoAcceptIndicator	N	Indicates if Allocation has been automatically accepted on behalf of the Carry Firm by the Clearing House
58	Text	N	
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
157	NumDaysInterest	N	Applicable for Convertible Bonds and fixed income
158	AccruedInterestRate	N	Applicable for Convertible Bonds and fixed income
159	AccruedInterestAmt	N	Applicable for Convertible Bonds and fixed income (REMOVED FROM THIS LOCATION AS OF FIX 4.4, REPLACED BY AllocAccruedInterest)
540	TotalAccruedInterestAmt	N	(Deprecated in FIX.4.4)
738	InterestAtMaturity	N	
920	EndAccruedInterestAmt	N	For repurchase agreements the accrued interest on termination.
921	StartCash	N	For repurchase agreements the start (dirty) cash consideration
922	EndCash	N	For repurchase agreements the end (dirty) cash consideration
650	LegalConfirm	N	
component block <Stipulations>		N	
component block <YieldData>		N	

© Copyright, 2006, FIX Protocol, Limited

component block <PositionAmountData>			N	Insert here here the set of "Position Amount Data" fields defined in "Common Components of Application Messages"
892	TotNoAllocs		N	Indicates total number of allocation groups (used to support fragmentation). Must equal the sum of all NoAllocs values across all message fragments making up this allocation instruction. Only required where message has been fragmented.
893	LastFragment		N	Indicates whether this is the last fragment in a sequence of message fragments. Only required where message has been fragmented.
78	NoAllocs		N	Conditionally required except when AllocTransType = Cancel, or when AllocType = Ready-to-book or Warehouse instruction
→	**79**	**AllocAccount**	N	May be the same value as BrokerOfCredit if ProcessCode is step-out or soft-dollar step-out and Institution does not wish to disclose individual account breakdowns to the ExecBroker. Required if NoAllocs > 0. Must be first field in repeating group. Conditionally required except when for AllocTransType="Cancel", or when AllocType= "Ready-To-Book" or "Warehouse instruction".
→	**661**	**AllocAcctIDSource**	N	
→	**573**	**MatchStatus**	N	
→	**366**	**AllocPrice**	N	Used when performing "executed price" vs. "average price" allocations (e.g. Japan). AllocAccount plus AllocPrice form a unique Allocs entry. Used in lieu of AllocAvgPx.
→	**80**	**AllocQty**	N	Conditionally required except when for AllocTransType="Cancel", or when AllocType= "Ready-To-Book" or "Warehouse instruction".
→	**467**	**IndividualAllocID**	N	
→	**81**	**ProcessCode**	N	
→	**989**	**SecondaryIndividualAllocID**	N	Can be used by an intermediary to specify an allocation ID assigned by the intermediary's system.
→	**1002**	**AllocMethod**	N	Specifies the method under which a trade quantity was allocated.
→	**993**	**AllocCustomerCapacity**	N	Can be used for granular reporting of separate allocation detail within a single trade report or allocation message.
→	**1047**	**AllocPositionEffect**	N	
→	**992**	**IndividualAllocType**	N	
→	**component block <NestedParties>**		N	Insert here the set of "Nested Parties" (firm identification "nested" within additional repeating group) fields defined in "Common Components of Application Messages"

© Copyright, 2006, FIX Protocol, Limited

				Used for NestedPartyRole=BrokerOfCredit, ClientID, Settlement location (PSET), etc. Note: this field can be used for settlement location (PSET) information.
→	**208**	**NotifyBrokerOfCredit**	N	
→	**209**	**AllocHandlInst**	N	
→	**161**	**AllocText**	N	Free format text field related to this AllocAccount
→	**360**	**EncodedAllocTextLen**	N	Must be set if EncodedAllocText field is specified and must immediately precede it.
→	**361**	**EncodedAllocText**	N	Encoded (non-ASCII characters) representation of the AllocText field in the encoded format specified via the MessageEncoding field.
→	**component block <CommissionData>**		N	Insert here the set of "CommissionData" fields defined in "Common Components of Application Messages"
→	**153**	**AllocAvgPx**	N	AvgPx for this AllocAccount. For F/X orders, should be the "all-in" rate (spot rate adjusted for forward points) for this allocation. For Fixed Income always express value as "percent of par".
→	**154**	**AllocNetMoney**	N	NetMoney for this AllocAccount ((AllocQty * AllocAvgPx) - Commission - sum of MiscFeeAmt + AccruedInterestAmt) if a Sell ((AllocQty * AllocAvgPx) + Commission + sum of MiscFeeAmt + AccruedInterestAmt) if a Buy
→	**119**	**SettlCurrAmt**	N	Replaced by AllocSettlCurrAmt
→	**737**	**AllocSettlCurrAmt**	N	AllocNetMoney in AllocSettlCurrency for this AllocAccount if AllocSettlCurrency is different from "overall" Currency
→	**120**	**SettlCurrency**	N	Replaced by AllocSettlCurrency SettlCurrency for this AllocAccount if different from "overall" Currency. Required if SettlCurrAmt is specified.
→	**736**	**AllocSettlCurrency**	N	AllocSettlCurrency for this AllocAccount if different from "overall" Currency. Required if AllocSettlCurrAmt is specified.
→	**155**	**SettlCurrFxRate**	N	Foreign exchange rate used to compute AllocSettlCurrAmt from Currency to AllocSettlCurrency
→	**156**	**SettlCurrFxRateCalc**	N	Specifies whether the SettlCurrFxRate should be multiplied or divided
→	**742**	**AllocAccruedInterestAmt**	N	Applicable for Convertible Bonds and fixed income
→	**741**	**AllocInterestAtMatu**	N	Applicable for securities that pay interest in lump-sum at

© Copyright, 2006, FIX Protocol, Limited

		rity			maturity
→	**136**	**NoMiscFees**		N	Required if any miscellaneous fees are reported. Indicates number of repeating entries. Repeating group. ** Nested Repeating Group follows **
→	→	**137**	**MiscFeeAmt**	N	Required if NoMiscFees > 0
→	→	**138**	**MiscFeeCurr**	N	
→	→	**139**	**MiscFeeType**	N	Required if NoMiscFees > 0
→	→	**891**	**MiscFeeBasis**	N	
→	**576**	**NoClearingInstructions**		N	** Nested Repeating Group follows **
→	→	**577**	**ClearingInstruction**	N	Required if NoClearingInstructions > 0
→	**635**	**ClearingFeeIndicator**		N	
→	**780**	**AllocSettlInstType**		N	Used to indicate whether settlement instructions are provided on this message, and if not, how they are to be derived. Absence of this field implies use of default instructions.
→	**component block <SettlInstructionsData>**			N	Insert here the set of "SettlInstructionsData" fields defined in "Common Components of Application Messages" Used to communicate settlement instructions for this AllocAccount detail. Required if AllocSettlInstType = 2 or 3.
819	AvgPxIndicator			N	Indicates if an allocation is to be average priced. Is also used to indicate if average price allocation group is complete or incomplete.
715	ClearingBusinessDate			N	Indicates Clearing Business Date for which transaction will be settled.
828	TrdType			N	Indicates Trade Type of Allocation.
829	TrdSubType			N	Indicates TradeSubType of Allocation. Necessary for defining groups.
582	CustOrderCapacity			N	Indicates CTI of original trade marked for allocation.
578	TradeInputSource			N	Indicates input source of original trade marked for allocation.
442	MultiLegReportingType			N	Indicates MultiLegReportType of original trade marked for allocation.
1011	MessageEventSource			N	Used to identify the event or source which gave rise to a message.
991	RndPx			N	Specifies the rounded price to quoted precision.
	StandardTrailer			Y	

© Copyright, 2006, FIX Protocol, Limited

Note: Req'd = "Y*" indicates that the field is not required for AllocTransType=Cancel

Note: Req'd = "Y**" indicates that the field is not required for AllocTransType=Cancel, nor is it required for AllocType=" Ready-To-Book " or AllocType="Warehouse instruction.

FIXML Definition for this message – see http://www.fixprotocol.org for details
Refer to FIXML element AllocInstrctn

© Copyright, 2006, FIX Protocol, Limited

Allocation Instruction Ack

In versions of FIX prior to version 4.4, this message was known as the Allocation ACK message.

The Allocation Instruction Ack message is used to acknowledge the receipt of and provide status for an Allocation Instruction message.

The status is indicated by the AllocStatus field as follows:

AllocStatus value	Description
3 = received, not yet processed	Used to acknowledge receipt of an Allocation Instruction message. This should always be followed by a second Allocation Instruction Ack of status 0, 1 or 2 as follows **or** an Allocation Report message.
0 = accepted	The Allocation Instruction has been validated and processed successfully.
1 = block level reject	The entire Allocation Instruction has been rejected. The AllocRejCode (88) field must be populated when performing a block level reject; this gives the reason for rejecting the Allocation Instruction.
2 = account level reject	The Allocation Instruction has been validated and one or more of the AllocAccount details in the NoAllocs repeating group has failed validation (e.g. account not found). In this case, it is possible (though not mandatory) to include a list of the AllocAccount details that failed validation together with reject reasons.

For an Allocation Instruction Ack message with AllocStatus of 'Accepted' in response to an Allocation Instruction with AllocType of ‘Calculated, it is recommended that the MatchStatus field be used to denote whether any financial details provided in the ‘Calculated’ Allocation Instruction were matched by the Respondent. If a match takes place and succeeds, then the match status will be '0-Compared and affirmed'. If the match takes place and fails, or no match takes place, then the match status will be '1-Uncompared or unaffirmed'.

Allocation Instruction Ack

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = P
70	AllocID	Y	
component block <Parties>		N	Insert here the set of "Parties" (firm identification) fields defined in "Common Components of Application Messages"
793	SecondaryAllocID	N	Optional second identifier for the allocation instruction being acknowledged (need not be unique)
75	TradeDate	N	
60	TransactTime	N	Date/Time Allocation Instruction Ack generated
87	AllocStatus	Y	Denotes the status of the allocation instruction; received (but not yet processed), rejected (at block or account level) or accepted (and processed).
88	AllocRejCode	N	Required for AllocStatus = 1 (block level reject) and for AllocStatus 2 (account level reject) if the individual accounts and reject reasons are not provided in this message

© Copyright, 2006, FIX Protocol, Limited

626	AllocType		N	
808	AllocIntermedReqType		N	Required if AllocType = 8 (Request to Intermediary) Indicates status that is requested to be transmitted to counterparty by the intermediary (i.e. clearing house)
573	MatchStatus		N	Denotes whether the financial details provided on the Allocation Instruction were successfully matched.
460	Product		N	
167	SecurityType		N	
58	Text		N	Can include explanation for AllocRejCode = 7 (other)
354	EncodedTextLen		N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText		N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
78	NoAllocs		N	This repeating group is optionally used for messages with AllocStatus = 2 (account level reject), AllocStatus = 0 (accepted), to provide details of the individual accounts that were accepted or rejected. In the case of a reject, the reasons for the rejection should be specified. This group should not be populated where AllocStatus has any other value. Indicates number of allocation groups to follow.
→	**79**	**AllocAccount**	N	Required if NoAllocs > 0. Must be first field in repeating group.
→	**661**	**AllocAcctIDSource**	N	
→	**366**	**AllocPrice**	N	Used when performing "executed price" vs. "average price" allocations (e.g. Japan). AllocAccount plus AllocPrice form a unique Allocs entry. Used in lieu of AllocAvgPx.
→	**1047**	**AllocPositionEffect**	N	
→	**467**	**IndividualAllocID**	N	
→	**776**	**IndividualAllocRejCode**	N	Required if NoAllocs > 0.
→	**component block <NestedParties>**		N	
→	**161**	**AllocText**	N	Free format text field related to this AllocAccount (can be used here to hold text relating to the rejection of this AllocAccount)
→	**360**	**EncodedAllocTextLen**	N	Must be set if EncodedAllocText field is specified and must immediately precede it.
→	**361**	**EncodedAllocText**	N	Encoded (non-ASCII characters) representation of the AllocText field in the encoded format specified via the

© Copyright, 2006, FIX Protocol, Limited

				MessageEncoding field.
→	**989**	**SecondaryIndividualAllocID**	N	Will allow the intermediary to specify an allocation ID generated by the system
→	**993**	**AllocCustomerCapacity**	N	Will allow for granular reporting of separate allocation detail within a single trade report or allocation message.
→	**992**	**IndividualAllocType**	N	Identifies whether the allocation is to be sub-allocated or allocated to a third party.
→	**80**	**AllocQty**	N	Quantity to be allocated to specific sub-account
	StandardTrailer		Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details
Refer to FIXML element AllocInstrctnAck

© Copyright, 2006, FIX Protocol, Limited

Allocation Instruction Alert

This message is used in a 3-party allocation model where notification of group creation and group updates to counterparites is needed. The mssage will also carry trade information that comprised the group to the counterparites.

Allocation Instruction Alert

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = BM
70	AllocID	Y	Unique identifier for this allocation instruction alert message
71	AllocTransType	Y	i.e. New, Cancel, Replace
626	AllocType	Y	Specifies the purpose or type of Allocation message
793	SecondaryAllocID	N	Optional second identifier for this allocation instruction (need not be unique)
72	RefAllocID	N	Required for AllocTransType = Replace or Cancel
796	AllocCancReplaceReason	N	Required for AllocTransType = Replace or Cancel Gives the reason for replacing or cancelling the allocation instruction
808	AllocIntermedReqType	N	Required if AllocType = 8 (Request to Intermediary) Indicates status that is requested to be transmitted to counterparty by the intermediary (i.e. clearing house)
196	AllocLinkID	N	Can be used to link two different Allocation messages (each with unique AllocID) together, i.e. for F/X "Netting" or "Swaps"
197	AllocLinkType	N	Can be used to link two different Allocation messages and identifies the type of link. Required if AllocLinkID is specified.
466	BookingRefID	N	Can be used with AllocType=" Ready-To-Book "
857	AllocNoOrdersType	N	Indicates how the orders being booked and allocated by this message are identified, i.e. by explicit definition in the NoOrders group or not.
73	NoOrders	N	Indicates number of orders to be combined for allocation. If order(s) were manually delivered set to 1 (one).Required when AllocNoOrdersType = 1
→	**11** **ClOrdID**	N	Order ID assigned by client if order(s) were electronically delivered and executed. If order(s) were manually delivered this field should contain string "MANUAL".Note where an order has undergone one or more cancel/replaces, this should be the ClOrdID of the most recent version of the order Required when NoOrders > 0 and must be the first repeating field in the group.

© Copyright, 2006, FIX Protocol, Limited

→	**37**	**OrderID**	N	
→	**198**	**SecondaryOrderID**	N	Can be used to provide order id used by exchange or executing system.
→	**526**	**SecondaryClOrdID**	N	
→	**66**	**ListID**	N	Required for List Orders.
→	**component block <NestedParties2>**		N	Insert here the set of "NestedParties2" fields defined in "Common Components of Application Messages" This is used to identify the executing broker for step in/give in trades
→	**38**	**OrderQty**	N	
→	**799**	**OrderAvgPx**	N	Average price for this order
→	**800**	**OrderBookingQty**	N	Quantity of this order that is being booked out by this message (will be equal to or less than this order's OrderQty) Note that the sum of the OrderBookingQty values in this repeating group must equal the total quantity being allocated (in Quantity (53) field)
124	NoExecs		N	Indicates number of individual execution repeating group entries to follow. Absence of this field indicates that no individual execution entries are included. Primarily used to support step-outs.
→	**32**	**LastQty**	N	Amount of quantity (e.g. number of shares) in individual execution. Required if NoExecs > 0
→	**17**	**ExecID**	N	
→	**527**	**SecondaryExecID**	N	
→	**31**	**LastPx**	N	Price of individual execution. Required if NoExecs > 0
→	**669**	**LastParPx**	N	Last price expressed in percent-of-par. Conditionally required for Fixed Income trades when LastPx is expressed in Yield, Spread, Discount or any other price type
→	**29**	**LastCapacity**	N	Used to identify whether the trade was executed on an agency or principal basis.
→	**1003**	**TradeID**	N	
→	**1041**	**FirmTradeID**	N	
570	PreviouslyReported		N	
700	ReversalIndicator		N	
574	MatchType		N	
54	Side		Y	
component block <Instrument>			Y	Insert here the set of "Instrument" (symbology) fields defined in "common components of application messages"

© Copyright, 2006, FIX Protocol, Limited

component block <InstrumentExtension>		N	Insert here the set of "InstrumentExtension" fields defined in "common components of application messages"
component block <FinancingDetails>		N	Insert here the set of "FinancingDetails" fields defined in "common components of application messages"
711	NoUnderlyings	N	Number of underlyings
→	**component block <UnderlyingInstrument>**	N	Must be provided if Number of underlyings > 0
555	NoLegs	N	Number of legs Identifies a Multi-leg Execution if present and non-zero.
→	**component block <InstrumentLeg>**	N	Must be provided if Number of legs > 0
53	Quantity	Y	Total quantity (e.g. number of shares) allocated to all accounts, or that is Ready-To-Book
854	QtyType	N	
30	LastMkt	N	Market of the executions.
229	TradeOriginationDate	N	
336	TradingSessionID	N	
625	TradingSessionSubID	N	
423	PriceType	N	
6	AvgPx	N	For F/X orders, should be the "all-in" rate (spot rate adjusted for forward points). For 3rd party allocations used to convey either basic price or averaged price Optional for average price allocations in the listed derivatives markets where the central counterparty calculates and manages average price across an allocation group.
860	AvgParPx	N	
component block <SpreadOrBenchmarkCurveData>		N	Insert here the set of "SpreadOrBenchmarkCurveData" fields defined in "common components of application messages"
15	Currency	N	Currency of AvgPx. Should be the currency of the local market or exchange where the trade was conducted.
74	AvgPxPrecision	N	Absence of this field indicates that default precision arranged by the broker/institution is to be used
component block <Parties>		N	Insert here the set of "Parties" (firm identification) fields defined in "common components of application messages"
75	TradeDate	Y	
60	TransactTime	N	Date/time when allocation is generated

© Copyright, 2006, FIX Protocol, Limited

63	SettlType	N	
64	SettlDate	N	Takes precedence over SettlType value and conditionally required/omitted for specific SettlType values.
775	BookingType	N	Method for booking. Used to provide notification that this is to be booked out as an OTC derivative (e.g. CFD or similar). Absence of this field implies regular booking.
381	GrossTradeAmt	N	Expressed in same currency as AvgPx. Sum of (AllocQty * AllocAvgPx or AllocPrice).
238	Concession	N	
237	TotalTakedown	N	
118	NetMoney	N	Expressed in same currency as AvgPx. Sum of AllocNetMoney.
77	PositionEffect	N	
754	AutoAcceptIndicator	N	Indicates if Allocation has been automatically accepted on behalf of the Carry Firm by the Clearing House
58	Text	N	
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
157	NumDaysInterest	N	Applicable for Convertible Bonds and fixed income
158	AccruedInterestRate	N	Applicable for Convertible Bonds and fixed income
159	AccruedInterestAmt	N	Applicable for Convertible Bonds and fixed income (REMOVED FROM THIS LOCATION AS OF FIX 4.4, REPLACED BY AllocAccruedInterest)
540	TotalAccruedInterestAmt	N	(Deprecated) use AccruedInterestAmt Sum of AccruedInterestAmt within repeating group.
738	InterestAtMaturity	N	
920	EndAccruedInterestAmt	N	For repurchase agreements the accrued interest on termination.
921	StartCash	N	For repurchase agreements the start (dirty) cash consideration
922	EndCash	N	For repurchase agreements the end (dirty) cash consideration
650	LegalConfirm	N	
component block <Stipulations>		N	
component block <YieldData>		N	
component block <PositionAmountData>		N	Insert here here the set of "Position Amount Data" fields defined in "Common Components of Application

© Copyright, 2006, FIX Protocol, Limited

				Messages"
892	TotNoAllocs		N	Indicates total number of allocation groups (used to support fragmentation). Must equal the sum of all NoAllocs values across all message fragments making up this allocation instruction. Only required where message has been fragmented.
893	LastFragment		N	Indicates whether this is the last fragment in a sequence of message fragments. Only required where message has been fragmented.
78	NoAllocs		N	Conditionally required except when AllocTransType = Cancel, or when AllocType = Ready-to-book or Warehouse instruction
→	**79**	**AllocAccount**	N	May be the same value as BrokerOfCredit if ProcessCode is step-out or soft-dollar step-out and Institution does not wish to disclose individual account breakdowns to the ExecBroker. Required if NoAllocs > 0. Must be first field in repeating group. Conditionally required except when for AllocTransType="Cancel", or when AllocType= "Ready-To-Book" or "Warehouse instruction".
→	**661**	**AllocAcctIDSource**	N	
→	**573**	**MatchStatus**	N	
→	**366**	**AllocPrice**	N	Used when performing "executed price" vs. "average price" allocations (e.g. Japan). AllocAccount plus AllocPrice form a unique Allocs entry. Used in lieu of AllocAvgPx.
→	**80**	**AllocQty**	N	Conditionally required except when for AllocTransType="Cancel", or when AllocType= "Ready-To-Book" or "Warehouse instruction".
→	**467**	**IndividualAllocID**	N	
→	**81**	**ProcessCode**	N	
→	**989**	**SecondaryIndividualAllocID**	N	Can be used by an intermediary to specify an allocation ID assigned by the intermediary's system.
→	**1002**	**AllocMethod**	N	Specifies the method under which a trade quantity was allocated.
→	**993**	**AllocCustomerCapacity**	N	Can be used for granular reporting of separate allocation detail within a single trade report or allocation message.
→	**1047**	**AllocPositionEffect**	N	
→	**992**	**IndividualAllocType**	N	
→	**component block <NestedParties>**		N	Insert here the set of "Nested Parties" (firm identification "nested" within additional repeating group) fields defined in "Common Components of Application Messages" Used for NestedPartyRole=BrokerOfCredit, ClientID, Settlement location (PSET), etc.

© Copyright, 2006, FIX Protocol, Limited

				Note: this field can be used for settlement location (PSET) information.
→	**208**	**NotifyBrokerOfCredit**	N	
→	**209**	**AllocHandlInst**	N	
→	**161**	**AllocText**	N	Free format text field related to this AllocAccount
→	**360**	**EncodedAllocTextLen**	N	Must be set if EncodedAllocText field is specified and must immediately precede it.
→	**361**	**EncodedAllocText**	N	Encoded (non-ASCII characters) representation of the AllocText field in the encoded format specified via the MessageEncoding field.
→	**component block <CommissionData>**		N	Insert here the set of "CommissionData" fields defined in "Common Components of Application Messages"
→	**153**	**AllocAvgPx**	N	AvgPx for this AllocAccount. For F/X orders, should be the "all-in" rate (spot rate adjusted for forward points) for this allocation. For Fixed Income always express value as "percent of par".
→	**154**	**AllocNetMoney**	N	NetMoney for this AllocAccount ((AllocQty * AllocAvgPx) - Commission - sum of MiscFeeAmt + AccruedInterestAmt) if a Sell ((AllocQty * AllocAvgPx) + Commission + sum of MiscFeeAmt + AccruedInterestAmt) if a Buy
→	**119**	**SettlCurrAmt**	N	Replaced by AllocSettlCurrAmt
→	**737**	**AllocSettlCurrAmt**	N	AllocNetMoney in AllocSettlCurrency for this AllocAccount if AllocSettlCurrency is different from "overall" Currency
→	**120**	**SettlCurrency**	N	Replaced by AllocSettlCurrency SettlCurrency for this AllocAccount if different from "overall" Currency. Required if SettlCurrAmt is specified.
→	**736**	**AllocSettlCurrency**	N	AllocSettlCurrency for this AllocAccount if different from "overall" Currency. Required if AllocSettlCurrAmt is specified.
→	**155**	**SettlCurrFxRate**	N	Foreign exchange rate used to compute AllocSettlCurrAmt from Currency to AllocSettlCurrency
→	**156**	**SettlCurrFxRateCalc**	N	Specifies whether the SettlCurrFxRate should be multiplied or divided
→	**742**	**AllocAccruedInterestAmt**	N	Applicable for Convertible Bonds and fixed income
→	**741**	**AllocInterestAtMaturity**	N	Applicable for securities that pay interest in lump-sum at maturity
→	**136**	**NoMiscFees**	N	Required if any miscellaneous fees are reported.

© Copyright, 2006, FIX Protocol, Limited

					Indicates number of repeating entries. Repeating group. ** Nested Repeating Group follows **
→	→	**137**	**MiscFeeAmt**	N	Required if NoMiscFees > 0
→	→	**138**	**MiscFeeCurr**	N	
→	→	**139**	**MiscFeeType**	N	Required if NoMiscFees > 0
→	→	**891**	**MiscFeeBasis**	N	
→	**576**	**NoClearingInstructions**		N	** Nested Repeating Group follows **
→	→	**577**	**ClearingInstruction**	N	Required if NoClearingInstructions > 0
→	**635**	**ClearingFeeIndicator**		N	
→	**780**	**AllocSettlInstType**		N	Used to indicate whether settlement instructions are provided on this message, and if not, how they are to be derived. Absence of this field implies use of default instructions.
→	**component block <SettlInstructionsData>**			N	Insert here the set of "SettlInstructionsData" fields defined in "Common Components of Application Messages" Used to communicate settlement instructions for this AllocAccount detail. Required if AllocSettlInstType = 2 or 3.
819	AvgPxIndicator			N	Indicates if an allocation is to be average priced. Is also used to indicate if average price allocation group is complete or incomplete.
715	ClearingBusinessDate			N	Indicates Clearing Business Date for which transaction will be settled.
828	TrdType			N	Indicates Trade Type of Allocation.
829	TrdSubType			N	Indicates TradeSubType of Allocation. Necessary for defining groups.
582	CustOrderCapacity			N	Indicates CTI of original trade marked for allocation.
578	TradeInputSource			N	Indicates input source of original trade marked for allocation.
442	MultiLegReportingType			N	Indicates MultiLegReportType of original trade marked for allocation.
1011	MessageEventSource			N	Used to identify the event or source which gave rise to a message.
991	RndPx			N	Specifies the rounded price to quoted precision.
	StandardTrailer			Y	

© Copyright, 2006, FIX Protocol, Limited

Allocation Report (aka Allocation Claim)

Sent from sell-side to buy-side, sell-side to 3rd-party or 3rd-party to buy-side, the Allocation Report (Claim) provides account breakdown of an order or set of orders plus any additional follow-up front-office information developed post-trade during the trade allocation, matching and calculation phase. In versions of FIX prior to version 4.4, this functionality was provided through the Allocation message. Depending on the needs of the market and the timing of "confirmed" status, the role of Allocation Report can be taken over in whole or in part by the Confirmation message.

Note the response to the Allocation Report message is the Allocation Report Ack message. In versions of FIX prior to version 4.4, the Allocation ACK served this purpose.

An Allocation Report message can be submitted with AllocReportType of

- Sellside Calculated Using Preliminary (includes Misc Fees, Accrued Interest and Net Money)
- Sellside Calculated Without Preliminary (includes Misc Fees, Accrued Interest and Net Money). (AllocType=" Sellside Initiated"), i.e. where the allocations have been provided via some other mechanism or agreed earlier in the order process.
- Warehouse recap – sent unsolicited by sellside, used to communicate confirmation and current status of any warehoused position in a particular stock (see Volume 7 – PRODUCT: EQUITIES for specific usage guidance on this topic)

Settlement instructions are supported on the Allocation Report message to allow the Respondent (sell-side party or carry firm) to send an override of its own instructions to the Initiator.

General guidelines applicable to this message:

- AllocReportID should be unique for all Allocation Report messages.
- To reject an Allocation Report message, an Allocation Report Ack with AllocStatus 'Block level reject' or 'Account level reject' should be used. Use of 'Block level reject' means the entire message has been rejected (e.g. net money mismatch). 'Account level reject' is used when the block level matches successfully but one or more (or all) of the constituent account level details fails validation (e.g. account not found, incorrect MiscFees). In the latter case, the rejecting party can (optionally) notify the instructing party of those allocation details that are being rejected by listing the offending account numbers in the Allocation Instruction Ack message.
- A rejected Allocation Report must be resolved out-of-band.
- Where settling in markets where multiple alternative settlement locations exist, it is recommended that the settlement location (equivalent to ISO15022 'PSET' field) be identified on each allocation detail within the NoAllocs repeating group. A nested parties component block is provided which can be used for this purpose.

The allocation message contains repeating fields for each order, sub-account and individual execution. The repeating fields are shown in the message definition below in typeface ***Bold-Italic*** and indented with the → symbol. The field's relative position within the repeating group in the message is important. For example, each instance of allocation must be in the order as shown in the message definition below.

- The number of sub-account instances is indicated in NoAllocs.
- Multiple orders can be combined for allocation or for AllocType=" Ready-To-Book" or AllocType = "Warehouse instruction". Note that combined orders must refer to the same instrument and have the same trade date, settlement date and side. The identification of the orders to be combined can be achieved in one of two ways:

© Copyright, 2006, FIX Protocol, Limited

- By identifying the number of orders in the NoOrders field and each individual order in the OrderID fields. The AllocNoOrdersType field is used to denote that this is happening and takes value "1=Explicit list provided". If any orders were handled outside FIX, the ClOrdID must be set to 'MANUAL'. Regardless of whether the orders were handled within or outside FIX, the order quantity and average price must also be specified for each order. This is to assist in validating the message and, for manual orders, to help identify the correct orders to book.
- By stating that an unspecified group of orders is to be combined. The NoOrders field in this case is left blank. The AllocNoOrdersType field is set to "0=Not specified" to specify that this is happening. Note use of this approach is only recommended where either the number of orders being booked is extremely large or some kind of aggregation rule is being used.

- Multiple executions can be combined for allocation by identifying the number of executions in the NoExecs field and each individual execution in the ExecID fields. Combined executions must refer to the same instrument, trade date, settlement date and side.

Allocation Report (aka Allocation Claim)

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = AS
755	AllocReportID	Y	Unique identifier for this message
70	AllocID	N	
71	AllocTransType	Y	i.e. New, Cancel, Replace
795	AllocReportRefID	N	Required for AllocTransType = Replace or Cancel
796	AllocCancReplaceReason	N	Required for AllocTransType = Replace or Cancel Gives the reason for replacing or cancelling the allocation report
793	SecondaryAllocID	N	Optional second identifier for this allocation instruction (need not be unique)
794	AllocReportType	Y	Specifies the purpose or type of Allocation Report message
87	AllocStatus	Y	
88	AllocRejCode	N	Required for AllocStatus = 1 (rejected)
72	RefAllocID	N	Required for AllocTransType = Replace or Cancel
808	AllocIntermedReqType	N	Required if AllocReportType = 8 (Request to Intermediary) Indicates status that is requested to be transmitted to counterparty by the intermediary (i.e. clearing house)
196	AllocLinkID	N	Can be used to link two different Allocation messages (each with unique AllocID) together, i.e. for F/X "Netting" or "Swaps"
197	AllocLinkType	N	Can be used to link two different Allocation messages and identifies the type of link. Required if AllocLinkID is specified.
466	BookingRefID	N	

© Copyright, 2006, FIX Protocol, Limited

715	ClearingBusinessDate		N	Indicates Clearing Business Date for which transaction will be settled.
828	TrdType		N	Indicates Trade Type of Allocation.
829	TrdSubType		N	Indicates TradeSubType of Allocation. Necessary for defining groups.
442	MultiLegReportingType		N	Indicates MultiLegReportType of original trade marked for allocation.
582	CustOrderCapacity		N	Indicates CTI of original trade marked for allocation.
578	TradeInputSource		N	Indicates input source of original trade marked for allocation.
991	RndPx		N	Specifies the rounded price to quoted precision.
1011	MessageEventSource		N	Used to identify the event or source which gave rise to a message.
579	TradeInputDevice		N	Specific device number, terminal number or station where trade was entered
819	AvgPxIndicator		N	Indicates if an allocation is to be average priced. Is also used to indicate if average price allocation group is complete or incomplete.
857	AllocNoOrdersType		N	Indicates how the orders being booked and allocated by this message are identified, i.e. by explicit definition in the NoOrders group or not.
73	NoOrders		N	Indicates number of orders to be combined for allocation. If order(s) were manually delivered set to 1 (one).Required when AllocNoOrdersType = 1
→	**11**	**ClOrdID**	N	Order ID assigned by client if order(s) were electronically delivered and executed. If order(s) were manually delivered this field should contain string "MANUAL".Note where an order has undergone one or more cancel/replaces, this should be the ClOrdID of the most recent version of the order Required when NoOrders > 0 and must be the first repeating field in the group.
→	**37**	**OrderID**	N	
→	**198**	**SecondaryOrderID**	N	Can be used to provide order id used by exchange or executing system.
→	**526**	**SecondaryClOrdID**	N	
→	**66**	**ListID**	N	Required for List Orders.
→	**component block <NestedParties2>**		N	Insert here the set of "NestedParties2" fields defined in "Common Components of Application Messages" This is used to identify the executing broker for step in/give in trades
→	**38**	**OrderQty**	N	

© Copyright, 2006, FIX Protocol, Limited

→	**799**	**OrderAvgPx**	N	Average price for this order
→	**800**	**OrderBookingQty**	N	Quantity of this order that is being booked out by this message (will be equal to or less than this order's OrderQty) Note that the sum of the OrderBookingQty values in this repeating group must equal the total quantity being allocated (in Quantity (53) field)
124	NoExecs		N	Indicates number of individual execution repeating group entries to follow. Absence of this field indicates that no individual execution entries are included. Primarily used to support step-outs.
→	**32**	**LastQty**	N	Amount of quantity (e.g. number of shares) in individual execution. Required if NoExecs > 0
→	**17**	**ExecID**	N	
→	**527**	**SecondaryExecID**	N	
→	**31**	**LastPx**	N	Price of individual execution. Required if NoExecs > 0
→	**669**	**LastParPx**	N	Last price expressed in percent-of-par. Conditionally required for Fixed Income trades when LastPx is expressed in Yield, Spread, Discount or any other price type
→	**29**	**LastCapacity**	N	Used to identify whether the trade was executed on an agency or principal basis.
→	**1003**	**TradeID**	N	
→	**1041**	**FirmTradeID**	N	
570	PreviouslyReported		N	
700	ReversalIndicator		N	
574	MatchType		N	
54	Side		Y	
component block <Instrument>			Y	Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages"
component block <InstrumentExtension>			N	Insert here the set of "InstrumentExtension" fields defined in "Common Components of Application Messages"
component block <FinancingDetails>			N	Insert here the set of "FinancingDetails" fields defined in "Common Components of Application Messages"
711	NoUnderlyings		N	Number of underlyings
→	**component block <UnderlyingInstrument>**		N	Must be provided if Number of underlyings > 0
555	NoLegs		N	Number of legs Identifies a Multi-leg Execution if present and non-zero.

© Copyright, 2006, FIX Protocol, Limited

→	**component block <InstrumentLeg>**	N	Must be provided if Number of legs > 0
53	Quantity	Y	Total quantity (e.g. number of shares) allocated to all accounts, or that is Ready-To-Book
854	QtyType	N	
30	LastMkt	N	Market of the executions.
229	TradeOriginationDate	N	
336	TradingSessionID	N	
625	TradingSessionSubID	N	
423	PriceType	N	
6	AvgPx	Y	For F/X orders, should be the "all-in" rate (spot rate adjusted for forward points).
860	AvgParPx	N	
component block <SpreadOrBenchmarkCurveData>		N	Insert here the set of "SpreadOrBenchmarkCurveData" fields defined in "Common Components of Application Messages"
15	Currency	N	Currency of AvgPx. Should be the currency of the local market or exchange where the trade was conducted.
74	AvgPxPrecision	N	Absence of this field indicates that default precision arranged by the broker/institution is to be used
component block <Parties>		N	Insert here the set of "Parties" (firm identification) fields defined in "Common Components of Application Messages"
75	TradeDate	Y	
60	TransactTime	N	Date/time when allocation is generated
63	SettlType	N	
64	SettlDate	N	Takes precedence over SettlType value and conditionally required/omitted for specific SettlType values.
775	BookingType	N	Method for booking. Used to provide notification that this is to be booked out as an OTC derivative (e.g. CFD or similar). Absence of this field implies regular booking.
381	GrossTradeAmt	N	Expressed in same currency as AvgPx. Sum of (AllocQty * AllocAvgPx or AllocPrice).
238	Concession	N	
237	TotalTakedown	N	
118	NetMoney	N	Expressed in same currency as AvgPx. Sum of AllocNetMoney.
77	PositionEffect	N	
754	AutoAcceptIndicator	N	Indicates if Allocation has been automatically accepted on behalf of the Carry Firm by the Clearing House

© Copyright, 2006, FIX Protocol, Limited

58	Text		N	
354	EncodedTextLen		N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText		N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
157	NumDaysInterest		N	Applicable for Convertible Bonds and fixed income
158	AccruedInterestRate		N	Applicable for Convertible Bonds and fixed income
159	AccruedInterestAmt		N	Sum of AllocAccruedInterestAmt within repeating group.
540	TotalAccruedInterestAmt		N	(Deprecated in FIX.4.4)
738	InterestAtMaturity		N	
920	EndAccruedInterestAmt		N	For repurchase agreements the accrued interest on termination.
921	StartCash		N	For repurchase agreements the start (dirty) cash consideration
922	EndCash		N	For repurchase agreements the end (dirty) cash consideration
650	LegalConfirm		N	
component block <Stipulations>			N	
component block <YieldData>			N	
component block <PositionAmountData>			N	Insert here here the set of "Position Amount Data" fields defined in "Common Components of Application Messages"
892	TotNoAllocs		N	Indicates total number of allocation groups (used to support fragmentation). Must equal the sum of all NoAllocs values across all message fragments making up this allocation instruction. Only required where message has been fragmented.
893	LastFragment		N	Indicates whether this is the last fragment in a sequence of message fragments. Only required where message has been fragmented.
78	NoAllocs		N	Conditionally required except when AllocTransType = Cancel, or when AllocType = Ready-to-book or Warehouse instruction
→	**79**	**AllocAccount**	N	May be the same value as BrokerOfCredit if ProcessCode is step-out or soft-dollar step-out and Institution does not wish to disclose individual account breakdowns to the ExecBroker. Required if NoAllocs > 0. Must be first field in repeating group. Conditionally required except when for AllocTransType="Cancel", or when AllocType= "Ready-

© Copyright, 2006, FIX Protocol, Limited

				To-Book" or "Warehouse instruction".
→	**661**	**AllocAcctIDSource**	N	
→	**573**	**MatchStatus**	N	
→	**366**	**AllocPrice**	N	Used when performing "executed price" vs. "average price" allocations (e.g. Japan). AllocAccount plus AllocPrice form a unique Allocs entry. Used in lieu of AllocAvgPx.
→	**80**	**AllocQty**	N	Conditionally required except when for AllocTransType="Cancel", or when AllocType= "Ready-To-Book" or "Warehouse instruction".
→	**467**	**IndividualAllocID**	N	
→	**81**	**ProcessCode**	N	
→	**989**	**SecondaryIndividualAllocID**	N	Can be used by an intermediary to specify an allocation ID assigned by the intermediary's system.
→	**1002**	**AllocMethod**	N	Specifies the method under which a trade quantity was allocated.
→	**993**	**AllocCustomerCapacity**	N	Can be used for granular reporting of separate allocation detail within a single trade report or allocation message.
→	**1047**	**AllocPositionEffect**	N	
→	**992**	**IndividualAllocType**	N	
→	**component block <NestedParties>**		N	Insert here the set of "Nested Parties" (firm identification "nested" within additional repeating group) fields defined in "Common Components of Application Messages" Used for NestedPartyRole=BrokerOfCredit, ClientID, Settlement location (PSET), etc. Note: this field can be used for settlement location (PSET) information.
→	**208**	**NotifyBrokerOfCredit**	N	
→	**209**	**AllocHandlInst**	N	
→	**161**	**AllocText**	N	Free format text field related to this AllocAccount
→	**360**	**EncodedAllocTextLen**	N	Must be set if EncodedAllocText field is specified and must immediately precede it.
→	**361**	**EncodedAllocText**	N	Encoded (non-ASCII characters) representation of the AllocText field in the encoded format specified via the MessageEncoding field.
→	**component block <CommissionData>**		N	Insert here the set of "CommissionData" fields defined in "Common Components of Application Messages"
→	**153**	**AllocAvgPx**	N	AvgPx for this AllocAccount. For F/X orders, should be the "all-in" rate (spot rate adjusted for forward points) for this allocation. For Fixed Income always express value as "percent of par".

© Copyright, 2006, FIX Protocol, Limited

→	**154**	**AllocNetMoney**		N	NetMoney for this AllocAccount ((AllocQty * AllocAvgPx) - Commission - sum of MiscFeeAmt + AccruedInterestAmt) if a Sell ((AllocQty * AllocAvgPx) + Commission + sum of MiscFeeAmt + AccruedInterestAmt) if a Buy
→	**119**	**SettlCurrAmt**		N	Replaced by AllocSettlCurrAmt
→	**737**	**AllocSettlCurrAmt**		N	AllocNetMoney in AllocSettlCurrency for this AllocAccount if AllocSettlCurrency is different from "overall" Currency
→	**120**	**SettlCurrency**		N	Replaced by AllocSettlCurrency SettlCurrency for this AllocAccount if different from "overall" Currency. Required if SettlCurrAmt is specified.
→	**736**	**AllocSettlCurrency**		N	AllocSettlCurrency for this AllocAccount if different from "overall" Currency. Required if AllocSettlCurrAmt is specified.
→	**155**	**SettlCurrFxRate**		N	Foreign exchange rate used to compute AllocSettlCurrAmt from Currency to AllocSettlCurrency
→	**156**	**SettlCurrFxRateCalc**		N	Specifies whether the SettlCurrFxRate should be multiplied or divided
→	**742**	**AllocAccruedInterestAmt**		N	Applicable for Convertible Bonds and fixed income
→	**741**	**AllocInterestAtMaturity**		N	Applicable for securities that pay interest in lump-sum at maturity
→	**136**	**NoMiscFees**		N	Required if any miscellaneous fees are reported. Indicates number of repeating entries. Repeating group. ** Nested Repeating Group follows **
→	→	**137**	**MiscFeeAmt**	N	Required if NoMiscFees > 0
→	→	**138**	**MiscFeeCurr**	N	
→	→	**139**	**MiscFeeType**	N	Required if NoMiscFees > 0
→	→	**891**	**MiscFeeBasis**	N	
→	**576**	**NoClearingInstructions**		N	** Nested Repeating Group follows **
→	→	**577**	**ClearingInstruction**	N	Required if NoClearingInstructions > 0
→	**635**	**ClearingFeeIndicator**		N	
→	**780**	**AllocSettlInstType**		N	Used to indicate whether settlement instructions are provided on this message, and if not, how they are to be derived. Absence of this field implies use of default instructions.
→	**component block**			N	Insert here the set of "SettlInstructionsData" fields

© Copyright, 2006, FIX Protocol, Limited

	<SettlInstructionsData>		defined in "Common Components of Application Messages" Used to communicate settlement instructions for this AllocAccount detail. Required if AllocSettlInstType = 2 or 3.
	StandardTrailer	Y	

Note: Req'd = "Y*" indicates that the field is not required for AllocTransType=Cancel

Note: Req'd = "Y**" indicates that the field is not required for AllocTransType=Cancel, nor is it required for AllocReportType="Warehouse recap".

FIXML Definition for this message – see http://www.fixprotocol.org for details
Refer to FIXML element AllocRpt

© Copyright, 2006, FIX Protocol, Limited

Allocation Report Ack (aka Allocation Claim Ack)

The Allocation Report Ack message is used to acknowledge the receipt of and provide status for an Allocation Report message.

It is possible that multiple Allocation Report Ack messages can be generated for a single Allocation Report message to acknowledge the receipt and then to detail the acceptance or rejection of the Allocation Report message.

It is recommended, when appropriate, that the MatchStatus field be used in the Allocation Report Ack to denote whether any financial details provided in the Allocation Report with AllocStatus of 'Accepted' were matched by the Initiator. If a match takes place and succeeds, then the match status will be '0-Compared and affirmed'. If the match takes place and fails, or no match takes place, then the match status will be '1-Uncompared or unaffirmed'.

Allocation Report Ack (aka Allocation Claim Ack)-

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = AT
755	AllocReportID	Y	
70	AllocID	Y	
715	ClearingBusinessDate	N	Indicates Clearing Business Date for which transaction will be settled.
819	AvgPxIndicator	N	Indicates if an allocation is to be average priced. Is also used to indicate if average price allocation group is complete or incomplete.
53	Quantity	N	
71	AllocTransType	N	
component block <Parties>		N	Insert here the set of "Parties" (firm identification) fields defined in "Common Components of Application Messages"
793	SecondaryAllocID	N	Optional second identifier for the allocation report being acknowledged (need not be unique)
75	TradeDate	N	
60	TransactTime	N	Date/Time Allocation Report Ack generated
87	AllocStatus	N	Denotes the status of the allocation report; received (but not yet processed), rejected (at block or account level) or accepted (and processed). AllocStatus will be conditionally required in a 2-party model when used by a counterparty to convey a change in status. It will be optional in a 3-party model in which only the central counterparty may issue the status of an allocation
88	AllocRejCode	N	Required for AllocStatus = 1 (block level reject) and for AllocStatus 2 (account level reject) if the individual accounts and reject reasons are not provided in this message

© Copyright, 2006, FIX Protocol, Limited

794	AllocReportType		N	
808	AllocIntermedReqType		N	Required if AllocReportType = 8 (Request to Intermediary) Indicates status that is requested to be transmitted to counterparty by the intermediary (i.e. clearing house)
573	MatchStatus		N	Denotes whether the financial details provided on the Allocation Report were successfully matched.
460	Product		N	
167	SecurityType		N	
58	Text		N	Can include explanation for AllocRejCode = 7 (other)
354	EncodedTextLen		N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText		N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
78	NoAllocs		N	This repeating group is optionally used for messages with AllocStatus = 2 (account level reject), AllocStatus = 0 (accepted), to provide details of the individual accounts that were accepted or rejected. In the case of a reject, the reasons for the rejection should be specified. This group should not be populated where AllocStatus has any other value. Indicates number of allocation groups to follow.
→	**79**	**AllocAccount**	N	Required if NoAllocs > 0. Must be first field in repeating group.
→	**661**	**AllocAcctIDSource**	N	
→	**366**	**AllocPrice**	N	Used when performing "executed price" vs. "average price" allocations (e.g. Japan). AllocAccount plus AllocPrice form a unique Allocs entry. Used in lieu of AllocAvgPx.
→	**1047**	**AllocPositionEffect**	N	
→	**467**	**IndividualAllocID**	N	
→	**776**	**IndividualAllocRejCode**	N	Required if NoAllocs > 0.
→	**component block <NestedParties>**		N	
→	**161**	**AllocText**	N	Free format text field related to this AllocAccount (can be used here to hold text relating to the rejection of this AllocAccount)
→	**360**	**EncodedAllocTextLen**	N	Must be set if EncodedAllocText field is specified and must immediately precede it.
→	**361**	**EncodedAllocText**	N	Encoded (non-ASCII characters) representation of the

© Copyright, 2006, FIX Protocol, Limited

				AllocText field in the encoded format specified via the MessageEncoding field.
→	**989**	**SecondaryIndividualAllocID**	N	Will allow the intermediary to specify an allocation ID generated by the system
→	**993**	**AllocCustomerCapacity**	N	Will allow for granular reporting of separate allocation detail within a single trade report or allocation message.
→	**992**	**IndividualAllocType**	N	Identifies whether the allocation is to be sub-allocated or allocated to a third party.
→	**80**	**AllocQty**	N	Quantity to be allocated to specific sub-account
	StandardTrailer		Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details

Refer to FIXML element AllocRptAck

© Copyright, 2006, FIX Protocol, Limited

Example Usage of Allocations and Ready-To-Book Messaging

The Allocation Instruction message provides the the ability to specify how an order or set of orders should be subdivided amongst **one or more** accounts.

Allocation is typically communicated **Post-Trade** (after fills have been received and processed). It can, however, also be communicated **Pre-Trade** (at the time the order is being placed) to specify the account(s) and their respective order quantities which make up the order. This is a regulatory requirement in certain markets and for certain types of securities.

The Allocation Instruction message can also be sent by the buyside firm after execution to indicate to the sellside firm that one or a combined (aggregated) set of orders are "Ready-To-Book" without specifying individual account breakdowns. This can be used to trigger post-trade allocation, matching, and settlement processing via other channels (e.g. post-trade industry utilities). See "Ready-To-Book Processing" subsection below.

Please refer to the overview section at the start of this category for more details.

Ready-To-Book Processing:

The Ready-To-Book capability of the Allocation Instruction message is designed to provide a clean interface between the "trading" and "booking" spaces. This allows buyside firms to both trigger and provide suitable references which can be passed down to assist in the matching process within industry utilities (e.g. Virual Matching Utilities) or bilaterally with their sellside counterparts. Bookable units can be single fills, combinations of fills, single orders, or groups of orders for the same security, side, settlement date, etc. Automated booking instructions can be communicated either pre-trade or post-trade.

Booking instructions can be communicated **Pre-Trade** (at the time the order is being placed) to convey that as soon as the order is filled it can be considered by the acceptor as ready for booking (e.g. in particular when there is no additional quantity behind). This can be accomplished by specifying DayBookingInst="auto" on the new order message. In addition, BookingUnit and PreallocMethod can be used to fine tune the automated booking procedure to be taken.

Booking instructions can also be communicated **Post-Trade** (after fills have been received and processed) to signal that a particular order is now ready for booking or to signal that a set of orders for the same security, side, settlement date, etc., are to be aggregated as single booking unit which is now ready for booking.

- Buyside sends a New Order request message
- Sellside sends Execution Report messages for the “New” and resulting fills.
- Sellside sends Execution Report messages with OrdStatus = “Filled" or "Done For Day".
- Buyside sends Allocation Instruction message with AllocType="Ready-To-Book "
  - The order id information from the order and execution report processing is referenced within NoOrders repeating group
  - Note that the NoAllocs repeating group (group of AllocAccount) is not required for Ready-To-Book

Example flow for AllocType="Ready-To-Book " post-trade processing which books out a single order:

Initiator	→	New Order-Single (OrderQty=35000, ClOrdID=123)	Respondent
	←	Execution Report (ExecType = “0” [New]) (ClOrdID=123, OrderID=ABC)	

© Copyright, 2006, FIX Protocol, Limited

	←	Execution Report (ExecType = “F”) [Trade] (ClOrdID=123, OrderID=ABC) (optional Execution Report (ExecType = “3”) [Done for day] (ClOrdID=123, OrderID=ABC) (receive either OrdStatus="Filled" or "Done For Day") and buyside ready for sellside to initiate booking	
	→	Allocation Instruction (AllocType="Ready-To-Book ", NoOrders=1, OrderID=ABC, ClOrdID=123)	
	←	Allocation Instruction Ack (AllocStatus=Received Not Yet Processed")	
	←	Allocation Instruction Ack (AllocStatus="Accepted")	
	Post-Trade Matching and Allocation Processing occurs (e.g. via an industry utility)		

Example flow for AllocType="Ready-To-Book " post-trade processing which books out a number of orders as a single block:

Initiator	→	New Order-Single (OrderQty=35000, ClOrdID=123, Symbol=IBM, Side=1)	Respondent
	←	Execution Report (ExecType = “0” [New]) (ClOrdID=123, OrderID=ABC)	
	←	Execution Report (ExecType = “F”) [Trade] (ClOrdID=123, OrderID=ABC) (optional Execution Report (ExecType = “3”) [Done for day] (ClOrdID=123, OrderID=ABC)	
	→	New Order-Single (OrderQty=2000, ClOrdID=456, Symbol=IBM, Side=1)	
	←	Execution Report (ExecType = “0” [New]) (ClOrdID=456, OrderID=DEF)	
	←	Execution Report (ExecType = “F”) [Trade] (ClOrdID=456, OrderID=DEF) (optional Execution Report (ExecType = “3”) [Done for day] (ClOrdID=456, OrderID=DEF) (receive either OrdStatus="Filled" or "Done For Day") for all orders to be combined and buyside ready for sellside to initiate booking	
	→	Allocation Instruction (AllocType="Ready-To-Book ", NoOrders=2, OrderID=ABC, ClOrdID=123, OrderID=DEF, ClOrdID=456)	
	←	Allocation Instruction Ack (AllocStatus=Received Not Yet Processed")	
	←	Allocation Instruction Ack (AllocStatus="Accepted")	

© Copyright, 2006, FIX Protocol, Limited

	Post-Trade Matching and Allocation Processing occurs (e.g. via an industry utility)	

Pre-Trade Allocation

There are two models for pre-trade allocation in FIX

- Allocating using details on the New Order message (Pre-allocated order).
- Allocating at the time of placing the order using a separate allocation instruction message (Pre-trade allocation).

Example flow for Pre-allocated order

Initiator	→	New Order-Single (OrderQty=35000, NoAllocs=2, AllocID=50, AllocAccount=ACCT1, AllocQty=10000, AllocAccount=ACCT2, AllocQty=25000)	Respondent
	←	Execution Report (ExecType = “0” [New]	
	←	Execution Report (ExecType = “F”) [Trade] (optional Execution Report (ExecType = “3”) [Done for day]	
	→	These three messages are optional – used for buyside ready to book notification, e.g. to agree average price, quantity to book or any order combination requirements. Allocation Instruction (AllocType=" Preliminary", AllocAccounts provided without MiscFees or NetMoney)	
	←	Allocation Instruction Ack (AllocStatus=Received Not Yet Processed)	
	←	Allocation Instruction Ack (AllocStatus=Accepted)	
	←	These three messages are optional – used for sellside notification. Allocation Report (AllocReportType="Sellside Calculated using Preliminary", AllocStatus=Accepted)	
	→	Allocation Report Ack (AllocStatus=Received Not Yet Processed)	
	→	Allocation Report Ack (AllocStatus=Accepted or Rejected)	

Note this same flow can be used for other kinds of New Order message, e.g. New Order List.

© Copyright, 2006, FIX Protocol, Limited

Example flow for rejection of Pre-allocated order

There are two ways to reject the allocation details on a pre-allocated order. The first is simply to reject the entire order:

Initiator	→	New Order-Single (OrderQty=35000, NoAllocs=2, AllocID = 100, AllocAccount=ACCT1, AllocQty=10000, AllocAccount=ACCT2, AllocQty=25000)	Respondent
	←	Execution Report (ExecType = “8” [Rejected]	

The second is to send an Allocation Instruction Ack message:

Initiator	→	New Order-Single (OrderQty=35000, NoAllocs=2, AllocID = 100, AllocAccount=ACCT1, AllocQty=10000, AllocAccount=ACCT2, AllocQty=25000)	Respondent
	←	Execution Report (ExecType = “0” [New]	
	←	Execution Report (ExecType = “F”) [Trade] (optional Execution Report (ExecType = “3”) [Done for day]	
	←	Allocation Instruction Ack (AllocID = 100, AllocStatus=Received)	
	←	Allocation Instruction Ack (AllocID = 100, AllocStatus=Block level reject or Account level reject)	

Example flow for Pre-Trade Allocation (using Allocation Instruction message)

Initiator	→	New Order-Single (OrderQty=35000)	Respondent
	←	Execution Report (ExecType = “0” [New]	
	→	Allocation Instruction (AllocType=" Preliminary", AllocAccounts provided without MiscFees or NetMoney)	
	←	Allocation Instruction Ack (AllocStatus=Received Not Yet Processed)	
	←	Allocation Instruction Ack (AllocStatus=Accepted)	
	←	Execution Report (ExecType = “F”) [Trade] (optional Execution Report (ExecType = “3”) [Done for day]	

Note the Allocation Instruction can be sent any time after the New Order message, at the same time or even before (though only if the sellside is able to queue the message until the order arrives).

The message initiator may optionally send an Allocation Instruction message of type 'Ready to book' (if this is provided, the respondent should respond by accepting or rejecting the message before proceeding to the next step). The purpose of this message is to confirm the average price and quantity to allocate (especially if multiple orders are to be combined for booking).

© Copyright, 2006, FIX Protocol, Limited

Message flows for rejection of allocation details when communicated pre-trade are the same as for post-trade allocations and are covered in the next section.

Post-Trade Allocation

Post trade allocations can be computed via one of two methods:

1. **Using Average Price:** Each AllocAccount has a single AllocAvgPx (e.g. US and European) (see examples 1-1, 2-1, 3-1)
2. **Using Executed Price:** Combination of each AllocAccount **and** AllocPrice (unique LastPx) (e.g. Japan) (see examples 1-2, 2-2, 3-2)

Post-Trade Allocation supports three different message flows:

1. Buyside initiated with buyside-computed Misc Fees and NetMoney (see examples 1-1 and 1-2)

The typical flow for US domestic trading (withNetMoney and MiscFees provided by the buyside) is as follows:

Initiator	→	Allocation Instruction (AllocType=" Calculated")	Respondent
	←	Allocation Instruction Ack (AllocStatus=Received Not Yet Processed)	
	←	Allocation Instruction Ack (AllocStatus=Accepted)	

2. Buyside-initiated with Misc Fee computation by the sellside firm (see examples 2-1 and 2-2)

The typical flow for international equity trading is as follows:

Initiator	→	Allocation Instruction (AllocType=" Preliminary", AllocAccounts provided without MiscFees or NetMoney)	Respondent
	←	Allocation Instruction Ack (AllocStatus=Received Not Yet Processed)	
	←	Allocation Instruction ACK (AllocStatus=Accepted)	

3. Sellside-initiated (see examples 3-1 and 3-2)

The typical flow for sellside-initiated (unsolicited by the buyside) is as follows:

Initiator	←	Allocation Report (AllocReportType="Sellside Calculated without Preliminary"	Respondent
	→	Allocation Report Ack (AllocStatus=Received Not Yet Processed)	
	→	Allocation Report Ack (AllocStatus=Accepted)	

Note in all three of these flows, the following should be noted:

- The buyside may send fee and expense information (MiscFees) on the allocation instruction, or may elect not to do this. Either way, the sellside does not respond back with fee and expense information on the Allocation Instruction Ack; such information is transmitted via the Confirmation message. This is different to the flows used in earlier versions of FIX where the sellside was able to respond using an allocation message populated with the MiscFees.

© Copyright, 2006, FIX Protocol, Limited

- Settlement instructions have been removed from the flow (see Settlement Instructions section for further details). However, there is a Parties block in the NoAllocs group of the Allocation Instruction message which can be used to transmit settlement location information (equivalent to ISO15022 PSET field).

Rejection Scenarios

To reject an entire Allocation Instruction, use an Allocation Instruction Ack of status 'Block level reject'.

Initiator			Respondent
	→	Allocation Instruction (AllocTransType = New)	
	←	Allocation Instruction Ack (AllocStatus=Received Not Yet Processed)	
	←	Allocation Instruction Ack (AllocStatus=Block level reject)	
	→	The corrected allocation details are communicated using a new Allocation Instruction Allocation Instruction (AllocTransType = New)	
	←	Allocation Instruction Ack (AllocStatus=Received Not Yet Processed)	
	←	Allocation Instruction Ack (AllocStatus=Accepted)	

To reject one or more of the allocation account details in an Allocation Instruction, use an Allocation Instruction Ack of status 'Account level reject'.

Initiator			Respondent
	→	Allocation Instruction (AllocTransType = New)	
	←	Allocation Instruction Ack (AllocStatus=Received Not Yet Processed)	
	←	Allocation Instruction Ack (AllocStatus=Account level reject)	
		The corrected allocation details are communicated either by using a 'replace' Allocation Instruction	
	→	Allocation Instruction (AllocTransType = Replace)	
	←	Allocation Instruction Ack (AllocStatus=Received Not Yet Processed)	
	←	Allocation Instruction Ack (AllocStatus=Accepted)	
		OR by cancelling the original Allocation Instruction and submitting a new one	
	→	Allocation Instruction (AllocTransType = Cancel)	
	←	Allocation Instruction Ack (AllocStatus=Received Not Yet Processed)	
	←	Allocation Instruction Ack (AllocStatus=Accepted)	
	→	Allocation Instruction (AllocTransType = New)	
	←	Allocation Instruction Ack (AllocStatus=Received Not Yet	

© Copyright, 2006, FIX Protocol, Limited

		Processed)
	←	Allocation Instruction Ack (AllocStatus=Accepted)

Example 1-1: Buyside-initiated flow withbuyside calculated NetMoney and MiscFees, using Average Price (all AllocAccounts with same AvgPx)

Initiator	→	New Order-Single	Respondent
	←	Execution Report (ExecType = "0" [New]	
	←	Execution Report (ExecType = "F") [Trade] (optional Execution Report (ExecType = "3") [Done for day]	
Allocate			
	→	Allocation Instruction (AllocType=" Calculated")	
	←	Allocation Instruction Ack (AllocStatus=Received Not Yet Processed)	
	←	Allocation Instruction Ack (AllocStatus=Accepted , Block level reject or Account level reject)	

Symbol	B/S	Mkt	**Order Message**			**Execution Rpt Messages**		
			Account	OrdID	ClOrdID	ExecID	LastPx	LastQty
IBM	Buy	N		520	20	300	100.00	3000
						301	100.25	1000
						302	100.00	3000
						303	100.50	2000

Allocation Instruction Msg ↓

Symbol	B/S	Mkt	Order section			***AvgPx***	Repeating fields				Repeating fields	
			ID	OrdID	ClOrdID		ExecID	LastPx	LastQty	***AllocAccount***	AllocQty	Commission
IBM	Buy	N	999	520	20	***100.1389***	300	100.00	3000	***F1***	3000	150
							301	100.25	1000	***F2***	3000	150
							302	100.00	3000	***F3***	3000	150
							303	100.50	2000			

© Copyright, 2006, FIX Protocol, Limited

Example 1-2: Buyside-initiated flow withbuyside calculated NetMoney and MiscFees, using Executed Price

Initiator	→	New Order-Single	Respondent
	←	Execution Report (ExecType = “0” [New]	
	←	Execution Report (ExecType = “F”) [Trade] (optional Execution Report (ExecType = “3”) [Done for day]	
Allocate			
	→	Allocation Instruction (AllocType=" Calculated")	
	←	Allocation Instruction Ack (AllocStatus=Received Not Yet Processed)	
	←	Allocation Instruction Ack (AllocStatus=Accepted, Block level reject or Account level reject)	

Symbol	B/S	Mkt	**Order Message**			**Execution Rpt Messages**		
			Account	OrdID	ClOrdID	ExecID	LastPx	LastQty
IBM	Buy	N		520	20	300	100.00	3000
						301	100.25	1000
						302	100.00	3000
						303	100.50	2000

Allocation Instruction Msg ↓

Symbol	B/S	Mkt	Order section			Repeating fields			Repeating fields			
			ID	OrdID	ClOrdID	ExecID	***LastPx***	LastQty	***AllocAccount***	***AllocPrice***	AllocQty	Commission
IBM	Buy	N	999	520	20	300	**100.00**	3000	***F1***	***100.00***	2000	100
						301	**100.25**	1000	***F1***	***100.25***	1000	50
						302	**100.00**	3000	***F2***	***100.00***	2000	100
						303	**100.50**	2000	***F2***	***100.50***	1000	50
									F3	***100.00***	2000	100
									F3	***100.50***	1000	50

© Copyright, 2006, FIX Protocol, Limited

Example 2-1: Buyside-initiated flow without buyside calculated NetMoney and MiscFees, using Average Price (all AllocAccounts with same AvgPx)

Initiator	→	New Order-Single	Respondent
	←	Execution Report (ExecType = "0" [New]	
	←	Execution Report (ExecType = "F") [Trade] (optional Execution Report (ExecType = "3") [Done for day]	
Allocate			
	→	Allocation Instruction (AllocType=" Preliminary", AllocAccounts provided without MiscFees or NetMoney)	
	←	Allocation Instruction Ack (AllocStatus=Received Not Yet Processed)	
	←	Allocation Instruction Ack (AllocStatus=Accepted, Block level reject or Account level reject)	

Symbol	B/S	Mkt	**Order Message**			**Execution Rpt Messages**		
			Account	OrdID	ClOrdID	ExecID	LastPx	LastQty
HNS.L	Buy	L		520	20	300	3.9809	100000
						301	3.9809	25000

Allocation Instruction Msg ↓

Symbol	B/S	Mkt	Order section			Repeating fields			Repeating fields			
			ID	OrdID	ClOrdID	ExecID	LastPx	LastQty	***AllocAccount***	AllocQty	Commission	Repeating fields (NoMiscFees=2)
												MiscFeeType / MiscFeeAmt
HNS.L	Buy	L	999	520	20	300	3.9809	100000				
						301	3.9809	25000	***F1***	42200	335.988	5 / 830.9699
												6 / .25
									F2	82800	652.937	5 / 1648.0926
												6 / .25

Example 2-2: Buyside-initiated flow with MiscFee computation, using Executed Price

Initiator	→	New Order-Single	Respondent
	←	Execution Report (ExecType = "0" [New]	

© Copyright, 2006, FIX Protocol, Limited

	←	Execution Report (ExecType = “F”) [Trade] (optional Execution Report (ExecType = “3”) [Done for day]	
Allocate			
	→	Allocation Instruction (AllocType=" Preliminary", AllocAccounts provided without MiscFees or NetMoney)	
	←	Allocation Instruction Ack (AllocStatus=Received Not Yet Processed)	
	←	Allocation Instruction Ack (AllocStatus=Accepted, Block level reject or Account level reject)	

Symbol	B/S	Mkt	**Order Message**			**Execution Rpt Messages**		
			Account	OrdID	ClOrdID	ExecID	LastPx	LastQty
1234	Buy	T		520	20	300	1300	3000
						301	1313	1000
						302	1300	3000
						303	1320	2000

Allocation Instruction Msg ↓

Symbol	B/S	Mkt	Order section			Repeating fields			Repeating fields					
			ID	OrdID	ClOrdID	ExecID	***LastPx***	LastQty	***AllocAccount***	***AllocPrice***	AllocQty	Commission	Repeating fields (NoMiscFees=1)	
1234	Buy	T	999	520	20	300	***1300***	3000					MiscFeeType	MiscFeeAmt
						301	***1313***	1000	***F1***	***1300***	2000	25061	9	1253
						302	***1300***	3000	***F1***	***1313***	1000	12656	9	632
						303	***1320***	2000	***F2***	***1300***	2000	25058	9	1252
									F2	***1320***	1000	12722	9	636
									F3	***1300***	2000	25058	9	1252
									F3	***1320***	1000	12722	9	636

Note: This example’s values are for a Japanese Domestic Trade, and for actual use, you need to set any other required fields.

© Copyright, 2006, FIX Protocol, Limited

Example 3-1: Sellside-initiated flow, single Account, using Average Price

Initiator	→	New Order-Single	Respondent
	←	Execution Report (ExecType = "0" [New]	
	←	Execution Report (ExecType = "F") [Trade] (optional Execution Report (ExecType = "3") [Done for day]	
Allocate			
			Commission/ Fee Calc
	←	Allocation Report (AllocType="Sellside Calculated without Preliminary", optional MiscFees and NetMoney provided by AllocAccount)	
	→	Allocation Report Ack (AllocStatus=Received Not Yet Processed)	
	→	Allocation Report Ack (AllocStatus=Accepted , Block level reject or Account level reject)	

Symbol	B/S	Mkt	**Order Message**			**Execution Rpt Messages**		
			Account	OrdID	ClOrdID	ExecID	LastPx	LastQty
IBM	Buy	N	***F1***	520	20	300	1300	3000
						301	1313	1000
						302	1300	3000
						303	1320	2000

Allocation Report Msg ↓

Symbol	B/S	Mkt	Order section			***AvgPx***	Repeating fields				Repeating fields	
			ID	OrdID	ClOrdID		ExecID	LastPx	LastQty	***AllocAccount***	AllocQty	Commission
IBM	Buy	N	999	520	20	***1305.889***	300	1300	3000	***F1***	9000	113277
							301	1313	1000			
							302	1300	3000			
							303	1320	2000			

© Copyright, 2006, FIX Protocol, Limited

Example 3-2: Sellside-initiated flow, single Account, using Executed Price

Initiator	→	New Order-Single	Respondent
	←	Execution Report (ExecType = “0” [New]	
	←	Execution Report (ExecType = “F”) [Trade] (optional Execution Report (ExecType = “3”) [Done for day]	
Allocate			
			Commission/ Fee Calc
	←	Allocation Report (AllocType="Sellside Calculated without Preliminary", optional MiscFees and NetMoney provided by AllocAccount)	
	→	Allocation Report Ack (AllocStatus=Received Not Yet Processed)	
	→	Allocation Report Ack (AllocStatus=Accepted , Block level reject or Account level reject)	

Symbol	B/S	Mkt	Order Message			Execution Rpt Messages		
			Account	OrdID	ClOrdID	ExecID	LastPx	LastQty
1234	Buy	T	***F1***	520	20	300	1300	3000
						301	1313	1000
						302	1300	3000
						303	1320	2000

Allocation Report Msg ↓

Symbol	B/S	Mkt	Order section			Repeating fields			Repeating fields					
			ID	OrdID	ClOrdID	ExecID	***LastPx***	LastQty	***AllocAccount***	***AllocPrice***	AllocQty	Commission	Repeating fields (NoMiscFees=1)	
1234	Buy	T	999	520	20	300	***1300***	3000					MiscFeeType	MiscFeeAmt
						301	***1313***	1000	***F1***	***1300***	6000	61441	9	3072
						302	***1300***	3000	***F1***	***1313***	1000	10342	9	517
						303	***1320***	2000	***F1***	***1320***	2000	20796	9	1039

Note: This example’s values are for a Japanese Domestic Trade, and for actual use, you need to set any other required fields.

© Copyright, 2006, FIX Protocol, Limited

CATEGORY: CONFIRMATION

Overview

This section provides a overview on how the FIX protolcol can be used to support the process of Confirmation together with the appropriate responses.

A similar overview is also provided at start of the Category on FIX Allocations. These two overviews provide a summary on how FIX messaging can be used for booking, allocation and confirmation up to the the start of settlement processing.

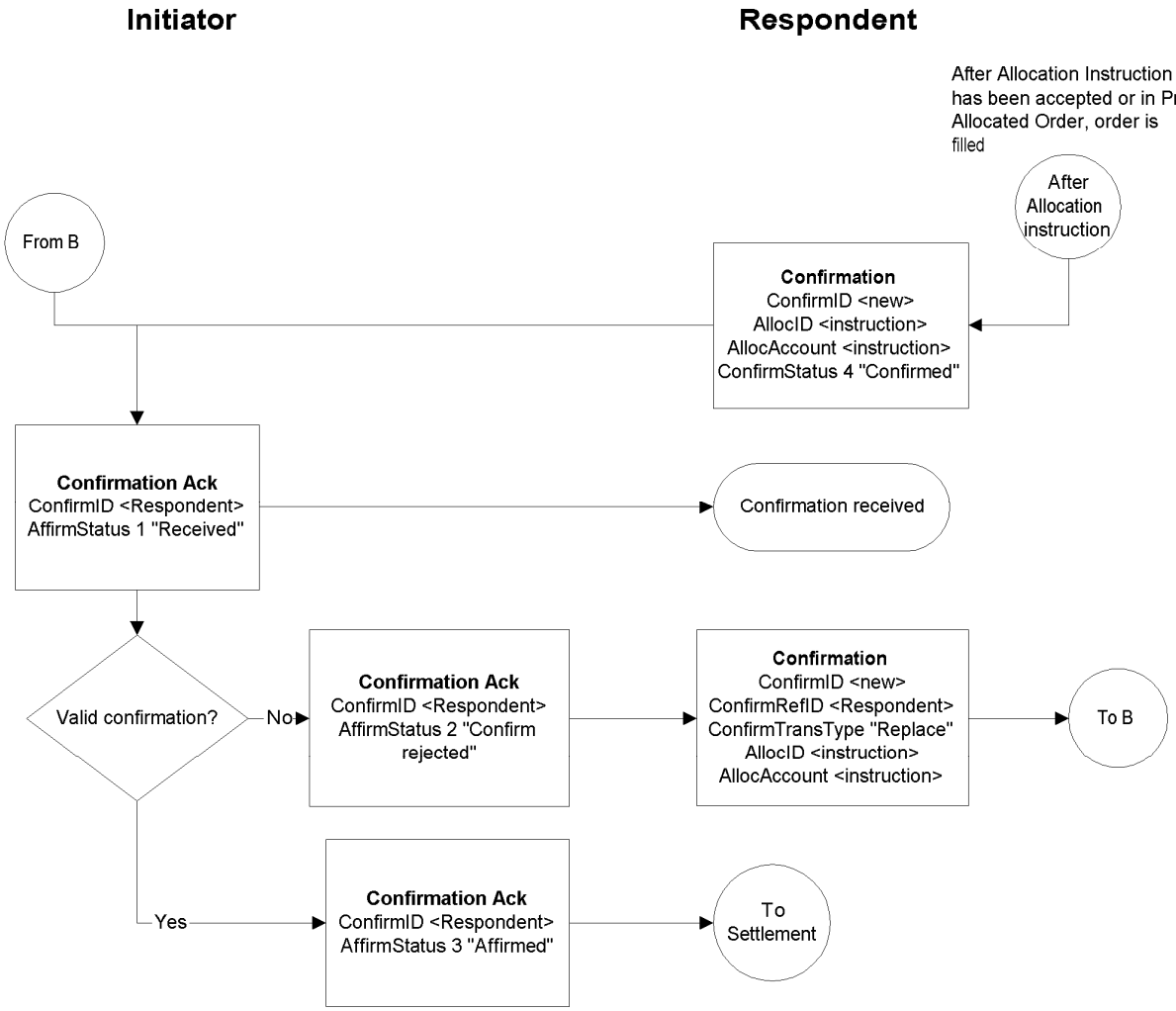
Further detail and additional optional flows for Confirmation are included in the Example Usage at the end of this category.

Confirmation via FIX

Confirmation processing within FIX takes place at an allocation account level, i.e. a single message for every account. Thus if the Allocation Instruction message was used to split a block into multiple accounts, then multiple FIX Confirmation messages would result. The Confirmation message can also be used as a trade status message in response to a Confirmation Request message.

© Copyright, 2006, FIX Protocol, Limited

Confirmation



It is always the Respondent that generates the FIX Confirmation message.

In the Pre-trade allocation scenario the Initiator would send the allocation instructions, after placing the order but before the Execution Report message indicated that the trade is completed, to the Respondent using a separate message - the Allocation Instruction message type. This scenario consists of the following steps:

- Respondent performs the calculation (i.e. net monies, etc.), and generate a FIX Confirmation message for each Allocation/Account within the validated Allocation Instruction.
- The Initiator can reject the validated/calculated confirmation, e.g. due to differences in calculations of net money, gross amounts, etc., for each of the allocated accounts.
- The Respondent can either:
    - Send a Confirmation message of type “cancel” followed by one of type “new”

      or

    - Send a Confirmation message of type “replace”
- Alternatively the Initiator can acknowledge back to the Respondent that the Confirmation is affirmed.
- At this point the message flow can be considered completed and all required information should have been collected and validated in order to proceeed to settlement processing.

© Copyright, 2006, FIX Protocol, Limited

The Confirmation message can also be used as a trade status message that allows the Respondent to report to the Initiator the status of each of the allocation or account as they work on it. The Initiator can request a booking status on an allocation or account using the optional Confirmation Request. This request could be raised when a confirmation has not been received for an allocation or account within an Allocation Instruction ("block") message.

© Copyright, 2006, FIX Protocol, Limited

Confirmation

The Confirmation messages are used to provide individual trade level confirmations from the sell side to the buy side. In versions of FIX prior to version 4.4, this role was performed by the allocation message. Unlike the allocation message, the confirmation message operates at an allocation account (trade) level rather than block level, allowing for the affirmation or rejection of individual confirmations.

This message is also used to report back, confirm or exception, the booking status of each allocation instance. When the buy-side, in response, "affirms" with the ConfirmationAck message, the trade is ready to settle.

Because each message reports the details of a single "ticket", Account names, fees, net money, and settlement information are reported using fields designated for single-account trades.

Every Confirmation message has a unique ConfirmID. It is recommended that the sellside system trade reference be used as ConfirmID where possible, in order to enable the ConfirmID to be used as a mutually understood trade reference (e.g. for use in manual conversations regarding specific trades).

The capacity or capacities of the firm executing the order or orders covered by this confirmation is represented in a repeating group. This is to support confirmations covering orders executed under more than one capacity (e.g. a mixture of agency and principal execution). The OrderCapacityQty field (inside this repeating group) gives the quantity executed under each OrderCapacity. The sum of the OrderCapacityQty values must equal the confirmation's AllocQty (field 80).

Confirmation

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = AK
664	ConfirmID	Y	Unique ID for this message
772	ConfirmRefID	N	Mandatory if ConfirmTransType is Replace or Cancel
859	ConfirmReqID	N	Only used when this message is used to respond to a confirmation request (to which this ID refers)
666	ConfirmTransType	Y	New, Cancel or Replace
773	ConfirmType	Y	Denotes whether this message represents a confirmation or a trade status message
797	CopyMsgIndicator	N	Denotes whether or not this message represents copy confirmation (or status message) Absence of this field indicates message is not a drop copy.
650	LegalConfirm	N	Denotes whether this message represents the legally binding confirmation Absence of this field indicates message is not a legal confirm.
665	ConfirmStatus	Y	
component block <Parties>		N	Insert here the set of "Parties" (firm identification) fields defined in "Common Components of Application Messages" Required for fixed income Also to be used in associated with ProcessCode for

© Copyright, 2006, FIX Protocol, Limited

				broker of credit (e.g. for directed brokerage trades) Also to be used to specify party-specific regulatory details (e.g. full legal name of contracting legal entity, registered address, regulatory status, any registration details)
73	NoOrders		N	Indicates number of orders to be combined for allocation. If order(s) were manually delivered set to 1 (one).Required when AllocNoOrdersType = 1
→	**11**	**ClOrdID**	N	Order ID assigned by client if order(s) were electronically delivered and executed. If order(s) were manually delivered this field should contain string "MANUAL".Note where an order has undergone one or more cancel/replaces, this should be the ClOrdID of the most recent version of the order Required when NoOrders > 0 and must be the first repeating field in the group.
→	**37**	**OrderID**	N	
→	**198**	**SecondaryOrderID**	N	Can be used to provide order id used by exchange or executing system.
→	**526**	**SecondaryClOrdID**	N	
→	**66**	**ListID**	N	Required for List Orders.
→	**component block <NestedParties2>**		N	Insert here the set of "NestedParties2" fields defined in "Common Components of Application Messages" This is used to identify the executing broker for step in/give in trades
→	**38**	**OrderQty**	N	
→	**799**	**OrderAvgPx**	N	Average price for this order
→	**800**	**OrderBookingQty**	N	Quantity of this order that is being booked out by this message (will be equal to or less than this order's OrderQty) Note that the sum of the OrderBookingQty values in this repeating group must equal the total quantity being allocated (in Quantity (53) field)
70	AllocID		N	Used to refer to an earlier Allocation Instruction.
793	SecondaryAllocID		N	Used to refer to an earlier Allocation Instruction via its secondary identifier
467	IndividualAllocID		N	Used to refer to an allocation account within an earlier Allocation Instruction.
60	TransactTime		Y	Represents the time this message was generated
75	TradeDate		Y	
component block <TrdRegTimestamps>			N	Time of last execution being confirmed by this message
component block <Instrument>			Y	Insert here the set of "Instrument" (symbology) fields

© Copyright, 2006, FIX Protocol, Limited

				defined in "Common Components of Application Messages"
component block <InstrumentExtension>			N	Insert here the set of "InstrumentExtension" fields defined in "Common Components of Application Messages"
component block <FinancingDetails>			N	Insert here the set of "FinancingDetails" fields defined in "Common Components of Application Messages"
711	NoUnderlyings		N	Number of underlyings
→	**component block <UnderlyingInstrument>**		N	Must be provided if Number of underlyings > 0
555	NoLegs		N	Number of legs Identifies a Multi-leg Execution if present and non-zero.
→	**component block <InstrumentLeg>**		N	Must be provided if Number of legs > 0
component block <YieldData>			N	If traded on Yield, price must be calculated "to worst" and the <Yield> component block must specify how calculated, redemption date and price (if not par). If traded on Price, the <Yield> component block must specify how calculated - "Worst", and include redemptiondate and price (if not par).
80	AllocQty		Y	The quantity being confirmed by this message (this is at a trade level, not block or order level)
854	QtyType		N	
54	Side		Y	
15	Currency		N	
30	LastMkt		N	
862	NoCapacities		Y	Indicates number of repeating entries. ** Nested Repeating Group follows **
→	**528**	**OrderCapacity**	Y	Specifies the capacity of the firm executing the order(s)
→	**529**	**OrderRestrictions**	N	
→	**863**	**OrderCapacityQty**	Y	The quantity that was executed under this capacity (e.g. quantity executed as agent, as principal etc.). Sum of OrderCapacityQty values must equal this message's AllocQty.
79	AllocAccount		Y	Account number for the trade being confirmed by this message
661	AllocAcctIDSource		N	
798	AllocAccountType		N	
6	AvgPx		Y	Gross price for the trade being confirmed Always expressed in percent-of-par for Fixed Income
74	AvgPxPrecision		N	Absence of this field indicates that default precision

© Copyright, 2006, FIX Protocol, Limited

			arranged by the broker/institution is to be used
423	PriceType	N	Price type for the AvgPx field
860	AvgParPx	N	
component block <SpreadOrBenchmarkCurveData>		N	Insert here the set of "SpreadOrBenchmarkCurveData" fields defined in "Common Components of Application Messages"
861	ReportedPx	N	Reported price (may be different to AvgPx in the event of a marked-up or marked-down principal trade)
58	Text	N	
354	EncodedTextLen	N	
355	EncodedText	N	
81	ProcessCode	N	Used to identify whether the trade was a soft dollar trade, step in/out etc. Broker of credit, where relevant, can be specified using the Parties nested block above.
381	GrossTradeAmt	Y	
157	NumDaysInterest	N	
230	ExDate	N	Optional "next coupon date" for Fixed Income
158	AccruedInterestRate	N	
159	AccruedInterestAmt	N	Required for Fixed Income products that trade with accrued interest
738	InterestAtMaturity	N	Required for Fixed Income products that pay lump sum interest at maturity
920	EndAccruedInterestAmt	N	For repurchase agreements the accrued interest on termination.
921	StartCash	N	For repurchase agreements the start (dirty) cash consideration
922	EndCash	N	For repurchase agreements the end (dirty) cash consideration
238	Concession	N	
237	TotalTakedown	N	
118	NetMoney	Y	
890	MaturityNetMoney	N	Net Money at maturity if Zero Coupon and maturity value is different from par value
119	SettlCurrAmt	N	
120	SettlCurrency	N	
155	SettlCurrFxRate	N	
156	SettlCurrFxRateCalc	N	
63	SettlType	N	
64	SettlDate	N	

© Copyright, 2006, FIX Protocol, Limited

component block <SettlInstructionsData>			N	Insert here the set of "SettlInstructionsData" fields defined in "Common Components of Application Messages" Used to communicate settlement instructions for this Confirmation.
component block <CommissionData>			N	
858	SharedCommission		N	Used to identify any commission shared with a third party (e.g. directed brokerage)
component block <Stipulations>			N	
136	NoMiscFees		N	Required if any miscellaneous fees are reported. Indicates number of repeating entries. Repeating group. ** Nested Repeating Group follows **
→	**137**	**MiscFeeAmt**	N	Required if NoMiscFees > 0
→	**138**	**MiscFeeCurr**	N	
→	**139**	**MiscFeeType**	N	Required if NoMiscFees > 0
→	**891**	**MiscFeeBasis**	N	
	StandardTrailer		Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details
Refer to the FIXML element Cnfm

© Copyright, 2006, FIX Protocol, Limited

Confirmation Ack (aka Affirmation)

The Confirmation Ack (aka Affirmation) message is used to respond to a Confirmation message.

Confirmation Ack (aka Affirmation)

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = AU
664	ConfirmID	Y	
75	TradeDate	Y	
60	TransactTime	Y	Date/Time Allocation Instruction Ack generated
940	AffirmStatus	Y	
774	ConfirmRejReason	N	Required for ConfirmStatus = 1 (rejected)
573	MatchStatus	N	Denotes whether the financial details provided on the Confirmation were successfully matched.
58	Text	N	Can include explanation for AllocRejCode = 7 (other)
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
	StandardTrailer	Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details
Refer to the FIXML element CnfmAck

© Copyright, 2006, FIX Protocol, Limited

Confirmation Request

The Confirmation Request message is used to request a Confirmation message.

Confirmation Request

Tag	*FieldName*		*Req'd*	*Comments*
	StandardHeader		Y	MsgType = BH
859	ConfirmReqID		Y	Unique identifier for this message
773	ConfirmType		Y	Denotes whether this message is being used to request a confirmation or a trade status message
73	NoOrders		N	Indicates number of orders to be combined for allocation. If order(s) were manually delivered set to 1 (one).Required when AllocNoOrdersType = 1
→	**11**	**ClOrdID**	N	Order ID assigned by client if order(s) were electronically delivered and executed. If order(s) were manually delivered this field should contain string "MANUAL".Note where an order has undergone one or more cancel/replaces, this should be the ClOrdID of the most recent version of the order Required when NoOrders > 0 and must be the first repeating field in the group.
→	**37**	**OrderID**	N	
→	**198**	**SecondaryOrderID**	N	Can be used to provide order id used by exchange or executing system.
→	**526**	**SecondaryClOrdID**	N	
→	**66**	**ListID**	N	Required for List Orders.
→	**component block <NestedParties2>**		N	Insert here the set of "NestedParties2" fields defined in "Common Components of Application Messages" This is used to identify the executing broker for step in/give in trades
→	**38**	**OrderQty**	N	
→	**799**	**OrderAvgPx**	N	Average price for this order
→	**800**	**OrderBookingQty**	N	Quantity of this order that is being booked out by this message (will be equal to or less than this order's OrderQty) Note that the sum of the OrderBookingQty values in this repeating group must equal the total quantity being allocated (in Quantity (53) field)
70	AllocID		N	Used to refer to an earlier Allocation Instruction.
793	SecondaryAllocID		N	Used to refer to an earlier Allocation Instruction via its secondary identifier
467	IndividualAllocID		N	Used to refer to an allocation account within an earlier

© Copyright, 2006, FIX Protocol, Limited

			Allocation Instruction.
60	TransactTime	Y	Represents the time this message was generated
79	AllocAccount	N	Account number for the trade being confirmed by this message
661	AllocAcctIDSource	N	
798	AllocAccountType	N	
58	Text	N	
354	EncodedTextLen	N	
355	EncodedText	N	
	StandardTrailer	Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details
Refer to the FIXML element CnfmReq

Example usage of Confirmations

The Confirmation message can be used in three ways:

1. As an electronic trade confirmation message (which requires affirmation or rejection from the recipient).
2. As an electronic copy of a confirmation to be sent to a third party (which does not require affirmation or rejection).
3. As a status message, to provide information regarding the state of an allocation level trade.

In all three cases, the final (successful) status of the Confirmation is "Affirmed" which can be taken to mean that the trade is ready to settle.

Affirmed Confirmation

Model 1 – Electronic Trade Confirmation Message

Initiator	←	Confirmation, (ConfirmType = "2" [Confirm], CopyMsgIndicator = "N", ConfirmTransType = "New", ConfirmStatus = "Confirmed"	Respondent
	→	Confirmation Ack (AffirmStatus = "Received")	
	→	Confirmation Ack (AffirmStatus = "Affirmed"	

Model 2 – Copy Confirmation Message

Initiator or 3rd party	←	Confirmation, (ConfirmType = "2" [Confirm], CopyMsgIndicator = "Y", ConfirmTransType = "New",	Respondent

© Copyright, 2006, FIX Protocol, Limited

		ConfirmStatus = "Confirmed"	
	→	Confirmation Ack (AffirmStatus = "Received")	

Where a copy confirm is to be sent to another interested third party (or even as a copy to the buyside), and the buyside is using Model 1 for electronic trade confirmation, the copy confirm should not be sent until the main confirm has been affirmed. In other words, the Model 2 flow should simply follow on from the end of the Model 1 flow. Note that the recipient of the copy confirm does not have the power to affirm or reject the message for business reasons (though a more technical level rejection is possible e.g. in the event of system failure and should read to mean message transmission/processing failure rather than rejection of content).

Model 3 – Trade Status Message

Initiator	←	Confirmation, (ConfirmType = "1" [Status], ConfirmTransType = "New", ConfirmStatus = "Confirmed", "Mismatched account", "Missing SSI" etc.	Respondent
	→	Confirmation Ack (AffirmStatus = "Received")	

This flow is used to report back, affirm or exception the booking status of each trade. A typical example of this flow would be where an order had been booked out and allocated successfully, but on attempting to enrich the trades with details required to produce a confirmation, some key information (e.g. settlement instructions) may be missing or incomplete. Should the sellside wish to notify the buyside of this electronically, this is the flow to use.

In all three cases, the sellside can cancel or replace the Confirmation message using ConfirmTransType of "Cancel" or "Replace" as appropriate.

Usage of the Confirmation Request Message

The Confirmation message can be used to request a specific confirmation message based on its AllocID and AllocAccount details.

Initiator			Respondent
	→	Confirmation Request	
	←	Confirmation, (ConfirmTransType = "New", ConfirmStatus = "Confirmed", ConfirmReqID = that of Confirmation Request message)	
	→	Confirmation Ack (AffirmStatus = "Received")	
	→	Confirmation Ack (AffirmStatus = "Affirmed"	

Rejected Confirmations

If the Confirmation is rejected by the buyside, The sellside can respond by either:

- sending a “cancel” for the original followed by a “new”

or

- sending a replace message.

© Copyright, 2006, FIX Protocol, Limited

Example flow using a "Cancel".

Initiator			Respondent
	←	Confirmation, (ConfirmType = "2" [Confirm], CopyMsgIndicator = "N", ConfirmTransType = "New", ConfirmStatus = "Confirmed"	
	→	Confirmation Ack (AffirmStatus = "Received")	
OR	→	Confirmation Ack (AffirmedStatus = "Confirm Rejected")	
		Cancelling the original Allocation Instruction and submitting a new one	
	←	Confirmation, (ConfirmType = "2" [Confirm], CopyMsgIndicator = "N", ConfirmTransType = "Cancel", ConfirmStatus = "Confirmed"	
	←	Confirmation, (ConfirmType = "2" [Confirm], CopyMsgIndicator = "N", ConfirmTransType = "New", ConfirmStatus = "Confirmed"	
	→	Confirmation Ack (AffirmedStatus = "Received")	
OR	→	Confirmation Ack (AffirmedStatus = "Confirm Rejected")	

Example flow using a "Replace" and "New"

Initiator			Respondent
	←	Confirmation, (ConfirmType = "2" [Confirm], CopyMsgIndicator = "N", ConfirmTransType = "New", ConfirmStatus = "Confirmed"	
	→	Confirmation Ack (AffirmedStatus = "Received")	
OR	→	Confirmation Ack (AffirmedStatus = "Confirm Rejected")	
		The corrected confirmation details are communicated by using a 'replace'	
	←	Confirmation, (ConfirmType = "2" [Confirm], CopyMsgIndicator = "N", ConfirmTransType = "Replace", ConfirmStatus = "Confirmed"	
	→	Confirmation Ack (AffirmStatus = "Received")	
OR	→	Confirmation Ack (AffirmStatus = "Confirm Rejected")	

© Copyright, 2006, FIX Protocol, Limited

CATEGORY: SETTLEMENT INSTRUCTIONS

Overview - Settlement Instructions

Settlement Instructions

The Settlement Instructions message provides the broker's, the institution's, or the intermediary's instructions for trade settlement. This message has been designed so that it can be sent from the broker to the institution, from the institution to the broker, or from either to an independent "standing instructions" database or matching system or, for CIV, from an intermediary to a fund manager.

The Settlement Instructions message can be used in one of three modes (SettlInstMode):

1) To provide "standing instructions" for the settlement of trades occurring in the future. The message could either be sent in an 'unsolicited' fashion (i.e. a 'push'-style update from one firm to that firm's counterparties) or in response to a Settlement Instruction Request message. In either of these scenarios, this message can provide multiple settlement instructions.

2) To reject a Settlement Instruction Request message (e.g. unable to process request, no matching settlement instructions found).

3) To provide settlement instructions for a specific Order with a single account either as overriding or standing instructions to support matching. The ClOrdID field should be used to link the settlement instructions to the corresponding Order message.

See VOLUME 7 - "PRODUCT: COLLECTIVE INVESTMENT VEHICLES"

The Settlement Instruction detail can be either explicitly specified (via the SettlInstructionsData component block) or can exist within an independent standing instructions database and can be referenced via the StandInstDbType, StandInstDbName, and StandInstDbID fields. See Volume 6 – Appendix 6-H for further details regarding the construction and formatting of settlement instruction details.

Settlement Instructions

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = T
777	SettlInstMsgID	Y	Unique identifier for this message
791	SettlInstReqID	N	Only used when this message is used to respond to a settlement instruction request (to which this ID refers)
160	SettlInstMode	Y	1=Standing Instructions, 2=Specific Allocation Account Overriding, 3=Specific Allocation Account Standing , 4=Specific Order, 5=Reject SSI request
792	SettlInstReqRejCode	N	Required for SettlInstMode = 5. Used to provide reason for rejecting a Settlement Instruction Request message.
58	Text	N	Can be used to provide any additional rejection text where rejecting a Settlement Instruction Request message.
354	EncodedTextLen	N	
355	EncodedText	N	

© Copyright, 2006, FIX Protocol, Limited

11	ClOrdID		N	Required for SettlInstMode=4.
60	TransactTime		Y	Date/time this message was generated
778	NoSettlInst		N	Required except where SettlInstMode is 5=Reject SSI request
→	**162**	**SettlInstID**	N	Unique ID for this settlement instruction. Required except where SettlInstMode is 5=Reject SSI request
→	**163**	**SettlInstTransType**	N	New, Replace, Cancel or Restate Required except where SettlInstMode is 5=Reject SSI request
→	**214**	**SettlInstRefID**	N	Required where SettlInstTransType is Cancel or Replace
→	**component block <Parties>**		N	Insert here the set of "Parties" (firm identification) fields defined in "Common Components of Application Messages" Used here for settlement location. Also used for executing broker for CIV settlement instructions
→	**54**	**Side**	N	Can be used for SettleInstMode 1 if SSIs are being provided for a particular side.
→	**460**	**Product**	N	Can be used for SettleInstMode 1 if SSIs are being provided for a particular product.
→	**167**	**SecurityType**	N	Can be used for SettleInstMode 1 if SSIs are being provided for a particular security type (as alternative to CFICode).
→	**461**	**CFICode**	N	Can be used for SettleInstMode 1 if SSIs are being provided for a particular security type (as identified by CFI code).
→	**120**	**SettlCurrency**	N	Can be used for SettleInstMode 1 if SSIs are being provided for a particular settlement currency
→	**168**	**EffectiveTime**	N	Effective (start) date/time for this settlement instruction. Required except where SettlInstMode is 5=Reject SSI request
→	**126**	**ExpireTime**	N	Termination date/time for this settlement instruction.
→	**779**	**LastUpdateTime**	N	Date/time this settlement instruction was last updated (or created if not updated since creation). Required except where SettlInstMode is 5=Reject SSI request
→	**component block <SettlInstructionsData>**		N	Insert here the set of "SettlInstructionsData" fields defined in "Common Components of Application Messages"
→	**492**	**PaymentMethod**	N	For use with CIV settlement instructions

© Copyright, 2006, FIX Protocol, Limited

→	**476**	**PaymentRef**	N	For use with CIV settlement instructions
→	**488**	**CardHolderName**	N	For use with CIV settlement instructions
→	**489**	**CardNumber**	N	For use with CIV settlement instructions
→	**503**	**CardStartDate**	N	For use with CIV settlement instructions
→	**490**	**CardExpDate**	N	For use with CIV settlement instructions
→	**491**	**CardIssNum**	N	For use with CIV settlement instructions
→	**504**	**PaymentDate**	N	For use with CIV settlement instructions
→	**505**	**PaymentRemitterID**	N	For use with CIV settlement instructions
	StandardTrailer		Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details

Refer to the FIXML element SettlInstrctns

© Copyright, 2006, FIX Protocol, Limited

Settlement Instruction Request

The Settlement Instruction Request message is used to request standing settlement instructions from another party. This could be:

- A buyside firm requesting standing instructions from a sellside firm.
- A sellside firm requesting standing instructions from a buyside firm.
- A sellside or buyside firm requesting standing instructions from a third party central static data database.
- A third party central static data database requesting standing instructions from a sellside or buyside firm.

Settlement instructions can be requested for any combination of the following parameters (in addition to the party whose instructions are being requested):

- AllocAccount
- Country (of settlement)
- Side
- SecurityType (and/or CFI code)
- SettlCurrency
- SettlDeliveryType (i.e. DVP vs. FOP)
- EffectiveTime (i.e. all instructions valid at any time from this date/time)
- Expiry Time (i.e. all instructions valid until this date/time)
- Last update time (i.e. all instructions created or updated since this date/time)

Alternatively, settlement instructions can be queried by reference to a database of standing instructions using the identifiers of that database as follows:

- Database id
- Database name
- Id of the settlement instructions on this database

The response to such a request should be a Settlement Instruction message with SettlInstTransType "New" containing all SSIs meeting the criteria specified in the Settlement Instruction request. If the request cannot be processed, the request should be rejected with a Settlement Instruction message with SettlInstTransType "Request rejected". Similarly, if the request returns no data, the request should be rejected with a Settlement Instruction message with SettlInstTransType "No matching data found".

Settlement Instruction Request

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = AV
791	SettlInstReqID	Y	Unique message ID
60	TransactTime	Y	Date/Time this request message was generated
component block <Parties>		N	Insert here the set of "Parties" (firm identification) fields defined in "Common Components of Application Messages"

© Copyright, 2006, FIX Protocol, Limited

			Used here for party whose instructions this message is requesting and (optionally) for settlement location Not required if database identifiers are being used to request settlement instructions. Required otherwise.
79	AllocAccount	N	Should not be populated if StandInstDbType is populated
661	AllocAcctIDSource	N	Required if AllocAccount populated Should not be populated if StandInstDbType is populated
54	Side	N	Should not be populated if StandInstDbType is populated
460	Product	N	Should not be populated if StandInstDbType is populated
167	SecurityType	N	Should not be populated if StandInstDbType is populated
461	CFICode	N	Should not be populated if StandInstDbType is populated
120	SettlCurrency	N	Should not be populated if StandInstDbType is populated
168	EffectiveTime	N	Should not be populated if StandInstDbType is populated
126	ExpireTime	N	Should not be populated if StandInstDbType is populated
779	LastUpdateTime	N	Should not be populated if StandInstDbType is populated
169	StandInstDbType	N	Should not be populated if any of AllocAccount through to LastUpdateTime are populated
170	StandInstDbName	N	Should not be populated if any of AllocAccount through to LastUpdateTime are populated
171	StandInstDbID	N	The identifier of the standing instructions within the database specified in StandInstDbType Required if StandInstDbType populated Should not be populated if any of AllocAccount through to LastUpdateTime are populated
	StandardTrailer	Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details

Refer to the FIXML element SettlInstrctnReq

© Copyright, 2006, FIX Protocol, Limited

CATEGORY: TRADE CAPTURE ("STREETSIDE") REPORTING

Overview:

Trade Capture Reporting allows sell-side firms (broker, exchange, ECN, central counter parties) to provide timely reporting of completed trades to parties involved in a trade as well as to external entities not involved in the execution of the trade. Trade Capture Reporting has been designed for several uses including sell-side trade reporting into an exchange or ECN, trade confirmation reporting by an exchange or clearing organization, and end of day trade reporting via static data files. For example, in the United States OCC (Options Clearing Corporation) and CME (Chicago Mercantile Exchange) both make extensive use of the Trade Capture Report for trade management, trade confirmation reporting, and end of day trade reconciliation via static data file. As settlement cycles reduce, such communication must be closer to real-time vs. an end-of-the day batch process. The Trade Capture Report and Trade Capture Report Request messages have been designed to facilitate such communication.

Trade Capture Reporting has been expanded to include support for two party (sell side - buy side) and three party (sell side - exchange/clearing house/VMU - buy side) communication. Appendix B contains an extensive set of message flow tables which illustrate Trade Capture Report usage for privately negotiated trades in an exchange setting. The tables also deomonstrate the appropriate models for cleared trade reporting by clearing organizations. Support for matched trades, unmatched trades, transfer, block trades, and exchange for physical (EFP) trades are supported.

Trade Capture Report Usages

Trade Capture Reports are used for various purposes including:

- Relaying Confirmed Trades to various parties not directly involved in the execution, e.g. CSD's, clearing houses, clearing firms and regulatory bodies. Those messages are **outbound** (from the marketplace).
- Relaying Confirmed Trades to counterparties of the trade. Where Execution Reports may be sufficient for front-office purposes, Trade Capture Reports can serve more demanding back-office processes better. Those messages are **outbound** (from the marketplace).
- Reporting of privately negotiated ("street-side") trades, i.e. trades formed outside of the marketplace. Those messages are **inbound** (to the marketplace) but may also be used as **outbound** (when the marketplace relays them to counterparties).
- Reporting of trades executed on the floor or from an automated order routing mechanism. These messages are **inbound**.
- Requesting a cancellation or amendment of a Confirmed Trade. Those messages are **inbound** (to the marketplace) but may also be used as **outbound** (when the marketplace relays them to counterparties).

In Exchange, ECN and Central Counter Party models, a TCR (Trade Capture Report) process ends with a Confirmed Trade. The process is triggered by a request to register a new trade, replace a trade or cancel a trade. The process can involve the counterparty and / or a marketplace official acknowledgement and can therefore take some time. During this time, the initiator may change his mind and withdraw or request a change to the request.

The following rules apply to TCR identifiers:

- TradeReportID is assigned by the submitter of the message and used as a pure message identifier.
- TradeID is assigned by the marketplace when it records a Confirmed Trade. It should be noted that some marketplaces will assign the TradeID earlier in the process, meaning that (in the case of sequential ID assignment) there will be gaps when a trade is not completed.
- TradeReportRefID is assigned by the submitter when it wants to link a new message to a previous message. This would normally apply only when it requests a replace or cancel of an ongoing process (i.e. the marketplace has not yet recorded the Confirmed Trade) and when the marketplace issues confirmed trades ending the process of reporting and acknowledging a privately negotiated trade.

© Copyright, 2006, FIX Protocol, Limited

- SecondaryTradeID can be assigned by the marketplace as an identifier for the process leading to a Confirmed Trade. It can be used by the submitter as an alternative to TradeReportRefID in a cancel or replace. Note that a prerequisite to use the SecondaryTradeID is that the marketplace issues TCR Ack messages providing that tag.

Trade Capture Reporting Business Workflows

Most markets see confirmed trades as a bookkeeping record of a finalized deal. Naturally there may be a chain of actors involved in the trade process, but from a marketplace point of view there is one such representation. In the following text the term "Confirmed Trade" is used to represent the fact that the marketplace has confirmed and stored a trade in its records. It may be that subsequent actors as clearing houses and depositories do not see that Confirmed Trade as finalized, but this is about their role in the transaction chain, not about the marketplace. A marketplace needs to communicate that trade to other interested parties.

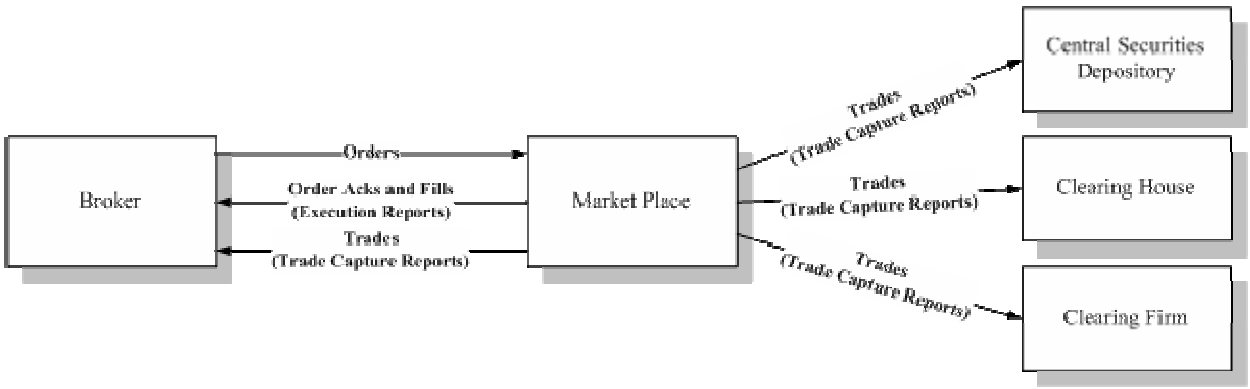
A Confirmed Trade can be produced by a marketplace as the result of various business processes, e.g.:

- Auto-matching of order and / or quotes
- Quote negotiations
- Hit / Lift of orders and / or quotes
- Reporting of trades in variants exemplified further in this section

From a marketplace point of view the outgoing trade confirmations are inherently different from any incoming representations of interest, including privately negotiated trades and other requests to report a trade. Once the marketplace has confirmed a trade and recorded it, it can submit it to downstream processes. Any post-trade management of that trade is often a matter between counterparties or their representatives and organizations specialized in downstream parts of the transaction chain.

The following tesxt describes a number of relevant high-level workflows. Detailed workflows are defined in greater detail in this section.

This diagram depicts a typical high level order flow:



A broker enters Order messages and receives Execution Reports in response. The Execution Reports are used to relay the status of an order, including:

- Confirm the acceptance of orders
- Relay fill information

© Copyright, 2006, FIX Protocol, Limited

- Inform about order expiry
- Etc

Execution Reports are sent to the order owner (the broker).

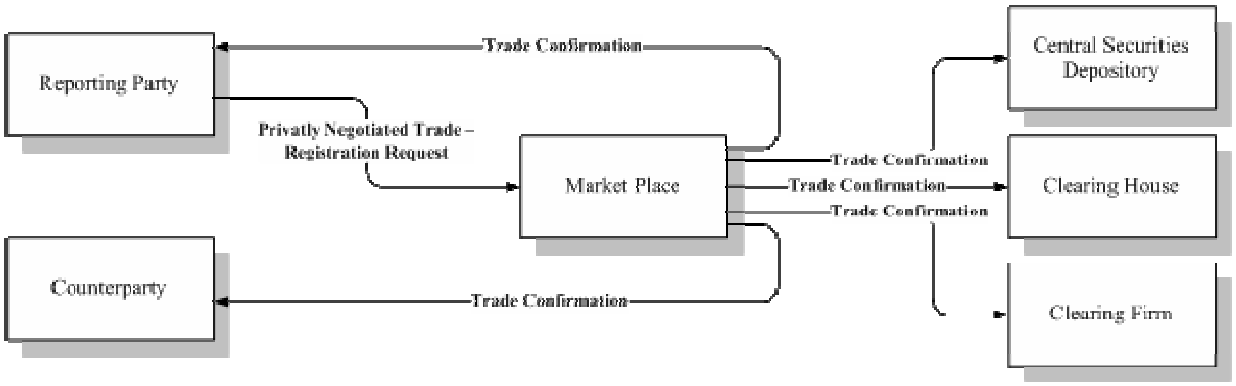
When an order is executed, Confirmed Trades are created. Those trades are published as Trade Capture Reports to various interested parties, including the broker (e.g. the back office). What trade information is made available to the respective parties varies.

Trades Reported to the Marketplace

Reporting privately negotiated trades to an exchange or a marketplace occurs when regulatory frameworks require it or when the marketplace provides complementary services. The following describe some of the business cases.

Privately Negotiated Trade, Two-Party Report

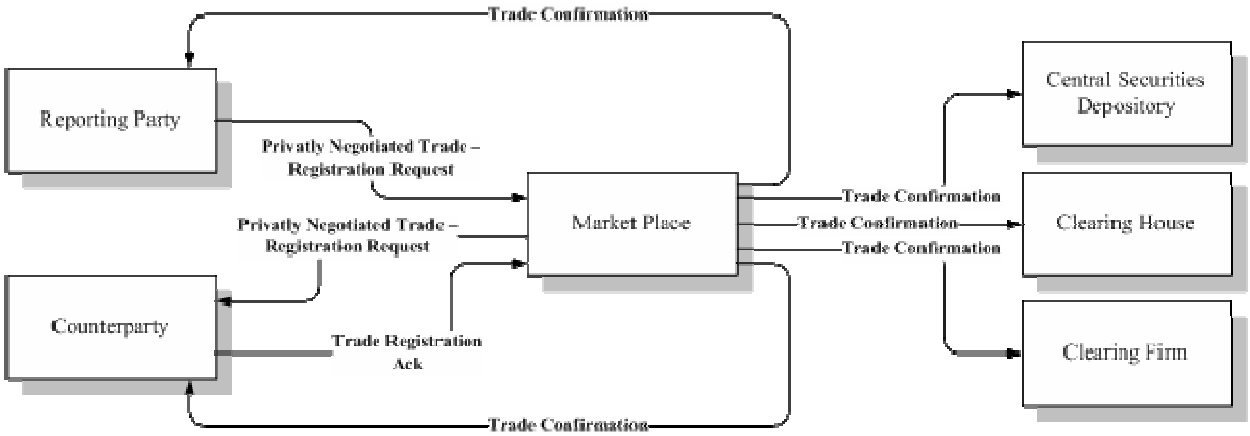
This diagram depicts the process:



A deal is typically struck between two parties, one of whom has an obligation to report the trade. The counterparty has an agreement with the reporting party. The reporting party sends the trade report to the market. The marketplace accepts the report and confirms the Confirmed Trade to all involved parties. The FIX Trade Capture Report is used for all involved messages.

Privately Negotiated Trade, One-Party Report for Passs-through to Counterparty

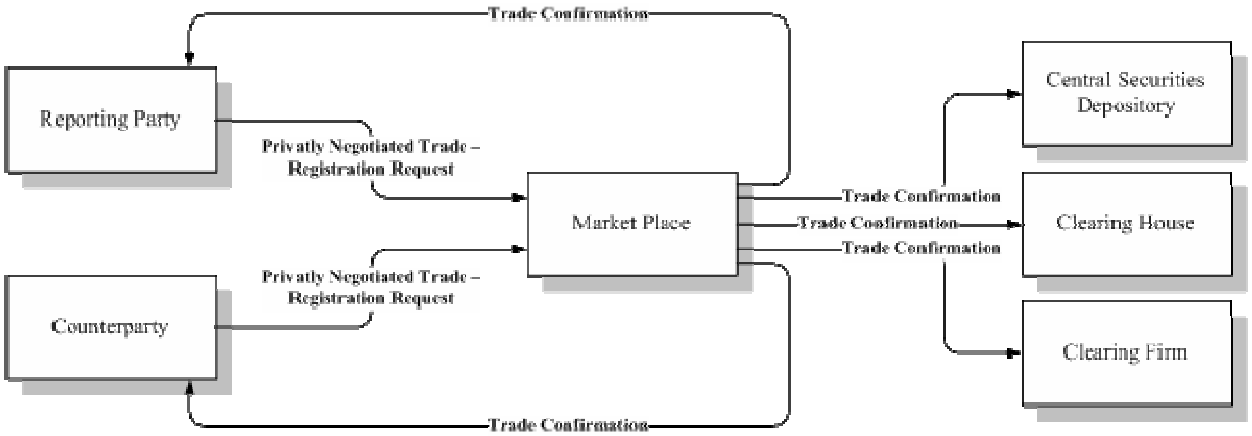
This diagram depicts the process:



Again the deal is struck between two parties, one of whom has an obligation to report the trade. The counterparty does not have agreement with the reporting party, so he must acknowledge the trade. The reporting party sends the trade report to the market. The market informs the counterparty of the report and the counterparty then accepts the trade. The marketplace confirms the Confirmed Trade to all involved parties. The FIX Trade Capture Report is used for all involved messages.

Privately Negotiated Trade, One Party Report for Matching

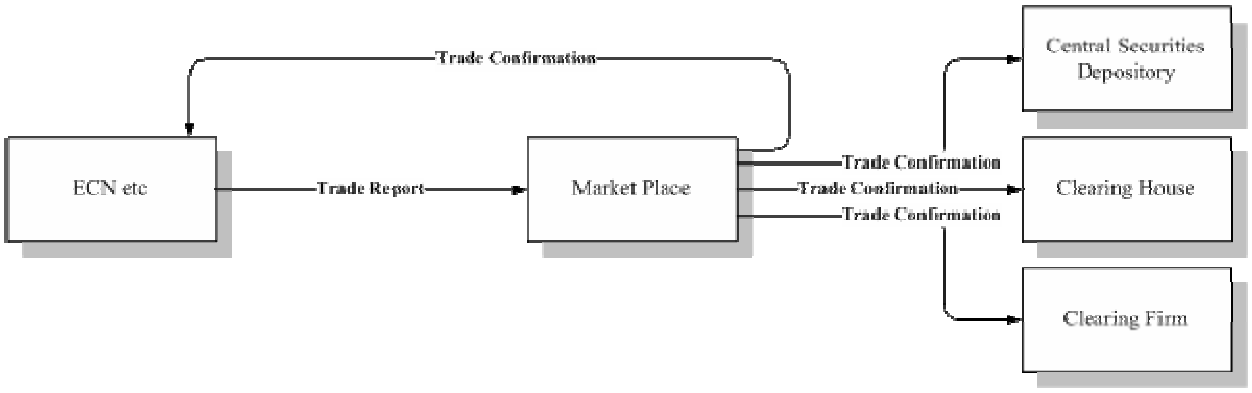
This diagram depicts the process:



This model is used by some markets for the same purpose as the one described in the pass-through model above. The model can also be used for trade negotiation (if allowed in the market). The deal is struck between two parties; both have an obligation to report the trade. Both parties send their view of the trade to the market. The market matches the two reports and confirms the Confirmed Trade to all involved parties. The FIX Trade Capture Report is used for all involved messages.

Reporting of Locked-In Trades (a.k.a. Three-Party Report)

This diagram depicts the process:



This model is typically used when external recognized markets such as ECNs or risk-less principals report trades to a marketplace. The case can for example be that the reporting party has arranged a trade between two parties but entered in between, so that he buys from one party and sells to the other. The reporting party can thereby report a trade which requires no matching, sometimes called a locked-in trade. The deal is struck at a market external from the marketplace; that market has an obligation to report the trade. External market reports and the marketplace confirm the Confirmed Trade to all involved parties. The FIX Trade Capture Report is used for all involved messages.

© Copyright, 2006, FIX Protocol, Limited

© Copyright, 2006, FIX Protocol, Limited

Proposed Message Flows

The message flow diagrams below show illustrate the primary models for Trade Capture Reporting in exchange, ECN, and central counter party environments. Supporting detail can be found in the Appendix B work flow and TCR usage tables.

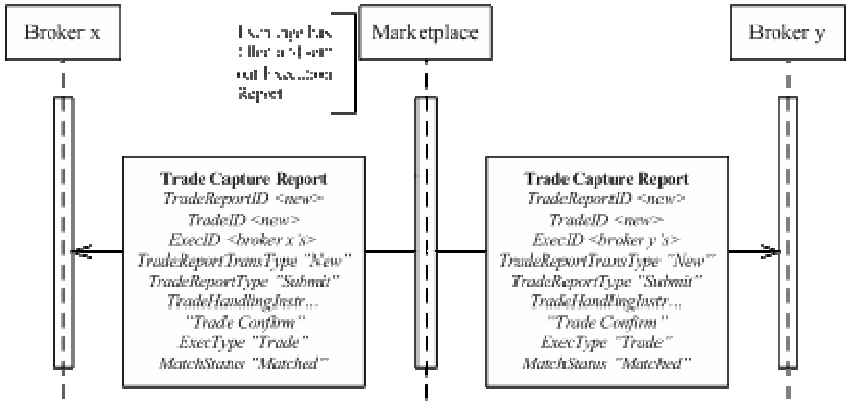
Reporting Confirmed Trades to Miscellaneous Parties

Trade Capture Reports can be used to report Confirmed Trades to actors not involved in the execution. Such parties include clearing houses, clearing firms, broker back-offices, depositories, vendors and regulatory bodies. The information that is reported to each party normally varies depending on their role.

Deals struck using Order and Quote messages are reported to the counterparties using Execution Reports. A marketplace may choose to send Trade Capture Reports to them as well, e.g. under the assumption that Execution Reports are used primarily by front-offices and Trade Capture Reports by back-offices.

Extension to Workflows resulting in Fills reported as Execution Reports

The diagram below depicts the trade confirmation part of the workflow defined in the Order Flow section above.

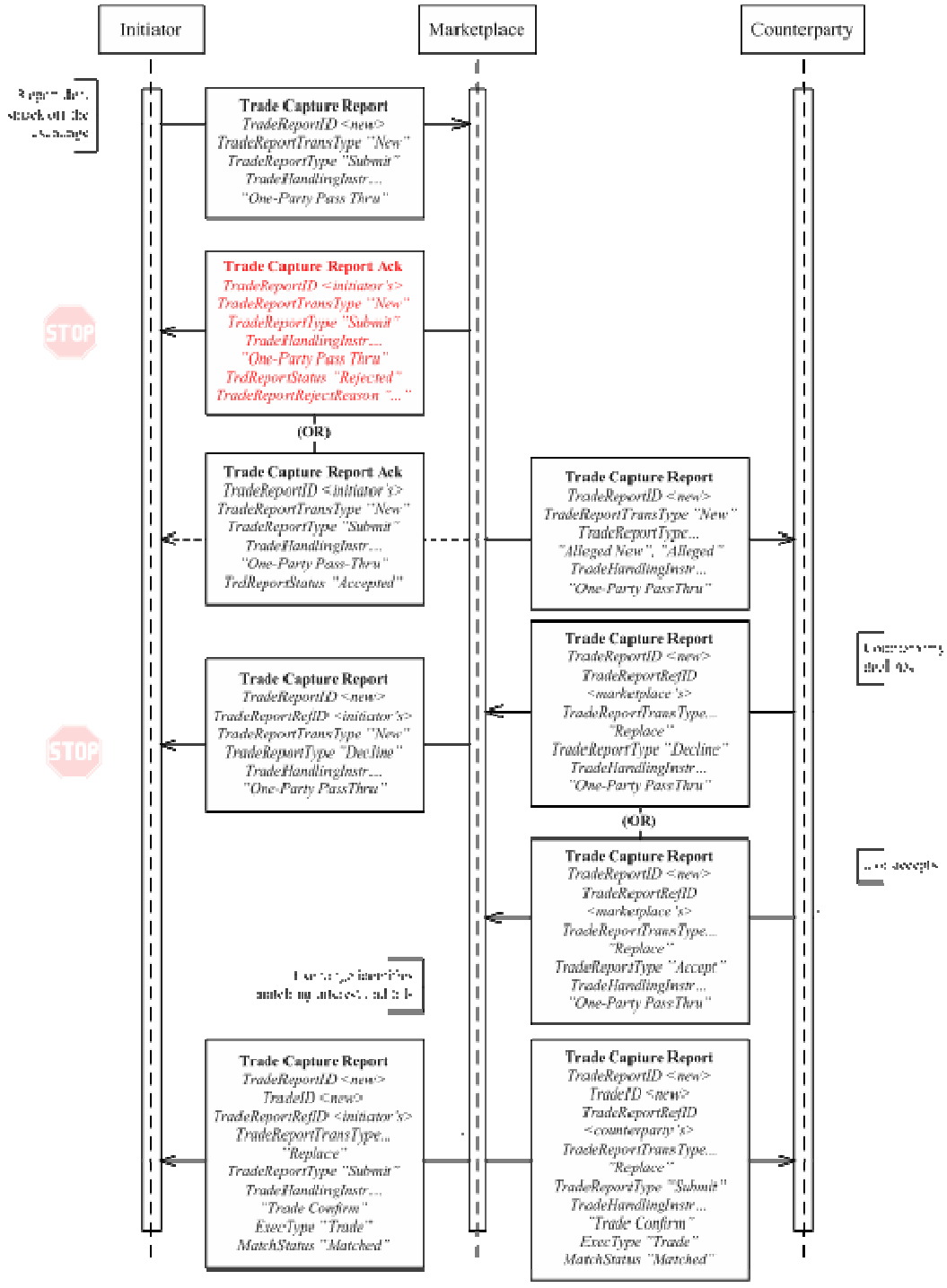


Please note that Confirmed Trades (in any of the flows of this section) can also be distributed to various other relevant parties as described above. Also note that OrderID, QuoteID and other message references could be specified in those trade confirmations.

© Copyright, 2006, FIX Protocol, Limited

One-Party Report for Pass-Through to Model

The following diagram depicts the core workflow in the One-Party pass-thru and accept model



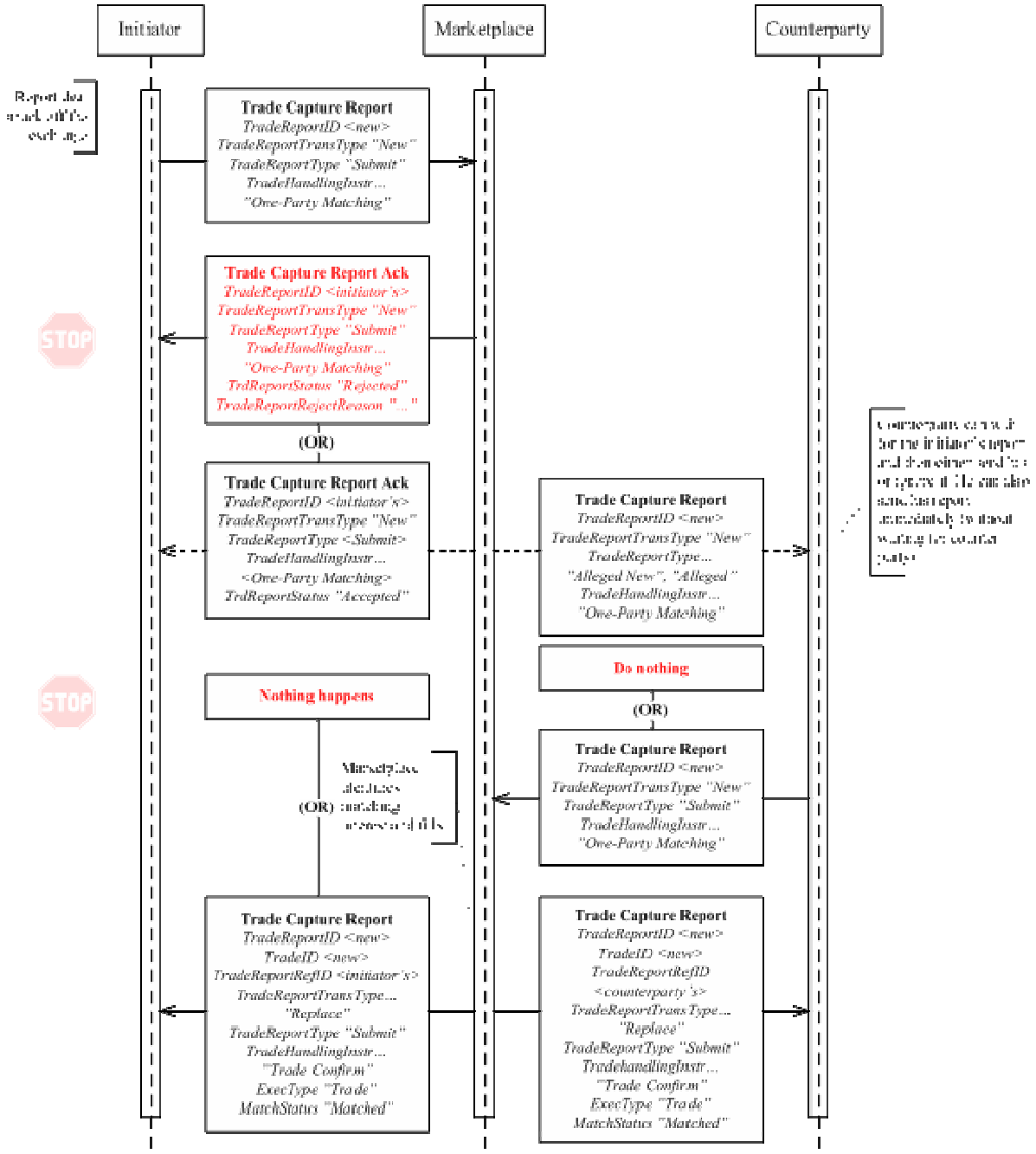
The counterparty receives the alleged trade report and accepts it. The counterparty can also complete the trade by entering private information for his side. Note that marketplace Ack messages for the counterparty response are not shown in the diagram due to space limitations.

Some marketplaces may choose to remind counterparties that take no action, others may let the TCR expire and leave it to the parties to take whatever action they deem relevant. Yet another variant is that the marketplace interprets the lack of a response as an acknowledgement and issues trade confirmations. Parties should bilaterally agree on behavioral aspects as the ones mentioned above.

© Copyright, 2006, FIX Protocol, Limited

One-Party Report for Matching Model

The following diagram depicts the workflow in the One-Party negotiation model.



In this model both parties submit their trade half and the marketplace matches the two. The marketplace may choose to forward the reporting party's TCR to the counterparty as a notification / trigger for action. Some counterparties may use that message to automatically prepare their response message.

© Copyright, 2006, FIX Protocol, Limited

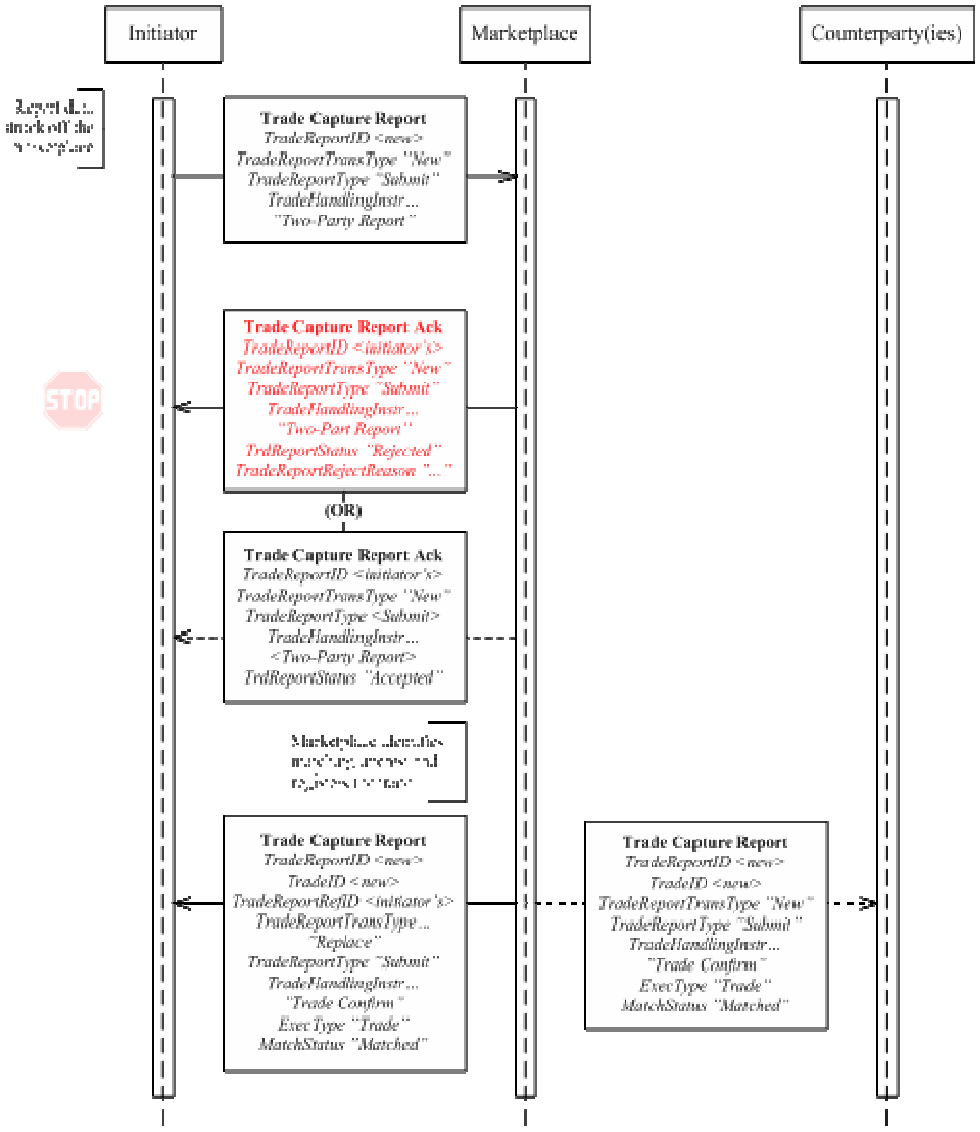
Some marketplaces may choose to remind counterparties that take no action, others may let the TCR expire and leave it to the parties to take whatever action they deem relevant. Yet another variant is that the marketplace interprets the lack of a response as an acknowledgement and issues trade confirmations.

Note that when the counterparty issues a TCR that does not match the TCR of the initiator (the second alternative back to the initiator in the diagram), a viable alternative is to let the initial TCR expire and start a new workflow with reversed roles. Parties should bilaterally agree on behavioral aspects as the ones mentioned above.

© Copyright, 2006, FIX Protocol, Limited

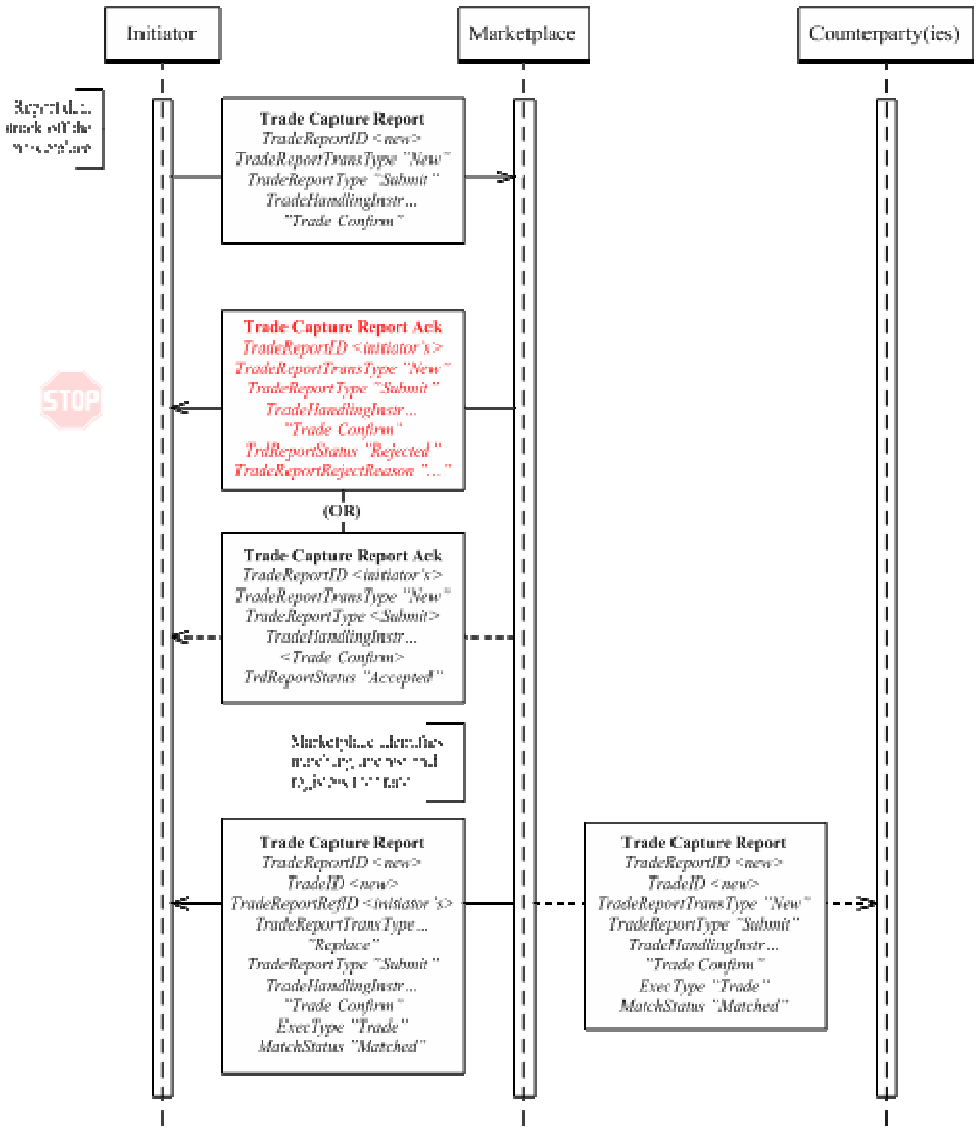
Two-Party Reporting

In this model one party reports a trade with one (a cross trade) or two counterparties. Counterparties are optionally informed by the marketplace of the completed trade. The workflow is depicted in the following diagram:



Confirmed Trade Reporting Model

In this model one party, itself a recognized marketplace as an ECN, ATS, exchange or clearing organization reports a trade with two counterparties. Counterparties are optionally informed by the marketplace of the completed trade. The workflow is depicted in the following diagram:



Note that this workflow is very similar to the Two-Party reporting model, the difference lies in the TradeHandlingInst ("Trade Confirm" instead of "Two-Party Report"). The reasoning behind this is that any marketplace records trades and thereby issues confirmed trades. Reporting to other marketplaces is done for regulatory and pure display purposes.

© Copyright, 2006, FIX Protocol, Limited

Trade Amendment

Marketplaces can allow brokers to request trade amendments. Trade amendments are normally limited to private properties for the side of the initiator (called Addendums) – i.e. can not affect the counterparty. Changes to bilateral trade terms can be indicated by using the "No/Was" value of the TradeReportType. Trade Addendums might not need acceptance by the counterparty. Marketplaces may limit what properties can be updated and also put a time limit for updates (e.g. up to fifteen minutes after the trade was created).

Trade amendment is done using the same models as for reporting, i.e. One-Party for Pass-Thru, One-Part for matching, Two-Party or reporting of confirmed trades. The workflows will thereby be very similar to the ones above, the difference being that other actions (TradeReportType) are used. Supporting work flow diagrams and usage tables can be found in Appendicies A and B respectively.

Trade Break / Trade Cancel

Marketplaces can allow brokers to request trade breaks (or cancellations). Marketplaces allowing brokers to request trade cancellation would require that all parties to the trade agree. Trade breaks may be limited to certain trades (e.g. privately negotiated ones), a limited time (e.g. up to fifteen minutes after the trade was created), etc.

Trade break is done using the same models as for reporting, i.e. One-Party for Pass-Thru, One-Part for matching, Two-Party or reporting of confirmed trades. The workflows will thereby be very similar to the ones above, the difference being that other actions (TradeReportType) are used. Supporting work flow diagrams and usage tables can be found in Appendicies A and B respectively.

Trade Capture Report Request

The Trade Capture Report Request can be used to:

- Request one or more trade capture reports based upon selection criteria provided on the trade capture report request
- Subscribe for trade capture reports based upon selection criteria provided on the trade capture report request.

The following criteria can be specified on the Trade Capture Report Request:

- All trades matching specified trade identification: TradeReportID, SecondaryTradeReportID
- All trades matching specified trade types: TrdType, TrdSubType, TransferReason, SecondaryTrdType, TradeLinkID
- All trades matching the order identification information: OrderId, ClOrdID, ExecID
- Trades that have specified MatchStatus
- All trades for the party defined in the component block <Parties>
    - This can be a trader id, firm, broker id, clearing firm
- All trades for a specific instrument, specified using the component block <Instrument>, the component block <UnderlyingInstrument>, and/or the component block <InstrumentLeg>.
- All unreported trades – Executions that have not been sent
- All unmatched trades – Trades that have not been matched
- All trades matching specific date and trading session criteria
- Trades entered via a specific TradeInputSource
- Trades entered via a specific TradeInputDevice
- All Advisories

© Copyright, 2006, FIX Protocol, Limited

Each field in the Trade Capture Report Request (other than TradeRequestID and SubscriptionRequestType) identify filters - trade reports that satisfy all Specified filters will be returned. Note that the filters are combined using an implied "and" - a trade report must satisfy every specified filter to be returned.

The optional date or time range-specific filter criteria (within NoDates repeating group) can be used in one of two modes:

- "Since" a time period. NoDates=1 with first TradeDate (and optional TransactTime) indicating the "since" (greater than or equal to operation) point in time.
- "Between" time periods. NoDates=2 with first TradeDate (and optional TransactTime) indicating the "beginning" (greater than or equal to operation) point in time and the second TradeDate (and optional TransactTime) indicating the "ending" (less than or equal to operation) point in time.

Trade Capture Report messages are the normal return type to a Trade Capture Report Request.

The response to a Trade Capture Report Request can be:

- One or more Trade Capture Reports
- A Trade Capture Report Request Ack followed by one or more Trade Capture Reports in two specific cases:
    - When the Trade Capture Reports are being delivered out of band (such as a file transfer),
    - When there is a processing delay between the time of the request and when the reports will be sent (for instance in a distributed trading environment where trades are distributed across multiple trading systems).
- A Trade Capture Report Ack only
- When no trades are found that match the selection criteria specified on the Trade Capture Report Request
- When the Trade Capture Report Request was deemed invalid for business reasons by the counterparty

Trade Capture Report Request

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = AD
568	TradeRequestID	Y	Identifier for the trade request
1003	TradeID	N	
1040	SecondaryTradeID	N	
1041	FirmTradeID	N	
1042	SecondaryFirmTradeID	N	
569	TradeRequestType	Y	
263	SubscriptionRequestType	N	Used to subscribe / unsubscribe for trade capture reports If the field is absent, the value 0 will be the default (snapshot only - no subscription)
571	TradeReportID	N	To request a specific trade report
818	SecondaryTradeReportID	N	(Deprecated in FIX.5.0)To request a specific trade report

© Copyright, 2006, FIX Protocol, Limited

17	ExecID	N	
150	ExecType	N	To requst all trades of a specific execution type
37	OrderID	N	
11	ClOrdID	N	
573	MatchStatus	N	
828	TrdType	N	To request all trades of a specific trade type
829	TrdSubType	N	To request all trades of a specific trade sub type
1123	TradeHandlingInstr	N	
830	TransferReason	N	To request all trades for a specific transfer reason
855	SecondaryTrdType	N	To request all trades of a specific trade sub type
820	TradeLinkID	N	To request all trades of a specific trade link id
880	TrdMatchID	N	To request a trade matching a specific TrdMatchID
component block <Parties>		N	Used to specify the parties for the trades to be returned (clearing firm, execution broker, trader id, etc.) ExecutingBroker ClearingFirm ContraBroker ContraClearingFirm SettlementLocation - depository, CSD, or other settlement party ExecutingTrader InitiatingTrader OrderOriginator
component block <Instrument>		N	Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages"
component block <InstrumentExtension>		N	Insert here the set of "InstrumentExtension" fields defined in "Common Components of Application Messages"
component block <FinancingDetails>		N	Insert here the set of "FinancingDetails" fields defined in "Common Components of Application Messages"
711	NoUnderlyings	N	Number of underlyings
→	**component block <UnderlyingInstrument>**	N	Must be provided if Number of underlyings > 0
555	NoLegs	N	Number of legs Identifies a Multi-leg Execution if present and non-zero.
→	**component block <InstrumentLeg>**	N	Must be provided if Number of legs > 0

© Copyright, 2006, FIX Protocol, Limited

580	NoDates		N	Number of date ranges provided (must be 1 or 2 if specified)
→	**75**	**TradeDate**	N	Used when reporting other than current day trades. Conditionally required if NoDates > 0
→	**779**	**LastUpdateTime**	N	
→	**60**	**TransactTime**	N	To request trades for a specific time.
715	ClearingBusinessDate		N	To request trades for a specific clearing business date.
336	TradingSessionID		N	To request trades for a specific trading session.
625	TradingSessionSubID		N	To request trades for a specific trading session.
943	TimeBracket		N	To request trades within a specific time bracket.
54	Side		N	To request trades for a specific side of a trade.
442	MultiLegReportingType		N	Used to indicate if trades are to be returned for the individual legs of a multileg instrument or for the overall instrument.
578	TradeInputSource		N	To requests trades that were submitted from a specific trade input source.
579	TradeInputDevice		N	To request trades that were submitted from a specific trade input device.
725	ResponseTransportType		N	Ability to specify whether the response to the request should be delivered inband or via pre-arranged out-of-band transport.
726	ResponseDestination		N	URI destination name. Used if ResponseTransportType is out-of-band.
58	Text		N	Used to match specific values within Text fields
354	EncodedTextLen		N	
355	EncodedText		N	
1011	MessageEventSource		N	Used to identify the event or source which gave rise to a message
	StandardTrailer		Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details

Refer to the FIXML element TrdCaptRptReq

© Copyright, 2006, FIX Protocol, Limited

Trade Capture Report Request Ack

The Trade Capture Request Ack message is used to:

- Provide an acknowledgement to a Trade Capture Report Request in the case where the Trade Capture Report Request is used to specify a subscription or delivery of reports via an out-of-band ResponseTransmissionMethod.
- Provide an acknowledgement to a Trade Capture Report Request in the case when the return of the Trade Capture Reports matching that request will be delayed or delivered asynchronously. This is useful in distributed trading system environments.
- Indicate that no trades were found that matched the selection criteria specified on the Trade Capture Report Request
- The Trade Capture Request was invalid for some business reason, such as request is not authorized, invalid or unknown instrument, party, trading session, etc.

NOTE: A Trade Capture Report Request Ack is not required if one or more Trade Capture Reports will be returned in-band immediately.

Trade Capture Report Request Ack

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = AQ
568	TradeRequestID	Y	Identifier for the trade request
1003	TradeID	N	
1040	SecondaryTradeID	N	
1041	FirmTradeID	N	
1042	SecondaryFirmTradeID	N	
569	TradeRequestType	Y	
263	SubscriptionRequestType	N	Used to subscribe / unsubscribe for trade capture reports If the field is absent, the value 0 will be the default
748	TotNumTradeReports	N	Number of trade reports returned
749	TradeRequestResult	Y	Result of Trade Request
750	TradeRequestStatus	Y	Status of Trade Request
component block <Instrument>		Y	Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages"
711	NoUnderlyings	N	Number of underlyings
→	**component block <UnderlyingInstrument>**	N	Must be provided if Number of underlyings > 0
555	NoLegs	N	Number of legs Identifies a Multi-leg Execution if present and non-zero.
→	**component block <InstrumentLeg>**	N	Must be provided if Number of legs > 0

© Copyright, 2006, FIX Protocol, Limited

442	MultiLegReportingType	N	Specify type of multileg reporting to be returned.
725	ResponseTransportType	N	Ability to specify whether the response to the request should be delivered inband or via pre-arranged out-of-band transport.
726	ResponseDestination	N	URI destination name. Used if ResponseTransportType is out-of-band.
58	Text	N	May be used by the executing market to record any execution Details that are particular to that market
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
1011	MessageEventSource	N	Used to identify the event or source which gave rise to a message
	StandardTrailer	Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details
Refer to the FIXML element TrdCaptRptReqAck

Trade Capture Report

The Trade Capture Report message can be:

- Used to report trades between counterparties.
- Used to report trades to a trade matching system
- Can be sent unsolicited between counterparties.
- Sent as a reply to a Trade Capture Report Request.
- Can be used to report unmatched and matched trades.

Trade Capture Report

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = AE
571	TradeReportID	N	TradeReportID is conditionally required in a message-chaining model in which a subsequent message may refer to a prior message via TradeReportRefID. The alternative to a message-chain model is an entity-based model in which TradeID is used to identify a trade. In this case, TradeID is required and TradeReportID can be optionally specified.
1003	TradeID	N	
1040	SecondaryTradeID	N	
1041	FirmTradeID	N	
1042	SecondaryFirmTradeID	N	
487	TradeReportTransType	N	Identifies Trade Report message transaction type.
856	TradeReportType	N	
939	TrdRptStatus	N	Status of Trade Report In 3 party listed derivatives model used to convey status of a trade to a counterparty. Used specifically in a "claim" model.
568	TradeRequestID	N	Request ID if the Trade Capture Report is in response to a Trade Capture Report Request
828	TrdType	N	
829	TrdSubType	N	
855	SecondaryTrdType	N	
1123	TradeHandlingInstr	N	
1124	OrigTradeHandlingInstr	N	
1125	OrigTradeDate	N	Used to preserve original trade date when original trade is being referenced in a subsequent trade transaction such as a transfer
1126	OrigTradeID	N	Used to preserve original trade id when original trade is

© Copyright, 2006, FIX Protocol, Limited

			being referenced in a subsequent trade transaction such as a transfer
1127	OrigSecondaryTradeID	N	Used to preserve original secondary trade id when original trade is being referenced in a subsequent trade transaction such as a transfer
830	TransferReason	N	
150	ExecType	N	Type of Execution being reported: Uses subset of ExecType for Trade Capture Reports
748	TotNumTradeReports	N	Number of trade reports returned - if this report is part of a response to a Trade Capture Report Request
912	LastRptRequested	N	Indicates if this is the last report in the response to a Trade Capture Report Request
325	UnsolicitedIndicator	N	Set to 'Y' if message is sent as a result of a subscription request or out of band configuration as opposed to a Position Request.
263	SubscriptionRequestType	N	Used to subscribe / unsubscribe for trade capture reports. If the field is absent, the value 0 will be the default
572	TradeReportRefID	N	The TradeReportID that is being referenced for some action, such as correction or cancelation
881	SecondaryTradeReportRefID	N	(Deprecated in FIX.5.0)
818	SecondaryTradeReportID	N	(Deprecated in FIX.5.0)
820	TradeLinkID	N	Used to associate a group of trades together. Useful for average price calculations.
880	TrdMatchID	N	
17	ExecID	N	Exchanged assigned Execution ID (Trade Identifier)
39	OrdStatus	N	Status of order as of this trade report
527	SecondaryExecID	N	
378	ExecRestatementReason	N	Reason for restatement
570	PreviouslyReported	N	Indicates if the trade capture report was previously reported to the counterparty
423	PriceType	N	Can be used to indicate cabinet trade pricing
component block <RootParties>		N	Insert here the set of "Root Parties" fields defined in "common components of application messages" Used for acting parties that applies to the whole message, not individual legs, sides, etc..
1015	AsOfIndicator	N	Indicates if the trade is an outtrade from a previous day.
716	SettlSessID	N	Intraday(ITD), Regular Trading Hours(EOD),
717	SettlSessSubID	N	
component block <Instrument>		Y	Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application

© Copyright, 2006, FIX Protocol, Limited

			Messages"
component block <FinancingDetails>		N	Insert here the set of "FinancingDetails" fields defined in "Common Components of Application Messages"
component block <OrderQtyData>		N	Insert here the set of "OrderQtyData" fields defined in "Common Components of Application Messages" Note: OrderQty field is required unless rejecting or an order ack for a CashOrderQty or PercentOrder.
854	QtyType	N	
component block <YieldData>		N	Insert here the set of "YieldData" fields defined in "Common Components of Application Messages"
711	NoUnderlyings	N	Number of underlyings
→	**component block <UnderlyingInstrument>**	N	Must be provided if Number of underlyings > 0
822	UnderlyingTradingSessionID	N	
823	UnderlyingTradingSessionSubID	N	
32	LastQty	Y	Trade Quantity.
31	LastPx	Y	Trade Price.
1056	CalculatedCcyLastQty	N	
669	LastParPx	N	Last price expressed in percent-of-par. Conditionally required for Fixed Income trades when LastPx is expressed in Yield, Spread, Discount or any other price type that is not percent-of-par.
194	LastSpotRate	N	Applicable for F/X orders
195	LastForwardPoints	N	Applicable for F/X orders
1071	LastSwapPoints	N	
30	LastMkt	N	
75	TradeDate	Y	Used when reporting other than current day trades.
715	ClearingBusinessDate	N	
6	AvgPx	N	Average Price - if present then the LastPx will contain the original price on the execution
component block <SpreadOrBenchmarkCurveData>		N	Insert here the set of "SpreadOrBenchmarkCurveData" fields defined in "Common Components of Application Messages"
819	AvgPxIndicator	N	Average Pricing indicator
component block <PositionAmountData>		N	Insert here here the set of "Position Amount Data" fields defined in "Common Components of Application Messages"
442	MultiLegReportingType	N	Type of report if multileg instrument. Provided to support a scenario for trades of multileg

© Copyright, 2006, FIX Protocol, Limited

				instruments between two parties.
824	TradeLegRefID		N	Reference to the leg of a multileg instrument to which this trade refers Used when MultiLegReportingType = 2 (Single Leg of a Multileg security)
555	NoLegs		N	Number of legs Identifies a Multi-leg Execution if present and non-zero.
→	**component block <InstrumentLeg>**		N	Must be provided if Number of legs > 0
→	**687**	**LegQty**	N	
→	**690**	**LegSwapType**	N	Instead of LegQty – requests that the sellside calculate LegQty based on opposite Leg
→	**990**	**LegReportID**	N	Additional attribute to store the Trade ID of the Leg.
→	**component block <LegStipulations>**		N	
→	**564**	**LegPositionEffect**	N	Provide if the PositionEffect for the leg is different from that specified for the overall multileg security
→	**565**	**LegCoveredOrUncovered**	N	Provide if the CoveredOrUncovered for the leg is different from that specified for the overall multileg security.
→	**component block <NestedParties>**		N	Insert here the set of "Nested Parties" (firm identification "nested" within additional repeating group) fields defined in "Common Components of Application Messages" Used for NestedPartyRole=Leg Clearing Firm/Account, Leg Account/Account Type
→	**654**	**LegRefID**	N	Used to identify a specific leg.
→	**566**	**LegPrice**	N	Provide only if a Price is required for a specific leg. Used for anchoring the overall multileg security price to a specific leg Price.
→	**587**	**LegSettlType**	N	
→	**588**	**LegSettlDate**	N	Takes precedence over LegSettlmntTyp value and conditionally required/omitted for specific LegSettlType values.
→	**637**	**LegLastPx**	N	Used to report the execution price assigned to the leg of the multileg instrument
→	**675**	**LegSettlCurrency**	N	
→	**1073**	**LegLastForwardPoints**	N	
→	**1074**	**LegCalculatedCcyLastQty**	N	
→	**1075**	**LegGrossTradeAmt**	N	For FX Futures can be used to express the notional value of a trade when LegLastQty and other quantity fields are

© Copyright, 2006, FIX Protocol, Limited

				expressed in terms of number of contracts - LegContractMultiplier (231) is required in this case.
60	TransactTime		N	Time the transaction represented by this Trade Capture Report occurred
component block <TrdRegTimestamps>			N	
63	SettlType		N	
64	SettlDate		N	Takes precedence over SettlType value and conditionally required/omitted for specific SettlType values.
987	UnderlyingSettlementDate		N	The settlement date for the underlying instrument of a derivatives security.
573	MatchStatus		N	
574	MatchType		N	
1115	OrderCategory		N	
552	NoSides		Y	Number of sides
→	**54**	**Side**	Y	
→	**37**	**OrderID**	N	OrderID should be conditionally required when Trade Capture Report is used for back office processing.
→	**198**	**SecondaryOrderID**	N	Can be used to provide order id used by exchange or executing system.
→	**11**	**ClOrdID**	N	Required for executions against electronically submitted orders which were assigned an ID by the institution or intermediary. Not required for orders manually entered by the broker or fund manager (for CIV orders).
→	**19**	**ExecRefID**	N	
→	**526**	**SecondaryClOrdID**	N	Can be used to provide secondary client order identifiers associated with this trade.
→	**66**	**ListID**	N	
→	**1009**	**SideQty**	N	Used to indicate the quantity on one side of a multi-sided Trade Capture Report
→	**1005**	**SideTradeReportID**	N	Used to indicate the report ID on one side of a multi-sided Trade Capture Report
→	**1006**	**SideFillStationCd**	N	Used for order routing to indicate the Fill Station Code on one side of a multi-sided Trade Capture Report
→	**1007**	**SideReasonCd**	N	Used to indicate the reason of a multi-sided Trade Capture Report
→	**83**	**RptSeq**	N	Used for order routing to indicate the fill sequence on one side of a multi-sided Trade Capture Report
→	**1008**	**SideTrdSubTyp**	N	Used to support multi-sided orders of different trade types
→	**component block <Parties>**		N	Insert here the set of "Parties" (firm identification) fields

© Copyright, 2006, FIX Protocol, Limited

				defined in "Common Components of Application Messages" Range of values on report:
→	**1**	**Account**	N	Required for executions against electronically submitted orders which were assigned an account by the institution or intermediary
→	**660**	**AcctIDSource**	N	
→	**581**	**AccountType**	N	Specifies type of account
→	**81**	**ProcessCode**	N	Used to specify Step-out trades
→	**1093**	**LotType**	N	
→	**575**	**OddLot**	N	
→	**576**	**NoClearingInstructions**	N	** Nested Repeating Group follows **
→	→	**577** **ClearingInstruction**	N	Required if NoClearingInstructions > 0
→	**578**	**TradeInputSource**	N	
→	**579**	**TradeInputDevice**	N	
→	**821**	**OrderInputDevice**	N	
→	**15**	**Currency**	N	
→	**376**	**ComplianceID**	N	
→	**377**	**SolicitedFlag**	N	
→	**528**	**OrderCapacity**	N	The capacity of the participant for this trade (principal or agent for example).
→	**529**	**OrderRestrictions**	N	Restrictions associated with the participant and their capacity for this trade.
→	**582**	**CustOrderCapacity**	N	The customer capacity for this trade
→	**40**	**OrdType**	N	Order type from the order associated with the trade
→	**18**	**ExecInst**	N	Execution Instruction from the order associated with the trade
→	**483**	**TransBkdTime**	N	A date and time stamp to indicate when this order was booked. For Equities, this is the time at which an order was received by an Exchange or Marketplace. For CIV, this is the time that a Fund Manager booked an order for execution at the next valuation point.
→	**336**	**TradingSessionID**	N	
→	**625**	**TradingSessionSubID**	N	
→	**943**	**TimeBracket**	N	
→	**component block <CommissionData>**		N	Insert here the set of "CommissionData" fields defined in "Common Components of Application Messages"

© Copyright, 2006, FIX Protocol, Limited

				Note: On a fill/partial fill messages, it represents value for that fill/partial fill, on ExecType=Calculated, it represents cumulative value for the order. Monetary commission values are expressed in the currency reflected by the Currency field.
→	**157**	**NumDaysInterest**	N	
→	**230**	**ExDate**	N	
→	**158**	**AccruedInterestRate**	N	
→	**159**	**AccruedInterestAmt**	N	
→	**738**	**InterestAtMaturity**	N	
→	**920**	**EndAccruedInterest Amt**	N	For repurchase agreements the accrued interest on termination.
→	**921**	**StartCash**	N	For repurchase agreements the start (dirty) cash consideration
→	**922**	**EndCash**	N	For repurchase agreements the end (dirty) cash consideration
→	**238**	**Concession**	N	
→	**237**	**TotalTakedown**	N	
→	**118**	**NetMoney**	N	Note: On a fill/partial fill messages, it represents value for that fill/partial fill, on ExecType=Calculated, it represents cumulative value for the order. Value expressed in the currency reflected by the Currency field.
→	**119**	**SettlCurrAmt**	N	Used to report results of forex accommodation trade
→	**120**	**SettlCurrency**	N	Used to report results of forex accommodation trade
→	**155**	**SettlCurrFxRate**	N	Foreign exchange rate used to compute SettlCurrAmt from Currency to SettlCurrency
→	**156**	**SettlCurrFxRateCalc**	N	Specifies whether the SettlCurrFxRate should be multiplied or divided
→	**77**	**PositionEffect**	N	For use in derivatives omnibus accounting
→	**58**	**Text**	N	May be used by the executing market to record any execution Details that are particular to that market
→	**354**	**EncodedTextLen**	N	Must be set if EncodedText field is specified and must immediately precede it.
→	**355**	**EncodedText**	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
→	**752**	**SideMultiLegReportingType**	N	Default is a single security if not specified. Provided to support the scenario where a single leg instrument trades against an individual leg of a multileg instrument.
→	**518**	**NoContAmts**	N	Number of contract details in this message (number of

© Copyright, 2006, FIX Protocol, Limited

					repeating groups to follow)
→	→	**519**	**ContAmtType**	N	Must be first field in the repeating group.
→	→	**520**	**ContAmtValue**	N	
→	→	**521**	**ContAmtCurr**	N	
→	**component block <Stipulations>**			N	
→	**136**	**NoMiscFees**		N	Required if any miscellaneous fees are reported. Indicates number of repeating entries. Repeating group. ** Nested Repeating Group follows **
→	→	**137**	**MiscFeeAmt**	N	Required if NoMiscFees > 0
→	→	**138**	**MiscFeeCurr**	N	
→	→	**139**	**MiscFeeType**	N	Required if NoMiscFees > 0
→	→	**891**	**MiscFeeBasis**	N	
→	**825**	**ExchangeRule**		N	Used to report any exchange rules that apply to this trade.
→	**826**	**TradeAllocIndicator**		N	Identifies if the trade is to be allocated
→	**591**	**PreallocMethod**		N	
→	**70**	**AllocID**		N	Used to assign an ID to the block of preallocations
→	**78**	**NoAllocs**		N	Number of repeating groups for trade allocation
→	→	**79**	**AllocAccount**	N	Required if NoAllocs > 0. Must be first field in repeating group.
→	→	**661**	**AllocAcctIDSource**	N	
→	→	**736**	**AllocSettlCurrency**	N	
→	→	**467**	**IndividualAllocID**	N	
→	→	**component block <NestedParties2>**		N	Insert here the set of "NestedParties2" (firm identification "nested" within additional repeating group) fields defined in "Common Components of Application Messages"
→	→	**80**	**AllocQty**	N	
→	→	**993**	**AllocCustomerCapacity**	N	Can be used for granular reporting of separate allocation detail within a single trade report or allocation message.
→	→	**1002**	**AllocMethod**	N	Specifies the method under which a trade quantity was allocated.
→	→	**989**	**SecondaryIndividualAllocI**	N	Provides support for an intermediary assigned allocation ID

© Copyright, 2006, FIX Protocol, Limited

			D		
→	→	**1136**	**AllocClearingFeeIndicator**	N	
→	**component block <SideTrdRegTS>**			N	Used to indicate the regulatory time stamp on one side of a multi-sided Trade Capture Report.
→	**1072**	**SideGrossTradeAmt**		N	
→	**1057**	**AggressorIndicator**		N	
→	**1139**	**ExchangeSpecialInstructions**		N	
797	CopyMsgIndicator			N	Indicates drop copy.
852	PublishTrdIndicator			N	
853	ShortSaleReason			N	
994	TierCode			N	Indicates the algorithm (tier) used to match a trade
1011	MessageEventSource			N	Used to identify the event or source which gave rise to a message
779	LastUpdateTime			N	Used to indicate reports after a specific time
991	RndPx			N	Specifies the rounded price to quoted precision.
1132	TZTransactTime			N	
1134	ReportedPxDiff			N	The reason(s) for the price difference should be stated by using field (Tag 828) TrdType and, if required, field (Tag 829) TrdSubType as well
381	GrossTradeAmt			N	
	StandardTrailer			Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details

Refer to the FIXML element TrdCaptRpt

© Copyright, 2006, FIX Protocol, Limited

Trade Capture Report Ack

The Trade Capture Report Ack message can be:

- Used to acknowledge trade capture reports received from a counterparty
- Used to reject a trade capture report received from a counterparty

Trade Capture Report Ack

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = AR
571	TradeReportID	N	Unique identifier for the Trade Capture Report
1003	TradeID	N	
1040	SecondaryTradeID	N	
1041	FirmTradeID	N	
1042	SecondaryFirmTradeID	N	
487	TradeReportTransType	N	Identifies Trade Report message transaction type.
856	TradeReportType	N	Indicates action to take on trade
828	TrdType	N	
829	TrdSubType	N	
855	SecondaryTrdType	N	
1123	TradeHandlingInstr	N	
1124	OrigTradeHandlingInstr	N	
1125	OrigTradeDate	N	Used to preserve original trade date when original trade is being referenced in a subsequent trade transaction such as a transfer
1126	OrigTradeID	N	Used to preserve original trade id when original trade is being referenced in a subsequent trade transaction such as a transfer
1127	OrigSecondaryTradeID	N	Used to preserve original secondary trade id when original trade is being referenced in a subsequent trade transaction such as a transfer
830	TransferReason	N	
component block <RootParties>		N	Insert here the set of "Root Parties" (firm identification) fields defined in "common components of application messages" Range of values on report:
150	ExecType	N	Type of Execution being reported: Uses subset of ExecType for Trade Capture Reports
572	TradeReportRefID	N	The TradeReportID that is being referenced for some action, such as correction or cancelation
881	SecondaryTradeReportRefID	N	(Deprecated in FIX.5.0)The SecondaryTradeReportID

© Copyright, 2006, FIX Protocol, Limited

			that is being referenced for some action, such as correction or cancelation
939	TrdRptStatus	N	Status of Trade Report
751	TradeReportRejectReason	N	Reason for Rejection of Trade Report
818	SecondaryTradeReportID	N	(Deprecated in FIX.5.0)
263	SubscriptionRequestType	N	Used to subscribe / unsubscribe for trade capture reports If the field is absent, the value 0 will be the default
820	TradeLinkID	N	Used to associate a group of trades together. Useful for average price calculations.
880	TrdMatchID	N	
17	ExecID	N	Exchanged assigned Execution ID (Trade Identifier)
527	SecondaryExecID	N	
39	OrdStatus	N	
378	ExecRestatementReason	N	
570	PreviouslyReported	N	
423	PriceType	N	
822	UnderlyingTradingSessionID	N	
823	UnderlyingTradingSessionSubID	N	
716	SettlSessID	N	Intraday(ITD), Regular Trading Hours(EOD),
717	SettlSessSubID	N	
854	QtyType	N	
32	LastQty	N	
31	LastPx	N	
component block <Instrument>		Y	Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages"
669	LastParPx	N	
1056	CalculatedCcyLastQty	N	
1071	LastSwapPoints	N	
194	LastSpotRate	N	
195	LastForwardPoints	N	
30	LastMkt	N	
75	TradeDate	N	
715	ClearingBusinessDate	N	
6	AvgPx	N	

© Copyright, 2006, FIX Protocol, Limited

819	AvgPxIndicator		N	
442	MultiLegReportingType		N	
824	TradeLegRefID		N	
60	TransactTime		N	Time ACK was issued by matching system, trading system or counterparty
63	SettlType		N	
711	NoUnderlyings		N	Number of underlyings
→	**component block <UnderlyingInstrument>**		N	Must be provided if Number of underlyings > 0
573	MatchStatus		N	
574	MatchType		N	
797	CopyMsgIndicator		N	
852	PublishTrdIndicator		N	
853	ShortSaleReason		N	
555	NoLegs		N	Number of legs Identifies a Multi-leg Execution if present and non-zero.
→	**component block <InstrumentLeg>**		N	Must be provided if Number of legs > 0
→	**687**	**LegQty**	N	
→	**690**	**LegSwapType**	N	Instead of LegQty – requests that the sellside calculate LegQty based on opposite Leg
→	**990**	**LegReportID**	N	Additional attribute to store the Trade ID of the Leg.
→	**component block <LegStipulations>**		N	
→	**564**	**LegPositionEffect**	N	Provide if the PositionEffect for the leg is different from that specified for the overall multileg security
→	**565**	**LegCoveredOrUncovered**	N	Provide if the CoveredOrUncovered for the leg is different from that specified for the overall multileg security.
→	**component block <NestedParties>**		N	Insert here the set of "Nested Parties" (firm identification "nested" within additional repeating group) fields defined in "Common Components of Application Messages" Used for NestedPartyRole=Leg Clearing Firm/Account, Leg Account/Account Type
→	**654**	**LegRefID**	N	Used to identify a specific leg.
→	**566**	**LegPrice**	N	Provide only if a Price is required for a specific leg. Used for anchoring the overall multileg security price to a specific leg Price.
→	**587**	**LegSettlType**	N	

© Copyright, 2006, FIX Protocol, Limited

→	**588**	**LegSettlDate**	N	Takes precedence over LegSettlmntTyp value and conditionally required/omitted for specific LegSettlType values.
→	**637**	**LegLastPx**	N	Used to report the execution price assigned to the leg of the multileg instrument
→	**675**	**LegSettlCurrency**	N	
→	**1073**	**LegLastForwardPoints**	N	
→	**1074**	**LegCalculatedCcyLastQty**	N	
→	**1075**	**LegGrossTradeAmt**	N	For FX Futures can be used to express the notional value of a trade when LegLastQty and other quantity fields are expressed in terms of number of contracts - LegContractMultiplier (231) is required in this case.
component block <TrdRegTimestamps>			N	
725	ResponseTransportType		N	Ability to specify whether the response to the request should be delivered inband or via pre-arranged out-of-band transport.
726	ResponseDestination		N	URI destination name. Used if ResponseTransportType is out-of-band.
58	Text		N	May be used by the executing market to record any execution Details that are particular to that market
354	EncodedTextLen		N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText		N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
1015	AsOfIndicator		N	Indicates if the trade is an outtrade from a previous day
635	ClearingFeeIndicator		N	
component block <PositionAmountData>			N	Insert here here the set of "Position Amount Data" fields defined in "Common Components of Application Messages"
994	TierCode		N	Indicates the algorithm (tier) used to match a trade
1011	MessageEventSource		N	Used to identify the event or source which gave rise to a message
779	LastUpdateTime		N	Used to indicate reports after a specific time
991	RndPx		N	Specifies the rounded price to quoted precision.
552	NoSides		Y	
→	**54**	**Side**	Y	
→	**37**	**OrderID**	N	

© Copyright, 2006, FIX Protocol, Limited

→	**198**	**SecondaryOrderID**		N	
→	**11**	**ClOrdID**		N	
→	**526**	**SecondaryClOrdID**		N	
→	**66**	**ListID**		N	
→	**component block <Parties>**			N	Insert here here the set of "Parties" fields defined in "Common Components of Application Messages"
→	**1**	**Account**		N	
→	**660**	**AcctIDSource**		N	
→	**581**	**AccountType**		N	
→	**81**	**ProcessCode**		N	
→	**575**	**OddLot**		N	
→	**1093**	**LotType**		N	
→	**576**	**NoClearingInstructions**		N	** Nested Repeating Group follows **
→	→	**577**	**ClearingInstruction**	N	Required if NoClearingInstructions > 0
→	**578**	**TradeInputSource**		N	
→	**579**	**TradeInputDevice**		N	
→	**821**	**OrderInputDevice**		N	
→	**15**	**Currency**		N	
→	**376**	**ComplianceID**		N	
→	**377**	**SolicitedFlag**		N	
→	**528**	**OrderCapacity**		N	
→	**529**	**OrderRestrictions**		N	
→	**582**	**CustOrderCapacity**		N	
→	**40**	**OrdType**		N	
→	**18**	**ExecInst**		N	
→	**483**	**TransBkdTime**		N	
→	**336**	**TradingSessionID**		N	
→	**625**	**TradingSessionSubID**		N	
→	**943**	**TimeBracket**		N	
→	**component block <CommissionData>**			N	Insert here here the set of "Commission Data" fields defined in "Common Components of Application Messages"
→	**157**	**NumDaysInterest**		N	

© Copyright, 2006, FIX Protocol, Limited

→	**230**	**ExDate**		N	
→	**158**	**AccruedInterestRate**		N	
→	**159**	**AccruedInterestAmt**		N	
→	**738**	**InterestAtMaturity**		N	
→	**920**	**EndAccruedInterestAmt**		N	
→	**921**	**StartCash**		N	
→	**922**	**EndCash**		N	
→	**238**	**Concession**		N	
→	**237**	**TotalTakedown**		N	
→	**118**	**NetMoney**		N	
→	**119**	**SettlCurrAmt**		N	
→	**120**	**SettlCurrency**		N	
→	**155**	**SettlCurrFxRate**		N	
→	**156**	**SettlCurrFxRateCalc**		N	
→	**77**	**PositionEffect**		N	
→	**752**	**SideMultiLegReportingType**		N	
→	**518**	**NoContAmts**		N	Number of contract details in this message (number of repeating groups to follow)
→	→	**519**	**ContAmtType**	N	Must be first field in the repeating group.
→	→	**520**	**ContAmtValue**	N	
→	→	**521**	**ContAmtCurr**	N	
→	**component block <Stipulations>**			N	Insert here here the set of "Stipulations" fields defined in "Common Components of Application Messages"
→	**136**	**NoMiscFees**		N	Required if any miscellaneous fees are reported. Indicates number of repeating entries. Repeating group. ** Nested Repeating Group follows **
→	→	**137**	**MiscFeeAmt**	N	Required if NoMiscFees > 0
→	→	**138**	**MiscFeeCurr**	N	
→	→	**139**	**MiscFeeType**	N	Required if NoMiscFees > 0
→	→	**891**	**MiscFeeBasis**	N	
→	**825**	**ExchangeRule**		N	
→	**826**	**TradeAllocIndicator**		N	

© Copyright, 2006, FIX Protocol, Limited

→	**591**	**PreallocMethod**		N	
→	**70**	**AllocID**		N	
→	**78**	**NoAllocs**		N	Number of repeating groups for trade allocation
→	→	**79**	**AllocAccount**	N	Required if NoAllocs > 0. Must be first field in repeating group.
→	→	**661**	**AllocAcctIDSource**	N	
→	→	**736**	**AllocSettlCurrency**	N	
→	→	**467**	**IndividualAllocID**	N	
→	→	**component block <NestedParties2>**		N	Insert here the set of "NestedParties2" (firm identification "nested" within additional repeating group) fields defined in "Common Components of Application Messages"
→	→	**80**	**AllocQty**	N	
→	→	**993**	**AllocCustomerCapacity**	N	Can be used for granular reporting of separate allocation detail within a single trade report or allocation message.
→	→	**1002**	**AllocMethod**	N	Specifies the method under which a trade quantity was allocated.
→	→	**989**	**SecondaryIndividualAllocID**	N	Provides support for an intermediary assigned allocation ID
→	→	**1136**	**AllocClearingFeeIndicator**	N	
→	**1072**	**SideGrossTradeAmt**		N	
→	**1057**	**AggressorIndicator**		N	
→	**1009**	**SideQty**		N	
→	**1005**	**SideTradeReportID**		N	
→	**1006**	**SideFillStationCd**		N	
→	**1007**	**SideReasonCd**		N	
→	**83**	**RptSeq**		N	
→	**1008**	**SideTrdSubTyp**		N	
→	**component block <SideTrdRegTS>**			N	
1135	RptSys			N	
381	GrossTradeAmt			N	
64	SettlDate			N	
	StandardTrailer			Y	

© Copyright, 2006, FIX Protocol, Limited

FIXML Definition for this message – see http://www.fixprotocol.org for details
Refer to the FIXML element TrdCaptRptAck

CATEGORY: REGISTRATION INSTRUCTIONS

Registration Instructions

The Registration Instructions message type may be used by institutions or retail intermediaries wishing to electronically submit registration information to a broker or fund manager (for CIV) for an order or for an allocation.

A Registration Instructions message can be submitted as new, cancel or replace. The RegistTransType field indicates the purpose of the message. When submitting replace or cancel RegistTransType messages the RegistRefID field is required. Replacement Registration Instructions messages must contain all data for the replacement registration.

See VOLUME 7 - "PRODUCT: COLLECTIVE INVESTMENT VEHICLES"

The Registration Instructions message contains repeating fields for each of several joint registrants. The number of registration details instances is indicated in NoRegistDtls. The repeating fields are shown in the message definition below in typeface ***Bold-Italic*** and indented with the → symbol. The field's relative position within the repeating group in the message is important. For example, each instance of registration must be in the order as shown in the message definition below.

The format of the Registration Instructions message is as follows:

Registration Instructions

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = o (lowercase O)
513	RegistID	Y	
514	RegistTransType	Y	
508	RegistRefID	Y	Required for Cancel and Replace RegistTransType messages
11	ClOrdID	N	Unique identifier of the order as assigned by institution or intermediary to which Registration relates
component block <Parties>		N	Insert here the set of "Parties" (firm identification) fields defined in "Common Components of Application Messages"
1	Account	N	
660	AcctIDSource	N	
493	RegistAcctType	N	
495	TaxAdvantageType	N	
517	OwnershipType	N	
473	NoRegistDtls	N	Number of registration details in this message (number of repeating groups to follow)

© Copyright, 2006, FIX Protocol, Limited

→	**509**	**RegistDtls**	N	Must be first field in the repeating group
→	**511**	**RegistEmail**	N	
→	**474**	**MailingDtls**	N	
→	**482**	**MailingInst**	N	
→	**component block <NestedParties>**		N	Insert here the set of "Nested Parties" (firm identification "nested" within additional repeating group) fields defined in "Common Components of Application Messages" Used for NestedPartyRole=InvestorID
→	**522**	**OwnerType**	N	
→	**486**	**DateOfBirth**	N	
→	**475**	**InvestorCountryOfResidence**	N	
510	NoDistribInsts		N	Number of Distribution instructions in this message (number of repeating groups to follow)
→	**477**	**DistribPaymentMethod**	N	Must be first field in the repeating group if NoDistribInsts > 0.
→	**512**	**DistribPercentage**	N	
→	**478**	**CashDistribCurr**	N	
→	**498**	**CashDistribAgentName**	N	
→	**499**	**CashDistribAgentCode**	N	
→	**500**	**CashDistribAgentAcctNumber**	N	
→	**501**	**CashDistribPayRef**	N	
→	**502**	**CashDistribAgentAcctName**	N	
	StandardTrailer		Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details

Refer to the FIXML element RgstInstrctns

© Copyright, 2006, FIX Protocol, Limited

Registration Instructions Response

The Registration Instructions Response message type may be used by broker or fund manager (for CIV) in response to a Registration Instructions message submitted by an institution or retail intermediary for an order or for an allocation.

The Registration Instructions Response message is used to:

1. confirm the receipt of a Registration Instructions message
2. confirm changes to an existing Registration Instructions message (i.e. accept cancel and replace requests)
3. relay Registration Instructions status information
4. relay assigned client and account Ids for Registration Instructions messages with RegTransType=New
5. reject Registration Instructions message

Each Registration Instructions Response message contains a RegistStatus field which is used to communicate the current state of the Registration Instructions as understood by the broker or fund manager. The Registration Instruction statuses are as follows (in highest to lowest precedence):

RegistStatus	**Description**
Accepted	Registration details are acceptable to the receiving broker, intermediary or fund manager. Assigned client and account Ids may be returned.
Rejected	Registration details have been rejected by the receiving broker, intermediary or fund manager.
Held	Registration details have been held by the receving broker, intermediary or fund manager. Assigned (possibly provisional) client and account Ids may be returned.

The format of the Registration Instructions Response message is as follows:

Registration Instructions Response

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = p (lowercase P)
513	RegistID	Y	Unique identifier of the original Registration Instructions details
514	RegistTransType	Y	Identifies original Registration Instructions transaction type
508	RegistRefID	Y	Required for Cancel and Replace RegistTransType messages
11	ClOrdID	N	Unique identifier of the order as assigned by institution or intermediary.
component block <Parties>		N	Insert here the set of "Parties" (firm identification) fields

© Copyright, 2006, FIX Protocol, Limited

			defined in "Common Components of Application Messages"
1	Account	N	
660	AcctIDSource	N	
506	RegistStatus	Y	
507	RegistRejReasonCode	N	
496	RegistRejReasonText	N	
	StandardTrailer	Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details
Refer to the FIXML element RgstInstrctnsRsp

CATEGORY: POSITIONS MAINTENANCE

Overview

Clearing Services for Position Management

The Position Management Clearing Services can be used to invoke the following business functions. If requested, message-based response confirmations will be provided to the client.

1. Position Change Submission (Final Position Instructions)
2. Position Adjustment
3. Exercise Notice
4. Abandonment Notice
5. Margin Disposition
6. Position Pledge
7. Request for Position

Clearing Services for Post-Trade Processing

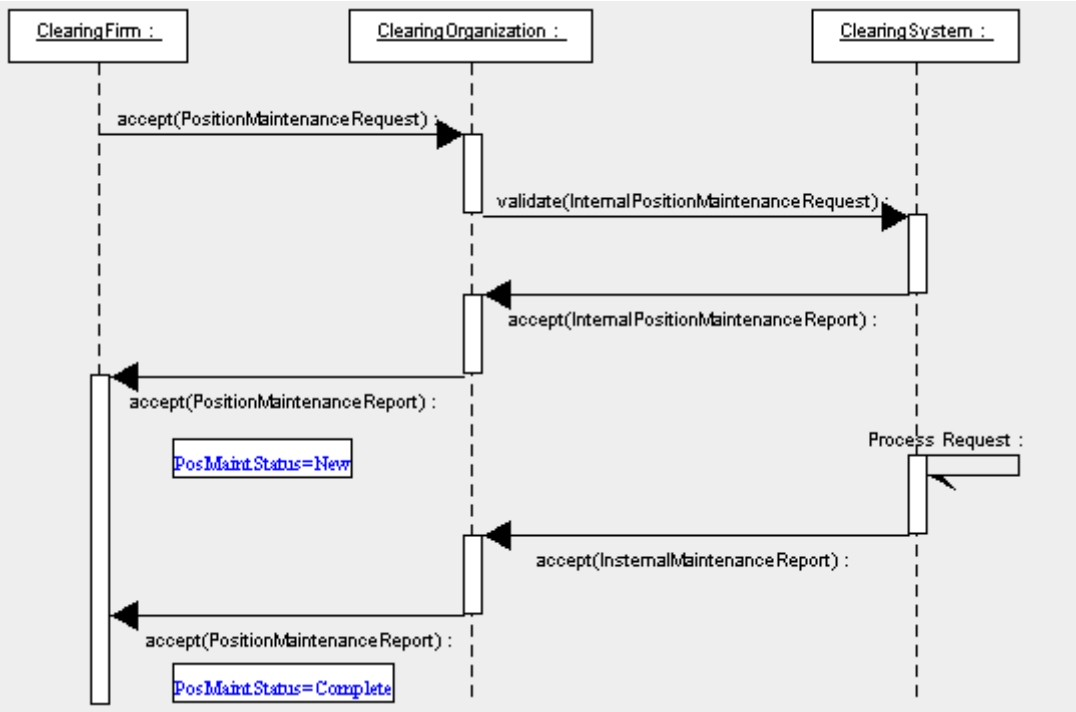
The Post-Trade Processing Clearing Services can be used to invoke the following business functions. If requested, message-based response confirmations will be provided to the client.

1. ETP message format: Trade Change
2. Give-up message format: Allocation, Accept, Reject, Release, Change, Delete
3. Exchange for Physical (EFP) message format: Allocation, Accept, Reject, Change, Delete
4. Average Price (APS) message format: Allocation, Accept, Change, Delete
5. Mutual Offset (MOS) message format: Allocation, Accept, Reject, Change, Delete
6. Trade Entry Edit message format: Trade Add, Transfer, Change

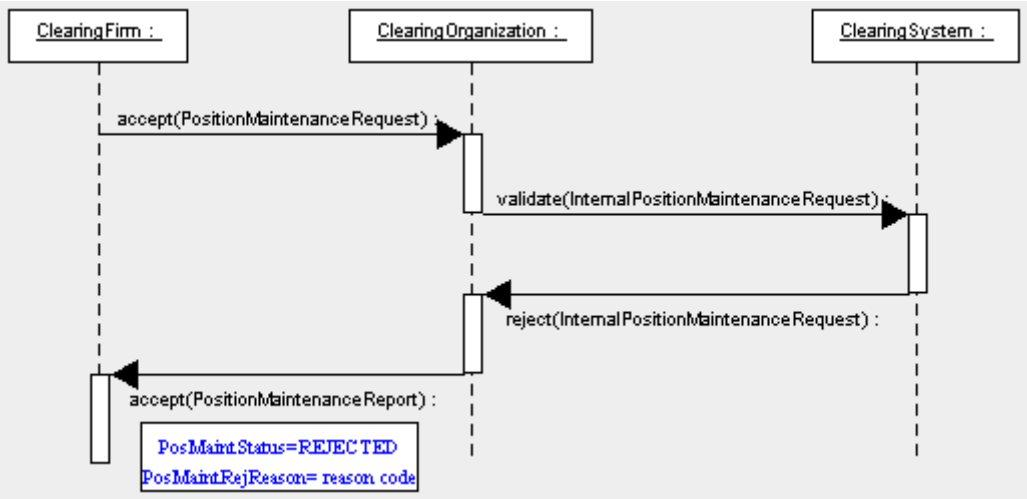
© Copyright, 2006, FIX Protocol, Limited

Position Maintenance Sequence Diagrams

Nominal Scenario - Valid Position Maintenance Request Accepted



Alternative Scenario - Invalid Position Maintenance Request - Rejected



Position Maintenance Request

The Position Maintenance Request message allows the position owner to submit requests to the holder of a position which will result in a specific action being taken which will affect the position. Generally, the holder of the position is a central counter party or clearing organization but can also be a party providing investment services. Submission of a request may result in the following:

- adjustement of both the long and short start of day position quantity
- exercise of an option position into a position in the instrument underlying the option
- abandonment of an option position that would otherwise exercise
- netting of current day trades to change to the end of day long and short position
- spreading of a position against other position in order to reduce margin requirements
- pledge of a position for collateral purposes
- large trader submission of the long and short quantities

The request may be submitted as either new, replace or cancel and may refer to a specific position or the previously submitted message. The request is always submitted as of a Clearing Business Date and is therefore required. The parties both owning and holding the position are specified in the parties block.

Position Maintenance Request

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = AL
710	PosReqID	N	Unique identifier for the position maintenance request as assigned by the submitter. Conditionally required when used in a request/reply scenario (i.e. not required in batch scenario)
709	PosTransType	Y	
712	PosMaintAction	Y	
713	OrigPosReqRefID	N	Reference to the PosReqID of a previous maintenance request that is being replaced or canceled.
714	PosMaintRptRefID	N	Reference to a PosMaintRptID from a previous Position Maintenance Report that is being replaced or canceled.
715	ClearingBusinessDate	Y	The Clearing Business Date referred to by this maintenance request
716	SettlSessID	N	
717	SettlSessSubID	N	
component block <Parties>		Y	The Following PartyRoles can be specified: ClearingOrganization Clearing Firm Position Account
1	Account	N	

© Copyright, 2006, FIX Protocol, Limited

660	AcctIDSource	N	
581	AccountType	N	Type of account associated with the order (Origin)
component block <Instrument>		Y	
15	Currency	N	
555	NoLegs	N	Number of legs Identifies a Multi-leg Execution if present and non-zero.
→	**component block <InstrumentLeg>**	N	Must be provided if Number of legs > 0
711	NoUnderlyings	N	Number of underlyings
→	**component block <UnderlyingInstrument>**	N	Must be provided if Number of underlyings > 0
386	NoTradingSessions	N	Specifies the number of repeating TradingSessionIDs
→	**336 TradingSessionID**	N	Required if NoTradingSessions is > 0.
→	**625 TradingSessionSubID**	N	
60	TransactTime	N	Time this order request was initiated/released by the trader, trading system, or intermediary.
component block <PositionQty>		Y	
component block <PositionAmountData>		N	
718	AdjustmentType	N	Type of adjustment to be applied, used for PCS & PAJ Delta_plus, Delta_minus, Final, If Adjustment Type is null, the request will be processed as Margin Disposition
719	ContraryInstructionIndicator	N	Boolean - if Y then indicates you are requesting a position maintenance that acting
720	PriorSpreadIndicator	N	Boolean - Y indicates you are requesting rollover of prior day's spread submissions
834	ThresholdAmount	N	
58	Text	N	
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
120	SettlCurrency	N	
	StandardTrailer	Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details

© Copyright, 2006, FIX Protocol, Limited

Refer to the FIXML element PosMntReq

© Copyright, 2006, FIX Protocol, Limited

Position Maintenance Report

The Position Maintenance Report message is sent by the holder of a positon in response to a Position Maintenance Request and is used to confirm that a request has been successfully processed or rejected.

Position Maintenance Report

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = AM
721	PosMaintRptID	Y	Unique identifier for this position report
709	PosTransType	Y	
710	PosReqID	N	Unique identifier for the position maintenance request associated with this report
712	PosMaintAction	Y	
713	OrigPosReqRefID	N	Reference to the PosReqID of a previous maintenance request that is being replaced or canceled.
722	PosMaintStatus	Y	Status of Position Maintenance Request
723	PosMaintResult	N	
715	ClearingBusinessDate	Y	The Clearing Business Date covered by this request
716	SettlSessID	N	Intraday(ITD), Regular Trading Hours(EOD),
717	SettlSessSubID	N	
component block <Parties>		N	Position Account
1	Account	N	
660	AcctIDSource	N	
581	AccountType	N	Type of account associated with the order (Origin)
714	PosMaintRptRefID	N	Reference to a PosMaintRptID (Tag 721) from a previous Position Maintenance Report that is being replaced or canceled
component block <Instrument>		Y	
15	Currency	N	
120	SettlCurrency	N	
719	ContraryInstructionIndicator	N	Can be set to true when a position maintenance request is being performed contrary to current money position, i.e. for an exercise of an out of the money position or an abandonement (do not exercise) of an in the money position
720	PriorSpreadIndicator	N	
555	NoLegs	N	Number of legs Identifies a Multi-leg Execution if present and non-zero.
→	**component block**	N	Must be provided if Number of legs > 0

© Copyright, 2006, FIX Protocol, Limited

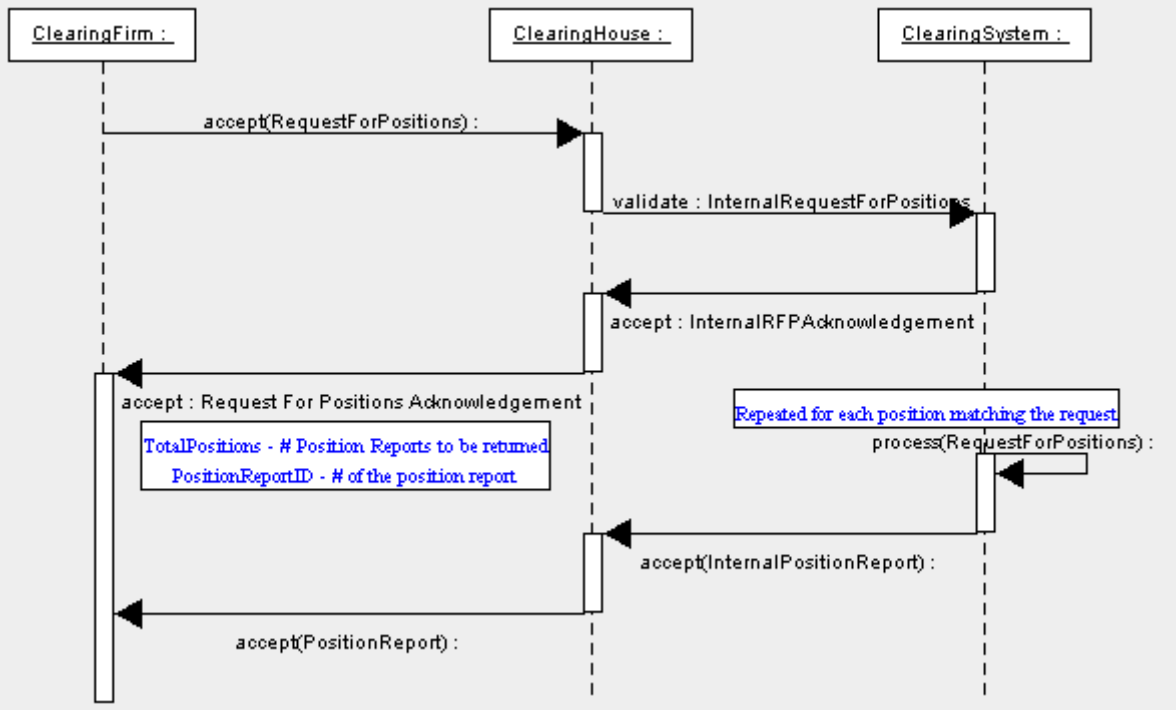
	<InstrumentLeg>			
711	NoUnderlyings		N	Number of underlyings
→	**component block <UnderlyingInstrument>**		N	Must be provided if Number of underlyings > 0
386	NoTradingSessions		N	Specifies the number of repeating TradingSessionIDs
→	**336**	**TradingSessionID**	N	Required if NoTradingSessions is > 0.
→	**625**	**TradingSessionSubID**	N	
60	TransactTime		N	Time this order request was initiated/released by the trader, trading system, or intermediary. Conditionally required except when requests for reports are processed in batch, transaction time is not available, or when PosReqID is not present.
component block <PositionQty>			Y	See definition for Position Quantity in the Proposed Component Block section above
component block <PositionAmountData>			N	Insert here here the set of "Position Amount Data" fields defined in "Common Components of Application Messages"
718	AdjustmentType		N	Type of adjustment to be applied Delta_plus, Delta_minus, Final. If Adjustment Type is null, the PCS request will be processed as Margin Disposition only
834	ThresholdAmount		N	
58	Text		N	
354	EncodedTextLen		N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText		N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
	StandardTrailer		Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details
Refer to the FIXML element PosMntRpt

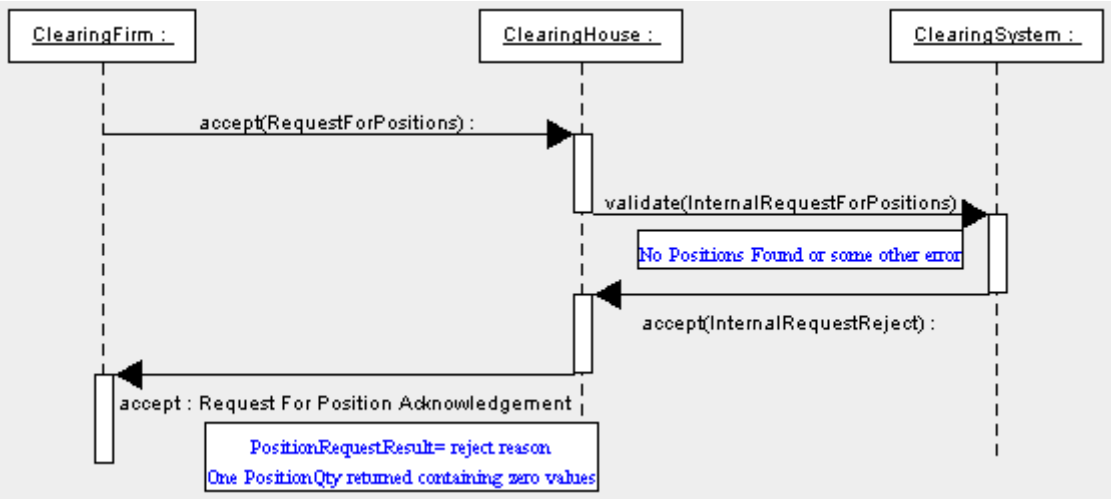
© Copyright, 2006, FIX Protocol, Limited

Request for Positions Sequence Diagrams

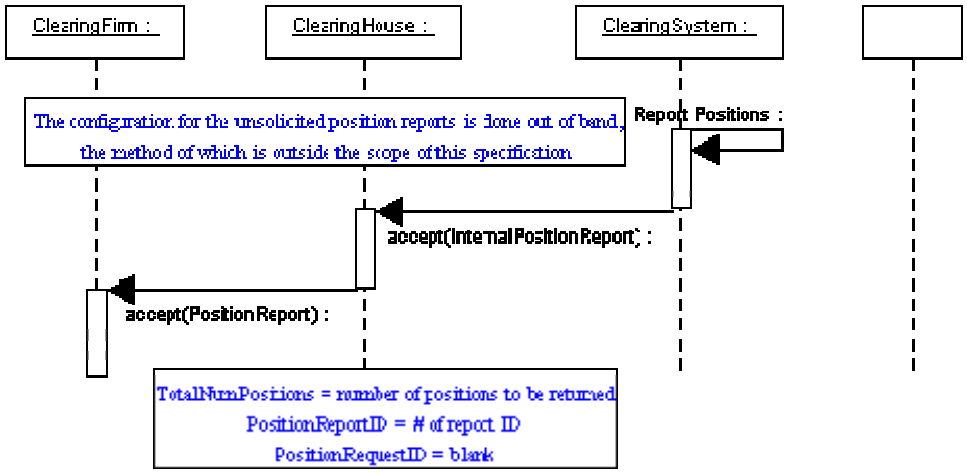
Nominal Scenario - Request for Positions



Alternative Scenario - Invalid Request for Positions



Alternative Scenario - Unsolicited Position Reports



© Copyright, 2006, FIX Protocol, Limited

Request For Positions

The Request For Positions message is used by the owner of a position to request a Position Report from the holder of the position, usually the central counter party or clearing organization. The request can be made at several levels of granualarity.

- Position Report only
- Positions and related Trades
- Exercises only
- Assignments only
- Settlements activity

The message can be used to request a one time snapshot of positions or to subscribe to updates as they occur using the SubscriptionRequestType (tag 263). The ResponseTransportType (tag 725) can be used to specify if the reports are to be sent inband over the session transport or out-of-band of band over an alternative transport such as FTP.

Request For Positions

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = AN
710	PosReqID	Y	Unique identifier for the Request for Positions as assigned by the submitter
724	PosReqType	Y	
573	MatchStatus	N	
263	SubscriptionRequestType	N	Used to subscribe / unsubscribe for trade capture reports If the field is absent, the value 0 will be the default
120	SettlCurrency	N	
component block <Parties>		Y	Position Account
1	Account	N	
660	AcctIDSource	N	
581	AccountType	N	Type of account associated with the order (Origin)
component block <Instrument>		N	
15	Currency	N	
555	NoLegs	N	Number of legs Identifies a Multi-leg Execution if present and non-zero.
→	**component block <InstrumentLeg>**	N	Must be provided if Number of legs > 0
711	NoUnderlyings	N	Number of underlyings
→	**component block <UnderlyingInstrument>**	N	Must be provided if Number of underlyings > 0

© Copyright, 2006, FIX Protocol, Limited

715	ClearingBusinessDate		Y	The Clearing Business Date referred to by this request
716	SettlSessID		N	Intraday(ITD), Regular Trading Hours(EOD)
717	SettlSessSubID		N	
386	NoTradingSessions		N	Specifies the number of repeating TradingSessionIDs
→	**336**	**TradingSessionID**	N	Required if NoTradingSessions is > 0.
→	**625**	**TradingSessionSubID**	N	
60	TransactTime		Y	Time this order request was initiated/released by the trader, trading system, or intermediary.
725	ResponseTransportType		N	Ability to specify whether the response to the request should be delivered inband or via pre-arranged out-of-band transport.
726	ResponseDestination		N	URI destination name. Used if ResponseTransportType is out-of-band.
58	Text		N	
354	EncodedTextLen		N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText		N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
	StandardTrailer		Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details
Refer to the FIXML element ReqForPoss

© Copyright, 2006, FIX Protocol, Limited

Request for Positions Ack

The Request for Positions Ack message is returned by the holder of the position in response to a Request for Positions message. The purpose of the message is to acknowledge that a request has been received and is being processed.

Request for Positions Ack

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = AO
721	PosMaintRptID	Y	Unique identifier for this position report
710	PosReqID	N	Unique identifier for the Request for Position associated with this report This field should not be provided if the report was sent unsolicited.
727	TotalNumPosReports	N	Total number of Position Reports being returned
325	UnsolicitedIndicator	N	Set to 'Y' if message is sent as a result of a subscription request or out of band configuration as opposed to a Position Request.
728	PosReqResult	Y	
729	PosReqStatus	Y	
724	PosReqType	N	
573	MatchStatus	N	
715	ClearingBusinessDate	N	
263	SubscriptionRequestType	N	
716	SettlSessID	N	
717	SettlSessSubID	N	
120	SettlCurrency	N	
component block <Parties>		Y	Position Account
1	Account	N	
660	AcctIDSource	N	
581	AccountType	N	Type of account associated with the order (Origin)
component block <Instrument>		N	
15	Currency	N	
555	NoLegs	N	Number of legs Identifies a Multi-leg Execution if present and non-zero.
→	**component block <InstrumentLeg>**	N	Must be provided if Number of legs > 0
711	NoUnderlyings	N	Number of underlyings

© Copyright, 2006, FIX Protocol, Limited

→	**component block <UnderlyingInstrument>**	N	Must be provided if Number of underlyings > 0
725	ResponseTransportType	N	Ability to specify whether the response to the request should be delivered inband or via pre-arranged out-of-band transport.
726	ResponseDestination	N	URI destination name. Used if ResponseTransportType is out-of-band.
58	Text	N	
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
	StandardTrailer	Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details

Refer to the FIXML element ReqForPossAck

© Copyright, 2006, FIX Protocol, Limited

Position Report

The Position Report message is returned by the holder of a position in response to a Request for Position message. The purpose of the message is to report all aspects of a position and may be provided on a standing basis to report end of day positions to an owner.

Position Report

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = AP
721	PosMaintRptID	Y	Unique identifier for this position report
710	PosReqID	N	Unique identifier for the Request for Positions associated with this report This field should not be provided if the report was sent unsolicited.
724	PosReqType	N	
263	SubscriptionRequestType	N	Used to subscribe / unsubscribe for trade capture reports If the field is absent, the value 0 will be the default
727	TotalNumPosReports	N	Total number of Position Reports being returned
728	PosReqResult	N	Result of a Request for Position
325	UnsolicitedIndicator	N	Set to 'Y' if message is sent as a result of a subscription request or out of band configuration as opposed to a Position Request.
715	ClearingBusinessDate	Y	The Clearing Business Date referred to by this maintenance request
716	SettlSessID	N	
717	SettlSessSubID	N	
423	PriceType	N	
120	SettlCurrency	N	
1011	MessageEventSource	N	Used to identify the event or source which gave rise to a message
component block <Parties>		Y	Position Account
1	Account	N	Account may also be specified through via Parties Block using Party Role 27 which signifies Account
660	AcctIDSource	N	
581	AccountType	N	Type of account associated with the order (Origin). Account may also be specified through via Parties Block using Party Role 27 which signifies Account
component block <Instrument>		N	
15	Currency	N	
730	SettlPrice	N	

© Copyright, 2006, FIX Protocol, Limited

731	SettlPriceType		N	Values = Final, Theoretical
734	PriorSettlPrice		N	
573	MatchStatus		N	Used to indicate if a Position Report is matched or unmatched
555	NoLegs		N	Number of legs Identifies a Multi-leg Execution if present and non-zero.
→	**component block <InstrumentLeg>**		N	Must be provided if Number of legs > 0
711	NoUnderlyings		N	
→	**component block <UnderlyingInstrument>**		N	Insert here the set of "Underlying Instrument" (underlying symbology) fields defined in "Common Components of Application Messages" Required if NoUnderlyings > 0
→	**732**	**UnderlyingSettlPrice**	N	
→	**733**	**UnderlyingSettlPriceType**	N	Values = Final, Theoretical
→	**1037**	**UnderlyingDeliveryAmount**	N	
→	**component block <UnderlyingAmount>**		N	Insert here the set of "Underlying Amount" fields defined in "Common Components of Application Messages"
component block <PositionQty>			N	Insert here the set of "Position Qty" fields defined in "Common Components of Application Messages"
component block <PositionAmountData>			N	Insert here the set of "Position Amount Data" fields defined in "Common Components of Application Messages"
506	RegistStatus		N	RegNonRegInd
743	DeliveryDate		N	
58	Text		N	
354	EncodedTextLen		N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText		N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
	StandardTrailer		Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details

Refer to the FIXML element PosRpt

Adjusted Position Report

© Copyright, 2006, FIX Protocol, Limited

Used to report changes in position, primarily in equity options, due to modifications to the underlying due to corporate actions

Adjusted Position Report

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = BL
721	PosMaintRptID	Y	Unique identifier for this Adjusted Position report
724	PosReqType	N	
715	ClearingBusinessDate	Y	The Clearing Business Date referred to by this maintenance request
716	SettlSessID	N	
714	PosMaintRptRefID	N	
component block <Parties>		Y	Position Account
component block <PositionQty>		Y	Insert here here the set of "Position Qty" fields defined in "Common Components of Application Messages"
component block <Instrument>		N	
730	SettlPrice	N	Settlement Price
734	PriorSettlPrice	N	Prior Settlement Price
	StandardTrailer	Y	

© Copyright, 2006, FIX Protocol, Limited

Assignment Report

Assignment Reports are sent from a clearing house to counterparties, such as a clearing firm as a result of the assignment process.Communication Scenarios

Assignment Report can be sent unsolicited from the clearing house to a clearing firm.

Assignment Report can be returned in response to a Request for Positions message with a PosReqType(tag 724) set to 3 (Assignment).

Assignment Report

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = AW
833	AsgnRptID	Y	Unique identifier for the Assignment report
832	TotNumAssignmentReports	N	Total Number of Assignment Reports being returned to a firm
912	LastRptRequested	N	
component block <Parties>		Y	Clearing Organization Clearing Firm Contra Clearing Organization Contra Clearing Firm Position Account
1	Account	N	Customer Account
581	AccountType	N	Type of account associated with the order (Origin)
component block <Instrument>		N	CFI Code - Market Indicator (col 4) used to indicate Market of Assignment
15	Currency	N	
555	NoLegs	N	Number of legs Identifies a Multi-leg Execution if present and non-zero.
→	**component block <InstrumentLeg>**	N	Must be provided if Number of legs > 0
711	NoUnderlyings	N	Number of underlyings
→	**component block <UnderlyingInstrument>**	N	Must be provided if Number of underlyings > 0
component block <PositionQty>		N	"Insert here here the set of "Position Qty" fields defined in "Common Components of Application Messages"
component block <PositionAmountData>		N	Insert here here the set of "Position Amount Data" fields defined in "Common Components of Application Messages"
834	ThresholdAmount	N	
730	SettlPrice	N	Settlement Price of Option
731	SettlPriceType	N	Values = Final, Theoretical

© Copyright, 2006, FIX Protocol, Limited

732	UnderlyingSettlPrice	N	Settlement Price of Underlying
734	PriorSettlPrice	N	
432	ExpireDate	N	Expiration Date of Option
744	AssignmentMethod	N	Method under which assignment was conducted Valid values: R = Random P = ProRata
745	AssignmentUnit	N	Quantity Increment used in performing assignment
746	OpenInterest	N	Open interest that was eligible for assignment
747	ExerciseMethod	N	Exercise Method used to in performing assignment Values = Automatic, Manual
716	SettlSessID	N	Settlement Session - EOD or Intraday
717	SettlSessSubID	N	Settlement Session enumerator
715	ClearingBusinessDate	Y	Business date of assignment
58	Text	N	
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
	StandardTrailer	Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details

Refer to the FIXML element AsgnRpt

© Copyright, 2006, FIX Protocol, Limited

Contrary Intention Report

The Contrary Intention Report is used for reporting of contrary expiration quantities for Saturday expiring options. This information is required by options exchanges for regulatory purposes.

Contrary Intention Report

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = BO
977	ContIntRptID	Y	Unique identifier for the Contrary Intention report
60	TransactTime	N	Time the contrary intention was received by clearing organization.
978	LateIndicator	N	Indicates if the contrary intention was received after the exchange imposed cutoff time
979	InputSource	N	Source of the contrary intention
715	ClearingBusinessDate	Y	Business date of contrary intention
component block <Parties>		Y	Clearing Organization Clearing Firm Position Account
component block <ExpirationQty>		Y	Expiration Quantities
component block <Instrument>		Y	
711	NoUnderlyings	N	Number of underlyings
→	**component block <UnderlyingInstrument>**	N	Must be provided if Number of underlyings > 0
58	Text	N	
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
	StandardTrailer	Y	

© Copyright, 2006, FIX Protocol, Limited

CATEGORY: COLLATERAL MANAGEMENT

Overview

A set of collateral management messages are provided to manage collateral associated with positions resulting from trading activity. The Collateral Management messages have been designed to address both two party and three party interaction. The two party model addresses communication between two counterparties to a trade. The three party model supports communication involving an intermediary acting as a facilitator or guarantor to the trade, such as a clearing house or ATS.

The following messages are provided to support collateral management transactions.

Collateral Request

Request collateral from counterparty

The response to the Collateral Request message is a Collateral Assignment message

Collateral Assignment

Used to make assignment, replenishment, or substitution to collateral for a trade

The response to a Collateral Assignment message is a Collateral Response message

Collateral Response

Reply from recipient (or market) to a Collateral Assignment message

Collateral Report

Reports status of collateral

Collateral Inquiry

Query collateral

Multiple criteria supported

The response to a Collateral Inquiry is one or more Collateral Report messages

Collateral Management Usage

Collateral management messages have been designed for the following uses:

Securities financing (such as Repurchase Agreements and Securities lending)

Clearing House collateralization

© Copyright, 2006, FIX Protocol, Limited

Collateral Request

An initiator that requires collateral from a respondent sends a Collateral Request. The initiator can be either counterparty to a trade in a two party model or an intermediary such as an ATS or clearinghouse in a three party model. A Collateral Assignment is expected as a response to a request for collateral.

Collateral Request

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = AX
894	CollReqID	Y	Unique identifier for collateral request
895	CollAsgnReason	Y	Reason collateral assignment is being requested
60	TransactTime	Y	
126	ExpireTime	N	Time until when Respondent has to assign collateral
component block <Parties>		N	
1	Account	N	Customer Account
581	AccountType	N	Type of account associated with the order (Origin)
11	ClOrdID	N	Identifier fo order for which collateral is required
37	OrderID	N	Identifier fo order for which collateral is required
198	SecondaryOrderID	N	Identifier fo order for which collateral is required
526	SecondaryClOrdID	N	Identifier fo order for which collateral is required
124	NoExecs	N	Executions for which collateral is required
→	**17 ExecID**	N	Required if NoExecs > 0
897	NoTrades	N	Trades for which collateral is required
→	**571 TradeReportID**	N	Required if NoTrades > 0
→	**818 SecondaryTradeReportID**	N	
component block <Instrument>		N	Instrument that was traded for which collateral is required
component block <FinancingDetails>		N	Details of the Agreement and Deal
64	SettlDate	N	
53	Quantity	N	
854	QtyType	N	
15	Currency	N	
555	NoLegs	N	Number of legs Identifies a Multi-leg Execution if present and non-zero.
→	**component block <InstrumentLeg>**	N	Must be provided if Number of legs > 0

© Copyright, 2006, FIX Protocol, Limited

711	NoUnderlyings		N	Number of legs that make up the Security
→	**component block <UnderlyingInstrument>**		N	Insert here the set of "Underlying Instrument" fields defined in "Common Components of Application Messages" Required if NoUnderlyings > 0
→	**944**	**CollAction**	N	Required if NoUnderlyings > 0
899	MarginExcess		N	
900	TotalNetValue		N	
901	CashOutstanding		N	
component block <TrdRegTimestamps>			N	Insert here the set of "TrdRegTimestamps" fields defined in "Common Components of Application Messages"
54	Side		N	
136	NoMiscFees		N	Required if any miscellaneous fees are reported. Indicates number of repeating entries. Repeating group. ** Nested Repeating Group follows **
→	**137**	**MiscFeeAmt**	N	Required if NoMiscFees > 0
→	**138**	**MiscFeeCurr**	N	
→	**139**	**MiscFeeType**	N	Required if NoMiscFees > 0
→	**891**	**MiscFeeBasis**	N	
44	Price		N	
423	PriceType		N	
159	AccruedInterestAmt		N	
920	EndAccruedInterestAmt		N	
921	StartCash		N	
922	EndCash		N	
component block <SpreadOrBenchmarkCurveData>			N	Insert here the set of "SpreadOrBenchmarkCurveData" fields defined in "Common Components of Application Messages"
component block <Stipulations>			N	Insert here the set of "Stipulations" fields defined in "Common Components of Application Messages"
336	TradingSessionID		N	Trading Session in which trade occurred
625	TradingSessionSubID		N	Trading Session Subid in which trade occurred
716	SettlSessID		N	
717	SettlSessSubID		N	
715	ClearingBusinessDate		N	
58	Text		N	
354	EncodedTextLen		N	Must be set if EncodedText field is specified and must

© Copyright, 2006, FIX Protocol, Limited

			immediately precede it.
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
	StandardTrailer	Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details
Refer to the FIXML element CollReq

Collateral Assignment

Used to assign collateral to cover a trading position. This message can be sent unsolicited or in reply to a Collateral Request message.

The Collateral Assignment message can be used to perform the following:

- Assign initial collateral
- Replace collateral

Collateral Assignment

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = AY
902	CollAsgnID	Y	Unique Identifer for collateral assignment
894	CollReqID	N	Identifer of CollReqID to which the Collateral Assignment is in response
895	CollAsgnReason	Y	Reason for collateral assignment
903	CollAsgnTransType	Y	Collateral Transaction Type
907	CollAsgnRefID	N	Collateral assignment to which this transaction refers
60	TransactTime	Y	
126	ExpireTime	N	For an Initial assignment, time by which a response is expected
component block <Parties>		N	
1	Account	N	Customer Account
581	AccountType	N	Type of account associated with the order (Origin)
11	ClOrdID	N	Identifier fo order for which collateral is required
37	OrderID	N	Identifier fo order for which collateral is required
198	SecondaryOrderID	N	Identifier fo order for which collateral is required
526	SecondaryClOrdID	N	Identifier fo order for which collateral is required
124	NoExecs	N	Executions for which collateral is required
→	**17** **ExecID**	N	Required if NoExecs > 0
897	NoTrades	N	Trades for which collateral is required
→	**571** **TradeReportID**	N	Required if NoTrades > 0
→	**818** **SecondaryTradeReportID**	N	
component block <Instrument>		N	Insert here the set of "Instrument" fields defined in "Common Components of Application Messages"
component block <FinancingDetails>		N	Insert here the set of "FinancingDetails" fields defined in "Common Components of Application Messages"
64	SettlDate	N	

© Copyright, 2006, FIX Protocol, Limited

53	Quantity		N	
854	QtyType		N	
15	Currency		N	
555	NoLegs		N	Number of legs Identifies a Multi-leg Execution if present and non-zero.
→	**component block <InstrumentLeg>**		N	Must be provided if Number of legs > 0
711	NoUnderlyings		N	Number of legs that make up the Security
→	**component block <UnderlyingInstrument>**		N	Insert here the set of "Underlying Instrument" fields defined in "Common Components of Application Messages" Required if NoUnderlyings > 0
→	**944**	**CollAction**	N	Required if NoUnderlyings > 0
899	MarginExcess		N	
900	TotalNetValue		N	
901	CashOutstanding		N	
component block <TrdRegTimestamps>			N	Insert here the set of "TrdRegTimestamps" fields defined in "Common Components of Application Messages"
54	Side		N	
136	NoMiscFees		N	Required if any miscellaneous fees are reported. Indicates number of repeating entries. Repeating group. ** Nested Repeating Group follows **
→	**137**	**MiscFeeAmt**	N	Required if NoMiscFees > 0
→	**138**	**MiscFeeCurr**	N	
→	**139**	**MiscFeeType**	N	Required if NoMiscFees > 0
→	**891**	**MiscFeeBasis**	N	
44	Price		N	
423	PriceType		N	
159	AccruedInterestAmt		N	
920	EndAccruedInterestAmt		N	
921	StartCash		N	
922	EndCash		N	
component block <SpreadOrBenchmarkCurveData>			N	Insert here the set of "SpreadOrBenchmarkCurveData" fields defined in "Common Components of Application Messages"
component block <Stipulations>			N	Insert here the set of "Stipulations" fields defined in "Common Components of Application Messages"

© Copyright, 2006, FIX Protocol, Limited

component block <SettlInstructionsData>		N	Insert here the set of "SettlInstructionsData" fields defined in "Common Components of Application Messages"
336	TradingSessionID	N	Trading Session in which trade occurred
625	TradingSessionSubID	N	Trading Session Subid in which trade occurred
716	SettlSessID	N	
717	SettlSessSubID	N	
715	ClearingBusinessDate	N	
58	Text	N	
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
	StandardTrailer	Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details

Refer to the FIXML element CollAsgn

Collateral Response

Used to respond to a Collateral Assignment message.

Collateral Response

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = AZ
904	CollRespID	Y	Unique identifer for the collateral response
902	CollAsgnID	N	Conditionally required when responding to a Collateral Assignment message
894	CollReqID	N	Identifer of CollReqID to which the Collateral Assignment is in response
895	CollAsgnReason	N	Conditionally required when responding to a Collateral Assignment message
903	CollAsgnTransType	N	Collateral Transaction Type - not recommended because it causes confusion
905	CollAsgnRespType	Y	Collateral Assignment Response Type
906	CollAsgnRejectReason	N	Reason Colllateral Assignment was rejected
60	TransactTime	Y	
1043	CollApplType	N	
291	FinancialStatus	N	Tells whether security has been restricted.
715	ClearingBusinessDate	N	
component block <Parties>		N	
1	Account	N	Customer Account
581	AccountType	N	Type of account associated with the order (Origin)
11	ClOrdID	N	Identifier fo order for which collateral is required
37	OrderID	N	Identifier fo order for which collateral is required
198	SecondaryOrderID	N	Identifier fo order for which collateral is required
526	SecondaryClOrdID	N	Identifier fo order for which collateral is required
124	NoExecs	N	Executions for which collateral is required
→	**17 ExecID**	N	Required if NoExecs > 0
897	NoTrades	N	Trades for which collateral is required
→	**571 TradeReportID**	N	Required if NoTrades > 0
→	**818 SecondaryTradeReportID**	N	
component block <Instrument>		N	Insert here the set of "Instrument" fields defined in "Common Components of Application Messages"
component block <FinancingDetails>		N	Insert here the set of "FinancingDetails" fields defined in

© Copyright, 2006, FIX Protocol, Limited

				"Common Components of Application Messages"
64	SettlDate		N	
53	Quantity		N	
854	QtyType		N	
15	Currency		N	
555	NoLegs		N	Number of legs Identifies a Multi-leg Execution if present and non-zero.
→	**component block <InstrumentLeg>**		N	Must be provided if Number of legs > 0
711	NoUnderlyings		N	Number of legs that make up the Security
→	**component block <UnderlyingInstrument>**		N	Insert here the set of "Underlying Instrument" fields defined in "Common Components of Application Messages" Required if NoUnderlyings > 0
→	**944**	**CollAction**	N	Required if NoUnderlyings > 0
899	MarginExcess		N	
900	TotalNetValue		N	
901	CashOutstanding		N	
component block <TrdRegTimestamps>			N	Insert here the set of "TrdRegTimestamps" fields defined in "Common Components of Application Messages"
54	Side		N	
136	NoMiscFees		N	Required if any miscellaneous fees are reported. Indicates number of repeating entries. Repeating group. ** Nested Repeating Group follows **
→	**137**	**MiscFeeAmt**	N	Required if NoMiscFees > 0
→	**138**	**MiscFeeCurr**	N	
→	**139**	**MiscFeeType**	N	Required if NoMiscFees > 0
→	**891**	**MiscFeeBasis**	N	
44	Price		N	
423	PriceType		N	
159	AccruedInterestAmt		N	
920	EndAccruedInterestAmt		N	
921	StartCash		N	
922	EndCash		N	
component block <SpreadOrBenchmarkCurveData>			N	Insert here the set of "SpreadOrBenchmarkCurveData" fields defined in "Common Components of Application Messages"

© Copyright, 2006, FIX Protocol, Limited

component block <Stipulations>		N	Insert here the set of "Stipulations" fields defined in "Common Components of Application Messages"
58	Text	N	
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
	StandardTrailer	Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details
Refer to the FIXML element CollRsp

Collateral Report

Used to report collateral status when responding to a Collateral Inquiry message.

Collateral Report

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = BA
908	CollRptID	Y	Unique Identifer for collateral report
909	CollInquiryID	N	Identifier for the collateral inquiry to which this message is a reply
60	TransactTime	N	
1043	CollApplType	N	Differentiates collateral pledged specifically against a position from collateral pledged against an entire portfolio on a valued basis.
291	FinancialStatus	N	Tells whether security has been restricted.
910	CollStatus	Y	Collateral status
911	TotNumReports	N	
912	LastRptRequested	N	
component block <Parties>		N	
1	Account	N	Customer Account
581	AccountType	N	Type of account associated with the order (Origin)
11	ClOrdID	N	Identifier fo order for which collateral is required
37	OrderID	N	Identifier fo order for which collateral is required
198	SecondaryOrderID	N	Identifier fo order for which collateral is required
526	SecondaryClOrdID	N	Identifier fo order for which collateral is required
124	NoExecs	N	Executions for which collateral is required
→	**17 ExecID**	N	Required if NoExecs > 0
897	NoTrades	N	Trades for which collateral is required
→	**571 TradeReportID**	N	Required if NoTrades > 0
→	**818 SecondaryTradeReportID**	N	
component block <Instrument>		N	Insert here the set of "Instrument" fields defined in "Common Components of Application Messages"
component block <FinancingDetails>		N	Insert here the set of "FinancingDetails" fields defined in "Common Components of Application Messages"
64	SettlDate	N	
53	Quantity	N	
854	QtyType	N	

© Copyright, 2006, FIX Protocol, Limited

15	Currency		N	
555	NoLegs		N	Number of legs Identifies a Multi-leg Execution if present and non-zero.
→	**component block <InstrumentLeg>**		N	Must be provided if Number of legs > 0
711	NoUnderlyings		N	Number of underlyings
→	**component block <UnderlyingInstrument>**		N	Must be provided if Number of underlyings > 0
899	MarginExcess		N	
900	TotalNetValue		N	
901	CashOutstanding		N	
component block <TrdRegTimestamps>			N	Insert here the set of "TrdRegTimestamps" fields defined in "Common Components of Application Messages"
54	Side		N	
136	NoMiscFees		N	Required if any miscellaneous fees are reported. Indicates number of repeating entries. Repeating group. ** Nested Repeating Group follows **
→	**137**	**MiscFeeAmt**	N	Required if NoMiscFees > 0
→	**138**	**MiscFeeCurr**	N	
→	**139**	**MiscFeeType**	N	Required if NoMiscFees > 0
→	**891**	**MiscFeeBasis**	N	
44	Price		N	
423	PriceType		N	
159	AccruedInterestAmt		N	
920	EndAccruedInterestAmt		N	
921	StartCash		N	
922	EndCash		N	
component block <SpreadOrBenchmarkCurveData>			N	Insert here the set of "SpreadOrBenchmarkCurveData" fields defined in "Common Components of Application Messages"
component block <Stipulations>			N	Insert here the set of "Stipulations" fields defined in "Common Components of Application Messages"
component block <SettlInstructionsData>			N	Insert here the set of "SettlInstructionsData" fields defined in "Common Components of Application Messages"
336	TradingSessionID		N	Trading Session in which trade occurred
625	TradingSessionSubID		N	Trading Session Subid in which trade occurred
716	SettlSessID		N	

© Copyright, 2006, FIX Protocol, Limited

717	SettlSessSubID	N	
715	ClearingBusinessDate	N	
58	Text	N	
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
	StandardTrailer	Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details
Refer to the FIXML element CollRpt

Collateral Inquiry

Used to inquire for collateral status.

Collateral Inquiry

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = BB
909	CollInquiryID	N	Identifier for the collateral inquiry to which this message is a reply
938	NoCollInquiryQualifier	N	Number of qualifiers to inquiry
→	**896 CollInquiryQualifier**	N	Required if NoCollInquiryQualifier > 0 Type of collateral inquiry
263	SubscriptionRequestType	N	Used to subscribe / unsubscribe for collateral status reports. If the field is absent, the default will be snapshot request only - no subscription.
725	ResponseTransportType	N	Ability to specify whether the response to the request should be delivered inband or via pre-arranged out-of-band transport.
726	ResponseDestination	N	URI destination name. Used if ResponseTransportType is out-of-band.
component block <Parties>		N	
1	Account	N	Customer Account
581	AccountType	N	Type of account associated with the order (Origin)
11	ClOrdID	N	Identifier fo order for which collateral is required
37	OrderID	N	Identifier fo order for which collateral is required
198	SecondaryOrderID	N	Identifier fo order for which collateral is required
526	SecondaryClOrdID	N	Identifier fo order for which collateral is required
124	NoExecs	N	Executions for which collateral is required
→	**17 ExecID**	N	Required if NoExecs > 0
897	NoTrades	N	Trades for which collateral is required
→	**571 TradeReportID**	N	Required if NoTrades > 0
→	**818 SecondaryTradeReportID**	N	
component block <Instrument>		N	Insert here the set of "Instrument" fields defined in "Common Components of Application Messages"
component block <FinancingDetails>		N	Insert here the set of "FinancingDetails" fields defined in "Common Components of Application Messages"
64	SettlDate	N	

© Copyright, 2006, FIX Protocol, Limited

53	Quantity	N	
854	QtyType	N	
15	Currency	N	
555	NoLegs	N	Number of legs Identifies a Multi-leg Execution if present and non-zero.
→	**component block <InstrumentLeg>**	N	Must be provided if Number of legs > 0
711	NoUnderlyings	N	Number of underlyings
→	**component block <UnderlyingInstrument>**	N	Must be provided if Number of underlyings > 0
899	MarginExcess	N	
900	TotalNetValue	N	
901	CashOutstanding	N	
component block <TrdRegTimestamps>		N	Insert here the set of "TrdRegTimestamps" fields defined in "Common Components of Application Messages"
54	Side	N	
44	Price	N	
423	PriceType	N	
159	AccruedInterestAmt	N	
920	EndAccruedInterestAmt	N	
921	StartCash	N	
922	EndCash	N	
component block <SpreadOrBenchmarkCurveData>		N	Insert here the set of "SpreadOrBenchmarkCurveData" fields defined in "Common Components of Application Messages"
component block <Stipulations>		N	Insert here the set of "Stipulations" fields defined in "Common Components of Application Messages"
component block <SettlInstructionsData>		N	Insert here the set of "SettlInstructionsData" fields defined in "Common Components of Application Messages"
336	TradingSessionID	N	Trading Session in which trade occurred
625	TradingSessionSubID	N	Trading Session Subid in which trade occurred
716	SettlSessID	N	
717	SettlSessSubID	N	
715	ClearingBusinessDate	N	
58	Text	N	
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must

© Copyright, 2006, FIX Protocol, Limited

			immediately precede it.
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
	StandardTrailer	Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details
Refer to the FIXML element CollInq

Collateral Inquiry Ack

Used to respond to a Collateral Inquiry in the following situations:

- When the CollateralInquiry will result in an out of band response (such as a file transfer).
- When the inquiry is otherwise valid but no collateral is found to match the criteria specified on the Collateral Inquiry message.
- When the Collateral Inquiry is invalid based upon the business rules of the counterparty.

Collateral Inquiry Ack

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = BG
909	CollInquiryID	Y	Identifier for the collateral inquiry to which this message is a reply
945	CollInquiryStatus	Y	Status of the Collateral Inquiry referenced by CollInquiryID
946	CollInquiryResult	N	Result of the Collateral Inquriy referenced by CollInquiryID - specifies any errors or warnings
938	NoCollInquiryQualifier	N	Number of qualifiers to inquiry
→	**896 CollInquiryQualifier**	N	Required if NoCollInquiryQualifier > 0 Type of collateral inquiry
911	TotNumReports	N	Total number of reports generated in response to this inquiry
component block <Parties>		N	
1	Account	N	Customer Account
581	AccountType	N	Type of account associated with the order (Origin)
11	ClOrdID	N	Identifier fo order for which collateral is required
37	OrderID	N	Identifier fo order for which collateral is required
198	SecondaryOrderID	N	Identifier fo order for which collateral is required
526	SecondaryClOrdID	N	Identifier fo order for which collateral is required
124	NoExecs	N	Executions for which collateral is required
→	**17 ExecID**	N	Required if NoExecs > 0
897	NoTrades	N	Trades for which collateral is required
→	**571 TradeReportID**	N	Required if NoTrades > 0
→	**818 SecondaryTradeReportID**	N	
component block <Instrument>		N	Insert here the set of "Instrument" fields defined in "Common Components of Application Messages"
component block <FinancingDetails>		N	Insert here the set of "FinancingDetails" fields defined in

			"Common Components of Application Messages"
64	SettlDate	N	
53	Quantity	N	
854	QtyType	N	
15	Currency	N	
555	NoLegs	N	Number of legs Identifies a Multi-leg Execution if present and non-zero.
→	**component block <InstrumentLeg>**	N	Must be provided if Number of legs > 0
711	NoUnderlyings	N	Number of underlyings
→	**component block <UnderlyingInstrument>**	N	Must be provided if Number of underlyings > 0
336	TradingSessionID	N	Trading Session in which trade occurred
625	TradingSessionSubID	N	Trading Session Subid in which trade occurred
716	SettlSessID	N	
717	SettlSessSubID	N	
715	ClearingBusinessDate	N	
725	ResponseTransportType	N	Ability to specify whether the response to the request should be delivered inband or via pre-arranged out-of-band transport.
726	ResponseDestination	N	URI destination name. Used if ResponseTransportType is out-of-band.
58	Text	N	
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
	StandardTrailer	Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details

Refer to the FIXML element CollInqAck

© Copyright, 2006, FIX Protocol, Limited

Appendix A – Trade Amendment and Trade Cancel Work Flow Diagrams

Trade Amendment for Trade Capture Report

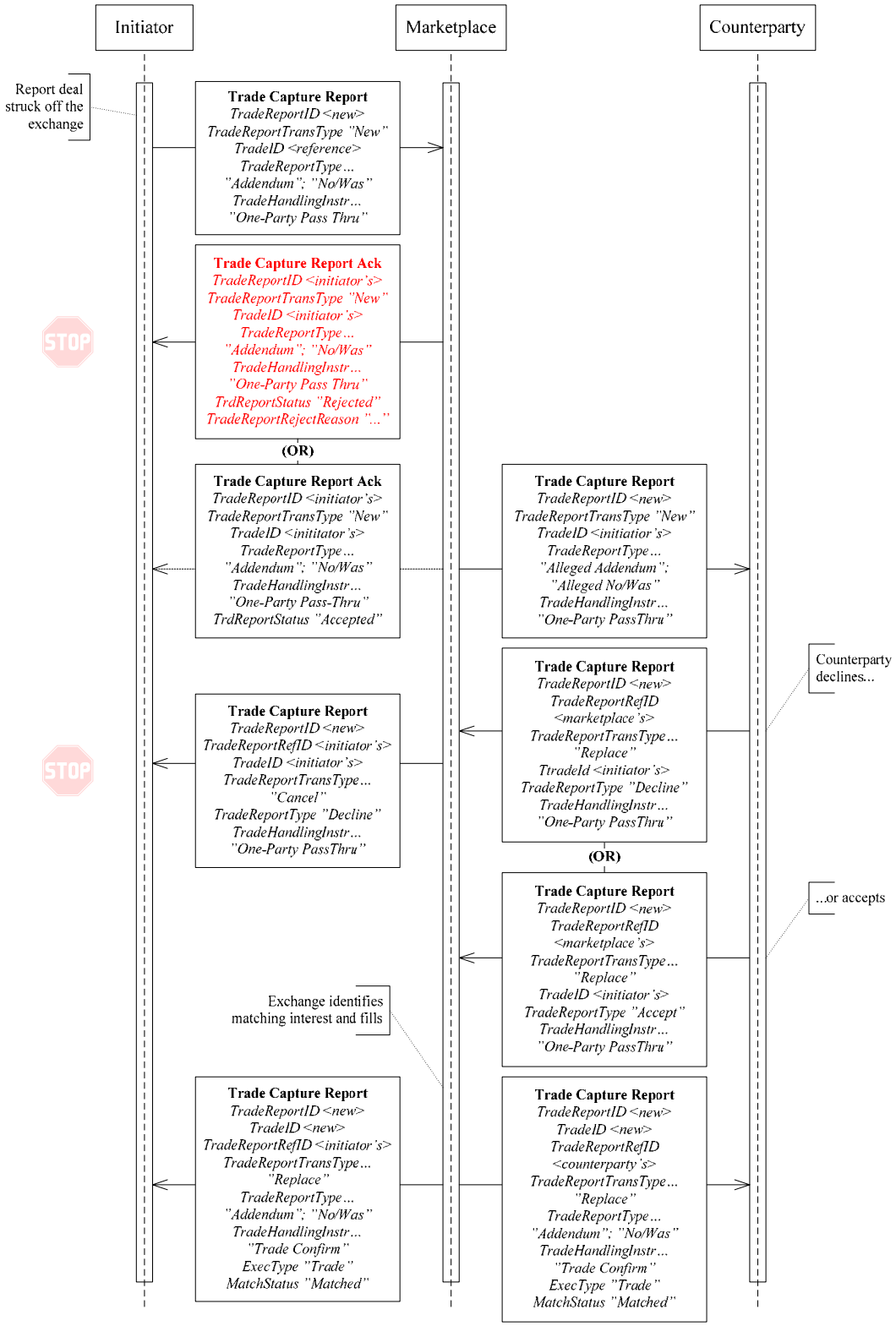
Marketplaces can allow brokers to request trade amendments. Trade amendments are normally limited to private properties for the side of the initiator (called Addendums) – i.e. can not affect the counterparty. Changes to bilateral trade terms can be indicated by using the "No/Was" value of the TradeReportType. Trade Addendums might not need acceptance by the counterparty. Marketplaces may limit what properties can be updated and also put a time limit for updates (e.g. up to fifteen minutes after the trade was created).

Note that marketplace Ack messages for the counterparty response are not shown in the diagrams due to space limitations. The table in Appendix B exemplifies the various messages in this process.

© Copyright, 2006, FIX Protocol, Limited

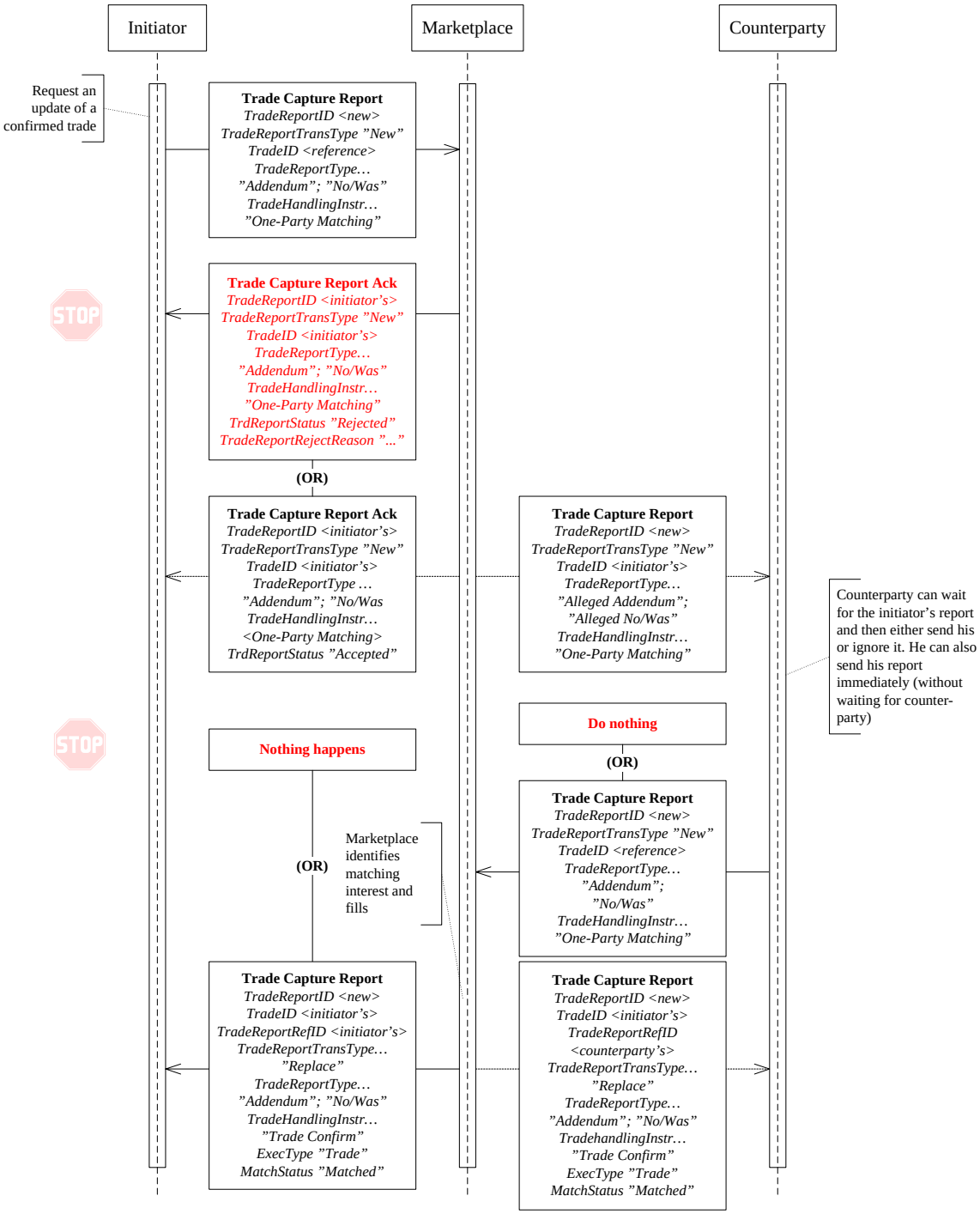
Trade Amendment - One-Party Report for Pass-Thru Model

The following diagram depicts the trade amendment part of the workflow.



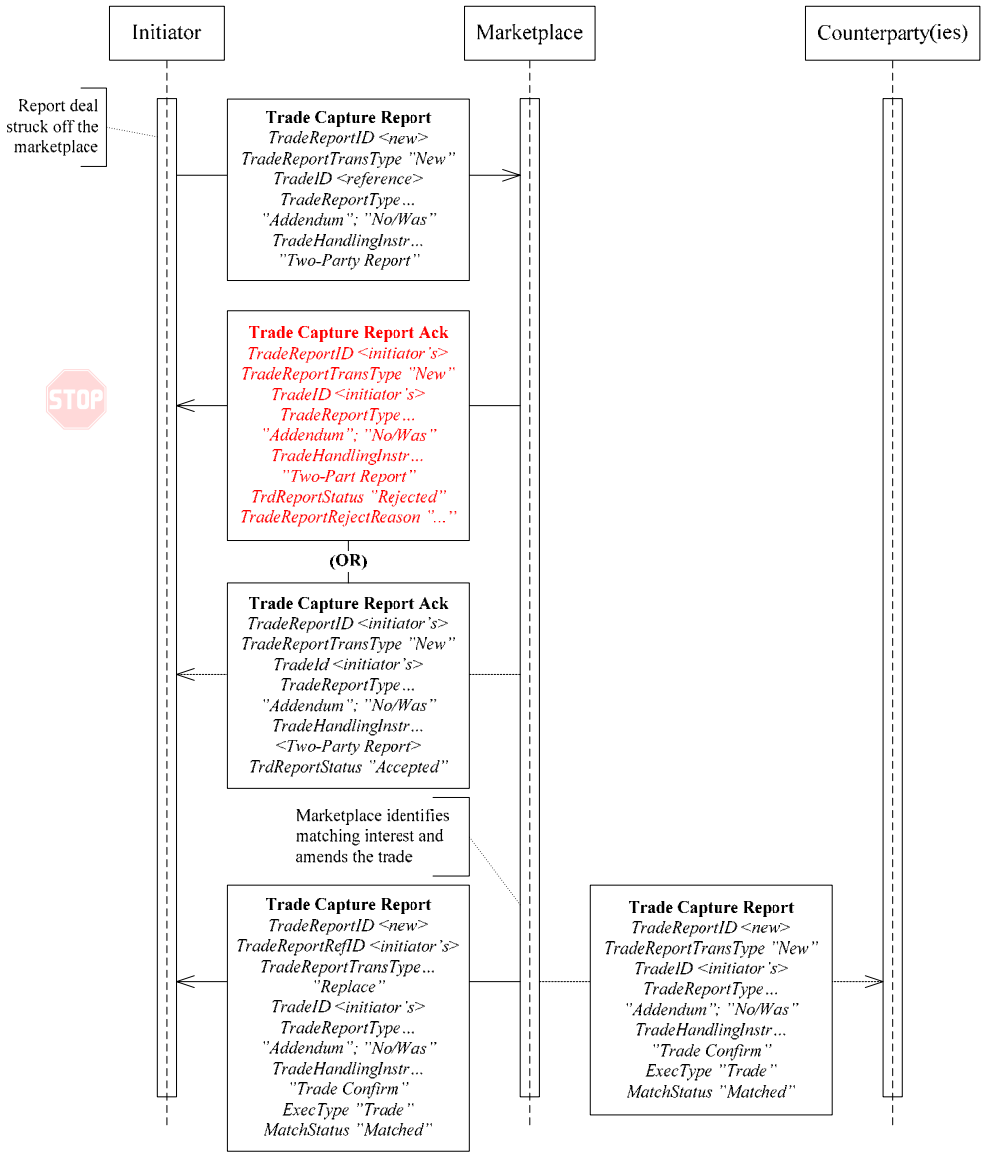
Trade Amendment - One-Party Matching Model

The following diagram depicts the trade amendment part of the workflow.



Trade Amendment - Two-Party Report

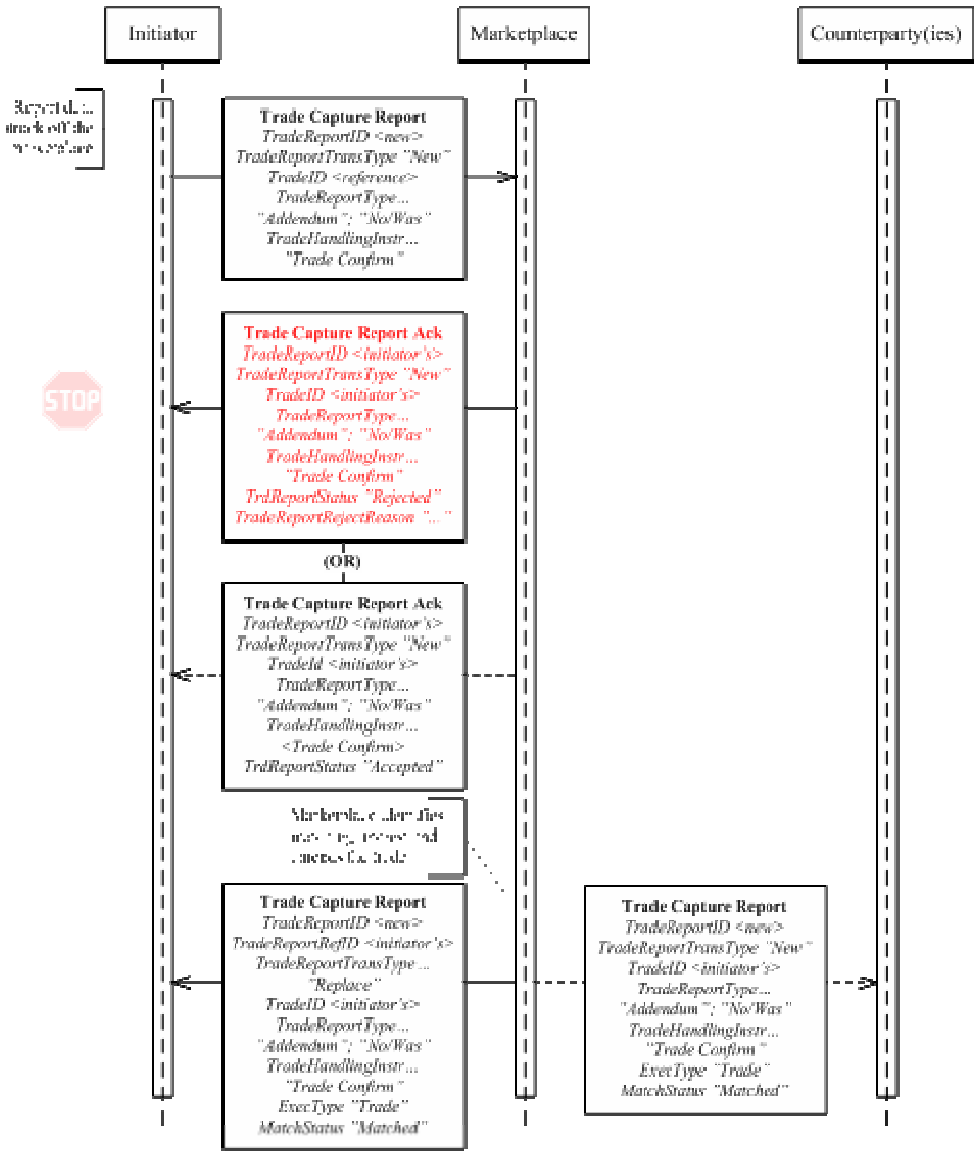
In this model one party reports an amendment to a confirmed trade with one (a cross trade) or two counterparties. Counterparties are optionally informed by the marketplace of the completed amendment. The workflow is depicted in the following diagram:



© Copyright, 2006, FIX Protocol, Limited

Trade Amendment - Confirmed Trade Reporting Model

In this model one party, itself a recognized marketplace as an ECN or ATS, exchange or clearing organization reports an amendment to a confirmed trade with two counterparties. Counterparties are optionally informed by the marketplace of the completed trade. The workflow is depicted in the following diagram:



Trade Capture Report Trade Cancel

Marketplaces can allow brokers to request trade breaks (or cancellations). Marketplaces allowing brokers to request trade cancellation would require that all parties to the trade agree. Trade breaks may be limited to certain trades (e.g. privately negotiated ones), a limited time (e.g. up to fifteen minutes after the trade was created), etc.

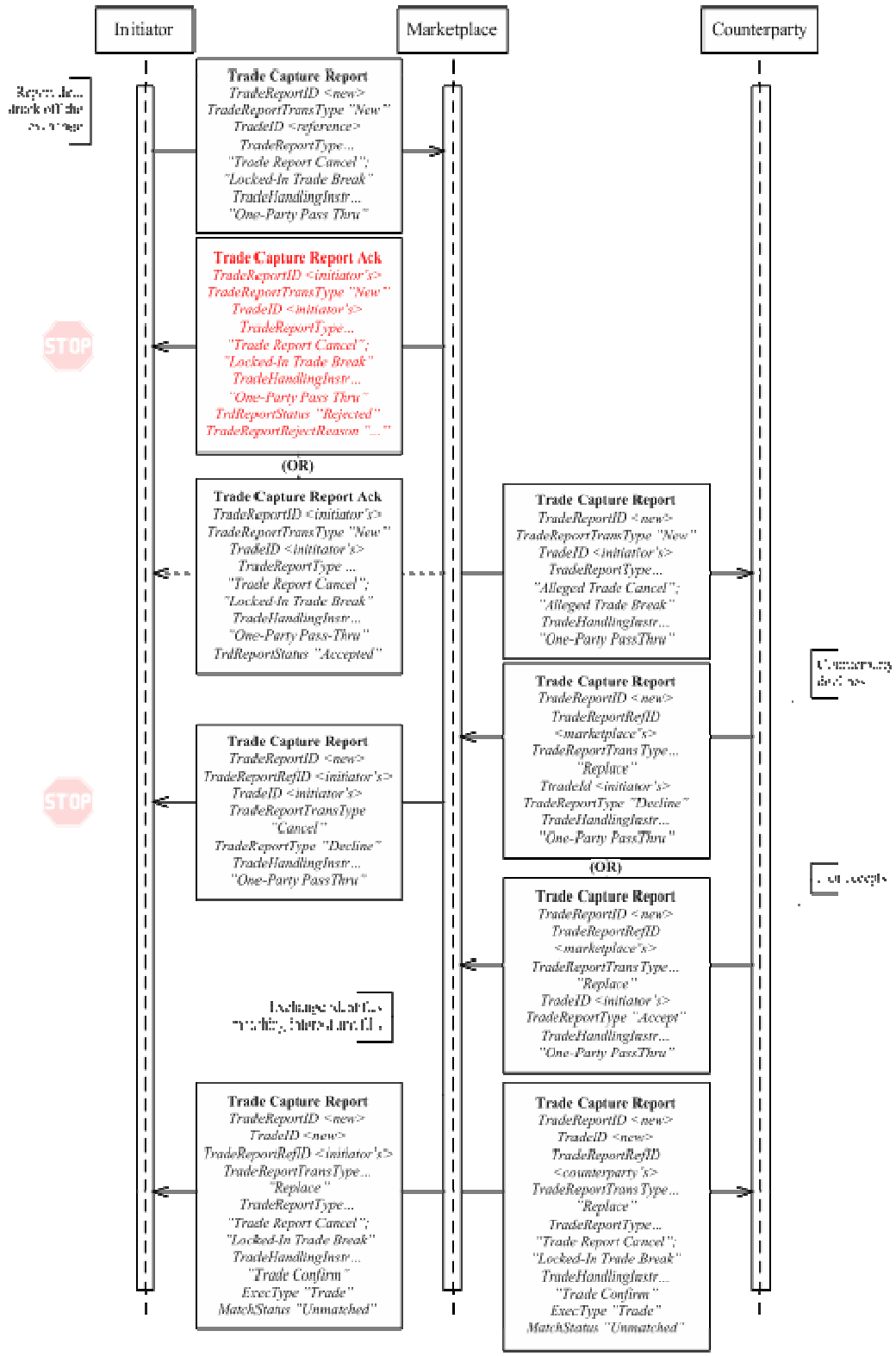
Trade break is done using the same models as for reporting, i.e. One-Party for Pass-Thru, One-Part for matching, Two-Party or reporting of confirmed trades. The workflows will thereby be very similar to the ones above, the difference being that other actions (TradeReportType) are used.

Note that marketplace Ack messages for the counterparty response are not shown in the diagrams due to space limitations. The table in Appendix B exemplifies the various messages in this process.

© Copyright, 2006, FIX Protocol, Limited

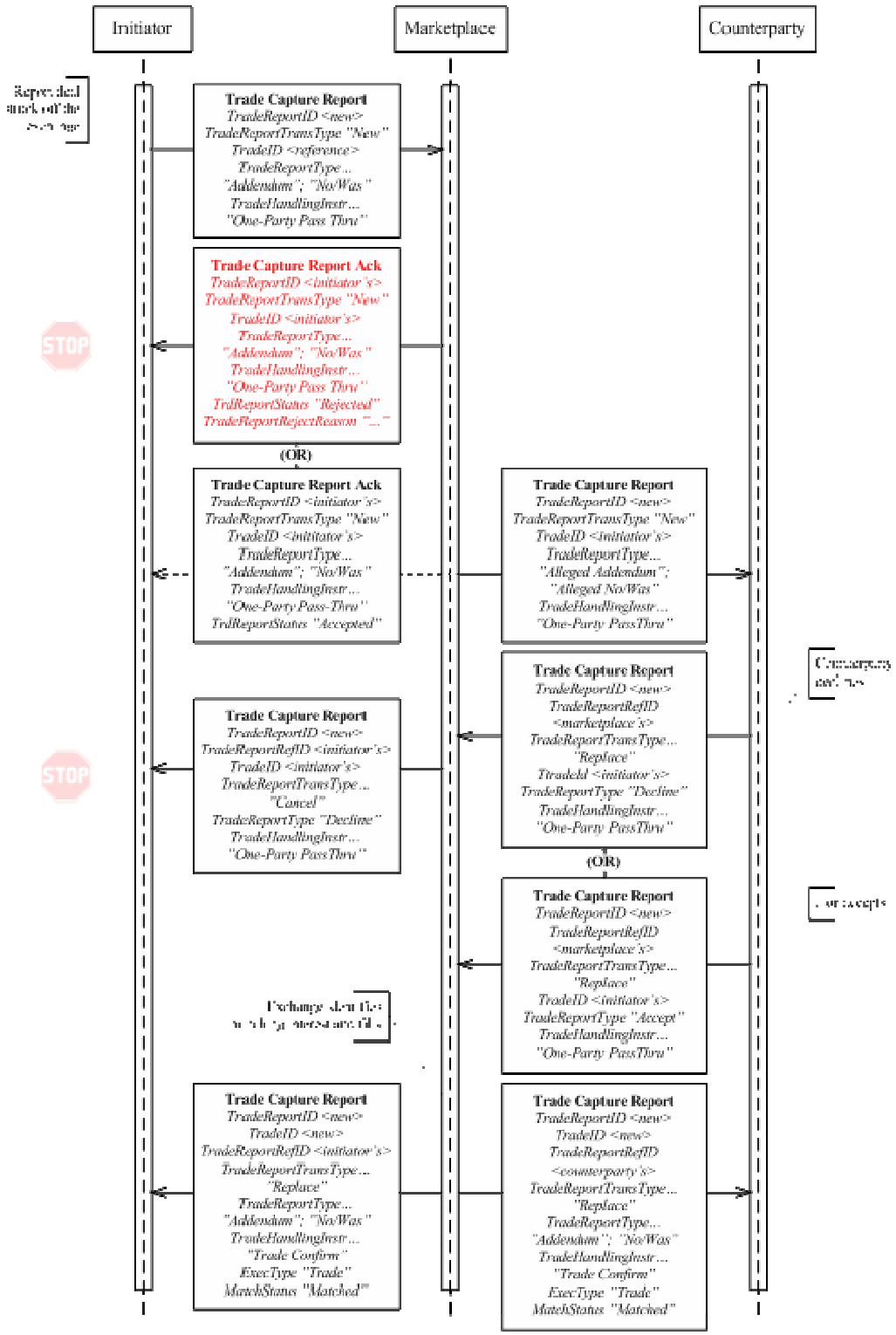
Trade Cancel - One-Party Pass-Thru Model

The following diagram depicts the trade break part of the workflow.



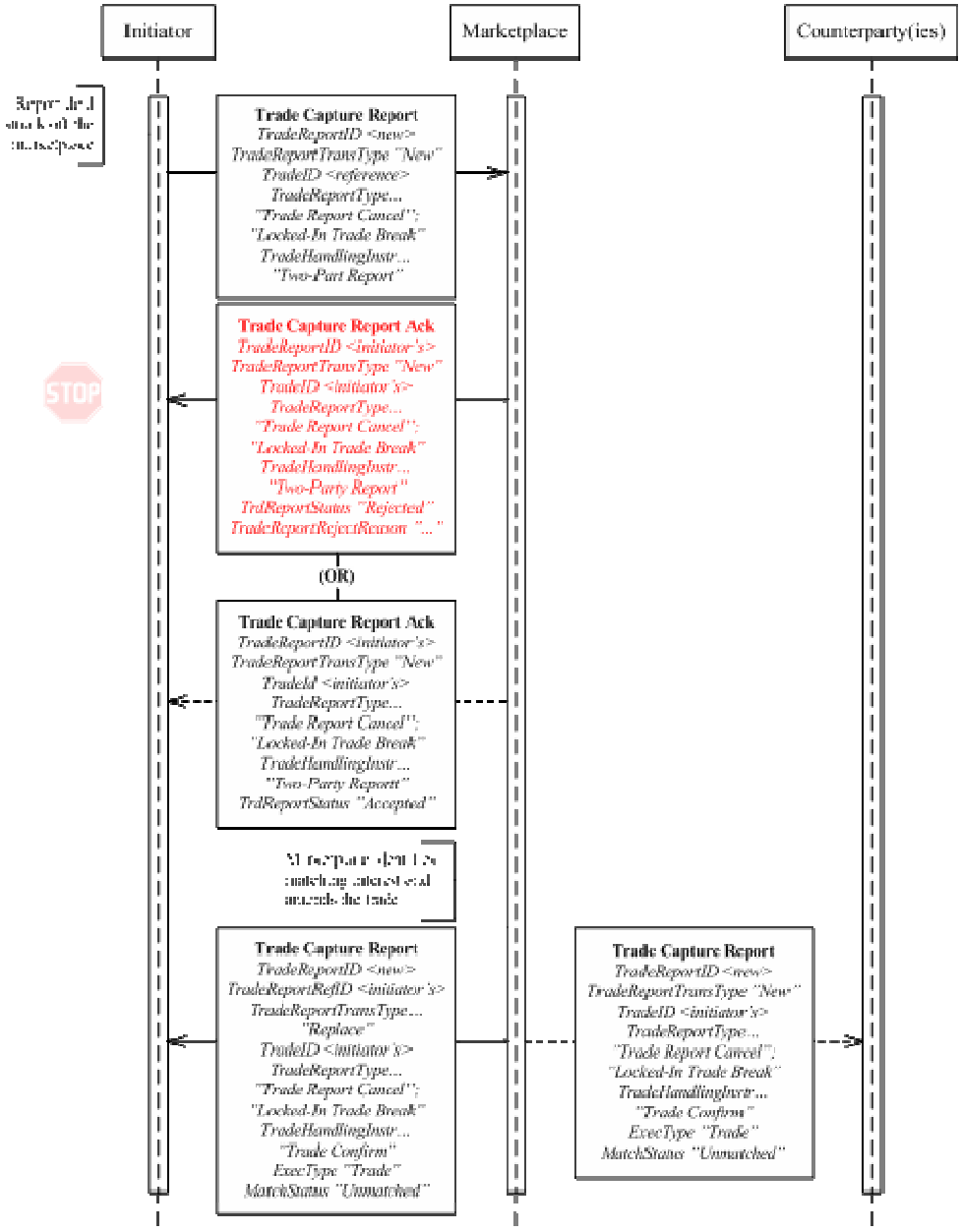
Trade Cancel - One-Party Matching Model

The following diagram depicts the trade break part of the workflow.



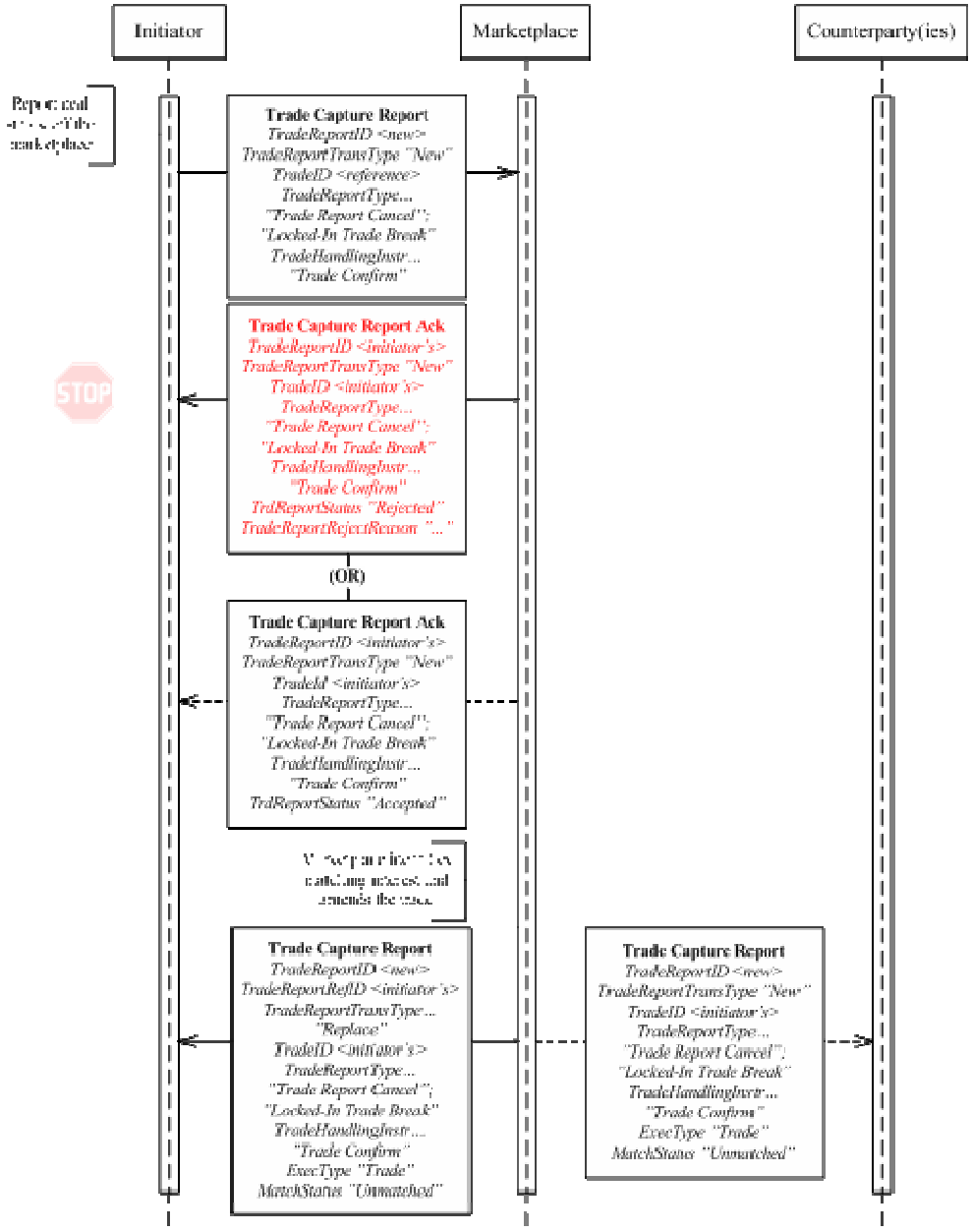
Trade Cancel - Two-Party Report

In this model one party reports a break to a confirmed trade with one (a cross trade) or two counterparties. Counterparties are optionally informed by the marketplace of the completed amendment. The workflow is depicted in the following diagram:



Trade Cancel - Confirmed Trade Reporting Model

In this model one party, itself a recognized marketplace as an ECN or ATS, reports a break of a confirmed trade with two counterparties. Counterparties are optionally informed by the marketplace of the completed trade. The workflow is depicted in the following diagram:

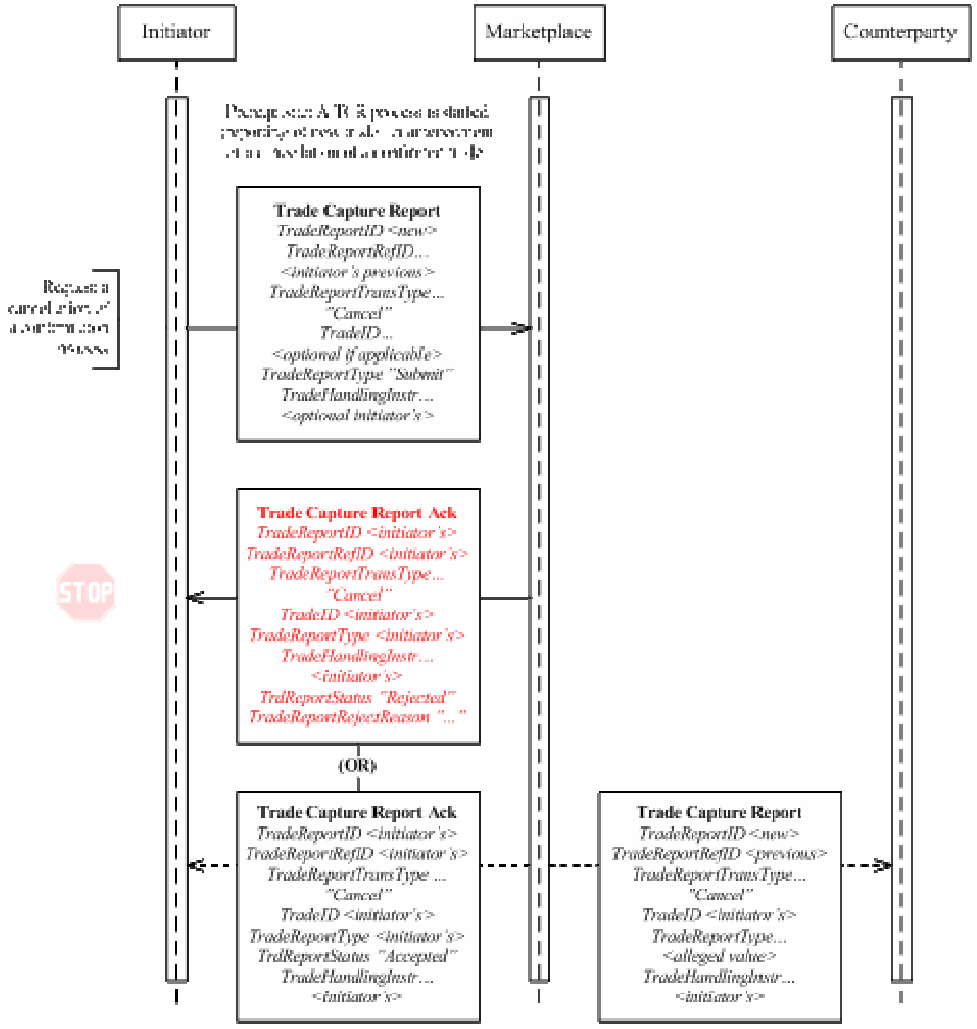


Generic Sub-Workflows

The following sub-workflows define generic parts of other flows as special cases. They are separated out from the main flows defined above so those can be focused on the core workflow.

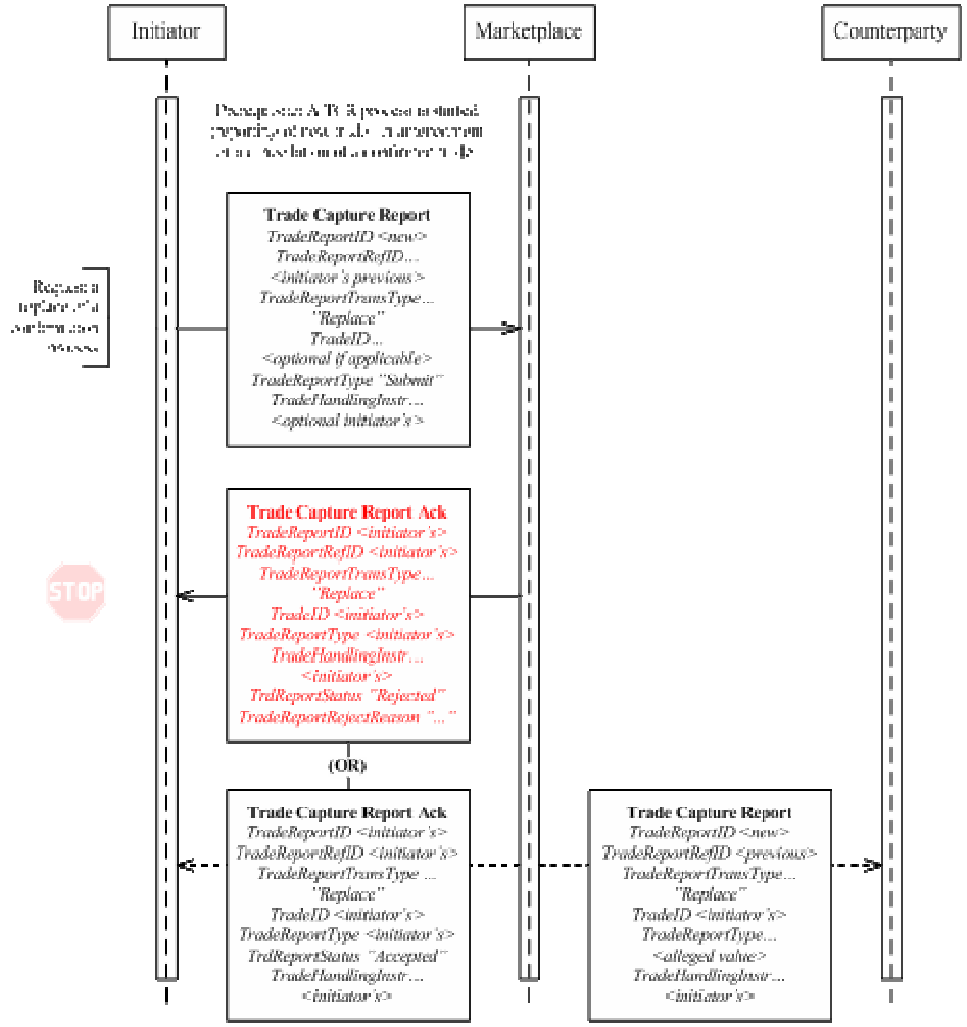
Canceling a Pre-confirmed Trade Capture Report

The following workflow is relevant when a trade confirmation process is started at the marketplace, but has not yet resulted in the marketplace confirming the resultant trade. Note that a cancellation of the ongoing process includes any confirmed trade action (TradeReportType = Addendum; No/Was; Trade Report Cancel; or Locked-In Trade Break). The initiator is the only user who can request a cancellation of the process. Marketplaces may choose not to support cancellations at all or only under certain circumstances.



Updating (Replacing) a Trade Capture Report

The following workflow is relevant when a trade confirmation is started at the marketplace, but has not yet resulted in the marketplace confirming the resultant trade. Note that an update of the ongoing process includes any confirmed trade action (TradeReportType = Addendum; No/Was; Trade Report Cancel; or Locked-In Trade Break). The initiator is the only user who can request an update of the process. Marketplaces may choose not to support updates at all or only under certain circumstances.



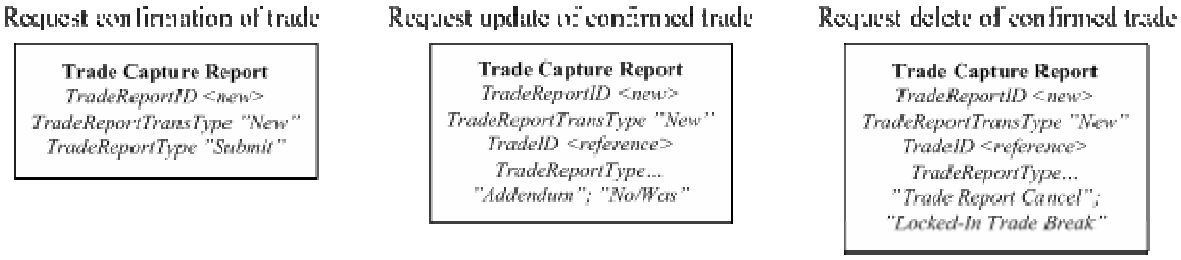
Appendix B - Trade Capture Report (TCR) Work Flow and Usage Tables

Trade Capture Processing Guidelines and Rules

In Exchange, ECN, and Centeral Counter Party models, TCRs have two related purposes; confirmed trades and the process to confirm a trade. Different message tags are used depending on what you want to achieve:

- The Confirmed Trade
    - Identifier = TradeID
    - Action = TradeReportType
- The **process** of confirming the trade
    - Identifier = TradeReportID. Each actor issues their own id for every message sent (excluding TCR Ack messages). Note that the TradeReportID is optional; it need only be used in cases where subsequent messages should refer to previous ones. Normally this means all messages discussed in this proposal have a TradeReportID. Marketplace issued confirmed trades however do not need it as future reference needs are covered by the TradeID instead.
        - Reference to previous message in the same process = TradeReportRefID.
            - The initiator always chains messages so that TradeReportRefID refers to the previous TradeReportID.
            - The marketplace uses the reference for confirmation messages (TradeHandlingInstr = Trade Confirm) and then refers to the external actors via TradeReportID.
            - The counterparty uses the reference in Accept / Decline messages and then refers to the TradeReportID of the TCR they received from the marketplace.
    - Action = TradeReportTransType
    - Method = TradeHandlingInstr (Shown in tables but not diagrams)

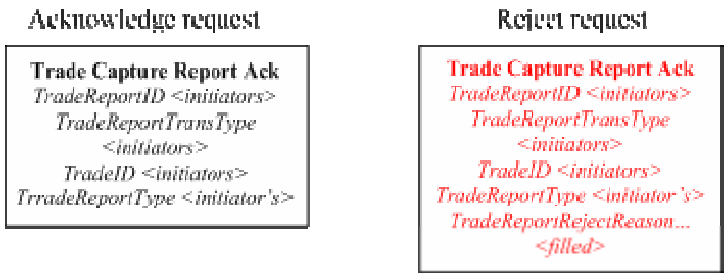
The basic method of identification varies with the purpose of the message. A user sending a request to a marketplace must specify as depicted in the following diagram:



As can be seen, a request for a new trade does not specify a TradeID (as this will be assigned by the marketplace). Requests to update or delete a confirmed trade naturally must specify the TradeID the marketplace has assigned.

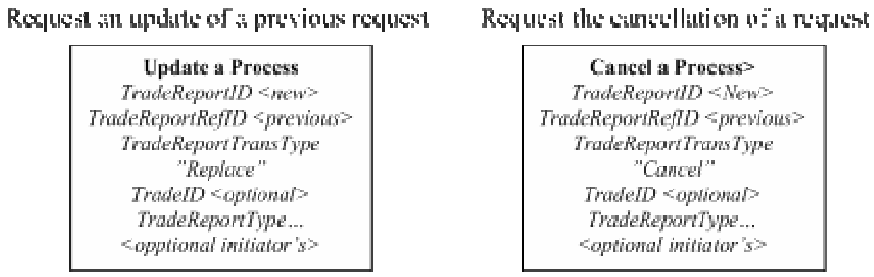
The marketplace will respond with a TCR Ack as depicted in the following diagram:

© Copyright, 2006, FIX Protocol, Limited



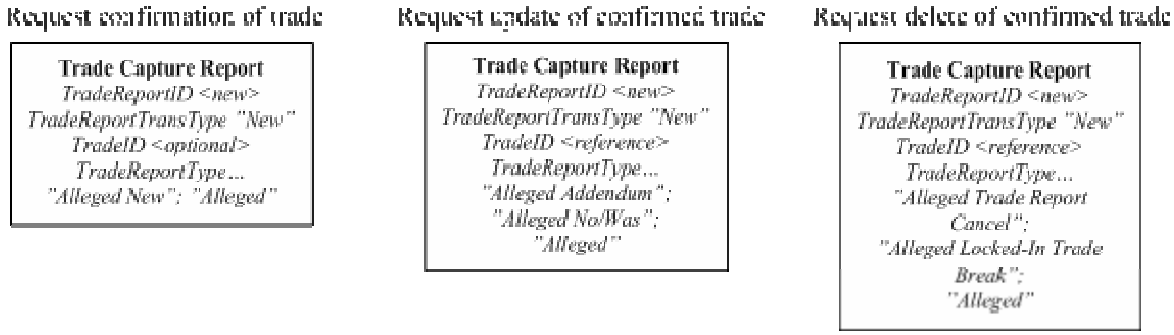
Note that the TradeID (value <initiators>) will not be part of the Ack message unless the request referred to an existing confirmed trade.

If a user wants to update or cancel the TCR during the process of the marketplace confirming the trade, he sends messages as depicted in the following diagram:



Note that if the TradeReportRefID is used for reference, the TradeReportType need not be provided as it can be retained throughout the confirmation process.

In cases where the marketplace forwards the request to the counterparty for acceptance, the messages are identified as depicted in the following diagram:

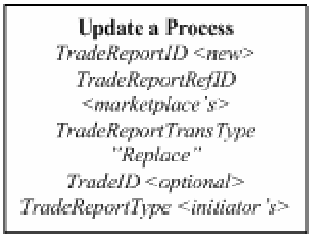


The TradeReportID is the one assigned for the process by the marketplace. Note that TradeReportTypes "Alleged…" tells the counterparty that a response is required. Note that the "Alleged" value itself can be used as an alternative to the more specific and new "Alleged…" values.

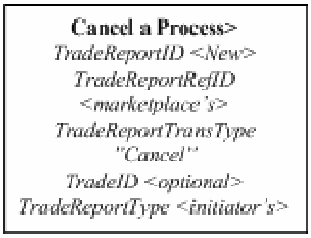
In cases the initiator want to cancel or update the TCR during the process, the marketplace forward the new state to the counterparty as depicted in the following diagram:

© Copyright, 2006, FIX Protocol, Limited

Request an update of a previous request



Request the cancellation of a request



Considering "in flight" modifications, it is worth noting that the marketplace representation of the state takes precedence.

Notes on the Following Tables

All tables below exemplify messages described in the TCR Section above

Please note that the grayed out actions in the below tables are considered less applicable in practice.

Note that the TradeID is not necessarily specified in cases where the user wants to cancel or replace a request; in such cases the TradeReportRefID must be specified!

The TradeID can be used as an alternative to the TradeReportRefID if the marketplace assigns the permanent trade id at the beginning of the trade flow.

Prior to confirmation of a trade by the marketplace, a replace or cancel request can be submitted at the level of either the TradeID or TradeReportRefID.

The SecondaryTradeID can also be used as an alternative to the TradeReportRefID. This is bilaterally agreed between the parties and applicable in the cases where the marketplace assigns a separate id to the process of confirming the Trade.

When the marketplace reports a confirmed new, busted or amended trade:

- If the trade originates in a counterparty system, as most privately negotiated trades do, then TradeReportTransType should be set to 'Replace' in order to update the representation of the trade on that side to a 'confirmed' status. Copies sent to other parties should carry "New" since the action specified in TradeReportType is being reported for the first time. Whether a trade confirm is being routed to a third party can be determined by comparing the third party identifier to the parties on the trade itself..

© Copyright, 2006, FIX Protocol, Limited

Trade Handling Usage Tables for Regular and Privately Negotiated Trades

Requesting the Market Place for a New Trade

Confirmed Trade – Published by Marketplace

	Action	Message	Trade Report Trans Type	Trade Report Type	Trade Handling Instr	Trade ID	Trade Report Ref ID	Comment
1	**Trade confirmation (always from marketplace to relevant parties)**							
1.1	Publish new trade	TCR	New (0)	Submit (0)	Trade Confirm (0)	Filled	Reference	MatchStatus = 0 (Compared, matched or affirmed) The TradeReportRefID depends on who the receiver is. Note that TCR are sent to various parties (some of whom need a reference and some who do not).
1.2	Publish amended trade	TCR	Replace (2)	Submit (0), Addendum (4) No/Was (5)	Trade Confirm (0)	Filled	Reference	MatchStatus = 0 (Compared, matched or affirmed) The TradeReportRefID depends on who the receiver is. Note that TCR are sent to various parties (some of whom need a reference and some who do not).
1.3	Publish canceled trade	TCR	Cancel (1)	Submit (0), Trade Report Cancel (6) (Locked In) Trade Break (7)	Trade Confirm (0)	Filled	Reference	MatchStatus = 1 (Uncompared, unmatched or unaffirmed)– The TradeReportRefID depends on who the receiver is. Note that TCR are sent to various parties (some of whom need a reference and some who do not).

© Copyright, 2006, FIX Protocol, Limited

	Action	Message	Trade Report Trans Type	Trade Report Type	Trade Handling Instr	Trade ID	Trade Report Ref ID	Comment
1.4	Publish trade waiting for manual approval	TCR	New (0), Cancel (1), Replace (2)	Submit (0), Addendum (4), No/Was (5) Trade Report Cancel (6), (Locked In) Trade Break (7)	Trade Confirm (0)	Filled	Reference	Also note that "Reversal" and "Release" might be relevant (although this Gap Analysis does not address them) MatchStatus = 1 (Uncompared, unmatched or unaffirmed)

One-Party Report for Pass-Thru

	Action	Message	Trade Report Trans Type	Trade Report Type	Trade Handling Instr	Trade ID	Trade ReportID (Trade Report Ref ID)	Comment
1	**Initiator requests (main flow from initiator to marketplace)**							
1.1	Enter new report	TCR	New (0)	Submit (0)	One-Party Report for Pass-Thru (1)	N/A	New (N/A)	Starts a process involving a counterparty response
1.2	Update earlier report	TCR	Replace (2)	Submit (0)	One-Party Report for Pass-Thru (1)	Optionally filled	New (Initiator's previous)	
1.3	Cancel earlier report	TCR	Cancel (1)	Submit (0)	One-Party Report for Pass-Thru (1)	Optionally filled	New (Initiator's previous)	Cancel request may be submitted using TradeID if provided by marketplace on initial Ack in which case the pre-final trade entity is being canceled.

© Copyright, 2006, FIX Protocol, Limited

	Action	Message	Trade Report Trans Type	Trade Report Type	Trade Handling Instr	Trade ID	Trade ReportID (Trade Report Ref ID)	Comment
								Otherwise, TradeReportRefID must be used to cancel the initial request.
1.4	Acknowledgement from marketplace	TCR Ack	New (0), Cancel (1) or Replace (2)	Initiators	One-Party Report for Pass-Thru (1)	Optionally filled	Initiator's (Initiator's)	Optional. If ack message is used for One-Party Reports.
1.5	Reject from marketplace	TCR Ack	New (0), Cancel (1) or Replace (2)	Initiators	One-Party Report for Pass-Thru (1)	Optionally filled	Initiator's (Initiator's)	TradeReportRejectReason is specified.
2	**Counterparty responses (bi-directional flow)**							
2.1	Marketplace forward of initiators "New" report (Alleged)	TCR	New (0)	Alleged New (tbd) or Alleged (1)	One-Party Report for Pass-Thru (1)	Optionally filled	New (N/A)	
2.2	Marketplace forward of initiators "Cancel" or "Replace" Report (Alleged)	TCR	Cancel (1) or Replace (2)	Alleged New (tbd) or Alleged (1)	One-Party Report for Pass-Thru (1)	Optionally filled	New (Marketplace previous)	
2.3.	Accept report (sent to marketplace)	TCR	Replace (2)	Accept (2)	One-Party Report for Pass-Thru (1)	Optionally filled	New (Marketplace's)	Counterparty can reference either TradeReportRefID or TradeID if assigned by marketplace
2.4	Decline report (sent to marketplace)	TCR	Replace (2)	Decline (3)	One-Party Report for Pass-Thru (1)	Optionally filled	New (Marketplace's)	- " -
2.5	Acknowledgement from marketplace	TCR Ack	Replace (2)	Accept (2), Decline (3)	One-Party Report for Pass-Thru (1)	Optionally filled	Counterparty's (Counterparty's)	Optional. If ack message is used for One-Party Reports.
2.6	Reject from marketplace	TCR Ack	Replace (2)	Accept (2), Decline (3)	One-Party Report for Pass-Thru (1)	Optionally filled	Counterparty's (Counterparty's)	TradeReportRejectReason is specified.

© Copyright, 2006, FIX Protocol, Limited

	Action	Message	Trade Report Trans Type	Trade Report Type	Trade Handling Instr	Trade ID	Trade ReportID (Trade Report Ref ID)	Comment
3	**Marketplace relay of counterparty Decline (flow from marketplace to initiator)**							
3.1	Marketplace cancels the process due to a decline from the counterparty	TCR	Cancel (1)	Decline (3)	One-Party Report for Pass-Thru (1)	Optionally filled	New (Initiator's)	
4	**Marketplace publication of confirmed trade (flow from marketplace to external actors)**							
4.1	Publication to the initiator and counterparty	TCR	Replace (2)	Submit (0)	Trade Confirm (0)	Filled	Reference	MatchStatus = 0 (Compared, matched or affirmed) The TradeReportRefID depends on who the receiver is.
4.2	Publication to other parties	TCR	New (0)	Submit (0)	Trade Confirm (0)	Filled	N/A	MatchStatus = 0 (Compared, matched or affirmed)

One-Party Report for Matching

	Action	Message	Trade Report Trans Type	Trade Report Type	Trade Handling Instr	Trade ID	Trade ReportID (Trade Report Ref ID)	Comment
1	**Either side requests (main flow from party to marketplace)**							
1.1	Enter new report	TCR	New (0)	Submit (0)	One-Party Report for Matching (2)	N/A	New (N/A)	Starts a process involving a counterparty response
1.2	Update earlier report	TCR	Replace (2)	Submit (0)	One-Party Report for Matching (2)	Optionally filled	New (Initiator's previous)	

© Copyright, 2006, FIX Protocol, Limited

	Action	Message	Trade Report Trans Type	Trade Report Type	Trade Handling Instr	Trade ID	Trade ReportID (Trade Report Ref ID)	Comment
1.3	Cancel earlier report	TCR	Cancel (1)	Submit (0)	One-Party Report for Matching (2)	Optionally filled	New (Initiator's previous)	Cancel request may be submitted using TradeID if provided by marketplace on initial Ack. Otherwise, TradeReportRefID must be used to reference initial request.
1.4	Acknowledgement from marketplace	TCR Ack	New (0), Cancel (1) or Replace (2)	Initiator's	One-Party Report for Matching (2)	Optionally filled	Initiator's (Initiator's)	Optional. If ack message is used for One-Party Reports.
1.5	Reject from marketplace	TCR Ack	New (0), Cancel (1) or Replace (2)	Initiator's	One-Party Report for Matching (2)	Optionally filled	Initiator's (Initiator's)	TradeReportRejectReason is specified.
2	**Optional relay of request to counterparty**							
2.1	Marketplace forward of initiators "New" report (Alleged)	TCR	New (0)	Alleged New (tbd) or Alleged (1)	One-Party Report for Matching (2)	N/A	New (N/A)	
2.2	Marketplace forward of initiators "Cancel" or " Replace" report (Alleged)	TCR	Cancel (1) or Replace (2)	Alleged New (tbd) or Alleged (1)	One-Party Report for Matching (2)	N/A	New (Marketplace's Previous)	
3	**Marketplace publication of confirmed trade (flow from marketplace to external actors)**							
3.1	Publication to the initiator and counterparty	TCR	Replace (2)	Submit (0)	Trade Confirm (0)	Filled	Reference	MatchStatus = 0 (Compared, matched or affirmed) The TradeReportRefID depends on who the receiver is.
3.2	Publication to other	TCR	New (0)	Submit (0)	Trade Confirm (0)	Filled	N/A	MatchStatus = 0 (Compared,

© Copyright, 2006, FIX Protocol, Limited

	Action	Message	Trade Report Trans Type	Trade Report Type	Trade Handling Instr	Trade ID	Trade ReportID (Trade Report Ref ID)	Comment
	parties							matched or affirmed)

Two-Party Report

	Action	Message	Trade Report Trans Type	Trade Report Type	Trade Handling Instr	Trade ID	Trade ReportID (Trade Report Ref ID)	Comment
1	**Actor requests (main flow from actor to marketplace):**							
1.1	Enter new report	TCR	New (0)	Submit (0)	Two-Party Report (3)	N/A	New (N/A)	
1.2	Update earlier report	TCR	Replace (2)	Submit (0)	Two-Party Report (3)	Optionally filled	New (Initiator's previous)	
1.3	Cancel earlier report	TCR	Cancel (1)	Submit (0)	Two-Party Report (3)	Optionally filled	New (Initiator's previous)	Cancel request may be submitted using TradeID if provided by marketplace on initial Ack. Otherwise, TradeReportRefID must be used to reference initial request.
1.4	Acknowledgement from marketplace	TCR Ack	New (0), Cancel (1), Replace (2)	Initiator's	Two-Party Report (3)	Optionally filled	Initiator's (Initiator's)	Optional. If ack message is used for Two-Party Reports.
1.5	Reject from marketplace	TCR Ack	New (0), Cancel (1), Replace (2)	Initiator's	Two-Party Report (3)	Optionally filled	Initiator's (Initiator's)	TradeReportRejectReason is specified.

© Copyright, 2006, FIX Protocol, Limited

	Action	Message	Trade Report Trans Type	Trade Report Type	Trade Handling Instr	Trade ID	Trade ReportID (Trade Report Ref ID)	Comment
2	**Marketplace publication of confirmed trade (flow from marketplace to external actors)**							
2.1	Publication to the initiator	TCR	Replace (2)	Submit (0)	Trade Confirm (0)	Filled	Initiator's	MatchStatus = 0 (Compared, matched or affirmed)
2.2	Publication to other parties	TCR	New (0)	Submit (0)	Trade Confirm (0)	Filled	N/A	MatchStatus = 0 (Compared, matched or affirmed)

Reporting of Locked-In Trade to Marketplace

	Action	Message	Trade Report Trans Type	Trade Report Type	Trade Handling Instr	Trade ID	Trade ReportID (Trade Report Ref ID)	Comment
1	**Actor requests (main flow from actor to marketplace):**							
1.1	Enter new report	TCR	New (0)	Submit (0)	Trade Confirm (0)	N/A	New (N/A)	
1.2	Update earlier report	TCR	Replace (2)	Submit (0)	Trade Confirm (0)	Optionally filled	New (Initiator's previous)	
1.3	Cancel earlier report	TCR	Cancel (1)	Submit (0)	Trade Confirm (0)	Optionally filled	New (Initiator's previous)	Cancel request may be submitted using TradeID if provided by marketplace on initial Ack. Otherwise, TradeReportRefID must be used to reference initial request.
1.4	Acknowledgement from	TCR Ack	New (0),	Initiator's	Trade Confirm (0)	Optionall	Initiator's	Optional. If ack message is used

© Copyright, 2006, FIX Protocol, Limited

	marketplace		Cancel (1), Replace (2)			y filled	(Initiator's)	for Locked-In Reports.
1.5	Reject from marketplace	TCR Ack	New (0), Cancel (1), Replace (2)	Initiator's	Trade Confirm (0)	Optionally filled	Initiator's (Initiator's)	
2	**Marketplace publication of confirmed trade (flow from marketplace to external actors)**							
2.1	Publication to the initiator	TCR	Replace (2)	Submit (0)	Trade Confirm (0)	Filled	Initiator's	MatchStatus = 0 (Compared, matched or affirmed)
2.2	Publication to other parties	TCR	New (0)	Submit (0)	Trade Confirm (0)	Filled	N/A	MatchStatus = 0 (Compared, matched or affirmed)

Requesting the Marketplace to Cancel a Trade

Trade Cancel - One-Party Report for Pass-Thru

	Action	**Message**	**Trade Report Trans Type**	**Trade Report Type**	**Trade Handling Instr**	**Trade ID**	**Trade ReportID (Trade Report Ref ID)**	**Comment**
1	**Initiator requests (main flow from initiator to marketplace):**							
1.1	Enter request to cancel trade	TCR	New (0)	Trade Report Cancel (6), (Locked In) Trade Break (7)	One-Party Report for Pass-Thru (1)	Market-place issued	New (N/A)	Note: The TradeReportType depends on the type of Cancellation (Submit can be used when the other two types are not distinguished). Also note that "Release" and "Reversal" may be relevant (t.b.d.) - or are those for the marketplace to decide?
1.2	Update earlier request	TCR	Replace (2)	Trade Report Cancel (6), (Locked In) Trade Break (7)	One-Party Report for Pass-Thru (1)	Market-place issued (optional)	New (Initiator's previous)	

© Copyright, 2006, FIX Protocol, Limited

	Action	Message	Trade Report Trans Type	Trade Report Type	Trade Handling Instr	Trade ID	Trade ReportID (Trade Report Ref ID)	Comment
1.3	Cancel earlier request	TCR	Cancel (1)	Trade Report Cancel (6), (Locked In) Trade Break (7)	One-Party Report for Pass-Thru (1)	Market-place issued (optional)	New (Initiator's previous)	Always applied to prior request, and never to the trade entity itself., When the TradeID is used for reference (and not the TradeReprotRefID), the presence of TradeReportType = 6 or 7 indicates that this is a request to cancel a prior Break or Amend request on a confirmed trade
1.4	Acknowledgement from marketplace	TCR Ack	New (0); Cancel (1); Replace (2)	Trade Report Cancel (6), (Locked In) Trade Break (7)	One-Party Report for Pass-Thru (1)	Initiator's	Initiator's (Initiator's)	Optional. If ack message is used for One-Party Reports.
1.5	Reject from marketplace	TCR Ack	New (0) ; Cancel (1); Replace (2)	Trade Report Cancel (6), (Locked In) Trade Break (7)	One-Party Report for Pass-Thru (1)	Initiator's	Initiator's (Initiator's)	
2	**Counterparty responses (bi-directional flow):**							
2.1	Marketplace forward of initiators "New" cancellation request	TCR	New (0)	Alleged Trade Report Cancel (tbd); Alleged (Locked In) Trade Break (tbd)	One-Party Report for Pass-Thru (1)	Initiator's	New (N/A)	

© Copyright, 2006, FIX Protocol, Limited

	Action	Message	Trade Report Trans Type	Trade Report Type	Trade Handling Instr	Trade ID	Trade ReportID (Trade Report Ref ID)	Comment
2.2	Marketplace forward of initiators "Replace" or "Cancel" cancellation request	TCR	Cancel (1) or Replace (2)	Alleged Trade Report Cancel (tbd); Alleged (Locked In) Trade Break (tbd)	One-Party Report for Pass-Thru (1)	Initiator's (optional)	New (Marketplace previous)	The TradeID is not needed when the request itself is cancelled or replaced! When the TradeID is used for reference (and not the TradeReportRefID), TradeReportType is provided to distinguish a cancel request of confirmed trade from a cancel request of an unconfirmed trade
2.3	Accept cancellation (sent to marketplace)	TCR	Replace (2)	Accept (2)	One-Party Report for Pass-Thru (1)	Initiator's	New (Marketplace's)	
2.4	Decline cancellation (sent to marketplace)	TCR	Replace (2)	Decline (3)	One-Party Report for Pass-Thru (1)	Initiator's	New (Marketplace's)	
2.5	Acknowledgement from marketplace	TCR Ack	Replace (2)	Accept (2), Decline (3)	One-Party Report for Pass-Thru (1)	Initiator's	Counterparty's (Counterparty's)	Optional. If ack message is used for One-Party Reports.
2.6	Reject from marketplace	TCR Ack	Replace (2)	Accept (2), Decline (3)	One-Party Report for Pass-Thru (1)	Initiator's	Counterparty's (Counterparty's)	
3	**Marketplace relay of counterparty Decline (flow from marketplace to initiator)**							
3.1	Marketplace cancels the process due to a decline from the counterparty	TCR	Cancel (1)	Decline (3)	One-Party Report for Pass-Thru (1)	N/A	New (Initiator's)	

© Copyright, 2006, FIX Protocol, Limited

	Action	Message	Trade Report Trans Type	Trade Report Type	Trade Handling Instr	Trade ID	Trade ReportID (Trade Report Ref ID)	Comment
4	**Marketplace publication of confirmed trade bust (flow from marketplace to external actors)**							
4.1	Publication to the initiator and counterparty	TCR	Replace (2)	Trade Report Cancel (6), (Locked In) Trade Break (7)	Trade Confirm (0)	Filled	Reference	MatchStatus = 0 (Compared, matched or affirmed) The TradeReportRefID depends on who the receiver is.
4.2	Publication to other parties	TCR	New (0)	Trade Report Cancel (6), (Locked In) Trade Break (7)	Trade Confirm (0)	Filled	N/A	MatchStatus = 0 (Compared, matched or affirmed)

Trade Cancel - One-Party Report for Matching

	Action	Message	Trade Report Trans Type	Trade Report Type	Trade Handling Instr	Trade ID	Trade ReportID (Trade Report Ref ID)	Comment
1	**Either side requests (main flow from party to marketplace):**							
1.1	Enter request to cancel trade	TCR	New (0)	Trade Report Cancel (6), (Locked In) Trade Break (7)	One-Party Report for Matching (2)	Market-place issued	New (N/A)	

© Copyright, 2006, FIX Protocol, Limited

	Action	Message	Trade Report Trans Type	Trade Report Type	Trade Handling Instr	Trade ID	Trade ReportID (Trade Report Ref ID)	Comment
1.2	Update earlier request	TCR	Replace (2)	Trade Report Cancel (6), (Locked In) Trade Break (7)	One-Party Report for Matching (2)	Market-place issued (optional)	New (Initiator's previous)	Always applied to prior request, and never to the trade entity itself. The presence of TradeReportType=6 or 7 indicates that this is a request to cancel a prior Break request on a confirmed trade
1.3	Cancel earlier request	TCR	Cancel (1)	Trade Report Cancel (6), (Locked In) Trade Break (7)	One-Party Report for Matching (2)	Market-place issued (optional)	New (Initiator's previous)	
1.4	Acknowledgement from marketplace	TCR Ack	New (0); Cancel (1); Replace (2)	Trade Report Cancel (6), (Locked In) Trade Break (7)	One-Party Report for Matching (2)	Initiator's	Initiator's (Initiator's)	Optional. If ack message is used for One-Party Reports.
1.5	Reject from marketplace	TCR Ack	New (0); Cancel (1); Replace (2)	Trade Report Cancel (6), (Locked In) Trade Break (7)	One-Party Report for Matching (2)	Initiator's	Initiator's (Initiator's)	
2	**Optional relay of request to counterparty**							
2.1	Marketplace forward of initiators "New" report (Alleged)	TCR	New (0)	Alleged Trade Report Cancel (tbd); Alleged (Locked In) Trade Break (tbd)	One-Party Report for Matching (2)	Initiator's	New (N/A)	

© Copyright, 2006, FIX Protocol, Limited

	Action	Message	Trade Report Trans Type	Trade Report Type	Trade Handling Instr	Trade ID	Trade ReportID (Trade Report Ref ID)	Comment
2.2	Marketplace forward of initiators "Replace" or "Cancel" report (Alleged)	TCR	Cancel (1) or Replace (2)	Alleged Trade Report Cancel (tbd); Alleged (Locked In) Trade Break (tbd)	One-Party Report for Matching (2)	Initiator's (optional)	New (Marketplace's Previous)	The TradeID is not needed when the request itself is cancelled or replaced!
3	**Marketplace publication of confirmed trade bust (flow from marketplace to external actors)**							
3.1	Publication to the initiator and counterparty	TCR	Replace (2)	Trade Report Cancel (6), (Locked In) Trade Break (7)	Trade Confirm (0)	Filled	Reference	MatchStatus = 0 (Compared, matched or affirmed) The TradeReportRefID depends on who the receiver is.
3.2	Publication to other parties	TCR	New (0)	Trade Report Cancel (6), (Locked In) Trade Break (7)	Trade Confirm (0)	Filled	N/A	MatchStatus = 0 (Compared, matched or affirmed)

Trade Cancel - Two-Party Report

	Action	Message	Trade Report Trans Type	Trade Report Type	Trade Handling Instr	Trade ID	Trade ReportID (Trade Report Ref ID)	Comment
1	**Actor requests (main flow from actor to marketplace):**							

© Copyright, 2006, FIX Protocol, Limited

	Action	Message	Trade Report Trans Type	Trade Report Type	Trade Handling Instr	Trade ID	Trade ReportID (Trade Report Ref ID)	Comment
1.1.	Enter new report	TCR	New (0)	Trade Report Cancel (6), (Locked In) Trade Break (7)	Two-Party Report (3)	Market-place issued	New (N/A)	
1.2	Update earlier request	TCR	Replace (2)	Trade Report Cancel (6), (Locked In) Trade Break (7)	Two-Party Report (3)	Market-place issued (optional)	New (Initiator's previous)	
1.3	Cancel earlier request	TCR	Cancel (1)	Trade Report Cancel (6), (Locked In) Trade Break (7)	Two-Party Report (3)	Market-place issued (optional)	New (Initiator's previous)	Always applied to prior request, and never to the trade entity itself. The presence of TradeReportType=6 or 7 indicates that this is a request to cancel a prior Break request on a confirmed trade
1.4	Acknowledgement from marketplace	TCR Ack	New (0); Cancel (1); Replace (2)	Trade Report Cancel (6), (Locked In) Trade Break (7)	Two-Party Report (3)	Initiator's	Initiator's (Initiator's)	Optional. If ack message is used for Two-Party Reports.
1.5	Reject from marketplace	TCR Ack	New (0); Cancel (1); Replace (2)	Trade Report Cancel (6), (Locked In) Trade Break (7)	Two-Party Report (3)	Initiator's	Initiator's (Initiator's)	
2	**Marketplace publication of confirmed trade bust (flow from marketplace to external actors)**							
2.1	Publication to the initiator	TCR	Replace (2)	Trade Report Cancel (6),	Trade Confirm (0)	Filled	Initiator's	MatchStatus = 0 (Compared, matched or affirmed)

© Copyright, 2006, FIX Protocol, Limited

	Action	Message	Trade Report Trans Type	Trade Report Type	Trade Handling Instr	Trade ID	Trade ReportID (Trade Report Ref ID)	Comment
2.2	Publication to other parties	TCR	New (0)	(Locked In) Trade Break (7) Trade Report Cancel (6), (Locked In) Trade Break (7)	Trade Confirm (0)	Filled	N/A	MatchStatus = 0 (Compared, matched or affirmed)

Trade Cancel - Locked-In Cancellation

	Action	Message	Trade Report Trans Type	Trade Report Type	Trade Handling Instr	Trade ID	Trade ReportID (Trade Report Ref ID)	Comment
1	**Actor requests (main flow from actor to marketplace):**							
1.1	Enter new report	TCR	New (0)	Trade Report Cancel (6), (Locked In) Trade Break (7)	Trade Confirm (0)	Market-place issued	New (N/A)	
1.2	Update earlier request	TCR	Replace (2)	Trade Report Cancel (6), (Locked In) Trade Break (7)	Trade Confirm (0)	Market-place issued (optional)	New (Initiator's previous)	
1.3	Cancel earlier request	TCR	Cancel (1)	Trade Report Cancel (6), (Locked In) Trade Break (7)	Trade Confirm (0)	Marketplace issued (optional)	New (Initiator's previous)	Always applied to prior request, and never to the trade entity itself. The presence of TradeReportType=6 or 7 indicates that this is a request to cancel a

© Copyright, 2006, FIX Protocol, Limited

	Action	Message	Trade Report Trans Type	Trade Report Type	Trade Handling Instr	Trade ID	Trade ReportID (Trade Report Ref ID)	Comment
								prior Break request on a confirmed trade
1.4	Acknowledgement from marketplace	TCR Ack	New (0); Cancel (1); Replace (2)	Trade Report Cancel (6), (Locked In) Trade Break (7)	Trade Confirm (0)	Initiator's	Initiator's (Initiator's)	Optional. If ack message is used for Locked-In Reports.
1.5	Reject from marketplace	TCR Ack	New (0); Cancel (1); Replace (2)	Trade Report Cancel (6), (Locked In) Trade Break (7)	Trade Confirm (0)	Initiator's	Initiator's (Initiator's)	
2	**Marketplace publication of confirmed trade bust (flow from marketplace to external actors)**							
2.1	Publication to the initiator	TCR	Replace (2)	Trade Report Cancel (6), (Locked In) Trade Break (7)	Trade Confirm (0)	Filled	Initiator's	MatchStatus = 0 (Compared, matched or affirmed)
2.2	Publication to other parties	TCR	New (0)	Trade Report Cancel (6), (Locked In) Trade Break (7)	Trade Confirm (0)	Filled	N/A	MatchStatus = 0 (Compared, matched or affirmed)

© Copyright, 2006, FIX Protocol, Limited

Trade Amendment

Trade Amendment - One-Party Report for Pass-Thru

	Action	Message	Trade Report Trans Type	Trade Report Type	Trade Handling Instr	Trade ID	Trade ReportID (Trade Report Ref ID)	Comment
1	**Initiator requests (main flow from initiator to marketplace):**							
1.1	Enter request to amend trade	TCR	New (0)	Addendum (4), No/Was (5)	One-Party Report for Pass-Thru (1)	Market-place issued	New (N/A)	Note: The TradeReportType depends on the type of update needed (Submit can be used when the other two types are not distinguished). Also note: That "addendums" normally would not need counterparty acknowledgement (as the update is limited to initiators side)
1.2	Update earlier request	TCR	Replace (2)	Addendum (4), No/Was (5)	One-Party Report for Pass-Thru (1)	Market-place issued (optional)	New (Initiator's previous)	
1.3	Cancel earlier request	TCR	Cancel (1)	Addendum (4), No/Was (5)	One-Party Report for Pass-Thru (1)	Market-place issued (optional)	New (Initiator's previous)	
1.4	Acknowledgement from marketplace	TCR Ack	New (0); Cancel (1); Replace (2)	Addendum (4), No/Was (5)	One-Party Report for Pass-Thru (1)	Initiator's	Initiator's (Initiator's)	Optional. If ack message is used for One-Party Reports.
1.5	Reject from marketplace	TCR Ack	New (0); Cancel (1);	Addendum (4), No/Was (5)	One-Party Report for Pass-Thru (1)	Initiator's	Initiator's (Initiator's)	

© Copyright, 2006, FIX Protocol, Limited

	Action	Message	Trade Report Trans Type	Trade Report Type	Trade Handling Instr	Trade ID	Trade ReportID (Trade Report Ref ID)	Comment
			Replace (2)					
2	**Counterparty responses (bi-directional flow):**							
2.1	Marketplace forward of initiators "New" amendment request	TCR	New (0)	Alleged Addendum (tbd); Alleged No/Was (tbd)	One-Party Report for Pass-Thru (1)	Initiator's	New (N/A)	
2.2	Marketplace forward of initiators "Cancel" or "Replace" amendment request	TCR	Cancel (1) or Replace (2)	Alleged Addendum (tbd); Alleged No/Was (tbd)	One-Party Report for Pass-Thru (1)	Initiator's (optional)	New (Marketplace previous)	The TradeID is not needed when the request itself is cancelled or replaced!
2.3	Accept amendment (sent to marketplace)	TCR	Replace (2)	Accept (2)	One-Party Report for Pass-Thru (1)	Initiator's	New (Marketplace's)	
2.4	Decline amendment (sent to marketplace)	TCR	Replace (2)	Decline (3)	One-Party Report for Pass-Thru (1)	Initiator's	New (Marketplace's)	
2.5	Acknowledgement from marketplace	TCR Ack	Cancel (1); Replace (2)	Accept (2), Decline (3)	One-Party Report for Pass-Thru (1)	Initiator's	Counterparty's (Counterparty's)	Optional. If ack message is used for One-Party Reports.
2.6	Reject from marketplace	TCR Ack	Cancel (1); Replace (2)	Accept (2), Decline (3)	One-Party Report for Pass-Thru (1)	Initiator's	Counterparty's (Counterparty's)	
3	**Marketplace relay of counterparty Decline (flow from marketplace to initiator)**							

© Copyright, 2006, FIX Protocol, Limited

	Action	Message	Trade Report Trans Type	Trade Report Type	Trade Handling Instr	Trade ID	Trade ReportID (Trade Report Ref ID)	Comment
3.1	Marketplace cancels the process due to a decline from the counterparty	TCR	Cancel (1)	Decline (3)	One-Party Report for Pass-Thru (1)	N/A	New (Initiator's)	
4	**Marketplace publication of confirmed trade amendment (flow from marketplace to external actors)**							
4.1	Publication to the initiator and counterparty	TCR	Replace (2)	Addendum (4), No/Was (5)	Trade Confirm (0)	Filled	Reference	MatchStatus = 0 (Compared, matched or affirmed) The TradeReportRefID depends on who the receiver is.
4.2	Publication to other parties	TCR	New (0)	Addendum (4), No/Was (5)	Trade Confirm (0)	Filled	N/A	MatchStatus = 0 (Compared, matched or affirmed)

© Copyright, 2006, FIX Protocol, Limited

Trade Amendment – One-Party Report for Matching

	Action	Message	Trade Report Trans Type	Trade Report Type	Trade Handling Instr	Trade ID	Trade ReportID (Trade Report Ref ID)	Comment
1	**Either side requests (main flow from party to marketplace):**							
1.1	Enter request to amend trade	TCR	New (0)	Addendum (4), No/Was (5)	One-Party Report for Matching (2)	Market-place issued	New (N/A)	Note: The TradeReportType depends on the type of update is needed (Submit can be used when the other two types are not distinguished). Also note: That "addendums" normally would not need counterparty acknowledgement (as the update is limited to initiators side)
1.2	Update earlier request	TCR	Replace (2)	Addendum (4), No/Was (5)	One-Party Report for Matching (2)	Market-place issued (optional)	New (Initiator's previous)	
1.3	Cancel earlier request	TCR	Cancel (1)	Addendum (4), No/Was (5)	One-Party Report for Matching (2)	Market-place issued (optional)	New (Initiator's previous)	
1.4	Acknowledgement from marketplace	TCR Ack	New (0); Cancel (1); Replace (2)	Addendum (4), No/Was (5)	One-Party Report for Matching (2)	Initiator's	Initiator's (Initiator's)	Optional. If ack message is used for One-Party Reports.
1.5	Reject from marketplace	TCR Ack	New (0); Cancel (1); Replace (2)	Addendum (4), No/Was (5)	One-Party Report for Matching (2)	Initiator's	Initiator's (Initiator's)	
2	**Optional relay of request to counterparty**							

© Copyright, 2006, FIX Protocol, Limited

2.1	Marketplace forward of initiators "New" report (Alleged)	TCR	New (0)	Alleged Addendum (tbd); Alleged No/Was (tbd)	One-Party Report for Matching (2)	Initiator's	New (N/A)	
2.2	Marketplace forward of initiators "Cancel" or "Replace" report (Alleged)	TCR	Cancel (1) or Replace (2)	Alleged (1)	One-Party Report for Matching (2)	Initiator's (optional)	New (Marketplace's Previous)	The TradeID is not needed when the request itself is cancelled or replaced!
3	**Marketplace publication of confirmed trade amendment (flow from marketplace to external actors)**							
3.1	Publication to the initiator and counterparty	TCR	Replace (2)	Addendum (4), No/Was (5)	Trade Confirm (0)	Filled	Reference	MatchStatus = 0 (Compared, matched or affirmed) The TradeReportRefID depends on who the receiver is.
3.2	Publication to other parties	TCR	New (0)	Addendum (4), No/Was (5)	Trade Confirm (0)	Filled	N/A	MatchStatus = 0 (Compared, matched or affirmed)

Trade Amendment - Two-Party Report

	Action	**Message**	**Trade Report Trans Type**	**Trade Report Type**	**Trade Handling Instr**	**Trade ID**	**Trade ReportID (Trade Report Ref ID)**	**Comment**
2	**Actor requests (main flow from actor to marketplace):**							
1.1	Enter request to amend trade	TCR	New (0)	Addendum (4), No/Was (5)	Two-Party Report (3)	Market-place issued	New (N/A)	Note: The TradeReportType depends on the type of update is needed (Submit can be used when the other two types are not distinguished).

© Copyright, 2006, FIX Protocol, Limited

	Action	Message	Trade Report Trans Type	Trade Report Type	Trade Handling Instr	Trade ID	Trade ReportID (Trade Report Ref ID)	Comment
1.2	Update earlier request	TCR	Replace (2)	Addendum (4), No/Was (5)	Two-Party Report (3)	Market-place issued (optional)	New (Initiator's previous)	
1.3	Cancel earlier request	TCR	Cancel (1)	Addendum (4), No/Was (5)	Two-Party Report (3)	Market-place issued (optional)	New (Initiator's previous)	
1.4	Acknowledgement from marketplace	TCR Ack	New (0); Cancel (1); Replace (2)	Addendum (4), No/Was (5)	Two-Party Report (3)	Initiator's	Initiator's (Initiator's)	Optional. If ack message is used for Two-Party Reports.
1.5	Reject from marketplace	TCR Ack	New (0); Cancel (1); Replace (2)	Addendum (4), No/Was (5)	Two-Party Report (3)	Initiator's	Initiator's (Initiator's)	
2	**Marketplace publication of confirmed trade amendment (flow from marketplace to external actors)**							
2.1	Publication to the initiator	TCR	Replace (2)	Addendum (4), No/Was (5)	Trade Confirm (0)	Filled	Initiator's	MatchStatus = 0 (Compared, matched or affirmed)
2.2	Publication to other parties	TCR	New (0)	Addendum (4), No/Was (5)	Trade Confirm (0)	Filled	N/A	MatchStatus = 0 (Compared, matched or affirmed)

© Copyright, 2006, FIX Protocol, Limited

Trade Amendment - Locked-In Amendment

	Action	Message	Trade Report Trans Type	Trade Report Type	Trade Handling Instr	Trade ID	Trade ReportID (Trade Report Ref ID)	Comment
1	**Actor requests (main flow from actor to marketplace):**							
1.1	Enter request to amend trade	TCR	New (0)	Addendum (4), No/Was (5)	Trade Confirm (0)	Market-place issued	New (N/A)	Note: The TradeReportType depends on the type of update is needed (Submit can be used when the other two types are not distinguished).
1.2	Update earlier request	TCR	Replace (2)	Addendum (4), No/Was (5)	Trade Confirm (0)	Market-place issued (optional)	New (Initiator's previous)	
1.3	Cancel earlier request	TCR	Cancel (1)	Addendum (4), No/Was (5)	Trade Confirm (0)	Market-place issued (optional)	New (Initiator's previous)	
1.4	Acknowledgement from marketplace	TCR Ack	New (0); Cancel (1); Replace (2)	Submit (0), Addendum (4), No/Was (5)	Trade Confirm (0)	Initiator's	Initiator's (Initiator's)	Optional. If ack message is used for Locked-In Reports.
1.5	Reject from marketplace	TCR Ack	New (0); Cancel (1); Replace (2)	Submit (0), Addendum (4), No/Was (5)	Trade Confirm (0)	Initiator's	Initiator's (Initiator's)	
	Marketplace publication of confirmed trade amendment (flow from marketplace to external actors)							
2.1	Publication to the initiator	TCR	Replace (2)	Addendum (4), No/Was (5)	Trade Confirm (0)	Filled	Initiator's	MatchStatus = 0 (Compared, matched or affirmed)
2.2	Publication to other parties	TCR	New (0)	Addendum (4), No/Was (5)	Trade Confirm (0)	Filled	N/A	MatchStatus = 0 (Compared, matched or affirmed)

© Copyright, 2006, FIX Protocol, Limited

© Copyright, 2006, FIX Protocol, Limited