Post-Trade Processing via FIX Recommended Practices - FX

February 07, 2019
Revision 0.08
Proposal Status: Approved Draft

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.

DRAFT OR NOT RATIFIED PROPOSALS (REFER TO PROPOSAL STATUS AND/OR SUBMISSION STATUS ON COVER PAGE) ARE PROVIDED "AS IS" TO INTERESTED PARTIES FOR DISCUSSION ONLY. PARTIES THAT CHOOSE TO IMPLEMENT THIS DRAFT PROPOSAL DO SO AT THEIR OWN RISK. IT IS A DRAFT DOCUMENT AND MAY BE UPDATED, REPLACED, OR MADE OBSOLETE BY OTHER DOCUMENTS AT ANY TIME. THE FPL GLOBAL TECHNICAL COMMITTEE WILL NOT ALLOW EARLY IMPLEMENTATION TO CONSTRAIN ITS ABILITY TO MAKE CHANGES TO THIS SPECIFICATION PRIOR TO FINAL RELEASE. IT IS INAPPROPRIATE TO USE FPL WORKING DRAFTS AS REFERENCE MATERIAL OR TO CITE THEM AS OTHER THAN “WORKS IN PROGRESS”. THE FPL GLOBAL TECHNICAL COMMITTEE WILL ISSUE, UPON COMPLETION OF REVIEW AND RATIFICATION, AN OFFICIAL STATUS ("APPROVED") OF/FOR THE PROPOSAL AND A RELEASE NUMBER.

No proprietary or ownership interest of any kind is granted with respect to the FIX Protocol (or any rights therein).

Table of Contents

Document History

Revision	Date	Author/Editor	Revision Comments
0.01	1/14/2014	D. Tolman	
0.02	3/18/2014	D. Tolman	
0.03	5/28/2016	D. Tolman	- QuoteStatus(297) replaced by QuotEntryStatus(1167)
0.04	4/9/2018	D. Tolman	- added [FIX 5.0 or later] notes - updated the notes on pre-allocated part-fill cases. - miscellaneous cleanup
0.05	Sept. 9, 2018	GTC	Extensively updated and refactored to restrict the coverage of the initial release of this document to address FX spot, forward and NDF placements, allocations and confirmation only. Changes and examples are based on three separate in-person working group meetings in July and August 2018.
0.06	Sept. 17, 2018	GTC	Incorporating PT Working group feedback.
0.07	Sept. 20, 2018	GTC	Updated field name and values for ‘OffshoreIndicator(tbd)’, added it to AllocationInstruction(J) message. Updated field name and added description for ‘FXBenchmarkRateFix(tbd)’. Removed CFICode (461) from NewOrderSingle(D), ExecutionReport(8) and AllocationInstruction(J) messages. Updated footnote.
0.08	February 07, 2019	GTC	Added message tables for TradeAggregationRequest(DW) and TradeAggregationReport(DX) Updated field #s for tbd fields Updated workflow diagram for correct message types and fields #s

1 Preface

The purpose of the FIX Trading Community Post-Trade Processing via FIX Initiative is to define industry practices for common usage of the FIX Protocol for post-trade processing, for all asset classes, between buy-sides and sell-sides that can be used bi-laterally as well as through intermediary facilities.

This document is one of a series of Recommended Practices for Post-Trade Processing via FIX specifying guidelines for industry usage of the FIX standard to facilitate parallel implementation across buy-sides, sell-sides and intermediaries.

This document assumes an understanding of the FIX Protocol and post-trade processing in general. This document is written in the context of the Common Post-trade Framework.

Note: While the base protocol is FIX 4.4, additional tags or additional valid values from FIX 5.0 or later have been added as needed to meet industry post-trade processing requirements. These are identified in the message format tables ("[FIX 5.0 or later]") and may require specific exception configuration for FIX engines. The FIX Global Technical Committee has approved this as accepted practice.

2 Scope

This document includes the FIX recommended practices for the following FX asset types:

- spot
- vanilla forwards
- vanilla non-deliverable forwards (NDF)

Workflows covered for the above FX asset types are:

- post-trade allocations
- trade aggregation

The workflows are described primarily as bilateral workflows (i.e. directly between investment managers and bank/broker). The intent is for the recommended practices to be as agnostic as possible to an involvement of a third-party portal or service interface in between the investment manager and bank/broker, but the specifics with 3-party flows will not be explicitly covered.

2.1 *Out of scope*

Clearing and settlement workflows are outside the scope of this with the exception of data required to be passed to these phases.

Request for Quote, Streaming and Pre-trade allocation will be covered in a separate recommended practices document.

Placement, RFQ, spot roll, trade aggregation, allocation, confirmation, and netting workflows behaviors that are specific to how market participants interact with third party platforms/utilities, which may differ from a bilateral workflow, are currently out of scope.

3 References

The following are the associated documents that have been completed at time of distribution.

3.1 *Post-Trade via FIX Recommended Practices*

For the current complete set of recommended practices, please refer to the FIX Trading Community website link. The documents listed on this web page are produced by various working groups in collaboration with Global Technical Committee and FIX Community.

https://www.fixtrading.org/recommended-practicesguidelines/

3.1.1 Common Framework

For common usage of the FIX protocol for post-trade processing between buy-sides and sell-sides, please refer to the document "PostTradeViaFIX_RecommendedPractices_CommonFramework.doc". The document covers the technical details that are asset class agnostic, e.g. confirmation status values, which are not covered in this document.

3.1.2 Asset class specific

For asset class specific post-trade processing recommendations, please refer to the documents mentioned below:

Equities and Equity Swaps:

- PostTradeViaFIX_RecommendedPractices_Equities.doc
- PostTradeViaFIX_RecommendedPractices_Equities_ExamplesAndNotes.doc
- PostTradeViaFIX_RecommendedPractices_EquitySwaps.doc

For options and futures:

- PostTradeViaFIX_RecommendedPractices_EquityOptions.doc
- PostTradeViaFIX_RecommendedPractices_Futures.doc

For cross asset special cases:

- PostTradeViaFIX_RecommendedPractices_SEF.doc
- PostTradeViaFIX_RecommendedPractices_BrokerAllege_AMCConfirmationRequest.doc

3.1.3 Code Lists

For fields utilizing an external code list, please refer to the FIX Trading Community website link.

https://www.fixtrading.org/standards/codelists/

4 FIX Message Types

All messages adhere to the FIX standard unless otherwise noted. The following FIX message types are utilized for FX:

Tag 35	Name	Description
“D”	NewOrderSingle	Initiates a single placement
“F”	OrderCancelRequest	Initiates a request for a cancel.
“G”	OrderCancelReplaceRequest	Initiates a request to replace a placement
“8”	ExecutionReport	Broker response to placements reporting on placement status changes. Some changes to placement status may be "unsolicited" (i.e. initiated by the bank/broker)
“Q”	DontKnowTrade	Used to reject unknown execution report messages
“9”	OrderCancelReject	Broker cannot process requested cancel or replace
“J”	AllocationInstruction	Provides Allocation Instructions for post-trade allocation style workflow
“P”	AllocationInstructionAck	“accept” or “reject” the AllocationInstruction
“DW”	TradeAggregationRequest	Initiates a request to aggregate trades
“DX”	TradeAggregationReport	“accept” or “reject” the TradeAggregationRequest

5 FX Key Concepts

5.1 *General Concepts*

FX assets has some unique concepts that differ from other asset classes. A FX transaction typically consists of an exchange of one currency for another at an agreed upon rate. The following describes how FX concepts are expressed in FIX:

- **FX asset classification**

When identifying the type of FX asset, the SecurityType(167) field is used to provide the explicit classification. Optionally the Product(460) field may also be used to identify the broader classification when Product(460)=4 (Currency) is specified.

  - SecurityType(167)="FXSPOT" identifies an FX spot transaction
  - SecurityType(167)="FXFWD" identifies an FX forward
  - SecurityType(167)="FXNDF" identifies an FX non-deliverable forward

- **Currency pair and dealt currency**

The currency pair of the transaction is identified in the Instrument component's Symbol(55) field. The currency pair expressed is "ccy1/ccy2" in the convention used by the market.

The amount being traded is expressed in the denomination of the dealt currency, which is identified in the Currency(15) field.

- **Direction**

The transaction's direction, buy or sell, is specified in the Side(54) field. This identifies whether the initiator is buying or selling the dealt currency.

- **Rate expression**

Unless explicitly specified, the rate is expressed as normal market convention. Should the rate need to be expressed as an inverse of the specified currency pair, PriceType(423)=21 (Inverse rate representation) maybe used to express the rate expression. PriceType(423)=20 (Normal rate representation) could be used to explicitly indicate that the rate is expressed as normal market convention for the currency pair.

- **Rate or price convention**

FX transaction price in FIX is typically an "all-in" price and conveyed in the Price(44) or LastPx(31) fields.

- **Settlement type and settlement date**

Transactions settlement type and date is expressed in the SettlType(63) and SettlDate(64) fields. SettlType(63) maybe used to convey the settlement convention for the transaction, for example:

  - SettlType(63)=0 (Regular / FX spot settlement)
  - SettlType(63)=1 (Cash (TOD/T+0)) - per market convention for the currency pair
  - SettlType(63)=2 (Next day (TOM/T+1)) - per market convention for the currency pair

  - SettlType(63)=B (Broken date) - the actual settlement date is to be specified in SettlDate(64)
  - SettlType(63)=C (FX Spot Next settlement (Spot+1 / next day))

When both SettlType(63) and SettlDate(64) are included, SettlDate(64) value prevails when there is a "conflict" between SettlType(63) and SettlDate(64).

- **Tenor**

For pre-trade workflows (e.g. RFQ, streaming or order placements) the intended settlement may be expressed as a tenor, e.g. 1-month. Tenor expressions specified in the SettlType(63) field using a tenor pattern in the following manner:

  - Dx = FX tenor expression for "days", e.g. "D5", where "x" is any integer > 0 to denote the number of days
  - Mx = FX tenor expression for "months", e.g. "M3", where "x" is any integer > 0 to denote the number of months
  - Wx = FX tenor expression for "weeks", e.g. "W13", where "x" is any integer > 0 to denote the number of weeks
  - Yx = FX tenor expression for "years", e.g. "Y1", where "x" is any integer > 0 to denote the number of years

For FX the tenors expressed using Dx, Mx, Wx, and Yx values do not denote business days, but calendar days.

- **Settlement currency and amount**

The settlement currency of the transaction is specified in SettlCurrency(120). Typically this would be the non-dealt (i.e. the opposite currency) currency in the currency pair, except for non-deliverable forwards where the settlement maybe in a third currency from the specified currency pair.

For spot and forwards, the settlement currency is the opposite of the dealt currency.

For non-deliverable forwards, the settlement currency may be in the dealt currency or in a third (deliverable) currency

- **Fixing date**

For non-deliverable forwards the fixing date is specified in the MaturityDate(541) field.

5.2 *Trade Aggregation*

Trade aggregation occurs post execution but prior to allocation. This is an optional operational step that may occur between buy-side and sell-side firms. Trade aggregation is an instruction from the buy-side to effectively "block up" the individual trade executions or fills in order to be allocated.

5.3 *Allocation*

5.3.1 Post Trade Allocation

In post-trade allocation, allocation instructions are provided after execution is complete. Post-trade allocations may be used for expected fully filled placements instead of pre-trade allocation, but are required for cases where partially filled placements are expected.

Using the AllocationInstructions(35=J) message, the buy-side specifies allocation instructions specifying the following components and fields:

- OrdAllocGrp component - indicating a single instance of the repeating group to identify the placement or aggregated trade identifier in ClOrdID(11)
- ExecAllocGrp component - indicating the execution fill(s) that are part of the allocation
    - LastQty(32) - the amount of the transaction expressed in dealt currency
    - ExecID(17) used to identify the dealer/bank's identifier for the transaction
    - LastPx(31) - the rate at which the referenced fill was executed at
- AllocGrp component - specifies how the total amount is to be allocated across the specified accounts, including the amount to be allocated to each account. In general, the sum of the amounts at the allocated level should add up to the total amount.
    - AllocAccount(79) = < client account ID >
    - AllocAcctIDSource(661) - May optionally be used to identify the source scheme of the account identifier in AllocAccount(79)
    - AllocSettlCurrency(736) = <ISO currency code> - Optionally used to specify the settlement currency for the account
    - AllocQty(80) = <quantity> - The amount to be allocated to the specified account, expressed in dealt currency terms.
    - IndividualAllocID(467) = <unique transaction-id> - Uniquely identifies the allocation instance. An identifier specified here will be carried through to other messages during post-trade processing including the Confirmation(35=AK) message.

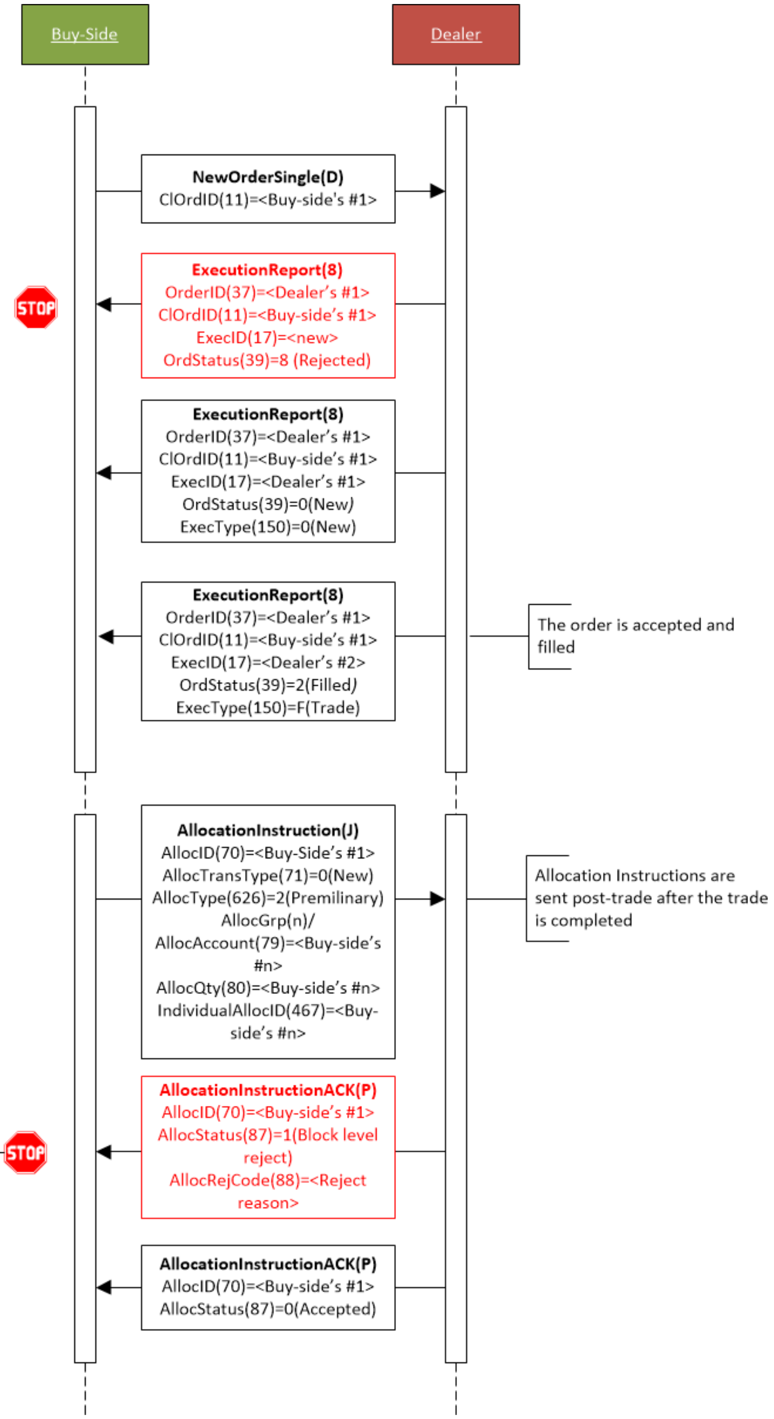
5.4 *Confirmation/Affirmation*

These steps are not covered in this document and may be undertaken by other messaging standards such as MT (SWIFT's implementation of ISO 15022) or MX (SWIFT's implementation of ISO 20022).

6 FX FIX Message Workflow

6.1 *Post-Trade Allocation*

Figure 2: Post trade allocation after trade execution.


Buy-Side
Dealer
NewOrderSingle(D)
ClOrdID(11)=<Buy-side's #1>
ExecutionReport(8)
OrderID(37)=<Dealer's #1>
ClOrdID(11)=<Buy-side's #1>
ExecID(17)=<new>
OrdStatus(39)=8 (Rejected)
STOP
ExecutionReport(8)
OrderID(37)=<Dealer's #1>
ClOrdID(11)=<Buy-side's #1>
ExecID(17)=<Dealer's #1>
OrdStatus(39)=0(New)
ExecType(150)=0(New)
ExecutionReport(8)
OrderID(37)=<Dealer's #1>
ClOrdID(11)=<Buy-side's #1>
ExecID(17)=<Dealer's #2>
OrdStatus(39)=2(Filled)
ExecType(150)=F(Trade)
AllocationInstruction(J)
AllocID(70)=<Buy-Side's #1>
AllocTransType(71)=0(New)
AllocType(626)=2(Premilinary)
AllocGrp(n)/
AllocAccount(79)=<Buy-side's #n>
AllocQty(80)=<Buy-side's #n>
IndividualAllocID(467)=<Buy-side's #n>
AllocationInstructionACK(P)
AllocID(70)=<Buy-side's #1>
AllocStatus(87)=1(Block level reject)
AllocRejCode(88)=<Reject reason>
STOP
AllocationInstructionACK(P)
AllocID(70)=<Buy-side's #1>
AllocStatus(87)=0(Accepted)

The order is accepted and filled

Allocation Instructions are sent post-trade after the trade is completed

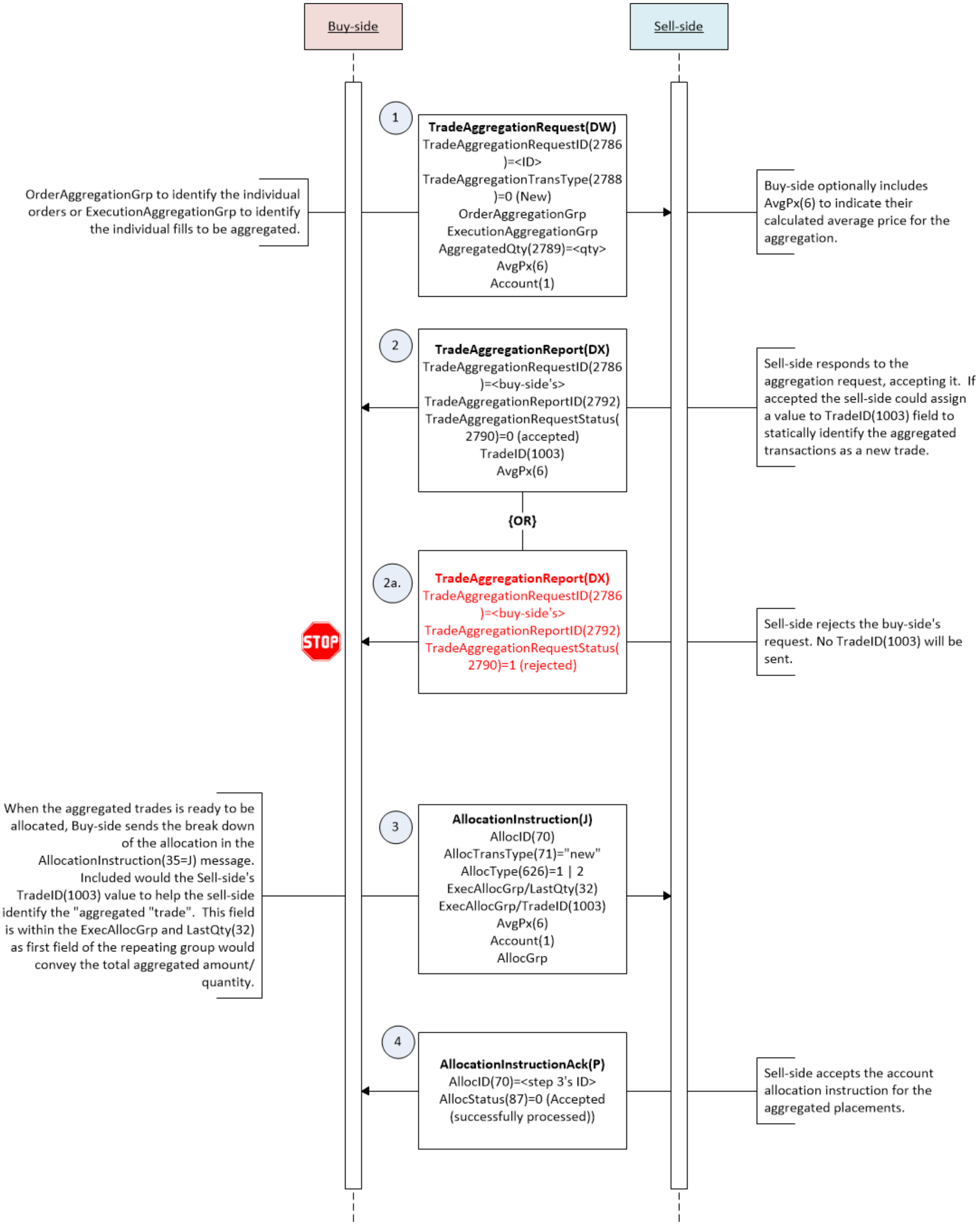
Execution stands, but remains unallocated. The dealer can send new allocation instructions with the correct information.

6.2 *Trade Aggregation*

The following diagram illustrates the typical trade aggregation workflow. In the initiation of the trade aggregation request, the buy-side may optionally specify their calculated average price for the entire aggregation - while optional it is recommended that the buy-side sends this if they are able to run the calculation.

The sell-side will either accept or reject the aggregation request. In the case of a rejection the current recommendation is to resolve the error out-of-band.

Figure 3: Basic trade aggregation


Buy-side
Sell-side
1
TradeAggregationRequest(DW)
TradeAggregationRequestID(2786)=<ID>
TradeAggregationTransType(2788)=0 (New)
OrderAggregationGrp
ExecutionAggregationGrp
AggregatedQty(2789)=<qty>
AvgPx(6)
Account(1)
2
TradeAggregationReport(DX)
TradeAggregationRequestID(2786)=<buy-side's>
TradeAggregationReportID(2792)
TradeAggregationRequestStatus(2790)=0 (accepted)
TradeID(1003)
AvgPx(6)
{OR}
2a.
TradeAggregationReport(DX)
TradeAggregationRequestID(2786)=<buy-side's>
TradeAggregationReportID(2792)
TradeAggregationRequestStatus(2790)=1 (rejected)
STOP
3
AllocationInstruction(J)
AllocID(70)
AllocTransType(71)="new"
AllocType(626)=1 | 2
ExecAllocGrp/LastQty(32)
ExecAllocGrp/TradeID(1003)
AvgPx(6)
Account(1)
AllocGrp
4
AllocationInstructionAck(P)
AllocID(70)=<step 3's ID>
AllocStatus(87)=0 (Accepted (successfully processed))

OrderAggregationGrp to identify the individual orders or ExecutionAggregationGrp to identify the individual fills to be aggregated.

Buy-side optionally includes AvgPx(6) to indicate their calculated average price for the aggregation.

Sell-side responds to the aggregation request, accepting it. If accepted the sell-side could assign a value to TradeID(1003) field to statically identify the aggregated transactions as a new trade.

Sell-side rejects the buy-side's request. No TradeID(1003) will be sent.

When the aggregated trades is ready to be allocated, Buy-side sends the break down of the allocation in the AllocationInstruction(35=J) message. Included would the Sell-side's TradeID(1003) value to help the sell-side identify the "aggregated "trade". This field is within the ExecAllocGrp and LastQty(32) as first field of the repeating group would convey the total aggregated amount/ quantity.

Sell-side accepts the account allocation instruction for the aggregated placements.

7 FIX 4.4 FX Message Formats

7.1 *Placement Messages*

7.1.1 NewOrderSingle(35=D)

The New Order – Single message is used by the buy-side to place an order with the dealer.

<NewOrderSingle> - Msgtype(35) = D				
Field Name	**Tag**	**FIX Data Type**	**Rq'd**	**Description**
<StandardHeader> *component*			Y	MsgType = D
ClOrdID	11	String	Y	Unique identifier for Order as assigned by the Participant.
<Parties> component			N	This is party information related to the submitter of the request.
NoPartyIDs	453	NumInGroup	N	The number of parties.
→ PartyID	448	String	N	The ID or code used to identify the party.
→ PartyIDSource	447	char	N	The class or source scheme of PartyID(448) value. Supported values: C = Generally accepted market participant identifier
→ PartyRole	452	int	N	The role of the party. Supported values: 11 = Order originating trader 13 = Order originating firm 56 = Acceptable counterparty – Added Standard FIX 5.0 value
end ***<Parties>*** *component*				
TradeDate	75	LocalMktDate	N	Indicates date of trading day. Absence of this field indicates current day.
Account	1	String	Y	Account mnemonic as agreed between the participants.

SettlType	63	String	N	Indicates the period for settlement of the trade.
				Supported values:
				0 = Regular - Default if not specified.
				B = Broken date – Use for FX to specify non-standard tenor. Specify the date in SettlDate(64).
				Dx = Tenor expression for "days", e.g. "D5", where "x" is any integer > 0
				Mx = Tenor expression for "months", e.g. "M3", where "x" is any integer > 0
				Wx = Tenor expression for "weeks", e.g. "W13", where "x" is any integer > 0
				Yx = Tenor expression for "years", e.g. "Y1", where "x" is any integer > 0

SettlDate	64	LocalMktDate	Y	Specific date of trade settlement in YYYYMMDD format.

<NewOrderSingle> - Msgtype(35) = D				
Field Name	**Tag**	**FIX Data Type**	**Rq'd**	**Description**
HandlInst	21	char	N	Instructions for Order handling. Supported values: 3 = Manual order best execution
ExecInst	18	MultipleCharValue	N	Instruction for order handling. Supported values: 1 = Not held – Default if not specified
<Instrument> component			Y	
Symbol	55	String	Y	"CCY1/CCY2", where CCY1 and CCY2 are ISO currency codes.
SecurityType	167	String	Y	Indicates type of security Supported values: FXSPOT FXFWD FXNDF
MaturityDate	541	LocalMktDate	N	For NDFs this represents the fixing date of the contract.
MaturityTime	1079	UTCTimestamp	N	*Standard FIX.5.0 tag added to the message For NDFs this represents the fixing time of the contract. It is optional to specify the fixing time.
*end **<Instrument>** component*				
Side	54	char	Y	Side of the order from the perspective of the submitter. Supported values: 1 = Buy 2 = Sell
TransactTime	60	UTCTimestamp	Y	Time this order request was initiated/released by the trader or trading system.
<OrderQtyData> component			Y	
OrderQty	38	Qty	Y	The currency traded amount.
*end **<OrderQtyData>** component*				
OrdType	40	char	Y	Indicates the type of order. Supported values: 1 = Market
Currency	15	Currency	Y	Denotes dealt currency.
TimeInForce	59	char	Y	Specifies how long the order remains in effect. Supported values: 1 = Good Till Cancel 6 = Good Till Date
ExpireDate	432	UTCTimestamp	N	Conditionally required if TimeInForce(59) = 6 (GTD).

OrderCapacity	528	char	N	Designates the capacity of the firm placing the order.
				Supported values:
				A = Agency
				P = Principal

<NewOrderSingle> - Msgtype(35) = D				
Field Name	**Tag**	**FIX Data Type**	**Rq'd**	**Description**
				R = Riskless Principal
SettlCurrency	120	Currency	Y	Counter currency of the transaction.
<RateSource> component			N	*GAP – Standard 5.0 FIX component added to NewOrderSingle(D) message Used to identify the FX fixing source for FX non-deliverable forward.
NoRateSources	1445	NumInGroup	N	The number of rate source instances.
→ RateSource	1446	int	N	Required if NoRateSources(1445) > 0. Identifies the reference source for FX spot rate. Supported values: 0 = Bloomberg 1 = Reuters 2 = Telerate tbd = EMTA 99 = Other
→ RateSourceType	1447	int	N	Required if NoRateSources(1445) > 0. Supported values: 0 = Primary 1 = Secondary
→ FXBenchmarkRateFix	2796	String	N	Specifies the foreign exchange benchmark rate fixing to be used in valuing the transaction. For Ex – London 4 p.m. or Tokyo 3 p.m.
*end **<RateSource>** component*				
OffshoreIndicator	2795	int	N	Indicates the type of the currency rate being used. This is relevant for currencies that have offshore rate that is different from the onshore rate. Supported values: 0 = Regular – Default if not specified 1 = Offshore 2 = Onshore
<StandardTrailer> component			Y	

7.1.2 ExecutionReport(35=8)

The Execution Report is used by the dealer to provide information regarding the status of the order.

<ExecutionReport> - Msgtype(35) = 8				
Field Name	**Tag**	**FIX Data Type**	**Rq'd**	**Description**
<StandardHeader> *component*			Y	MsgType = 8
OrderID	37	String	Y	Unique identifier for each chain of orders.
ClOrdID	11	String	N	Unique identifier for Order as assigned by the Participant when the order was created or last amended.

<ExecutionReport> - Msgtype(35) = 8				
Field Name	**Tag**	**FIX Data Type**	**Rq'd**	**Description**
<Parties> component			y	This is party information related to the submitter of the request.
NoPartyIDs	453	NumInGroup	Y	The number of parties.
→ PartyID	448	String	Y	The ID or code used to identify the party.
→ PartyIDSource	447	char	Y	The class or source scheme of PartyID(448) value. Supported values: C = Generally accepted market identifier D = Proprietary / Custom code
→ PartyRole	452	int	Y	The role of the party. Supported values: 1 = Executing firm 11 = Order originating trader 13 = Order originating firm 56 = Acceptable counterparty – Added Standard FIX.5.0 value
*end **<Parties>** component*				
ExecID	17	String	Y	Unique identifier of execution message as assigned by sell-side.
ExecRefID	19	String	N	Required for ExecType(150) = H (Trade Cancel) and ExecType(150) = G (Trade Correct).
ExecType	150	char	Y	Describes the purpose of the execution report. Supported values: 0 = New 4 = Canceled 5 = Replaced 6 = Pending Cancel 8 = Rejected A = Pending New E = Pending Replace F = Trade (partial fill or fill) G = Trade Correct H = Trade Cancel
OrdStatus	39	char	Y	Identifies current status of order. Supported values: 0 = New 1 = Partially filled 2 = Filled 4 = Canceled 6 = Pending Cancel 8 = Rejected A = Pending New E = Pending Replace
Account	1	String	N	Account mnemonic as agreed between the participants.
SettlType	63	String	N	Indicates the period for settlement of the trade. Supported values:

<ExecutionReport> - Msgtype(35) = 8				
Field Name	**Tag**	**FIX Data Type**	**Rq'd**	**Description**

				0 = Regular - Default if not specified.
				B = Broken date – Use for FX to specify non-standard tenor. Specify the date in SettlDate(64).
				Dx = Tenor expression for "days", e.g. "D5", where "x" is any integer > 0
				Mx = Tenor expression for "months", e.g. "M3", where "x" is any integer > 0
				Wx = Tenor expression for "weeks", e.g. "W13", where "x" is any integer > 0
				Yx = Tenor expression for "years", e.g. "Y1", where "x" is any integer > 0

SettlDate	64	LocalMktDate	Y	Specific date of trade settlement in YYYYMMDD format.
<Instrument> component			Y	
Symbol	55	String	Y	"CCY1/CCY2", where CCY1 and CCY2 are ISO currency codes.
SecurityType	167	String	Y	Indicates type of security Supported values: FXSPOT FXFWD FXNDF
MaturityDate	541	LocalMktDate	N	For NDFs this represents the fixing date of the contract.
MaturityTime	1079	UTCTimestamp	N	*Standard FIX 5.0 tag added to the message. For NDFs this represents the fixing time of the contract. It is optional to specify the fixing time.
*end **<Instrument>** component*				
Side	54	char	Y	Side of the order from the perspective of the submitter. Supported values: 1 = Buy 2 = Sell
<OrderQtyData> component			Y	
OrderQty	38	Qty	Y	The currency traded amount.
*end **<OrderQtyData>** component*				
OrdType	40	char	Y	Indicates the type of order. Supported values: 1 = Market
Currency	15	Currency	Y	Denotes dealt currency.
TimeInForce	59	char	Y	Specifies how long the order remains in effect. Supported values: 1 = Good Till Cancel 6 = Good Till Date
ExpireDate	432	UTCTimest	N	Conditionally required if TimeInForce(59) = 6 (GTD).

<ExecutionReport> - Msgtype(35) = 8				
Field Name	**Tag**	**FIX Data Type**	**Rq'd**	**Description**
		amp		
ExecInst	18	MultipleCharValue	N	Instruction for order handling. Supported values: 1 = Not held – Default if not specified

OrderCapacity	528	char	N	Designates the capacity of the firm placing the order.
				Supported values:
				A = Agency
				P = Principal
				R = Riskless Principal

LastQty	32	Qty	Y	Quantity filled, denominated in the dealt currency specified in Currency(15).
LastPx	31	Price	Y	"All-in" rate for the transaction.
MidPx	631	Price	N	Mid price/rate.
LastSpotRate	194	Price	N	The spot rate in the "all-in" rate in LastPx(31).
LastForwardPoints	195	PriceOffset	N	The forward points added to LastSpotRate(194) to determine the "all-in" rate in LastPx(31).
LeavesQty	151	Qty	Y	Quantity open for further execution.
CumQty	14	Qty	Y	Currently executed quantity.
AvgPx	6	Price	Y	Calculated average rate of all fills on this order for the currency pair in Symbol(55).
TradeDate	75	LocalMktDate	Y	Indicates date of trading day. Absence of this field indicates current day.
TransactTime	60	UTCTimestamp	Y	Time the transaction represented by this ExecutionReport(35=8) occurred.
<**CommissionDataGrp**> component			N	*Standard 5.0 component added to FIX.4.4 message
NoCommissions	2639	NumInGroup	N	The number of commissions in the repeating group.
CommissionAmount	2640	Amt	N	Required if NoCommissions(2639) > 0. The commission amount.
CommissionAmountType	2641	Int	N	Required if NoCommissions(2639) > 0. Indicates what type of commission is being expressed in CommissionAmount(2640) Supported values: 2 = Broker 3 = Clearing broker
CommissionBasis	2642	Char	N	Required if NoCommissions(2639) > 0. Specifies the basis or unit used to calculate the commission. Supported values: 1 = Amount per unit 2 = Percent 3 = Absolute - Total monetary amount.
end <CommissionDataGrp> component				

<ExecutionReport> - Msgtype(35) = 8				
Field Name	**Tag**	**FIX Data Type**	**Rq'd**	**Description**
SettlCurrency	120	Currency	Y	Counter currency of the transaction.
<RateSource> component			N	*Standard 5.0 component added to FIX.4.4 message Used to identify the FX fixing source for FX non-deliverable forward.
NoRateSources	1445	NumInGroup	N	The number of rate source instances.
→ RateSource	1446	int	N	Required if NoRateSources(1445) > 0. Identifies the reference source for FX spot rate. Supported values: 0 = Bloomberg 1 = Reuters 2 = Telerate 99 = Other
→ RateSourceType	1447	int	N	Required if NoRateSources(1445) > 0. Supported values: 0 = Primary 1 = Secondary
→ FXBenchmarkRateFix	2796	String	N	Specifies the foreign exchange benchmark rate fixing to be used in valuing the transaction. For Ex – London 4 p.m. or Tokyo 3 p.m.
*end **<RateSource>** component*				
OffshoreIndicator	2795	int	N	Indicates the type of the currency rate being used. This is relevant for currencies that have offshore rate that is different from the onshore rate. Supported values: 0 = Regular – Default if not specified 1 = Offshore 2 = Onshore
HandlInst	21	char	N	Instructions for Order handling. Supported values: 3 = Manual order best execution
<**MiscFeesGrp**> component			N	
NoMiscFees	136	NumInGroup	N	Used to report fees other than commission. Indicates the number of fee entries.
MiscFeeAmt	137	Amt	N	Required if NoMiscFees(136) > 0. The amount of the type specified in MiscFeeType(139).
MiscFeeCurr	138	Currency	N	Currency of miscellaneous fee.
MiscFeeType	139	String	N	Required if NoMiscFees(136) > 0. Specifies the type of fee in MiscFeeAmt(137). Supported values: 1 = Regulatory fee 2 = Tax 3 = Local commission 4 = Exchange fees

<ExecutionReport> - Msgtype(35) = 8				
Field Name	**Tag**	**FIX Data Type**	**Rq'd**	**Description**
				5 = Stamp 6 = Levy 7 = Other 8 = Markup 9 = Consumption Tax 10 = Per Transaction 11 = Conversion 12 = Agent
end <MiscFeesGrp> component				
<StandardTrailer> component			Y	

7.2 *Allocation Messages*

7.2.1 AllocationInstruction(35=J)

The Allocation Instruction is used by the buy-side to instruct the dealer on how a trade is to be allocated to the specified account(s).

<AllocationInstruction> - MsgType(35)=J				
Name	**Tag**	**FIX Data Type**	**Rq’d**	**Description**
<StandardHeader> component			Y	MsgType = 35
AllocID	70	String	Y	Unique identifier for this allocation instruction message
AllocTransType	71	char	Y	Indicates the message's type of transaction. Supported values: 0 = New 1 = Replace 2 = Cancel
AllocType	626	int	Y	Specifies the purpose or type of allocation message. 2 = Preliminary (without MiscFees and NetMoney)
AllocNoOrdersType	857	int	Y	Indicates how the orders being booked and allocated by this message are identified, i.e. by explicit definition in the OrdAllocGrp repeating group component or not. Supported values: 1 = Explicit list provided
<**OrdAllocGrp**> component			Y	
NoOrders	73	NumInGroup	Y	Required when AllocNoOrdersType(857) = 1 (Explict list provided).
→ ClOrdID	11	String	Y	Order identifier assigned by the submitter.
→ OrderID	37	String	Y	Order identifier assigned by the sell-side.
*end **<OrdAllocGrp>** component*				
<**ExecAllocGrp**> component			Y	

<AllocationInstruction> - MsgType(35)=J				
Name	**Tag**	**FIX Data Type**	**Rq'd**	**Description**
NoExecs	124	NumInGroup	Y	The number of executions included in the allocation.
→ LastQty	32	Qty	Y	The quantity for the execution.
→ ExecID	17	String	Y	Unique identifier of execution message as assigned by sell-side
→ LastPx	31	Price	Y	The price of the execution.
*end **<ExecAllocGrp>** component*				
Side	54	char	Y	Side of the trade. Supported values: 1 = Buy 2 = Sell
<**Instrument**> component			Y	
Symbol	55	String	Y	"CCY1/CCY2", where CCY1 and CCY2 are ISO currency codes.
SecurityType	167	String	Y	Indicates type of security Supported values: FXSPOT FXFWD FXNDF
*end **<Instrument>** component*				
Quantity	53	Qty	Y	Total quantity.
AvgPx	6	Price	Y	Weighted average price of all the executions.
<**Parties**> component			N	
NoPartyIDs	453	NumInGroup	N	The number of parties.
→ PartyID	448	String	N	The ID or code used to identify the party.
→ PartyIDSource	447	char	N	The class or source scheme of PartyID(448) value. Supported values: C = Generally accepted market identifier D = Proprietary / Custom code
→ PartyRole	452	int	N	The Role of the party. Supported values: 1 = Executing Firm 13 = Order origination firm
*end **<Parties>** component*				
TradeDate	75	LocalMktDate	Y	Indicates date of trade.
TransactTime	60	UTCTimestamp	N	Date/time when allocation is generated.
SettlType	63	String	N	Indicates the period for settlement of the trade. Supported values: 0 = Regular - Default if not specified. B = Broken date – Use for FX to specify non-standard

<AllocationInstruction> - MsgType(35)=J				
Name	**Tag**	**FIX Data Type**	**Rq'd**	**Description**

				tenor. Specify the date in SettlDate(64).
				Dx = Tenor expression for "days", e.g. "D5", where "x" is any integer > 0
				Mx = Tenor expression for "months", e.g. "M3", where "x" is any integer > 0
				Wx = Tenor expression for "weeks", e.g. "W13", where "x" is any integer > 0
				Yx = Tenor expression for "years", e.g. "Y1", where "x" is any integer > 0

SettlDate	64	LocalMktDate	Y	Specific date of trade settlement in YYYYMMDD format.
<**AllocGrp**> component			N	
NoAllocs	78	NumInGroup	Y	Indicates number of allocation groups to follow.
→ AllocAccount	79	String	Y	Sub-account mnemonic.
→ AllocAcctIDSource	661	int	Y	Used to identify the source of the AllocAccount(79) value. Supported values: 4 = OMGEO (AlertID)
→ AllocPrice	366	Price	N	Booking price for this account Note: if included for one account must be included for all accounts. Weighted average equals AvgPx(6).
→ AllocQty	80	Qty	Y	Amount to be allocated to the account.
→ IndividualAllocID	467	String	Y	Unique identifier for a specific NoAllocs(78) repeating group instance (e.g. for an AllocAccount).
→ ProcessCode	81	Char	Y	Processing code for sub-account. Supported values: 0 = Regular
→ **<NestedParties>** component			Y	
→ NoNestedPartyIDs	539	NumInGroup	Y	Indicates the number of party entries.
→→ NestedPartyID	524	String	Y	The ID or code used to identify the party.
→→ NestedPartyIDSource	525	char	Y	Identifies the class or source of the NestedPartyID(524) value. Supported values: B = BIC - Bank Identification Code C = Generally accepted market identifier N = Legal Entity Identifier - ISO 17442 LEI – Added Standard FIX.5.0 value
→→ NestedPartyRole	538	int	Y	The role of the party. Supported values: 4 = Clearing firm
→ *end **<NestedParties>** component*				
→ AllocAvgPx	153	Price	N	Average price for this allocation instance.

<AllocationInstruction> - MsgType(35)=J				
Name	**Tag**	**FIX Data Type**	**Rq'd**	**Description**
→ AllocSettlCurrency	736	Currency	Y	AllocSettlCurrency for this AllocAccount if different from "overall" Currency. Defaults to SettlementCurrency(120).
*end **<AllocGrp>** component*				
<RateSource> component			N	*Standard 5.0 component added to FIX.4.4 message Used to identify the FX fixing source for FX non-deliverable forward.
NoRateSources	1445	NumInGroup	N	The number of rate source instances.
→ RateSource	1446	int	N	Required if NoRateSources(1445) > 0. Identifies the reference source for FX spot rate. Supported values: 0 = Bloomberg 1 = Reuters 2 = Telerate 99 = Others
→ RateSourceType	1447	int	N	Required if NoRateSources(1445) > 0. Supported values: 0 = Primary 1 = Secondary
→ FXBenchmarkRateFix	2796	String	N	Specifies the foreign exchange benchmark rate fixing to be used in valuing the transaction. For Ex – London 4 p.m. or Tokyo 3 p.m.
*end **<RateSource>** component*				
OffshoreIndicator	2795	int	N	Indicates the type of the currency rate being used. This is relevant for currencies that have offshore rate that is different from the onshore rate. Supported values: 0 = Regular – Default if not specified 1 = Offshore 2 = Onshore
<**StandardTrailer**> component			Y	

7.2.2 AllocationInstructionAck(35=P) Message

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

<AllocationInstructionAck> - MsgType(35)=P				
Name	**Tag**	**FIX Data Type**	**Rq'd**	**Description**
<StandardHeader> component			Y	MsgType = P
AllocID	70	String	Y	Unique identifier for this allocation instruction message.

TradeDate	75	LocalMktDate	N	Indicates date of trade.
TransactTime	60	UTCTimestamp	Y	Date/Time AllocationInstructionAck(35=P) was generated by the application.
AllocStatus	87	int	Y	Denotes the status of the allocation instruction. Supported values: 0 = Accepted 1 = Block level reject 3 = Received (but not yet processed)
AllocRejCode	88	Int	N	Identifies reason for rejection. Supported values: 0 = unknown account(s) 1 = incorrect quantity 2 = incorrect average price 3 = unknown executing broker mnemonic 4 = commission difference 5 = unknown OrderID (37) 7 = other 8 = incorrect allocated quantity 9 = calculated difference 10 = unknown or stale ExecID(17) 11 = mismatched data value 12 = unknown ClOrdID(11) *Below are standard 5.0 FIX values added to FIX.4.4 version 14 = Duplicate or missing IndividualAllocId(467) 15 = Trade not recognized 16 = Trade previously allocated [DuplicateTrade] 17 = Incorrect or missing instrument 18 = Incorrect or missing settlement date 19 = Incorrect or missing fund ID or fund name 20 = Incorrect or missing settlement instructions 21 = Incorrect or missing fees 22 = Incorrect or missing tax 23 = Unknown or missing party 24 = Incorrect or missing side 25 = Incorrect or missing net-money 26 = Incorrect or missing trade date 27 = Incorrect or missing settlement currency instructions
RejectText	1328	String	N	*Standard FIX 5.0 tag added to the message. Identifies the reason for rejection.
<**StandardTrailer**> component			Y	

7.3 *Trade Aggregation Messages*

7.3.1 TradeAggregationRequest(35=DW)

The TradeAggregationRequest(35=DW) is used by the buy-side to request the dealer for aggregation of individual trades.

<TradeAggregationRequest> - MsgType(35)=DW				
Name	**Tag**	**FIX Data Type**	**Rq'd**	**Description**
<StandardHeader> component			Y	MsgType = DW
TradeAggregationRequestID	2786	String	Y	Unique identifier for this message.
TradeAggregationRefRequestID	2787	char	N	Required when TradeAggregationTransType(2788)=1 (Cancel) or 2 (Replace)
TradeAggregationTransType	2788	int	Y	Identifies the trade aggregation transaction type. Supported values: 0 = New 1 = Cancel 2 = Replace
AggregatedQty	2789	int	N	Total quantity of orders or fills quantity aggregated.
Currency	15	Currency	N	Identifies dealt currency for aggregation.
AvgPx	6	Price	N	The all-in average price, if specified, for the aggregation.
Side	54	char	Y	Side of the trade. Supported values: 1 = Buy 2 = Sell
PricePrecision	2349	int	N	Specifies the price decimal precision of the instrument.
<**OrderAggregationGrp**> component			N	Maybe used to specify the IDs of the orders being aggregated together.
NoOrders	73	NumInGroup	N	
→ ClOrdID	11	String	N	Order identifier assigned by the submitter.
→ OrderID	37	String	N	Order identifier assigned by the sell-side.
→ OrderQty	38	Qty	N	Order quantity.
→ OrderAvgPx	799	Price	N	Average price of the order.
*end **<OrderAggregationGrp>** component*				
<**ExecutionAggregationGrp**> component			N	Maybe used to specify the IDs of the execution fills being aggregated together.
NoExecs	124	NumInGroup	N	The number of executions or trade.
→ LastQty	32	Qty	N	Required if NoExecs(124) > 0
→ ExecID	17	String	N	Either ExecID(17) or TradeID(1003) must be specified.
→ TradeID	1003	String	N	Either ExecID(17) or TradeID(1003) must be specified.
→ LastPx	31	Price	N	The price of the execution.
*end **<ExecutionAggregationGrp>** component*				
Account	1	String	N	Aggregation or holding account identifier.
<**Instrument**> component			Y	
Symbol	55	String	Y	"CCY1/CCY2", where CCY1 and CCY2 are ISO currency codes.
SecurityType	167	String	Y	Indicates type of security

<TradeAggregationRequest> - MsgType(35)=DW				
Name	**Tag**	**FIX Data Type**	**Rq'd**	**Description**
				Supported values: FXSPOT FXFWD FXNDF
*end **<Instrument>** component*				
<**Parties**> component			N	
NoPartyIDs	453	NumInGroup	N	The number of parties.
→ PartyID	448	String	N	The ID or code used to identify the party.
→ PartyIDSource	447	char	N	The class or source scheme of PartyID(448) value. Supported values: C = Generally accepted market identifier D = Proprietary / Custom code
→ PartyRole	452	int	N	The Role of the party. Supported values: 1 = Executing Firm 13 = Order origination firm
*end **<Parties>** component*				
<**StandardTrailer**> component			Y	

7.3.2 TradeAggregationReport(35=DX)

The TradeAggregationReport(35=DX) is used by the dealer to accept or reject the buy-side's request for trade aggregation.

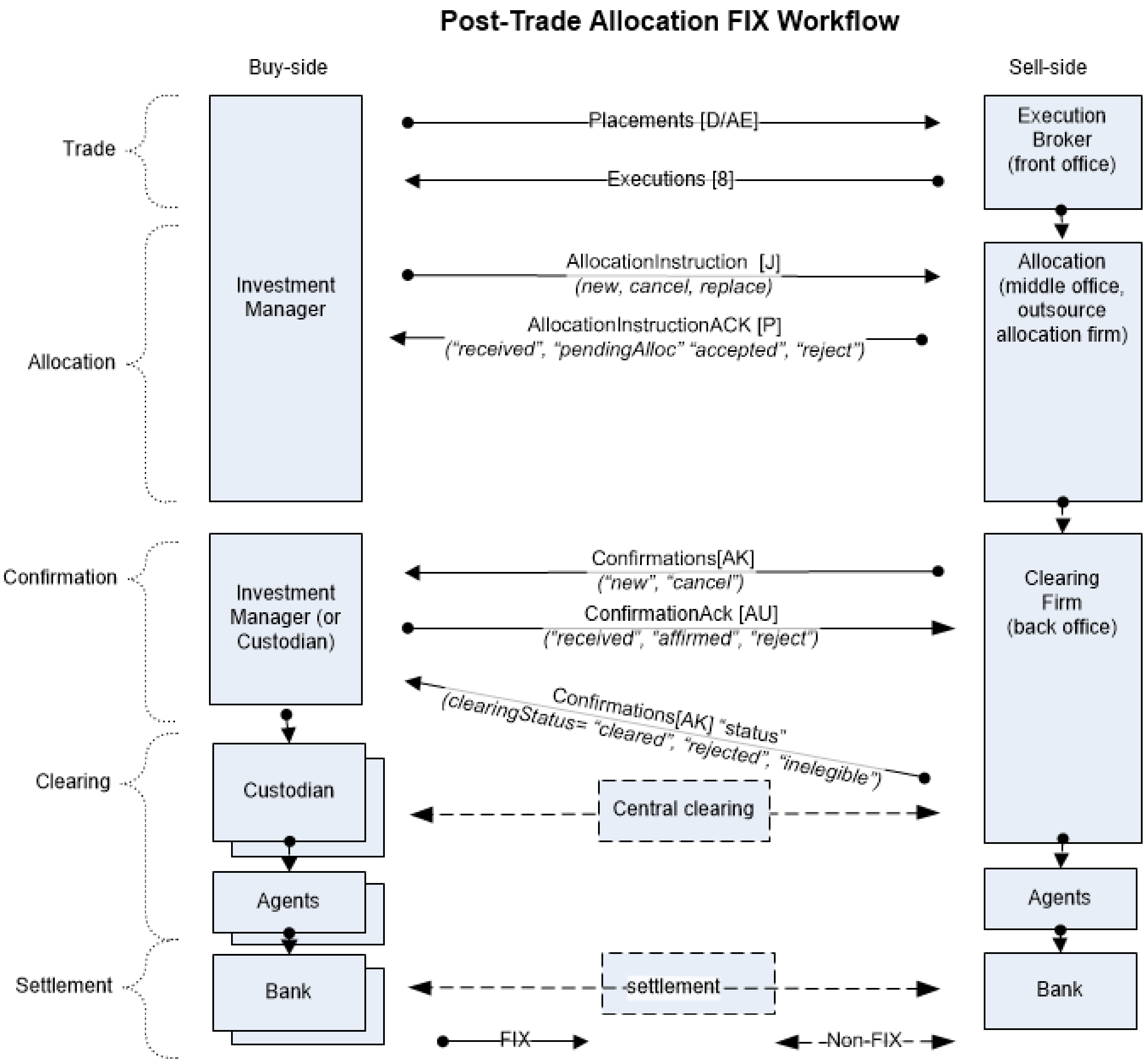
<TradeAggregationReport> - MsgType(35)=DX				
Name	**Tag**	**FIX Data Type**	**Rq'd**	**Description**
<StandardHeader> component			Y	MsgType = DW
TradeAggregationReportID	2792	String	Y	Unique identifier for the report message.
TradeAggregationRequestID	2786	char	N	Unique identifier for the TradeAggregationRequest(35=DW) message being responded to.
TradeAggregationRequestStatus	2790	int	Y	Status of the trade aggregation request. Supported values: 0 = Accepted 1 = Rejected
TradeID	1003	String	N	Conditionally required when TradeAggregationRequestStatus(2790)=0 (Accepted). The trade identifier for the group of aggregated trades.
TradeAggregationRejectReason	2791	int	N	Reason for trade aggregation request being rejected. Supported values: 0 = Unknown order(s)

<TradeAggregationReport> - MsgType(35)=DX				
Name	**Tag**	**FIX Data Type**	**Rq'd**	**Description**
				1 = Unknown execution/fill(s) 99 = Other
AggregatedQty	2789	int	N	Total quantity of orders or fills quantity aggregated.
AvgPx	6	Price	N	Average rate for the aggregated trades. Conditionally required when TradeAggregationRequestStatus(2790)=0 (Accepted).
AvgSpotRate	2793	Price	N	The average FX spot rate.
AvgForwardPoints	2794	PriceOffset	N	The average forward points.
SettlDate	64	LocalMktDate	N	Trade settlement date.
<**Instrument**> component			N	
Symbol	55	String	N	"CCY1/CCY2", where CCY1 and CCY2 are ISO currency codes.
SecurityType	167	String	N	Indicates type of security Supported values: FXSPOT FXFWD FXNDF
end ***<Instrument>*** *component*				
Side	54	char	N	Conditionally required when TradeAggregationRequestStatus(2790)=0 (Accepted).
RejectText	1328	String	N	Optionally used to provide a text narrative for rejecting the request.
EncodedRejectTextLen	1664	Length	N	Must be set if EncodedRejectText(1665) field is specified and must immediately precede it.
EncodedRejectText	1665	data	N	Encoded (non-ASCII characters) representation of the RejectText(1328) field in the encoded format specified via the MessageEncoding(347) field.
<**StandardTrailer**> component			Y	

8 Appendices

8.1 *FX Workflow Flow Diagram*

Figure 1 Post-Trade Allocation FIX Workflow


Post-Trade Allocation FIX Workflow
Buy-side
Sell-side
Trade
Allocation
Confirmation
Clearing
Settlement
Investment Manager
Execution Broker (front office)
Placements [D/AE]
Executions [8]
AllocationInstruction [J]
(new, cancel, replace)
AllocationInstructionACK [P]
("received", "pendingAlloc" "accepted", "reject")
Allocation (middle office, outsource allocation firm)
Investment Manager (or Custodian)
Confirmations[AK]
("new", "cancel")
ConfirmationAck [AU]
("received", "affirmed", "reject")
Confirmations[AK] "status"
(clearingStatus= "cleared", "rejected", "inelegible")
Clearing Firm (back office)
Custodian
Central clearing
Agents
Agents
Bank
settlement
Bank
FIX
Non-FIX

8.2 *Glossary*

Term	Description
Trade	Buy/sell of a given quantity (block) of a given currency pair.
Transaction	• One currency pair • One account • One price
Allocation	Process of assigning the desired quantities and fair prices, from a trade (block) to a series of accounts. Result is a series of transactions.
Average-Price Allocation	• Using the average price of multiple execution reports so that each account gets the same price. • Executions may only be average priced within a broker, and the price must be known to the broker, because each broker generates their own transactions.
Central Clearing	Currently no central clearing party (though that is changing) – done by the parties.
Settlement	Transfer of funds among parties
Netting	• **General definition** o Combining buy/sell placement within a currency pair or across currency pairs to a net placement. • Pre-trade netting o Aggregating buys and sells before market transaction for purposes of reducing the amount required to be traded o Logical trade prices are computed based on market prices and netted trade price. o Still have to provide allocation-preview for the"logical" trade across all the accounts o Bank still wants to see the logical trades and allocations along with the netted trade. ▪ Settlement cost of all the transactions is out of the spread on the netted trade. ▪ Net to zero issue: bank does us a favor. o Bank must know final transaction characteristics so that buy-side transactions match the sell-side transactions downstream. • Pre-trade netting on an EMS o OMS sends in multiple placements that get netted on the EMS before placement with broker. o EMS nets the placement and gets places netted placement with broker. o Executions: EMS sends back "fake" executions that match the OMS placements but with the average price from the "real" executions. ▪ EMS works out prices for "fake" executions based upon market o Allocation: transaction characteristics must be coordinated between buy-side and sell-side through one of the following:

Term	Description
	▪ Post-trade AllocationInstruction, including the “fake” execution report list. EMS communicates the allocations in terms of the “real” execution reports. ▪ Pre-trade allocation would require some sort of convention (e.g. transactions would be created that match the pre-trade allocations with the average price from the “fake” execution reports.) • Post-trade netting o Aggregating confirmed buy and sell transactions between counter-parties (bi-lateral or multilateral with CCP) for purposes of reducing amount of currency to be transferred and reducing risk.

8.3 *FIX 5.0 and Later Post-trade Fields and Valid Values*

The following table contains the tags and valid values from FIX 5.0 or later that are used in the FIX Post-trade Recommended Practices.

Tag	Field	Valid Values	FIX Version	EP
88	AllocRejectCode	14 = Duplicate or missing IndividualAllocId(467) 15 = Trade not recognized 16 = Trade previously allocated [DuplicateTrade] 17 = Incorrect or missing instrument 18 = Incorrect or missing settlement date 19 = Incorrect or missing fund ID or fund name 20 = Incorrect or missing settlement instructions 21 = Incorrect or missing fees 22 = Incorrect or missing tax 23 = Unknown or missing party 24 = Incorrect or missing side 25 = Incorrect or missing net-money 26 = Incorrect or missing trade date 27 = Incorrect or missing settlement currency instructions	5.0 SP2	170
447	PartyIDSource	N = Legal Entity Identifier (ISO 17442) LEI	5.0 SP2	140
452	PartyRole	56 = Acceptable counterparty	4.4	8
538	NestedPartyIDSource	N = Legal Entity Identifier (ISO 17442) LEI	5.0 SP2	140
1079	MaturityTime	N/A	4.4	21
1445	NoRateSources	N/A	5.0 SP1	82
1446	RateSource	0 = Bloomberg 1 = Reuters 2 = Telerate tbd = EMTA 99 = Others	5.0 SP1	82
1447	RateSourceType	0 = Primary 1 = Secondary	5.0 SP1	82
1328	RejectText	N/A	5.0 SP2	EP103
2639	NoCommissions	N/A	5.0 SP2	204
2640	CommissionAmount	N/A	5.0 SP2	204
2641	CommissionAmountType	2 = Broker 3 = Clearing broker	5.0 SP2	204
2642	CommissionBasis	1 = per unit 2 = percent 3 = absolute	5.0 SP2	204
2786	TradeAggregationRequestID		5.0 SP2	EP 247
2787	TradeAggregationRequestRefID		5.0 SP2	EP 247
2788	TradeAggregationTransType	0 = New 1 = Cancel 2 = Replace	5.0 SP2	EP 247
2789	AggregatedQty		5.0 SP2	EP 247
2790	TradeAggregationRequestStatus	0 = Accepted 1 = Rejected	5.0 SP2	EP 247

Tag	Field	Valid Values	FIX Version	EP
2791	TradeAggregationRejectReason	0 = Unknown order(s) 1 = Unknown execution/fill(s) 99 = Other	5.0 SP2	EP 247
2792	TradeAggregationReportID		5.0 SP2	EP 247
2793	AvgSpotRate		5.0 SP2	EP 247
2794	AvgForwardPoints		5.0 SP2	EP 247
2795	OffshoreIndicator	0 = Regular – Default if not specified 1 = Offshore 2 = Onshore	5.0 SP2	EP 247
2796	FXBenchmarkRateFix		5.0 SP2	EP 247