

## Appendix E

### Order Handling and Instruction Semantics

#### London SETS Order Types Matrix

The table below presents the representation of the London Stock Exchange Trading System (SETS) order types in the FIX protocol:

LSE Order Type	OrdType	TimeInForce	ExpireTime	Price	Comment
At Best	1	3	n/a	No	
Fill or Kill - no limit price	1	4	n/a	No	
Fill or Kill - limit price	2	4	n/a	Yes	
Limit - day	2	n/a, 0	n/a	Yes	
Limit - good until	2	6	Good Till Date	Yes	
Execute and Eliminate	2	3	n/a	Yes	
Market Orders - day	1	n/a, 0	N/a	No	SETS Release 3.1 Only
Market Orders -good until	1	6	Good Till Date	No	SETS Release 3.1 Only

#### Asia/Pacific Regional Order Handling

The following table identifies how to represent via FIX the commonly used and understood order handling instructions within the Asia/Pacific region.

Asia/Pacific Dealer Instruction	OrdType	ExecInst
Careful Discretion	1 (Market)	4 (Over the Day)
Market	1 (Market)	5 (Held)
Trader Discretion	1 (Market)	1 (Not Held)

#### Handling Instructions (HandlInst) field

The following identifies the meaning and expected usage of the HandlInst (Handling Instructions) field. This field has been required on the New Order messages since the inception of FIX. Usage of this field may vary by market and by broker. Buy side and sell side firms should confirm their mutual understanding of the usage and implementation of HandlInst.

1 = Automated execution order, private, no Broker intervention

Order is systematically routed to the market place, usually to an exchange or ECN or market maker, for execution. It is expected that no broker

intervention is required to accept or forward the order into the market.

Notes:

- Private does not mean broker cannot see buy side order flow. In many markets, the Broker has the legal requirement to monitor customer order flow and be responsible for those orders.
- Buy side firm may be expected to supply the symbology required by the market.
- Broker may require certain optional fields, such as ExDestination and/or Currency.
- Implies an immediate reject will be sent if order cannot be forwarded immediately into the market.

2 = Automated execution order, public, Broker intervention OK

Broker may stop order from flowing immediately into the market place. This would typically be done, if the broker can cross this order against another order to provide price improvement and / or liquidity.

If Broker does not choose to stop this order, it will automatically flow into the market for execution.

3 = Manual order, best execution

Order is routed to appropriate sell side broker who then accepts responsibility for the order. This should operate as though the buy side firm called the order into their broker.

Notes:

- Different than “not held”.
- Does not imply “call first” (an ExecInst value).

## Pegged Orders

The following are all pegging ExecInst values used when OrdType=P to specify the type of pegged order represented. Note that these fields cannot be combined; only one may be specified on a pegged order.

L = Last peg (last sale)

M = Mid-price peg (midprice of inside quote)

O = Opening peg

P = Market peg

R = Primary peg (primary market - buy at bid/sell at offer)

T = Fixed Peg to Local best bid or offer at time of order

W = Peg to VWAP

A pegged order acts like a limit order, except that the limit price fluctuates relative to another quantity, such as the last sale, midpoint, opening price, bid, offer, or VWAP (Volume Weighted Average Price). A primary peg order is priced relative to the bid if buying, the offer if selling. A market peg order is priced relative to the offer if buying, the bid if selling.

In the absence of the PegDifference field, or when PegDifference = 0, the price of the pegged order follows the referenced quantity exactly. If PegDifference is specified, the referenced quantity + the PegDifference = the price of the order.

Some systems allow pegged orders to be specified with a Price field. In this case, the Price field serves to put a limit on how far the pegged value can move. For instance, if the bid for a stock is 50, the offer is 50.10, the order is a primary peg to sell, PegDifference = -0.02, and Price = 45, the order will be priced to sell at the offer + (-0.02) or 50.08. If the offer falls, the order's price will fall such that it is always 0.02 less than the offer. However, once the order's price hits 45 (the limit specified in the Price field) it can fall no further.

A pegged order with ExecInst = T (Fixed Peg to Local best bid or offer at time of order) behaves differently in two ways. First, it acts like a Primary peg, but it is relative to the local best bid or offer, not the global best bid or offer. (For instance, in an ECN environment, this would be the best bid or offer on the particular ECN, not the best price available in the national marketplace.) Second, once an initial price for the order is set, unlike normal pegged orders, the price will not move even if the best bid or offer of the local marketplace moves.

#### “Reserve Quantity” Orders

MaxFloor: Traditionally used to indicate reserve quantity. To indicate a single level of reserve quantity, MaxFloor should be used.

MaxShow: Used when two levels of reserve quantities are needed, e.g. one level displayed to the world (MaxFloor) and another displayed to subscribers of their ECN (MaxShow.) I.e. MaxFloor <= MaxShow <= OrderQty.

- One may place an order for 100,000 shares (OrderQty), only want 1000 shares shown to NASDAQ at any one time (MaxFloor), but will allow other subscribers of that ECN to see 5000 shares (MaxShow).

## Appendix F

### Settlement Instructions Field Usage Matrix

Trade Settlement Type	F.I.X. Fields Required	F.I.X. Fields Optional
Standing Instructions Provided (i.e. to be stored in an internal or third-party standing instructions database)	SettlInstID SettlInstTransType SettlInstRefID (if SettlInstTransType=Cancel or Replace) SettlInstMode=1 SettlInstSource AllocAccount (some combination of) <ul style="list-style-type: none"> <li>• LastMkt</li> <li>• Side</li> <li>• SettlLocation</li> <li>• SecurityType</li> </ul>	ClientID ExecBroker Text StandInstDbName StandInstDbID SettlDepositoryCode SecuritySettlAgentName SecuritySettlAgentCode SecuritySettlAgentAcctNum SecuritySettlAgentContactName SecuritySettlAgentContactPhone (CashSettl* only if SecuritySettl*)

	<ul style="list-style-type: none"> <li>SettlDeliveryType</li> <li>EffectiveTime</li> </ul> TransactTime StandInstDbType (if SettlDepositoryCode is not specified, one of more of the SecuritySettl* fields are required) SettlBrkrCode SettlInstCode	fields provided) CashSettlAgentName CashSettlAgentCode CashSettlAgentAcctNum CashSettlAgentContactName CashSettlAgentContactPhone
Specific Allocation Account (trade) referencing existing Standing Instructions	SettlInstID SettlInstTransType SettlInstRefID (if SettlInstTransType=Cancel or Replace) SettlInstMode=2 SettlInstSource AllocAccount TradeDate AllocID LastMkt Side TransactTime StandInstDbType StandInstDbID SettlBrkrCode SettlInstCode	SettlLocation SecurityType ClientID ExecBroker Text StandInstDbName
Specific Allocation Account (trade) providing details for settlement at a depository	SettlInstID SettlInstTransType SettlInstRefID (if SettlInstTransType=Cancel or Replace) SettlInstMode=2 SettlInstSource AllocAccount SettlLocation TradeDate AllocID LastMkt Side TransactTime	SecurityType ClientID ExecBroker Text SettlDeliveryType

	SettlDepositoryCode SettlBrkrCode SettlInstCode	
Specific Allocation Account (trade) providing details for a Single Agent (bank) for the security	SettlInstID SettlInstTransType SettlInstRefID (if SettlInstTransType=Cancel or Replace) SettlInstMode=2 SettlInstSource AllocAccount SettlLocation TradeDate AllocID LastMkt Side TransactTime SettlBrkrCode SettlInstCode SecuritySettlAgentName SecuritySettlAgentCode SecuritySettlAgentAcctNum	SecurityType ClientID ExecBroker Text SettlDeliveryType SecuritySettlAgentContactName SecuritySettlAgentContactPhone
Specific Allocation Account (trade) providing details for a Two Agents (banks) one for the security and one for cash	SettlInstID SettlInstTransType SettlInstRefID (if SettlInstTransType=Cancel or Replace) SettlInstMode=2 SettlInstSource AllocAccount SettlLocation TradeDate AllocID LastMkt Side TransactTime SettlDeliveryType=Free SettlBrkrCode SettlInstCode	SecurityType ClientID ExecBroker Text SecuritySettlAgentName SecuritySettlAgentContactName SecuritySettlAgentContactPhone CashSettlAgentName CashSettlAgentContactName CashSettlAgentContactPhone

	SecuritySettlAgentCode SecuritySettlAgentAcctNum CashSettlAgentCode CashSettlAgentAcctNum	

## Appendix G

### Rule80A (aka OrderCapacity) Usage by Market

Note that the name of the Rule80A field is changing to “OrderCapacity” as Rule80A is a very US market-specific term. Other world markets need to convey similar information, however, often a subset of the US values. This appendix documents the market-specific usage of this field.

#### United States Listed Equity Markets:

Rule80A's values and usage details are documented in SEC Rule11Ac1-1/4. Note the purpose behind the rule is to restrict prices from rising or falling too fast providing more stability in the market. See Investments by Sharpe, 6<sup>th</sup> edition p. 50. Indicates the order type upon which exchange Rule 80A is applied.

The following values are valid and applicable when using FIX to communicate with the New York Stock Exchange (NYSE) or other US listed equity exchanges per the SuperDOT Notification document. The values and usage details when used for US trading are documented in SEC Rule11Ac1-1/4.

Valid values:

- A = Agency single order
- B = Short exempt transaction (refer to A type)
- C = Program Order, non-index arb, for Member firm/org
- D = Program Order, index arb, for Member firm/org
- E = Registered Equity Market Maker trades
- F = Short exempt transaction (refer to W type)
- H = Short exempt transaction (refer to I type)
- I = Individual Investor, single order
- J = Program Order, index arb, for individual customer
- K = Program Order, non-index arb, for individual customer
- L = Short exempt transaction for member competing market-maker affiliated with the firm clearing the trade (refer to P and O types)
- M = Program Order, index arb, for other member
- N = Program Order, non-index arb, for other member
- O = Competing dealer trades
- P = Principal
- R = Competing dealer trades
- S = Specialist trades
- T = Competing dealer trades
- U = Program Order, index arb, for other agency
- W = All other orders as agent for other member
- X = Short exempt transaction for member competing market-maker not affiliated with the firm clearing the trade (refer to W and T types)
- Y = Program Order, non-index arb, for other agency
- Z = Short exempt transaction for non-member competing market-maker (refer to A and R types)

#### Japanese Equity Markets:

Used to specify whether order is Agency or Principal.

Valid values:

- A = Agency single order
- P = Principal

**Other Markets:**

All or a subset of the Rule80A (aka OrderCapacity) field values defined in the field reference may be applicable for other markets. Future markets will be included in this section as they are defined and brought forward to the FIX Technical Committee.



## Appendix H

### Mass Quote Message Scenarios

#### Unsolicited quote(s) no response requested

Mass Quote is sent from first party to second party. The quote has the QuoteResponseLevel set to 0 or omitted. The second party does not acknowledge the quote. If the quote is later hit, resulting in a trade, an Execution Report is sent to the first party.

First Party		Second Party
<u>Mass Quote message</u> Options: One or more sets of quotes Set QuoteResponseLevel is set to 0 or omitted	→	Interprets quotes applies them to a market Interprets Response Level – provides response accordingly No response is sent
	←	<u>Execution Report</u> Quote Results in Trade

#### Unsolicited quote(s) negative response only requested

Mass Quote is sent from first party to second party. The quote has the QuoteResponseLevel set to 1. The second party only acknowledges the quote if there is an error. If an error is encountered by the second party while processing the quote a Quote Acknowledgement message is sent with the QuoteRejectReason set to the error encountered.

First Party		Second Party
<u>Mass Quote message</u> Options: One or more sets of quotes Set Response Level to 1	→	Interprets quotes applies them to a market
Interprets Quote Acknowledgement If error – then send revised quote	←	<u>Quote Acknowledgement</u> If an error is encountered
<u>Mass Quote message</u>	→	Interprets quotes applies them to a market

**Unsolicited quote(s) full response requested**

Mass Quote is sent from first party to second party. The quote has the QuoteResponseLevel set to 2. The second party acknowledges each quote.

First Party		Second Party
<u>Mass Quote message</u> Options: One or more sets of quotes Set Response Level to 2	→	Interprets quotes applies them to a market
Interpret Quote Acknowledgement	←	<u>Quote Acknowledgement</u>

**Cancel All Quotes**

The First Party asks the second party to cancel all quotes. A Quote Acknowledgement is sent back to the first party by the second party after quotes are canceled.

First Party		Second Party
<u>Quote Cancel message</u>  <u>QuoteCancelType = 4 (Cancel all quotes)</u>	→	Interprets Quote Cancel message and cancels quotes.
Interpret Quote Acknowledgement	←	<u>Quote Acknowledgement</u>

## Appendix I

### Security Definition, Security Status, and Trading Session Message Scenarios

#### Overview

A set of messages has been defined for the definition and dissemination of securities information traded between two parties. These messages allow for the ability to define complex, multi-leg financial securities, such as options strategies, futures spreads, underlying-derivative combinations, indexes, and baskets. *Security Definition Request* message is used to define a security to the counterparty for trading and to retrieve definitions for securities available for trading with the counterparty.

The *Security Definition* message can also be used to query a list of securities offered by a trading party. This message is useful for obtaining lists of products that are traded on a market. Although intended to support exchange style trading – this capability should also be of use in trading between any two trading partners.

Two additional messages have been added for status purposes: The *Security Status* message and the *Trading Session Status* message. The *Security Status* message is based upon the Trade Related message proposal from SIAC.

The *Security Status* message, based upon the SIAC Trade Related message proposal, has been added to provide solicited or unsolicited status information on securities. An exchange can use this message to transmit change in trading state of a product. The *Security Status Request* message can be used to query the state of a product.

The *Trading Session Status* message has been added to provide status on a market. An exchange can use this to indicate status on the overall market and to provide a list of securities traded during that trading session. Two trading parties can also use this message to communicate information on two-party trading. The *Trading Session Status Request* message is used to query the state of a product.

Both the *Security Status* message and *Trading Session Status* message include a *SubscriptionRequestType* field, which is used to tell the counterparty application if the requesting application wants to receive a snapshot of status or wants to subscribe for unsolicited messages as the status of the security (or trading session) changes.

#### Background

The motivation behind these messages was to identify a method to be able to trade derivative strategies (butterfly spread, vertical spread, calendar spread, covered write, etc.) and to provide a mechanism to define FLEX Options using the FIX protocol. Most exchange trading systems have some type of product definition service. Although the motivation for the new messages was to support the communication between trading party and exchange, it was important to make any message flexible enough to support a variety of applications, including the ability to retrieve information about securities available for trading with a counterparty. The ability to query for a list of securities is very important in an exchange environment – where the retrieval of “standing data” from the exchange is needed by many trading systems.

#### Definitions

- Strategy - A group of related securities that are traded atomically at a net price.

Examples:

- Vertical Spread
- Butterfly Spread

- Calendar Spread
- Covered Write
- Strategy Leg - One Security within a strategy
- Spread - combination of derivative securities whose maturity date or strike price is spread, creating a synthetic Security.
- Synthetic - A financial security that is the result of holding positions in multiple securities.
- Combination - alias for spread or strategy.

### Approach

A *Security Definition Request* message can be used to make the following requests:

- Define and/or Request a specific Security to be traded with the second party.
- Request a list of the Security Types that can be traded with the second party (these Securities can include derivative trading strategies).
- Request a list of Securities that can be traded with the second party.
- Request a list of Securities for a specified Security Type that can be traded with the second party.
- Request a list of Securities available for trading during a specific trading session.

The *Security Definition* message is used to:

- Indicate acceptance of a Security defined in a previous *Security Definition Request* message.
- Indicate acceptance of a Security defined in a previous *Security Definition Request* message with changes to the definition and/or symbol or security ID.
- Reject the request for security.
- Return a list of security types.
- Return a list of securities.
- Return a list of securities for a security type.

### Extensions to other messages

One additional field, *MultiLegReportingType*, is to be used on the Execution Report to indicate if the Execution Report is for the multileg security itself or an individual leg of the multileg security. Absence of this field in the Execution Report implies that the report pertains to the entire security – not an individual leg.

The agreement on how parties report multileg security execution is left to individual trading parties and is to be configured out of band. The FIX protocol will not provide a mechanism to specify how multileg execution reporting should be done.

For an example:

A straddle is an option strategy that consists of simultaneously buying a call option and a put option at the same strike price and maturity date. The straddle is defined for trading using *the Security Definition Request Message*. Once the straddle is defined, via receipt of the *Security Definition Message* from the counterparty (in this case an options exchange), a *New Order – Single* is used to submit the order to trade this newly defined multileg security. If the parties agree to report multileg execution by individual legs–

then an execution report will be generated for each leg of the option strategy. If the parties agree to report multileg execution by multileg security only, then only one *Execution Report* will be issued for the fill.

Reporting by leg is required for equity options as clearing houses will only understand the individual option series legs. Reporting by legs permits the trading parties to accurately maintain positions.

## Rules

- The Security identification negotiated during the session is, by default, assumed valid only during the session. This eliminates the requirement for, but does not prevent the use, of a service to define and keep Securities persistent.
- Once a Security is defined, it will be traded as a regular Security
- Once a Security is defined, it will be traded at a single net price
- Once a Security is defined, it can be traded by FIX 4.1 compatible systems (This provides for backward compatibility and the ability to maintain Security information outside of FIX so that FIX 4.1 engines can participate).

## Specifying Derivative Trading Strategies using the Security Definition message

The Security Definition message can be used to specify multiple legs of a derivative trading strategy. The first set of security related tags are used to name and identify the proposed strategy. This is followed by the NoRelatedSym tag (146), which indicates the number of legs in the proposed security. After the NoRelatedSym tag, security related fields are repeated for each leg in the proposed security.

Two additional pieces are needed specify the strategy.

- *RatioQty* is a quantity field that indicates the ratio of the leg to other legs in the strategy.
- *Side* indicates if that particular leg will be bought or sold as part of the strategy.

Example using *RatioQty* and *Side*:

A Butterfly strategy consists of simultaneously:

Buying 1 Call at Strike Price #1

Selling 2 Calls at the next higher strike price (Strike Price #2)

Buying 1 call at the next higher strike price (Strike Price #3)

The Legs that would describe this strategy are as follows:

PutOrCall	RatioQty	Side
1=Call	1	1=Buy
1=Call	2	2=Sell
1=Call	1	1=Buy

## Scenarios

### Scenario 1 - Typical use of Security Definition message in placing an Order

This scenario has the first party defining a strategy order using a Security Definition message.

First Party		Second Party
<u>Security Definition Request message</u> SecurityRequest = 1 Propose an identity for the Security or Request an identity for the Security from second party	→	Interprets Security request
If second party accepted Security then the first party is free to use the Security in a trade	←	<u>Security Definition message</u> SecurityResponse=0
<u>New Order – Single message</u> Product = Security information from the Security Definition message	→	Order is handled by exchange
	←	<u>Execution Report</u> Order received (Most likely will need to add Security information to the Execution report)
	←	<u>Execution Report</u> Fill Information on Order

### Scenario 2 - Inquire Securities Types Available

This scenario has the first party requesting a list of Security types supported by the second party

First Party		Second Party
<u>Security Definition Request message</u> SecurityRequest = 2	→	Processes Security Definition message
First party can use this to select a list of messages	←	<u>Security Definition message</u> In this scenario, the trading party only trades three types of securities SecurityResponseType= 2 NoRelatedSym=3 UnderlyingSecuritySymbol=SecurityType#1 UnderlyingSecuritySymbol=SecurityType#2 UnderlyingSecuritySymbol=SecurityType#3

### Scenario 3 – Inquire Common Stocks Available for Trading with Counterparty.

This example shows how the Security Definition Request Message and Security Definition Messages can be used to return a list of common stocks available for trading with a counterparty. The first party specifies the SecurityRequest equal to 3 and specifies the SecurityType of common stock. The second party returns a list of common stocks available on its market. Note: This is intended to return standing data (static data) or a list of products available for trading – it is **not** intended to return an order book (see Market Data messages for this purpose). This is most applicable but not limited, to the case when the second party is an exchange.

First Party		Second Party
<u>Security Definition Request message</u>  In this scenario the initiator wants to obtain a list of common stock available for trading with the counterparty.  SecurityRequest=3 SecurityType="CS"	→	Processes Security request  Create a list of common stocks that are available for trading.
First party can use this to select a list of messages	←	<u>Security Definition message</u>  Contains list of common stocks available for trading with the second party  SecurityResponse=3 NoRelatedSym=25 UnderlyingSecuritySymbol="AOL" ....Other tags for this security UnderlyingSecuritySymbol="GM" ....Other tags for this security UnderlyingSecuritySymbol="IBM" ....Other tags for this security

#### Scenario 4 - Inquire all securities traded by a trading party

This scenario has the first party requesting a list of Security types supported by the second party.

First Party		Second Party
<u>Security Definition Request message</u>  SecurityRequest=3	→	Processes Security request  Create a list of the Securities available for the specified SecurityType
First party can use this to select a list of messages	←	<u>Security Definition message</u>  Contains list of Securities available for the specified the Security Types supported by second party  SecurityResponse=3 NoRelatedSym=XX  Security information for each security is provided for each of the XX securities.

### Scenario 5 – Inquire Option Classes Available for Trading with Counterparty.

This example shows how the *Security Definition Request* Message and *Security Definition* Messages can be used to return a list of option classes available for trading with a counterparty. The first party specifies a Security Request Type equal to 3 (Request List of Securities) and the SecurityType of options. The second party returns a list of option classes available on its markets. Note: This is intended to return standing data (static data) or a list of products available for trading – it is not intended to return an order book (see Market Data messages).

First Party		Second Party
<u>Security Definition Request message</u>  In this scenario the initiator wants to see a list of option series for IBM that are traded by the counterparty (that may be an exchange)  SecurityRequest=3 SecurityType="OPT"	→	Processes Security request  Create a list of common stocks that are available for trading.
First party can use this to select a list of messages	←	<u>Security Definition message</u>  Contains list of common stocks available for trading with the second party  SecurityResponse=3  NoRelatedSym=25  UnderlyingSecuritySymbol="AOL"  UnderlyingSecuritySymbol="GM"  UnderlyingSecuritySymbol="IBM"

### Scenario 6 - Inquire list of option series for a class

This scenario has the first party requesting a list of option classes by setting the SecurityRequest equal to 3, the SecurityType to "OPT", and a security symbol = "IBM". Because a symbol is given, the second party sends back a list of option series for the class specified with a symbol or securityID.

First Party		Second Party
<u>Security Definition Request message</u>  SecurityRequest=3 SecurityType="OPT" Symbol="IBM"  Any of the security identification fields can be populated for this query	→	Processes Security request  Because a symbol is provided the second party sends back a list of option series.



First party can use this to select a list of messages	←	<u>Security Definition message</u> Contains list of option series available for the specified the class specified in the request. SecurityResponse=3 NoRelatedSym=XX Security information for each security is provided for each of the XX securities.
---	---	--

## Appendix J

### Example Usage of Encoded Fields For Japanese Language Support

**Example 1 - Specify the ASCII/English value as Issuer plus Japanese character set as EncodedIssuer**

Tag	Field Name	Value
... Other Standard Header fields		
347	MessageEncoding	Shift_JIS
... Other Standard Header fields		
... Other Message Body fields		
106	Issuer	HITACHI
348	EncodedIssuerLen	10
349	EncodedIssuer	日立製作所
... Other Message Body fields		

**Example 2 - Specify the ASCII/English value as Issuer plus Japanese character set as EncodedIssuer. Specify the ASCII/English value as Text plus Japanese character set as EncodedText.**

Tag	Field Name	Value
... Other Standard Header fields		
347	MessageEncoding	Shift_JIS
... Other Standard Header fields		
... Other Message Body fields		
106	Issuer	HITACHI
348	EncodedIssuerLen	10
349	EncodedIssuer	日立製作所
... Other Message Body fields		
58	Text	This is a test
356	EncodedTextLen	17
357	EncodedText	これはテストです。
... Other Message Body fields		

#### Precautions when using UNICODE

There is the possibility that an SOH may be included in the character data when using UNICODE encoding. To avoid parsing problems, a FIX engine should use the EncodedLen value to extract the proper number of bytes.

## Appendix K

### Example Usage of Allocations

The Allocation 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.

#### Orders involving Pre-Trade Allocation consist of the following steps:

- Buyside sends a New Order request message specifying one or more AllocAccount and AllocShares values within the repeating group designated by NoAllocs.
- Sellside sends Execution Report messages for the “New” and resulting fills.
- Post-Trade Allocation messaging takes place

The typical flow for Pre-Trade allocation is as follows:

Institution	➔	New Order-Single (OrderQty=35000, NoAllocs=2, AllocAccount=ACCT1, AllocShares=10000, AllocAccount=ACCT2, AllocShares=25000)	Broker
	←	Execution Report (ExecType = “0” [New])	
	←	Execution Report (ExecType = “1”) [Partial Fill] Execution Report (ExecType = “2”) [Filled] (optional Execution Report (ExecType = “3”) [Done for day])	
		Post-Trade Allocation Processing (see examples below)	

#### Post-Trade Allocation 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 without Misc Fees (see examples 1-1 and 1-2)**

The typical flow for US domestic trading (without MiscFees) is as follows:

	➔	Allocation (AllocTransTyp=New)	
--	---	--------------------------------	--

Institution	←	AllocationACK (AllocStatus=Received Not Yet Processed)	Broker
	←	AllocationACK (AllocStatus=Accepted or Rejected)	
	→	Settlement Instructions (optional) (SettlInstSource=Institution's)	
	←	Settlement Instructions (optional) (SettlInstSource=Broker's)	

\*Settlement Instructions may occur anywhere in the flow and may represent standing instructions.

## 2. Buyside-initiated with Misc Fee computation (see examples 2-1 and 2-2)

The typical flow for international trading (with MiscFees) is as follows:

Institution	→	Allocation (AllocTransTyp=Preliminary, AllocAccounts provided without MiscFees or NetMoney)	Broker
	←	AllocationACK (AllocStatus=Received Not Yet Processed)	
	←	Allocation (AllocTransTyp=Calculated, MiscFees and NetMoney provided by AllocAccount)	
	→	AllocationACK (AllocStatus=Received Not Yet Processed)	
	→	AllocationACK (AllocStatus=Accepted or Rejected)	
	→	Settlement Instructions (optional*) (SettlInstSource=Institution's)	
	←	Settlement Instructions (optional*) (SettlInstSource=Broker's)	

\*Settlement Instructions may occur anywhere in the flow and may represent standing instructions.

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

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

Institution	←	Allocation (AllocTransTyp=Calculated without preliminary, MiscFees and NetMoney provided by AllocAccount)	Broker
	→	AllocationACK (AllocStatus=Received Not Yet Processed)	
	→	AllocationACK (AllocStatus=Accepted or Rejected)	
	→	Settlement Instructions (optional*) (SettlInstSource=Institution's)	
	←	Settlement Instructions (optional*) (SettlInstSource=Broker's)	

\*Settlement Instructions may occur anywhere in the flow and may represent standing instructions.

**Example 1-1: Buyside-initiated flow without MiscFee computation, using Average Price (all AllocAccounts with same AvgPx)**

BUYSIDE			SELLSIDE
	➔	New Order-Single	
	⬅	Execution Report (ExecType = "0" [New])	
	⬅	Execution Report (ExecType = "1") [Partial Fill] Execution Report (ExecType = "2") [Filled] (optional Execution Report (ExecType = "3") [Done for day])	
Allocate			
	➔	Allocation (AllocTransTyp=New)	
	⬅	AllocationACK (AllocStatus=Received Not Yet Processed)	
	⬅	AllocationACK (AllocStatus=Accepted or Rejected)	

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

**Allocation Msg**



Sym bol	B/S	Mkt	Order section			AvgPx	Repeating fields				Repeating fields	
			ID	OrdID	ClOrdID		ExecID	LastPx	LastShares	AllocAccount	AllocShares	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			

**Example 1-2: Buyside-initiated flow without MiscFee computation, using Executed Price**

BUYSIDE			SELLSIDE
	➔	New Order-Single	
	➔	Execution Report (ExecType = "0" [New])	
	➔	Execution Report (ExecType = "1") [Partial Fill] Execution Report (ExecType = "2") [Filled] (optional Execution Report (ExecType = "3") [Done for day])	
Allocate			
	➔	Allocation (AllocTransTyp=New)	
	➔	AllocationACK (AllocStatus=Received Not Yet Processed)	
	➔	AllocationACK (AllocStatus=Accepted or Rejected)	

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

**Allocation Msg**

Symbol	B/S	Mkt	Order section			Repeating fields			Repeating fields			
			ID	OrdID	ClOrdID	ExecID	LastPx	LastShares	AllocAcct	AllocPrice	AllocShares	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

**Example 2-1: Buyside-initiated flow with MiscFee computation, using Average Price (all AllocAccounts with same AvgPx)**

BUYSIDE			SELLSIDE
	➔	New Order-Single	
	⬅	Execution Report (ExecType = "0" [New])	
	⬅	Execution Report (ExecType = "1") [Partial Fill] Execution Report (ExecType = "2") [Filled] (optional Execution Report (ExecType = "3") [Done for day])	
Allocate			
	➔	Allocation (AllocTransTyp=Preliminary, AllocAccounts provided without MiscFees or NetMoney)	
	⬅	AllocationACK (AllocStatus=Received Not Yet Processed)	
			Commission/ Fee Calc
	⬅	Allocation (AllocTransTyp=Calculated, MiscFees and NetMoney provided by AllocAccount)	
	➔	AllocationACK (AllocStatus=Received Not Yet Processed)	
	➔	AllocationACK (AllocStatus=Accepted or Rejected)	

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

**Allocation Msg**

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

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

BUYSIDE			SELLSIDE
	➔	New Order-Single	
	⬅	Execution Report (ExecType = "0" [New])	
	⬅	Execution Report (ExecType = "1") [Partial Fill] Execution Report (ExecType = "2") [Filled] (optional Execution Report (ExecType = "3") [Done for day])	
Allocate			
	➔	Allocation (AllocTransTyp=Preliminary, AllocAccounts provided without MiscFees or NetMoney)	
	⬅	AllocationACK (AllocStatus=Received Not Yet Processed)	
			Commission/ Fee Calc
	⬅	Allocation (AllocTransTyp=Calculated, MiscFees and NetMoney provided by AllocAccount)	
	➔	AllocationACK (AllocStatus=Received Not Yet Processed)	
	➔	AllocationACK (AllocStatus=Accepted or Rejected)	

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

**Allocation Msg**

Symbol	B/S	Mkt	Order section			Repeating fields			Repeating fields				
			ID	OrdID	ClOrdID	ExecID	LastPx	LastShares	AllocAccount	AllocPrice	AllocShares	Commission	Repeating fields (NoMiscFees=1)
1234	Buy	T	999	520	20	300	1300	3000					MiscFeeType
						301	1313	1000	F1	1300	2000	25061	9
						302	1300	3000	F1	1313	1000	12656	9
						303	1320	2000	F2	1300	2000	25058	9



<b>F2</b>	<b>1320</b>	1000	12722	9	636
<b>F3</b>	<b>1300</b>	2000	25058	9	1252
<b>F3</b>	<b>1320</b>	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.

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

BUYSIDE			SELLSIDE
	➔	New Order-Single	
	⬅	Execution Report (ExecType = "0" [New])	
	⬅	Execution Report (ExecType = "1") [Partial Fill] Execution Report (ExecType = "2") [Filled] (optional Execution Report (ExecType = "3") [Done for day])	
Allocate			
			Commission/ Fee Calc
	⬅	Allocation (AllocTransTyp=Calculated without preliminary, MiscFees and NetMoney provided by AllocAccount)	
	➔	AllocationACK (AllocStatus=Received Not Yet Processed)	
	➔	AllocationACK (AllocStatus=Accepted or Rejected)	

Sym bol	B/S	Mkt	Order Message			Execution Rpt Messages		
			Account	OrdID	ClOrdID	ExecID	LastPx	LastShares
IBM	Buy	N	<b>F1</b>	520	20	300	1300	3000
						301	1313	1000
						302	1300	3000
						303	1320	2000

#### Allocation Msg



Sym bol	B/S	Mkt	Order section			AvgPx	Repeating fields				Repeating fields	
			ID	OrdID	ClOrdID		ExecID	LastPx	LastShares		AllocShare s	Commission
IBM	Buy	N	999	520	20	<b>1305.889</b>	300	1300	3000	<b>F1</b>	9000	113277
							301	1313	1000			
							302	1300	3000			
							303	1320	2000			

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

BUYSIDE			SELLSIDE
	➔	New Order-Single	
	⬅	Execution Report (ExecType = "0" [New])	
	⬅	Execution Report (ExecType = "1") [Partial Fill] Execution Report (ExecType = "2") [Filled] (optional Execution Report (ExecType = "3") [Done for day])	
Allocate			
			Commission/ Fee Calc
	⬅	Allocation (AllocTransTyp=Calculated without preliminary, MiscFees and NetMoney provided by AllocAccount)	
	➔	AllocationACK (AllocStatus=Received Not Yet Processed)	
	➔	AllocationACK (AllocStatus=Accepted or Rejected)	

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

#### Allocation Msg

Symbol	B/S	Mkt	Order section			Repeating fields			Repeating fields					
			ID	OrdID	ClOrdID	ExecID	LastPx	LastShares	AllocAccount	AllocPrice	AllocShares	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.

## Appendix L

### Pre-Trade Message Targeting/Routing

Three fields, NoRoutingID, RoutingType, and RoutingID have been added to support list processing on third party networks. Vendor "indication of interest" systems generally have list management capabilities. These capabilities include blocking and targeting. To mirror the functionality of the vendor indication systems both blocking and targeting were supported.

#### Targeting

Targeting relates to the message that contains a list of targeted firms or targeted vendor maintained list identifiers to receive the indication. Generally, most vendor "indication of interest" systems maintain list identifiers that contain firm identifiers for their broker connections. For example, a broker has a list called "JapanList" that contains three institutions JapaneseFirm1, JapaneseFirm2, and JapaneseFirm3. The three firm identifiers are created by the vendor.

Targeting allows for the definition of the universe of firms to receive the indication of interest. A indication of interest message without the targeting identifiers (either firm or list) is assumed to be sent to the whole list of indication receiving firms managed by the vendor (i.e. every institution connected to the broker).

Specific targeting can be accomplished through the combination of firm identifiers and list identifiers. For example, a broker needs to send an indication of interest to a vendor maintained list of U.K. based clients called "UKList" and two U.S. based firms. The targeting section of the indication of interest would look as follows:

215=3^216=1^217=USFirm1^216=1^217=USFirm2^216=2^217=UKList^

Note: The ^ character represents the SOH delimiter.

#### Tag Explanation

215=3	Three pairs of routing types and IDs to be processed
216=1	Target ID to follow
217=USFirm1	Target ID named USFirm1
216=1	Target ID to follow
217=USFirm2	Target ID named USFirm2
216=2	Target list to follow
217=UKList	Target list named UKList

The vendor would assemble the destination list based on the two firm identifiers and the one list identifier.

## Blocking

An indication with blocking contains a list of firm identifiers or vendor maintained list identifiers that will be excluded from the targeted list of indication receiving firms managed by the vendor. Using the blocking fields without targeting fields implies that indication of interest is being blocked from the whole universe of institutions available to the broker (i.e. everyone on the vendor's system but these firms).

Many "indication of interest" systems have sophisticated list handling mechanisms that need to be replicated. Blocking is not always performed from the whole universe of firms on the system (i.e. ALL).

Using a combination of targeting and blocking fields can allow for sophisticated list management capabilities. For example, let's assume that the broker intends to send an indication of interest to the universe defined by the broker's UKList and two U.S. based firms. However, the broker needs to exclude one UK based firm from the UKList. The targeting and blocking section would appear as follows:

215=4^216=2^217=UKList^216=1^217=USFirm1^216=1^217=USFirm2^216=3^217=UKFirm1^

Note: The ^ character represents the SOH delimiter.

## Tag Explanation

215=4	Four pairs of routing types and IDs to be processed
216=2	Target list to follow
217=UKList	Target list named UKList
216=1	Target firm to follow
217=USFirm1	Target firm named USFirm1
216=1	Target firm to follow
217=USFirm2	Target firm named USFirm2
216=3	Blocked firm to follow
217=UKFirm1	UKFirm1 is blocked from receiving IOI

The vendor would assemble the targets based on the supplied UKList and two firm identifiers (USFirm1 and USFirm2) and then remove UKFirm1 from the combined list.

## Other Issues

It is expected that every indication of interest message will have a unique IOId for the fix session usually the trading day.

For canceling and replacing, the vendor system would cancel or replace every destination that has been identified on the previous indication of interest by the IOIid. Blocking and targeting information would not be required on the canceled or replaced indication of interest.

The use of vendor based firm identifiers requires periodic updates to the brokers to ensure proper blocking and targeting. It is expected that vendors will provide file base transfers of firm identifiers and company names until a more automated solution becomes available.

## **Appendix M**

### **FIXML Support**

The FIX Technical Committee has added two new fields XMLDataLen and XMLData to the Standard FIX Header to facilitate the creation of FIXML pilots.

The FIXML Working Group has been conducting a “proof of concept” pilot project to demonstrate evolution of FIX tag-value application messages to FIXML, a markup grammar based on the Extensible Markup Language (XML). This pilot, following the FIXML white paper and FIXML DTD, is the third deliverable in the process of creating a FIXML standard for the FIX Protocol.

The addition of these new fields will allow users to:

- Modify an existing FIX engine by adding a validating XML parser to allow FIXML to be sent enclosed in a traditional FIX header and trailer.
- Connect the modified FIX engine to an application that has the ability to send and receive FIX messages.
- Measure the ease with which FIX engines can be ported over to FIXML.
- Prove that FIXML messages will not break existing FIX engines. The same engine should be able to handle FIXML and FIX messages.

## **Appendix N**

### **Program/Basket/List Trading**

#### **Overview**

A set of messages allow for the automation of program trading. While it is hoped that the message set is comprehensive enough, to automate the complete cycle, it is expected that not all messages will be used in all transactions. Although the message set may appear to be quite complex at first glance, most of the complexity arises from developing one message set that can be used to support two different business models for list trading. The two models, the “Disclosed” and “Non Disclosed” models, are described in the next two sections. The “Disclosed” model is commonly used in Japan while the “Non Disclosed” model is commonly used in Europe and America.

#### ***“Non Disclosed” model (e.g. US/European)***

The buy-side details to the sell-side information about the sector, country and potential market impact of the stocks to be bought or sold. Using this information the sell-side firms bid for the trade. If successful the buy-side firm gives the sell-side firm a detailed list of the stocks to be traded and the sell-side firm executes the trades.

The important point in the “Non Disclosed” model is that the stocks in the list are not disclosed until a particular sell-side firm has won the portfolio.

#### ***“Disclosed” model (e.g. Japanese)***

The buy-side details the exact stocks and sizes to be traded and the sell-side firm offers the buy-side firm a two-way price, to buy or to sell the indicated stocks. The buy-side firm then tells the successful sell-side firm to buy or sell on its behalf.

The important point in the “Disclosed” model is that all sell-side firms see all of the stocks and quantities in the portfolio during the bidding phase regardless of whether or not they win the business.

The New Order - List message can be used, with the side omitted as part of the bidding process, as is the practice in “Disclosed” model or once the bidding has been completed to exchange the list of stocks that make up the program to be traded. Pre-trade allocation is handled via a repeating group within the repeating order block.

Order modification and cancelation of a portfolio is a major change to the agreement between the buy-side and sell-side firms and as such this change should be conducted by telephone. If an automated route for dealing with amendment/cancelation is required then the existing messages can be used – List Cancel, Order Cancel/Replace Request.

The New Order - List message is based on the single order message and message flows for canceling a single stock line within a program trade should be those used to support order cancelation in the single order model (e.g. Order Cancel/Replace Request, etc). The ListID in those systems should be used to assist in identifying the order as part of a list trade. Similarly, the Order Status Request message can be used to request the status of a single order in a portfolio trade.

The List Strike Price Message details the prices that a principal trade is being executed at. In some transactions this appears to be generated by the sell side and checked by the buy-side, in others the reverse is true, and in other cases this information is not passed until the final execution reports



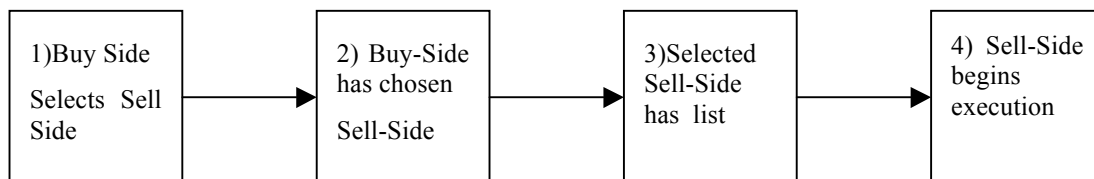
Pre-trade allocation is much more common place in the program trading community than it is in block trading. For the purposes of pre-trade allocation, a repeating group covering account and number of shares has been added to the order message. It is assumed that participants will use the either FIX allocation messages and message flows for post trade allocation or their existing allocation systems. (e.g. in the event of a pre-allocation basket trade).

At any stage in the processing of a list message the buy-side may request the status of the list from the sell-side using the List Status Request message. The sell-side responds with a List Status Response message. The sell-side can also send the List Status Response message in an unsolicited fashion according to the requirements passed in the bidding phase or in the List message. The List Status Response message provides summary detail about each of the orders in the List. The sell-side should acknowledge any list request from the buy-side with a List Status Response message providing the current state.

Once the portfolio has been executed by the sell-side and a List Status Response message has been sent to the buy-side indicating "DONE" for each of the orders in the List, the list can be allocated. If pre-allocation information was provided with the original orders and the orders were fully executed then the allocation information is already known to the sell-side. If the pre-trade allocations are no longer appropriate post trade allocation may be preformed either using FIX Allocation messages or existing allocation systems.

## Message Flow Diagrams

### *Overview of logical stages*



The diagram above shows the logical stages involved in the execution of a program trade.

Transition	Description
1->2	<p>This transition can occur in the following ways :</p> <ul style="list-style-type: none"> <li>Buy-side has preferred list which means the business will be directed to a specific Sell-Side</li> <li>Buy-Side provides details of the program to a number of sell-sides. This can be achieved using a mixture of telephone, fax, modem links, and FIX Bid messages.</li> </ul>
2->3	<p>Details of the program are transmitted to the chosen Sell-side using telephone, fax, modem links, or FIX New Order - List message.</p>
3->4	<p>This transition can occur in the following ways :</p> <ul style="list-style-type: none"> <li>Buy-Side and Sell-Side communicate by telephone to confirm content of the program and the Buy-Side instructs the Sell-Side to begin execution.</li> <li>Buy-Side sends a List Execute FIX message to instruct Sell-Side to begin execution</li> </ul>

4	Once the list is being executed the FIX List status messages may be used to notify/request status of the list
---	---

***Order Details sent via FIX (Bidding Outside of FIX)***

<b>Buy Side Institution</b>		New Order - List Message (1 broker)	→	<b>Sell Side Broker Dealer</b>
	←	2) List Order Status (ACK)		

***“Disclosed” Bid and Program Trade***

<b>Buy Side Institution</b>		1) New Order - List Message (N brokers)	→	<b>Sell Side Broker Dealer</b>
	←	2) List Order Status (ACK) (1 per broker)		
		3) Bid Request Message (N brokers)	→	
	←	4) Bid Response (1 per broker)		
		5) Cancel bid sent to (N – 1) brokers	→	

***“Non Disclosed” Bid and Program Trade***

Buy Side Institution	1) Bid Request Message (N brokers)		➔	Sell Side Broker Dealer
	➔	2) Bid Response (1 per broker)		
	Buy-side selects one Sell-side and sends order detail for execution			
	3a) Bid Request Cancel (N – 1 brokers)		➔	
	3b) New Order - List Message (1 sell-side)			
	➔	4) List Status Response (ACK)		

***Message Flows once a buy-side has chosen a single sell-side and transmitted New Order - List messages***

The following message flows can occur in any order relative to each other and some may occur many times.

***Optional notification to begin execution (may occur zero or one times)***

<b>Buy Side Institution</b>		1) List Execute message (1 sell-side)	→	<b>Sell Side Broker Dealer</b>
	←	2) List Order Status (Ack) Response		

***Optional transfer of Principal Portfolio Trade prices from buy-side to sell-side (may occur zero or one times)***

<b>Buy Side Institution</b>		1) List Strike Price message	→	<b>Sell Side Broker Dealer</b>
	←	2) List Status message		

***Optional transfer of Principal Portfolio Trade prices from sell-side to buy-side (may occur zero or one times)***

<b>Buy Side Institution</b>	←	1) List Strike Price message		<b>Sell Side Broker Dealer</b>
		2) List Status message	→	

***Optional Execution Report status update (may occur zero or N times)***

<b>Buy Side Institution</b>	←	1) Execution Reports (if requested)		<b>Sell Side Broker Dealer</b>
---------------------------------	---	-------------------------------------	--	------------------------------------

***Optional List Status Request (may occur zero or N times)***

<b>Buy Side Institution</b>		1) List Status Request message	→	<b>Sell Side Broker Dealer</b>
	←	2) List Status message		

***Optional Sell-Side unsolicited Status update (may occur zero or N times)***

<b>Buy Side Institution</b>	←	2) List Status message		<b>Sell Side Broker Dealer</b>
---------------------------------	---	------------------------	--	------------------------------------

**Scenario 1 Bidding performed by Telephone and List provided via FIX**

Message	Description	Purpose
	New Order - List Message from B/S to S/S	Details the list of stocks that an institution wishes to trade. Normally side is omitted and an indicator is set to show that this message is part of a bid

	List Status Response (Acknowledgement)	List status response indicates that the sell-side has received the New Order - List message. The status of each order in the list should indicate a status of <i>bid</i> or <i>rejected</i> . The former if the stock is recognised and the latter if the stock is not recognised.
may be omitted	List Execute Message	Details the specific bid that has been accepted. The specific bid indicates the direction of the list to be executed.
Required if previous provided	List Status Response	Details the status of each order in the list. The status should be <i>executing</i> for each order.
	Status updates may optionally follow	

#### Scenario 2 Fully Disclosed Program Trade – with bidding stage through FIX

Message	Description	Purpose
	New Order - List Message from B/S to S/S	Details the list of stocks that an institution wishes to trade. Normally side is omitted and an indicator is set to show that this message is part of a bid
	List Status Response (Acknowledgement)	List status response indicates that the sell-side has received the New Order - List message. The status of each order in the list should indicate a status of <i>bid</i> or <i>rejected</i> . The former if the stock is recognised and the latter if the stock is not recognised.
	Bid Request Message from B/S to S/S	Details the types of bids required, eg Side, Execution Type etc
may be omitted	Bid Response Message	Details the bid response for a program
may be omitted	List Execute Message	Details the specific bid that has been accepted. The specific bid indicates the direction of the list to be executed.
Required if previous provided	List Status Response	Details the status of each order in the list. The status should be <i>executing</i> for each order.
	Status updates may optionally follow	

#### Scenario 3 Non-Disclosed Program Trade – with bidding stage through FIX

Message	Description	Purpose
may be omitted	Bid Request from B/S to S/S	Details the liquidity information about the stocks that an institution wishes to trade. It does not identify the stocks in the

done by phone		program.
may be omitted done by phone	Bid Response Message from S/S to B/S	Details the bid response for a program
	List Message Detail Message from B/S to S/S	Details the list of stocks that an institution wishes to trade including the stock, quantity, and direction for each order.
Required if previous provided	List Status Response	Details the status of each order in the list. The status should be <i>awaiting execution, executing</i> or <i>rejected</i> for each order.
may be omitted done by phone	List Execute Message from B/S to S/S	Details the bid for a program
required if previous provided	List Status Response	Details the status of each order in the list. The status should be <i>executing</i> for each order.

#### Illustration of liquidity indicator fields usage

Normally details, by country and by sector, as number at <5%, no in 5-10%, no in 10-30% and number at > 30% eg 1@ 70%, 1 @ 600% For example

Country	<5%	5 – 10%	10 - 30%	> 30%
DEM	1 Sec \$1000000	4 Sec \$2000000	7 Sec \$1500000	1 Sec @60%, \$3000000 1 Sec @300% \$8000000
ESP	4 Sec \$3000000	5 Sec \$3000000	3 Sec \$3500000	
UK	3 Sec \$4500000	6 Sec \$3600000	2 Sec \$5000000	1 Sec @450% \$9000000

Sector	<5%	5 – 10%	10 - 30%	> 30%
Industrails	2 Sec \$1500000	5	4	1 Sec @300% \$8000000
Parmacetical	4 Sec	3	3	1 Sec @450% \$9000000

Hotels	2	7	5	1    Sec    @60%, \$3000000
--------	---	---	---	--------------------------------

**X1.. 9 and Y1 .. 9 simply represent the numbers of stocks in each category.**

#### Illustration of liquidity indicator fields usage

The liquidity indicator fields are used to describe the shape of a basket trade in terms of the liquidity and classification of the stocks contained within the list. Thus a list that may be described by the following to tables.

List liquidity information by country.

% columns refer to percentage of average daily volume.

Country	<5%	5 – 10%	10 - 30%	> 30%
DEM	1 Security Value \$1000000	4 Security Value \$2000000	7 Security Value \$1500000	1 Security Value \$3M @ 60%  1 Security Value \$8M @300%
ESP	4 Security Value \$3000000	5 Security Value \$3000000	3 Security Value \$3500000	
UK	3 Security Value \$4500000	6 Security Value \$3600000	2 Security Value \$5000000	1 Security Value \$9M @ 450%

List liquidity information by Security Sector.

% columns refer to percentage of average daily volume.

Sector	<5%	5 – 10%	10 - 30%	> 30%
Industrials	2 Security Value \$1500000	5 Security Value \$2600000	4 Security Value \$3000000	1 Security Value \$8M @300%
Pharmaceutical	4 Security Value \$3000000	3 Security Value \$3000000	3 Security Value \$1500000	1 Security Value \$9M @450%
Hotels	2 Security Value \$4000000	7 Security Value \$3000000	5 Security Value \$2500000	1 Security Value \$3M @60%

Would be represented by the following BidRequest Message.

BidRequest Message (Non Disclosed bid, basket of securities, not an exchange for physical trade)

Client Bid ID	Bid Request Trans Type	Total Num Securities	Bid Type	Side Value 1	Repeating fields						
					Bid Descr- iptor Type	Bid Descr- Iptor	Side Value Ind	Liquidity Value	Liquidity Num Secu- rities	Liqui- dity Pct Low	Liqui- dity Pct High
1001	N	38	1	37100000	2	DEM	1	1000000	1	0.00	0.05
					2	DEM	1	2000000	4	0.05	0.10
					2	DEM	1	1500000	7	0.10	0.30
					2	DEM	1	3000000	1	0.60	*- NP
					2	DEM	1	8000000	1	3.00	*- NP
					2	ESP	1	3000000	4	0.00	0.05
					2	ESP	1	3000000	5	0.05	0.10
					2	ESP	1	3500000	3	0.10	0.30
					2	UK	1	4500000	3	0.00	0.05
					2	UK	1	3600000	6	0.05	0.10
					2	UK	1	2000000	2	0.10	0.30
					2	UK	1	9000000	1	4.50	*- NP
					1	Ind	1	1500000	2	0.00	0.05
					1	Ind	1	2600000	5	0.05	0.10
					1	Ind	1	3000000	4	0.10	0.30
					1	Ind	1	8000000	1	3.00	*- NP
					1	Pharm	1	3000000	4	0.00	0.05
					1	Pharm	1	3000000	3	0.05	0.10
					1	Pharm	1	1500000	3	0.10	0.30
					1	Parm	1	9000000	1	4.50	*- NP
					1	Hotels	1	4000000	2	0.00	0.05
					1	Hotels	1	3000000	7	0.05	0.10
					1	Hotels	1	2500000	5	0.10	0.30
					1	Hotels	1	3000000	1	0.60	*- NP

Notes

- \*- NP field not present in repeating group as entry is describing a single stock at a specific liquidity
- Where the BidDescriptorType set to 1 the entry in the BidDescriptor field is free text, the sector names Pharmaceuticals and Industrials have been shorten to make every thing fit.



## Appendix O

### Foreign Exchange (F/X) Trading

Notes:

- The forex Symbol is defined in "EBS" (Electronic Banking System) format: "CCY1/CCY2".
  - Rates are expressed as "currency1 in currency2" (or "currency2 per currency1") and are calculated as CCY2 divided by CCY1 (**NOT** CCY1 divided by CCY2)
- (e.g. "GBP/USD" represents a rate expressed as USD per GBP, "USD/JPY" represents a rate expressed as JPY per USD, etc.).
  - CCY1 and CCY2 are ISO currency codes
- OrderQty represents the amount expressed in units of currency specified by the Currency field. Note OrderQty must be capable of representing large values with at least 2 decimal places.
- The value of the Currency field represents the denomination of the quantity field(s).
- The Rate (specified as a "Price" field) represents CCY2 divided by CCY1 (**NOT** CCY1 divided by CCY2)
- The "unknown quantity" can be calculated using the following rules:
  - If Currency field value = CCY1 then as: (OrderQty \* Rate)
  - If Currency field value = CCY2 then as: (OrderQty / Rate)

Verbal representation	Side	OrderQty	Currency	Symbol (CCY1/CCY2)	Rate	Rate "Style"	"Resulting" Quantity
Sell 1,000,000 USD for JPY	Sell	1,000,000	USD	USD/JPY	105.92	Normal	105,920,000
				JPY/USD	0.009441088	Inverted	
Sell 50,000,000 JPY for USD	Sell	50,000,000	JPY	USD/JPY	105.92	Normal	472,054.38
				JPY/USD	0.009441088	Inverted	
Buy 50,000,000 JPY for USD	Buy	50,000,000	JPY	USD/JPY	105.92	Normal	472,054.38
				JPY/USD	0.009441088	Inverted	
Buy 1,000,000 USD for JPY	Buy	1,000,000	USD	USD/JPY	105.92	Normal	105,920,000
				JPY/USD	0.009441088	Inverted	
Sell 1,000,000 USD for CAD	Sell	1,000,000	USD	USD/CAD	1.437	Normal	1,437,000.00
				CAD/USD	0.695894224	Inverted	
Sell 50,000,000 CAD for USD	Sell	50,000,000	CAD	USD/CAD	1.437	Normal	34,794,711.20
				CAD/USD	0.695894224	Inverted	
Buy 50,000,000 CAD for USD	Buy	50,000,000	CAD	USD/CAD	1.437	Normal	34,794,711.20
				CAD/USD	0.695894224	Inverted	

Buy 1,000,000 USD for CAD	Buy	1,000,000	USD	USD/CAD	1.437	Normal	1,437,000.00
				CAD/USD	0.695894224	Inverted	
Sell 1,000,000 USD for GBP	Sell	1,000,000	USD	GBP/USD	1.6368	Normal	610,948.19
				USD/GBP	0.610948192	Inverted	
Sell 50,000,000 GBP for USD	Sell	50,000,000	GBP	GBP/USD	1.6368	Normal	81,840,000.00
				USD/GBP	0.610948192	Inverted	
Buy 50,000,000 GBP for USD	Buy	50,000,000	GBP	GBP/USD	1.6368	Normal	81,840,000.00
				USD/GBP	0.610948192	Inverted	
Buy 1,000,000 USD for GBP	Buy	1,000,000	USD	GBP/USD	1.6368	Normal	610,948.19
				USD/GBP	0.610948192	Inverted	
Sell 1,000,000 USD for EUR	Sell	1,000,000	USD	EUR/USD	1.001	Normal	999,001.00
				USD/EUR	0.999000999	Inverted	
Sell 50,000,000 EUR for USD	Sell	50,000,000	EUR	EUR/USD	1.001	Normal	50,050,000.00
				USD/EUR	0.999000999	Inverted	
Buy 50,000,000 EUR for USD	Buy	50,000,000	EUR	EUR/USD	1.001	Normal	50,050,000.00
				USD/EUR	0.999000999	Inverted	
Buy 1,000,000 USD for EUR	Buy	1,000,000	USD	EUR/USD	1.001	Normal	999,001.00
				USD/EUR	0.999000999	Inverted	
Sell 1,000,000 EUR for GBP	Sell	1,000,000	EUR	EUR/GBP	.6111	Normal	611,100.00
				GBP/EUR	1.636393389	Inverted	
Sell 50,000,000 GBP for EUR	Sell	50,000,000	GBP	EUR/GBP	.6111	Normal	81,819,669.45
				GBP/EUR	1.636393389	Inverted	
Buy 50,000,000 GBP for EUR	Buy	50,000,000	GBP	EUR/GBP	.6111	Normal	81,819,669.45
				GBP/EUR	1.636393389	Inverted	
Buy 1,000,000 EUR for GBP	Buy	1,000,000	EUR	EUR/GBP	.6111	Normal	611,100.00
				GBP/EUR	1.636393389	Inverted	
Sell 1,000,000 EUR for CHF	Sell	1,000,000	EUR	EUR/CHF	1.6125	Normal	1,612,500.00
				CHF/EUR	0.620155039	Inverted	
Sell 50,000,000 CHF for EUR	Sell	50,000,000	CHF	EUR/CHF	1.6125	Normal	31,007,751.94
				CHF/EUR	0.620155039	Inverted	
Buy 50,000,000 CHF for EUR	Buy	50,000,000	CHF	EUR/CHF	1.6125	Normal	31,007,751.94
				CHF/EUR	0.620155039	Inverted	

Buy 1,000,000 EUR for CHF	Buy	1,000,000	EUR	EUR/CHF	1.6125	Normal	1,612,500.00
				CHF/EUR	0.620155039	Inverted	

## Glossary

### Business Terms

The following glossary is an attempt to identify business terms used in this document or related to implementing FIX globally. Requests for new terms and/or suggested definitions should be posted in the FIX Web Site's Discussion section.

All or None	A round-lot market or limit-price order that must be executed in its entirety or not at all; unlike Fill or Kill orders, AON orders are not treated as canceled if they are not executed as soon as represented in the Trading Crowd. [ExecInst]
At the Opening	A market or limit-price order to be executed at the opening of the stock or not at all; all or part of any order not executed at the opening is treated as canceled. [TimeInForce]
Basis Price	A price established by joint agreement of odd-lot dealers in 100-share-unit stocks when: <ul style="list-style-type: none"> <li>- no round-lot has occurred during the trading session,</li> <li>- the spread between the closing bid and offer is two points or more, and</li> <li>- on odd-lot the dealer has been given a "basis-price" order. [OrdType]</li> </ul>
Buy Minus	A round-lot market order to buy "minus" is an order to buy a stated amount of a stock provided that its price is: <ul style="list-style-type: none"> <li>- not higher than the last sale if the last sale was a "minus" or "zero minus" tick and</li> <li>- not higher than the last sale minus the minimum fractional change in the stock if the last sale was a "plus" or "zero plus" tick.</li> </ul> A limit price order to buy "minus" also states the highest price at which it can be executed. [Side]
Day Order	A buy or sell order that, if not executed expires at the end of the trading day on which it was entered. [TimeInForce]
Do Not Increase	A limit order to buy, a stop order to sell, or a stop-limit order to sell which is not to be increased in shares on the ex-dividend date as a result of a stock dividend or distribution. [ExecInst]
Do Not Reduce	A limit order to buy, a stop order to sell, or a stop-limit order to sell that is not to be reduced in price by the amount of an ordinary cash dividend on the ex-dividend date. A do-not-reduce order applies only to ordinary cash dividends; it should be reduced for other distributions - such as when a stock goes "ex" stock dividend or "ex" rights. [ExecInst]
Fill or Kill	A market or limit-price order that is to be executed in its entirety as soon as it is represented in the Trading Crowd; if not so executed, the order is to be canceled. Not to be confused with Immediate or Cancel. [TimeInForce]
Good Till Canceled	An order to buy or sell that remains in effect until it is either executed or canceled; sometimes called an "open order". [TimeInForce]
Good Till Executed	An order to buy or sell that remains in effect until it is executed.
Immediate or Cancel	A market or limit-price order that is to be executed in whole or in part as soon

	as it is represented in the Trading Crowd; any portion not so executed is to be canceled. Not to be confused with Fill or Kill. [TimeInForce]
Limit or Better	Indicates an order to - buy a security at the indicated limit price or lower, or to - sell a security at the indicated limit price or higher. [OrdType]
Limit With or Without	An order to be executed at a limit price, with or without round-lot sales; valid only for odd lot orders. [OrdType]
Market	Indicates an order to buy or sell a stated amount of a security at the most advantageous price obtainable after the order is represented in the Trading Crowd. [OrdType]
Market On Close	A round-lot order to be executed at - or as near to as practical - the close of the market. [OrdType]
Market Or Better	Indicates an order to buy or sell a stated amount of a security at the quoted market or better. [OrdType]
On Close	An odd-lot order to buy or sell to be filled at the price of the closing round-lot offer - plus the differential, for a buy order, or - minus the differential, for a sell order, or A crossing session order to buy or sell at the closing price. [OrdType]
Sell Plus	A round-lot market order to sell “plus” is an order to sell a stated amount of a stock provided that its price is: - not lower than the last sale if the last sale was a “plus” or “zero plus” tick and - not lower than the last sale minus the minimum fractional change in the stock if the last sale was a “minus” or “zero minus” tick. A limit-price order to sell “plus” also states the lowest price at which it can be executed. [OrdType]
Sell Short	An order to sell a security that the seller does not own; a sale effected by delivering a security borrowed by, or for the account of, the seller. Can only be executed on a “plus” or “zero plus” tick. [OrdType]
Sell Short Exempt	Short sale exempt from short-sale rules. [OrdType]
Stop	A stop order to buy which becomes a market order when the security trades at - or above - the stop price after the order is represented in the Trading Crowd. A stop order to sell which becomes a market order when the security trades at - or below - the stop price after the order is represented in the Trading Crowd. [OrdType]
Stop Limit	A stop order to buy which becomes a limit order at the limit price when the security trades at - or above - the stop price after the order is represented in the Trading Crowd. A stop order to sell which becomes a limit order at the limit price when the security trades at - or below - the stop price after the order is represented in the Trading Crowd. [OrdType]