FINANCIAL INFORMATION EXCHANGE PROTOCOL (FIX)

Version 5.0 Service Pack 1

VOLUME 7 – FIX USAGE NOTES

March 2008

© Copyright, 2007-2008, FIX Protocol, Limited

DISCLAIMER

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

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

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

© Copyright 2003-2008 FIX Protocol Limited, all rights reserved

REPRODUCTION

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

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

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

Contents – Volume 7

DISCLAIMER **2**

REPRODUCTION **2**

VOLUME INTRODUCTION **8**

PRODUCT: COLLECTIVE INVESTMENT VEHICLES (CIV) **9**

OVERVIEW AND SCOPE 9
MARKET ENVIRONMENT 9
CIV SECURITY TYPE IDENTIFICATION 10
TYPES OF CIV FIX MESSAGES 10
ORDER QUANTITIES 11
INTERMEDIARY IDENTIFICATION 11
INVESTOR DETAILS 11
INVESTOR IDENTIFICATION 12
NEW INVESTOR -> NEW ORDER -> REGISTRATION INSTRUCTION 12
FUND & UNIT IDENTIFICATION 12
ORDER DETAILS - SINGLE 13
ORDER DETAILS - LIST 13
COMMISSION INSTRUCTIONS 13
COMPLIANCE 14
SETTLEMENT INSTRUCTIONS 14
DISTRIBUTION INSTRUCTIONS 14
UNIT PRICES 15
Valuation point *15*
Single pricing *15*
Dual pricing *15*
EXECUTION REPORTS 16
CIV-specific use of OrdStatus: *16*
CIV EXAMPLES 18
CIV Example 1. Single order for a CIV fund for a known investor/nominee, to be dealt on a "historic" basis *18*
CIV Example 2. Single order for a CIV fund for a known investor/nominee, to be dealt on a "forward" basis *18*
CIV Example 3. Single order for a CIV fund for an investor/nominee not known to the fund manager - registration and settlement instructions after trade *19*
CIV Example 4. Single order for a CIV fund for an investor/nominee not known to the fund manager - registration and settlement instructions required before trade *20*
CIV Example 5. Single order for a CIV fund for a known investor/nominee – order modified before execution *21*
CIV Example 6. Single order for a CIV fund for a new investor/nominee to the fund manager - registration and settlement instructions rejected, then modified & accepted *21*
CIV Example 7. Exchange/switch order between several CIV funds from a single fund manager or via a funds supermarket *22*
CIV Example 8. Order for CIV fund by new or existing investor, routed via a client money/asset holding broker or funds supermarket to fund manager *23*
CIV Example 9. Order for CIV fund by an institutional investor, routed via a broker to a fund manager – possibly via a hub/exchange *24*
CIV Example 10. Order for CIV fund by new investor via non-client money/asset holding intermediary to fund manager *24*
CIV Example 11. Order for CIV fund by new investor, routed via non-client money/asset holding intermediary via a non-aggregating hub/exchange to fund manager *25*

© Copyright, 2007-2008, FIX Protocol, Limited

CIV Example 12. Order for CIV fund by new investor routed via intermediary to a funds supermarket – which places bulk/net orders to the fund manager 26
CIV Example 13. Exchange/switch order quantities – OrderPercent, Rounding, Sell Driven....27
CIV Example 14. CIV Bulk order – purchase of funds for multiple investors into a designated nominee....28
CIV Example 15. Registration Instruction – Joint Investors....29
CIV Example 16 Registration Instruction – Tenants in Common, ...31

PRODUCT: LISTED DERIVATIVES (FUTURES & OPTIONS) **32**

USE OF CFICODE TO IDENTIFY DERIVATIVES SECURITY 32
Single Leg Instruments *32*
Multileg Instrument Specification....*33*
US LISTED OPTIONS ORDER CAPACITY VALUES....33
Proposed option order capacity codes and their FIX 4.3 equivalents *34*
CUSTOMERORDERCAPACITY(TAG 582) MAPPINGS FOR FUTURES CTICODE 36
NEGATIVE PRICES PERMITTED FOR FUTURES AND OPTIONS STRATEGIES....37
DERIVATIVES MARKETS ORDER STATE TRANSITION 37
PARTY ROLES USED FOR DERIVATIVES MARKETS....38
MAPPING FIX 4.2 TO FIX 4.3 USAGE FOR OPTIONS MARKETS....39
GENERAL USAGE INFORMATION – US FUTURES MARKETS....40
EXECUTION TIME BRACKET REPORTING FOR US FUTURES MARKETS....40
EXAMPLE NEW ORDER – SINGLE FOR LISTED FUTURES MARKET 41
EXAMPLE NEW ORDER – SINGLE FOR LISTED OPTIONS MARKET....45
EXAMPLE NEW ORDER - MULTILEG FOR LISTED FUTURES MARKET (BUTTERFLY STRATEGY)....55
EXAMPLE MULTLILEG EXECUTION REPORT FOR LISTED FUTURES MARKET....62
Multlileg Execution Report Example for Futures Markets....*62*
OPTIONS BACK OFFICE PROCESSING....69
Background....*69*
Position Maintenance Report....*69*
Position Report....*70*
Trade Capture Report Ack....*70*
Trade Capture Report....*70*
Security Definition....*70*
Security List....*70*
Parties component block *70*
Contrary Intention Report....*71*
Security Definition Update Report *71*
Security List Update Report *71*
FIA TRADE IDENTIFICATION STANDARDS....71
Background....*71*
Trade Identification Fields....*71*
Additional Identifier Definitions....*72*
Trade Identification Usage Table....*76*
COLLATERAL MESSAGES FOR MARGIN MANAGEMENT....78
Background....*78*
Business Workflow....*78*
Message flow with a clearinghouse....*78*
Use of Instrument and UnderlyingInstrument component blocks *80*
Marginable vs. Valued Collateral *80*
COVERED SPREADS AND OTHER USER DEFINED SPREADS USING SECURITY DEFINITION MESSAGES 80
Usage examples....*81*
PRODUCT REFERENCE USAGE 83
Introduction....*83*
Business Workflow....*83*
Product Reference Model....*85*
Market Segment and Venue....*91*

© Copyright, 2007-2008, FIX Protocol, Limited

Message Flows *91*
Usage exmples *92*

PRODUCT: EQUITIES **97**

STEP-OUTS AND GIVE-UPS 97
CFDS 98
CFD with cash equity hedge executed by same broker as writing the CFD *98*
CFD with cash equity hedge executed by different broker from that writing the CFD *98*
COMMISSION SHARING ARRANGEMENTS 100
Soft Dollars *100*
Directed Brokerage *100*
MULTI-DAY AVERAGE PRICING 101
Introduction *101*
Flow Summary *101*
Example Warehouse Flows *102*
Decision Flows *106*
REGULATION SHO - SHORT-SELL SECURITY LOCATE 108
STRATEGY PARAMETERS FOR ALGORITHMIC TRADING 108
REGULATION NMS 109
Background *109*
Order Protection Rule Compliance *109*
Sub-penny Rule Compliance *112*
OATS PHASE 3 REQUIREMENTS 113
Background *113*
Meeting OATS 3 Requirements using FIX *113*
TrdRegTimestamp Usage Example for OATS 3 ..... *114*
EXTERNAL ORDER ROUTING CONTROL 117

PRODUCT: FIXED INCOME (FI) **118**

INTRODUCTION 118
MESSAGE DIALOG 118
Indication of Interest (Offerings) *119*
Negotiated Trade /Inquiry/Bid or Offer Request *120*
Out-of-Band Negotiated Order *123*
Allocation Instructions *125*
Post Trade Reporting to a 3rd Party or Virtual Matching Utility *129*
MESSAGE USAGE DETAILS 131
General Usage Rules *131*
Indication Of Interest *131*
Quote Request *131*
Quote Response *132*
Quote *132*
New Order - Single *132*
New Order - Multileg *132*
Execution Report *133*
Allocation Instruction *133*
Allocation Report *133*
Trade Capture Report *134*
Instrument component block *134*
OrderQtyData component block *134*
REPURCHASE AGREEMENTS (REPO) AND COLLATERAL MANAGEMENT 134
Repo Terminology *134*
Collateral Management *141*
IDENTIFYING EURO ISSUERS 147
Euro CountryOfIssue Codes: *147*

© Copyright, 2007-2008, FIX Protocol, Limited

Euro Issuer Values: *147*
EXAMPLE USAGE OF FI SPECIFIC COMPONENT BLOCKS 150
Example usage of BenchmarkCurve fields *150*
Example usage of Stipulation fields *151*

PRODUCT: FOREIGN EXCHANGE 152

INTRODUCTION 152
MESSAGE DIALOG 152
Price Discovery *152*
General Order and Execution Handling *159*
FX Settlement Obligation *162*
USAGE NOTES 164
General Usage Notes *164*
Quote Request *165*
Quote Response *166*
Quote *166*
Quote Request Reject *167*
Quote Cancel *167*
Market Data Request *168*
Market Data Snapshot/Full Refresh *168*
Market Data Incremental Refresh *169*
Market Data Request Reject *170*
New Order - Single *170*
New Order - Multileg *171*
Execution Report *172*
FX OTC Spot Option *173*
SettlDate and SettlType Required Usage Exception *175*
MESSAGE SAMPLES 175
Quote Request for FX Swap using NoLegs repeating group *176*
Quote for FX Swap using NoLegs repeating group *176*
Single Bank Market Data Request *178*
"Exchange" Market Data Request *179*
FX Swap Multi-legged Order *181*
Execution Report for FX Swap Multi-legged Order *182*
Settlement Obligation Report *183*

USER GROUP: EXCHANGES AND MARKETS 186

INTRODUCTION 186
ORDER STATE CHANGE MATRICES FOR EXCHANGES 186
A Vanilla *189*
I TimeInForce *192*
CONTINUOUS MARKET MAKER QUOTING 193
Quote Identifiers *193*
Quote Acknowledgement and Status *195*
Reporting a Mass Cancel *197*
Quote Cancel Scope *198*
Workflows *198*
QUOTE NEGOTIATION 215
Introduction *215*
Usage Notes *215*
QUOTE NEGOTIATION SCENARIOS 217
Public Tradable Quote *217*
Private Tradeable Quote - Three-Party Matching Model *217*
MULTILEG ORDERS 223
User-defined, Non-securitised Strategies *223*

© Copyright, 2007-2008, FIX Protocol, Limited

Multileg Price Method *225*
Delta Neutral Multileg Orders *225*
MARKET DATA 226
Books View Complexities *226*
Varying Book Depths *232*
Trade Statistics *233*
APPLICATION SEQUENCING AND RECOVERY 233
Business Workflow *233*
Business Cases *234*
Application Indentification *234*
Application Sequencing integration *234*
Interaction with Session Layer *235*
Application sequencing gap detection *236*
Use with Business-level Request Messages *237*

© Copyright, 2007-2008, FIX Protocol, Limited

VOLUME INTRODUCTION

Volume 7 of the FIX Protocol Specification aims to provide the FIX user community with notes and guidelines on the usage of FIX for specific areas, namely by security product or by user groups. Sections that are security product related has a section title that begins with "PRODUCT" while user group sections begin with "USER GROUP". A product is the broad security categorisation. A user group represents a particular type of user, for example exchanges, investment managers.

The "PRODUCT" sections are usage notes specific to a product. The "USER GROUP" sections are ussge notes specific to that type of user - this mostly covers FIX message flows and interpretation of key fields within FIX messages used by the user group.

© Copyright, 2007-2008, FIX Protocol, Limited

PRODUCT: COLLECTIVE INVESTMENT VEHICLES (CIV)

Overview and Scope

This Appendix summarises how FIX messages can be used to support order initiation / confirmation and to issue settlement / Registration Instructions for open-ended Collective Investment Vehicles ("CIVs") – known variously as Mutual Funds, Unit Trusts, Managed Investments. Open Ended Investment Companies (OEICs), Undertaking for Collective Investment in Transferable Securities (UCITs) etc.

Note that the FIX messages for CIV do not address Exchange Traded Funds, closed funds such as Investment Trusts or other scenarios where CIVs are traded as if they were equities.

Market environment

Units in funds are typically sold to Retail Investors on the recommendation of an Intermediary advisor (whose firm may not be authorised to hold client assets or settle transactions), or purchased at the Investors' initiative via a broker or funds supermarket (which may outsource settlement to a third party) or purchased by the Investor directly from the fund manager (who again may outsource fund administration to a third party).

Retail intermediaries (eg. Intermediary advisors) who don't hold client funds or settle transactions are rewarded by commission from the fund manager out of charges collected from the Investor. Commission and charges may be paid at the time of investment ("front-end load funds") and/or during the life of the investment ("no-load funds"). The latter may be called "renewal" or "trail" commission, and is typically paid directly to the intermediary at the end of each period.

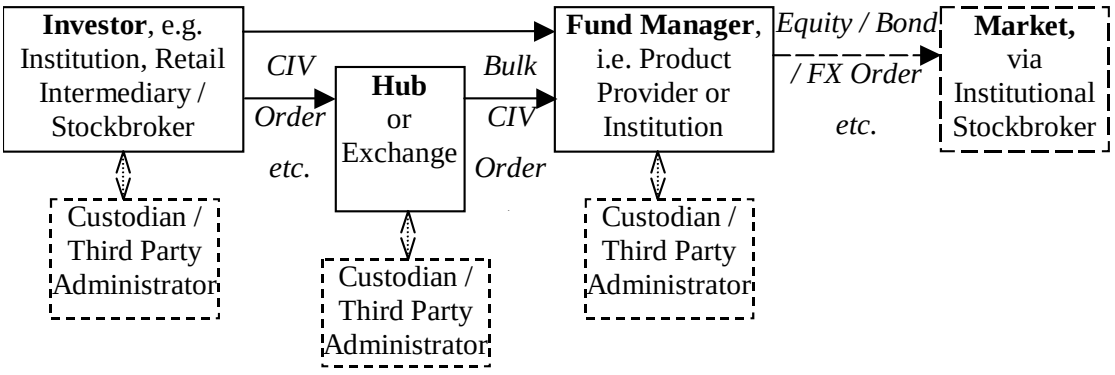
Intermediaries such as brokers and funds supermarkets may charge their own commission etc. directly to the Investor and instruct the fund manager not to deduct commission from the sum invested.

Institutional Investors typically purchase funds directly from the fund manager and no commission is payable.

In some regulatory environments the fund manager is responsible for making compliance and money laundering checks before a CIV order is executed, hence for new investors full details must be supplied with the order.

In some markets Hubs, Exchanges or Funds Supermarkets provide messaging, order matching/crossing, clearing and settlement services between Intermediaries/brokers, Fund managers etc.

FIX messages may be used between any of the participants. The fund manager may also use FIX messages to buy and sell fund assets with other participants in the relevant market(s) (eg. Equities):



Note that in a CIV scenario brokers, intermediaries etc. may be on the "buy side" and institutions may be on the "sell" side, i.e. a reversal of the situation in equity/fixed interest/FX transactions.

© Copyright, 2007-2008, FIX Protocol, Limited

CIV Security Type Identification

A Collective Investment Vehicle security type is designated by a CFICode field (ISO 10962 standards-based) value which starts with "EU". Note that if the Product field is specified, the value should be set to "Equity" to correspond with the "E" in the CFICode "EU" prefixed value, as presently defined. **See "Volume 6 – Appendix 6D" for CFICode details.**

Types of CIV FIX Messages

The FIX messages specifically supporting CIV trades are:

- "New Order – Single" – used to specify the buy or sell of a CIV fund. The message includes the ability to specify percentage of a holding to be sold, whether or not the order can be crossed or matched, compliance/money laundering status, commission instructions, etc. The New Order – Single comprises the major details:
    - Intermediary & Client Identification Information
    - Commission
    - Order Quantity
    - Registration and Reconciliation details
- "New Order – List" – used for an Investor to initiate exchanges or switches between CIV funds, or by a broker or Hub to place a bulk buy or sell order for several funds. New order List comprises one or more "New Order – Singles"
- Order Cancel Request – used for an Investor, Broker or Hub to request cancellation of an outstanding order
- Order Cancel Reject – used for a fund manager to reject Cancellation of an order
- Order Status Request – used for an Investor, Broker or Hub to request the status of an order
- "Settlement" – used to transmit Investors' payment details to the fund manager where the Intermediary does not settle trades
- "Registration Instructions" and "Registration response" – used to transmit Investors' registration details to the Fund manager, allow compliance checks and opening of the correct type of account. This may be sent before or after corresponding New Order messages. The Registration Instructions message type comprises the major details:
    - RegistrationID
    - OrderLink Fields
    - Registration Classification
    - Member Registration
    - Distribution Details
- "Execution Report" – used to transmit details of Unit price basis, charges, commission etc. to the Investor and Intermediary

Allocation messages are not required for CIV trading with Fund managers, but other FIX messages are unchanged and can be used as required, e.g. Market Data, Security Status Request, Quote, Order Status, Order Cancel / Replace, Don't Know, Business Reject etc.

(See ***CIV Examples 1 – 7*** below for examples of the use of these message types.)

© Copyright, 2007-2008, FIX Protocol, Limited

Order Quantities

Income on units may be credited as additional units on the Investor's account with the Fund manager, leading to uncertainty about the exact number of units when a holding is to be sold. Similarly when an exchange or switch is requested the cash value of investments realised and to be re-invested is not known. Hence it can be more convenient for Unit quantities to be expressed as a percentage of total holding, e.g. sell 50% or 100% of the existing holding, and reinvest 50% of the cash proceeds in Fund A, 25% in Fund B and 25% in Fund C.

"Percentage" amounts are indicated in the OrderPercent field.

Where an order is for investment of a money amount (CashOrderQty) or percentage (OrderPercent) the Intermediary may request that the resultant quantity is rounded up or down to a specific fraction or multiple of units by setting RoundingDirection and RoundingModulus.

(See ***CIV Example 13*** below for an example of the use of OrderPercent & Rounding to specify order quantity.)

Intermediary identification

Where messages are sent to or from a Fund manager via a Hub or Funds Supermarket on behalf of the Intermediary the IntroBroker field may be used to identify the Intermediary who is interfacing with the Investor.

This information is used by the Fund manager used to validate the Investor / Intermediary relationship on his records and to credit Commission to the correct Intermediary.

Investor details

If an Intermediary places a CIV Order for a new Investor (to the Fund manager) then the Registration instructions message can be used to transmit the details as required by the Fund manager:

- RegistAcctType – identifying which of the fund manager's account types should be opened
- TaxExemptType – identifying which of the (nationally defined) tax-exempt accounts or "plans" is required
- OwnershipType – indicates relationship between owners where there is more than one, e.g. tenants in common (i.e. equal interests), joint tenants with rights of survivorship.
- RegistDtls & RegistEmail – name and address into which purchases for this Investor should be registered, plus e-mail address where applicable.
- InvestorCountryOfResidence – identifying the country of residence of the investor, e.g. for compliance and/or tax purposes
- OwnerType – identifying whether the registered investor is an individual, corporation, nominee/street name, trustee etc. (This information may be required for regulatory purposes and/or to indicate which format of Registration name and address information is required)
- InvestorID and InvestorIDSource – containing identifiers issued by official organisations such as tax authorities, company registrar, regulators or national numbering agencies, together with an identifier for the source of the identifier
- MailingDtls – the name and address to which general correspondence should be sent (if different from the Registration address), semi-annual reports, marketing literature.
- MailingInst – e.g. instructions indicating what the mailing address is to be used for, whether marketing literature is acceptable etc.

(See ***CIV Examples 15-16*** below for examples of the use of registration instruction for new investors, accounts etc.)

© Copyright, 2007-2008, FIX Protocol, Limited

Having received this information the Fund manager responds with a Registration Instructions Response– which in addition to the RegistID of the Registration request should also contain the Account and/or ClientIDs allocated to the Investor.

(See ***CIV Examples 3, 4 & 6*** below for examples of the use of Registration instruction response message.)

Investor identification

A Fund manager may allocate an Account id and/or Client id to each Investor – depending on the architecture of his account database. These can be returned on the Registration status or Execution report message or by some other means (e.g. printed confirm or contract note), and should be quoted on subsequent New Order etc. messages.

(See ***CIV Examples 8-10*** below for examples of the use of identification fields for new and existing investors, accounts etc.)

New Investor -> New Order -> Registration instruction

Registration instruction messages can be sent before, after or both before and after a related New Order:

- before the New Order, e.g. to give details of a new investor / account (with name & address etc.). The RegistID specified on this Registration message must also be quoted on the subsequent New Order.
- after the New Order e.g. to give distribution payment details or to override previous Registration instructions for that specific New Order. This message should quote ClOrdID from the New Order (and Account and ClientID if available), but not the RegistID.

The Fund manager will respond to each Registration instruction with one or more Registration status messages, indicating whether the details are:

- Accepted – where possible including the Account and ClientID if these have been allocated by the Fund manager
- Rejected – in which case the RegistRejReasonCode and RegistRejReasonText fields should be populated to indicate the reason for rejection
- Held – e.g. pending receipt of the New Order or for later batch or manual processing, following which an “accepted” or “rejected” Registration status message will be sent

Note that the Designation field is available on the New Order message to provide supplementary registration information.

(See ***CIV Examples 6 & 14-16*** for examples of registration instructions and the designation field.)

Fund & Unit Identification

Many Funds offer several classes of units, e.g. front-end, back-end or no-load; income or accumulation units etc. In some tax regimes Fund managers are required to differentiate between units purchased before and after the most recent distribution. In markets where ticker symbols are allocated to unit types these are entered in the Symbol field; where tickers are not available an alternative identification such as ISIN is entered in the (mandatory) Symbol field and also the (optional) SecurityID field, with the code type in the SecurityIDSource field.

The Issuer and SecurityDesc fields may also be used to further confirm the Fund and Unit type required.

© Copyright, 2007-2008, FIX Protocol, Limited

Note that the Fund managers or regulators may impose restrictions on the Funds in an order, e.g. they must be available to the type of Investor, Account or Tax Exemption, or (for an exchange/switch) they may all have to be issued by the same Fund manager.

Order details - single

Order details for a CIV Order typically include:

- Side – "buy" (sometimes known as create, although creation may not actually be involved) or "sell" (sometimes known as a cancel, although cancellation may not actually be involved) - where "buy" or "sell" order can be matched or crossed by an intermediary, funds supermarket, broker/dealer etc. or forwarded to the fund manager. On the other hand a "subscribe" or "redeem" order must be forwarded to the fund manager, e.g. where the originator requires specific tax treatment and/or dealing charges.
- OrdType – Previous Fund Valuation Point (Historic pricing) or Next Fund Valuation Point –(Forward pricing)
- Order quantity expressed as one of:
    - OrderQty – number of units,
    - CashOrdQty– cash amount to be invested, or
    - OrderPercent – percentage of existing holding (for a sell) or percentage of available cash amount to be invested (for an exchange / switch)
- RoundDirection & RoundModulus – for cash amount or percentage, allows the investor or intermediary to request rounding up or down to the nearest 5, 10, 100 etc. or fractional units
- Currency & ForexReqd – e.g. for an off-shore fund settled in domestic currency
- Designation – supplementary registration information specific to this Order

Order details - list

A CIV "New Order – List" would typically be issued:

- by a retail intermediary to initiate an "exchange" or "switch" between funds on behalf of a single Investor
- by a broker, funds supermarket or hub/exchange to initiate "bulk buy" or "bulk sell" order of funds held for the account of several investors

For an exchange/switch:

- the ListNoOrds and ListSeqNo fields determine the order in which the deals are to be executed
- the ListExecInstType determines how the Order quantities and Settlement amount are to be calculated (i.e. sell-driven, buy-driven with additional cash available, buy-driven without additional cash)

For a bulk buy / bulk sell the Designation field can be used to supply supplementary registration information for each order line, to maintain segregation between the holdings for individual clients.

(See ***CIV Examples 13 & 14*** below for an example of the use of New Order – List.)

Commission Instructions

The Intermediary can indicate specific commission requirements using:

© Copyright, 2007-2008, FIX Protocol, Limited

- Commission & CommType – e.g. a specific commission rate or a waiver of the standard commission rate for the fund, the saving on standard commission being credited as for additional units or as a cash discount
- CommCurrency – to specify that commission on an overseas or offshore fund should be paid in domestic currency
- FundBasedRenewalWaived – to indicate whether or not the Intermediary accepts renewal/trail commission

Compliance

Depending on terms of business and the regulatory environment either or both of the Intermediary and Fund manager may be required to support money laundering status checking and/or right-to-cancel. The New Order message supports these with:

- MoneyLaundering – indicating whether or not checks are required and have already been carried out by the Intermediary
- CancellationRights - indicating whether or not a "right-to-cancel" applies

Settlement instructions

For CIV Orders retail settlement instructions may be transmitted using Settlement instruction features including:

- SettlInstMode – indicating that settlement instructions relate to a specific (retail) Order
- SettlInstSource –indicating the Investor as the source of settlement instructions
- PaymentMethod & SettCurrency – indicating cheque, bank transfer, payment card, cash account at depository etc.
- CardHolderName, CardNumber, CardStartDate, CardExpDate, CardIssNo, PaymentDate and PaymentRemitterID – details required for cash settlement by payment card
- SettlBrkrCode, SettlDepositoryCode – for cash settlement via central depositories
- CashSettlAgentName, CashSettlAgentCode, CashSettlAgentAcctNum, CashSettlAgentAcctName - for cash settlement by bank transfer
- PaymentRef – cross-reference or narrative information for bank transfers etc. to appear on bank statements, SWIFT MT950's etc. to assist reconciliation

Distribution instructions

The Registration instruction message can also carry Distribution instructions, including:

- NoDistribDetls & DistribSeqNo – the number of beneficiaries
- DistribPercent –the split of each distribution (by value) between several beneficiaries
- DistribPaymentMethod & CashDistribCurr – payment method and currency for a specific beneficiary
- CashDistribAgentName, AgentCode, AgentAcctName and AgentAcctNum – bank and account details for a specific beneficiary
- CashDistribPayRef - cross-reference or narrative information for bank statements

(See ***CIV Examples 15 & 16*** below for examples of the use of distribution instructions.)

© Copyright, 2007-2008, FIX Protocol, Limited

Unit Prices

Fund managers calculate a net asset value for each fund – typically at a fixed time each day, the "valuation point". They then quote either a single Unit price ("single pricing") or separate buying and selling prices ("dual pricing") – depending on the fund's constitution and regulatory environment.

Valuation point

The unit price applicable to a CIV trade depends on when the Order was received by the fund manager relative to a Valuation point, whether the Fund is normally dealt on a Historic or Forward basis, and possibly also on recent volatility on underlying fund assets and any specific instructions from the Investor.

Some of this information is indicated by fields on the New Order:

- TransactTime – the time at which the Investor placed the CIV Order directly, or at which Intermediary placed the Order on behalf of the Investor
- OrdType – whether Investor requires a Forward or (where available) a Historic price

Other times establishing the relevant valuation point are shown on the Execution Report:

- OrderBookedTime – the time at which the Fund manager provisionally accepted the order for execution (having completed any preliminaries, e.g. setting up an account, money laundering checks)
- ExecValuationPoint - the fund valuation point with respect to which a order was priced by the fund manager (may be before or after the OrderBookedTime).

Single pricing

The Unit price for single-priced funds is determined from the net asset value, based on the mid-price of the underlying assets of the fund, divided by the applicable number of units. For these funds ExecPriceType on the Execution Report should be set to "S" = Single.

The manager's Initial charge (if any) is then charged out separately. In addition a Dilution levy may be charged on large buy or sell transactions, e.g. to compensate for the difference between the mid- and buy/sell- price of the underlying investments. These charges can be notified on the Execution Report in the Contract amounts repeat group.

Dual pricing

For dual priced funds the manager calculates:

- Creation price – based on the "buy" price of the underlying assets (net of transaction taxes etc.)
- Cancellation price – based on the "sell" price of the underlying assets (net of transaction taxes etc.)

If the net cash flow is into the fund new units will be created:

- Offer or Buy price – will be no higher than the Creation price plus the manager's Initial charge
- Bid, Sell or Redemption price – will be the Offer price minus the manager's Dealing spread

If the net cash flow is out of the fund existing units will be cancelled:

- Bid, Sell or Redemption price – will be no lower than the Cancellation price
- Offer or Buy price – will be the Bid price plus the manager's Dealing spread, up to a limit of the Creation price plus the manager's Initial charge

The manager may sell to buyers units he has re-purchased from sellers (rather than cancelling and re-creating units), thus profiting from the Dealing spread.

The Initial charge covers any commission paid to Intermediaries as well as advertising, administration, dealing costs etc. It can be a money amount or percentage and may be waived on large investments, e.g. by institutional investors. Where the Initial charge is waived for a private investor an Exit charge (money amount or percentage) may be levied if an investment is sold within the first few years. (This is sometimes known as a Deferred contingent sales charge.) These charges can be notified on the Execution Report in the Contract amounts repeat group.

© Copyright, 2007-2008, FIX Protocol, Limited

The manager may offer an improved buying price by discounting the initial charge or reducing his dealing spread – the improved price is expressed as "Creation price plus" an amount or percentage, or "Offer price minus" an amount or percentage.

ExecPriceType and (where applicable) ExecPriceAdjustment on the Execution Report indicate how the actual buying or selling price was calculated from the fund valuation price(s).

Execution Reports

The Fund manager should send Execution Report messages to confirm receipt (OrdStatus="New") and execution (OrdStatus= "Filled" and/or "Calculated") of CIV Orders, plus other Order Status from the list below as agreed between the parties – individual Execution Reports being sent for each line of an New Order – List.

In markets where tax treatment and/or dealing charges depend on whether execution was by crossing / matching by an intermediary, or by subscription / redemption at the fund manager the LastMkt field should be used to indicate either the Exchange or 11 for an OTC trade, or omitted if execution was by the fund manager.

CIV-specific use of OrdStatus:

CIV orders to be executed by the fund manager do not use the TimeInForce field and only the following OrdStatus values are expected to be used:

***** This OrdStatus table lists CIV-specific values *****

Precedence	OrdStatus	Description
11	Pending Cancel	Order with an Order Cancel Request pending, used to confirm receipt of an Order Cancel Request. DOES NOT INDICATE THAT THE ORDER HAS BEEN CANCELED. (Where supported by the receving broker, intermediary, fund manager etc.)
10	Pending Replace	Order with an Order Cancel/Replace Request pending, used to confirm receipt of an Order Cancel/Replace Request. DOES NOT INDICATE THAT THE ORDER HAS BEEN REPLACED. (Where supported by receiving broker, intermediary, fund manager etc.)
8	Calculated	Order has been filled, settlement details, currency, commission, contract amounts etc. have been calculated and reported in this execution message
7	Filled	Order has been filled, execution valuation point, shares/unit quantity and price have been calculated and and reported in this execution message
4	Canceled	Canceled order without executions (where supported by receiving broker, intermediary, fund manager etc.).
2	New	Outstanding order which has not been executed. The OrderBookedTime field will be completed. For Forward priced orders or funds the order will be executed at the next Valuation Point. (This status may not be sent if the order can be executed immediately on a Historic pricing basis)

© Copyright, 2007-2008, FIX Protocol, Limited

2	Rejected	Order has been rejected by broker, intermediary or fund manager (for CIV orders). NOTE: An order can be rejected subsequent to order acknowledgment, i.e. an order can pass from New to Rejected status.
2	Pending New	Order has been received by broker's system but not yet accepted for execution. An execution message with this status will only be sent in response to a Status Request message. (Where supported by receiving broker, intermediary or fund manager etc.)

The CIV Fields included for each value of OrdStatus in Execution Report are listed below:

OrdStatus	CIV Fields included on Execution Report
Rejected	
Pending Cancel	ClOrdID, ListID & TransactTime – Intermediary's Order (and List) references and time of submission
Canceled	Other fields may be populated if available
Pending Replace	
Pending New	ClOrdID, ListID & TransactTime – Intermediary's Order (and List) references and time of submission All fields populated on the CIV Order (apart from Order fields not available in Execution Report)
New	Same as for "Pending New" plus: TranBkdTime – time at which the Fund manager accepted the CIV Order onto his books OrderId – order reference assigned by Fund manager (to each line in a New Order - List)
Filled	Same as for "New" plus: ExecID & DealTime – Fund manager's reference & Valuation point at which the Fund manager priced the CIV Order LastQty, LastPx & ExecPriceType – Unit quantity, price & basis of calculation of the price (e.g. Bid, Offer / Offer minus, Creation / Creation plus etc.)
Calculated	As for "Filled" plus: ContAmt, Type & Curr – type, currency and value of various contract amounts (Initial, Commission, Discount Exit, Dilution etc.)

© Copyright, 2007-2008, FIX Protocol, Limited

(See ***CIV Examples 1 – 7*** below for examples of the use of Execution Report messages.)

CIV EXAMPLES

The following examples illustrate how FIX messages can be used to process CIV fund orders and provide settlement and registration instructions to the fund manager.

NOTE that in the examples:

- "Buyside" refers to an institution or private investor investing in a CIV fund via broker, intermediary – or a hub and/or exchange transmitting messages to/from other buyside parties
- "Sellside" refers to a CIV fund manager or intermediary – or a hub and/or exchange transmitting messages to/from other sellside parties

CIV Example 1. Single order for a CIV fund for a known investor/nominee, to be dealt on a "historic" basis

A typical flow for an order for a CIV fund dealt on Historic price for an investor or nominee known to the fund manager – is as follows:

BUYSIDE			**SELLSIDE**
			Fund Valuation Point
	→	New Order-Single (IntroBroker, ClOrdID, Account & ClientID specified)	
	←	Execution Report (ExecType = "F") [Trade] (IntroBroker, ClOrdID, Account & ClientID echoed)	
			Commission/ Fee Calc
	←	Execution Report (ExecType = "B") [Calculated] (IntroBroker, ClOrdID, Account & ClientID echoed)	

CIV Example 2. Single order for a CIV fund for a known investor/nominee, to be dealt on a "forward" basis

A typical flow for an order for a CIV fund for an investor/nominee known to the fund manager that wishes to deal on a Forward price basis – is as follows:

BUYSIDE		**SELLSIDE**

© Copyright, 2007-2008, FIX Protocol, Limited

BUYSIDE			SELLSIDE
	→	New Order-Single (IntroBroker, ClOrdID, Account & ClientID specified) (OrdType="M") [Forward]	
	←	Execution Report (ExecType = “0” [New] (IntroBroker, ClOrdID, Account & ClientID echoed)	
			Fund Valuation Point
	←	Execution Report (ExecType = “F”) [Trade] (IntroBroker, ClOrdID, Account & ClientID echoed)	
			Commission/ Fee Calc
	←	Execution Report (ExecType = “B”) [Calculated] (IntroBroker, ClOrdID, Account & ClientID specified)	

CIV Example 3. Single order for a CIV fund for an investor/nominee not known to the fund manager - registration and settlement instructions after trade

A typical flow for an order for a CIV fund for an investor/nominee not known to the fund manager where the fund manager does not require settlement or registration instructions in advance – is as follows:

BUYSIDE			SELLSIDE
	→	New Order – Single (IntroBroker & ClOrdID specified, Account, ClientID & RegistID not specified)	
	←	Execution Report (ExecType = “0” [New] (IntroBroker & ClOrdID echoed)	
			Fund Valuation Point
	←	Execution Report (ExecType = “F”) [Trade] (IntroBroker & ClOrdID echoed)	
			Commission/ Fee Calc
	←	Execution Report (ExecType = “B”) [Calculated] (IntroBroker & ClOrdID echoed)	
	←	Registration Instruction Response (RegistStatus = “N”) [Reminder] (IntroBroker & ClOrdID echoed)	
	→	Settlement Instruction (SettInstTransType = “N”) [New] (SettlInstMode=”4”) [Specific Order] (IntroBroker & ClOrdID specified)	

© Copyright, 2007-2008, FIX Protocol, Limited

BUYSIDE			SELLSIDE
	→	Registration Instruction (RegistTransType = "0") [New] (IntroBroker, ClOrdID & RegistID specified)	
			Validate Registration Instruction
	←	Registration Instruction Response (RegistStatus = "A") [Accepted] (IntroBroker, ClOrdID & RegistID echoed, Account and/or ClientID returned)	

CIV Example 4. Single order for a CIV fund for an investor/nominee not known to the fund manager - registration and settlement instructions required before trade

A typical flow for an order for a CIV fund for an investor/nominee not known to the fund manager where the fund manager requires settlement and registration instructions in advance – is as follows:

BUYSIDE			SELLSIDE
	→	Registration Instruction (RegistTransType = "0") [New] (RegistID, IntroBroker & ClOrdID specified)	
	←	Registration Instruction Response (RegistStatus = "H") [Held] (IntroBroker, ClOrdID & RegistID echoed, Account and/or ClientID not returned)	
			Validate Registration Instruction
	←	Registration Instruction Response (RegistStatus = "A") [Accepted]	
	→	New Order – Single (IntroBroker & ClOrdID specified, Account, ClientID & RegistID not specified)	
	←	Execution Report (ExecType = "A" [Pending New]	
	→	Settlement Instruction (SettInstTransType = "A") [New] (SettlInstMode="4") [Specific Order] (IntroBroker & ClOrdID specified)	
			Validate Settlement Instruction
	←	Execution Report (ExecType = "0") [New]	
			Fund Valuation Point
	←	Execution Report (ExecType = "F") [Trade]	

© Copyright, 2007-2008, FIX Protocol, Limited

BUYSIDE			SELLSIDE
			Commission/ Fee Calc
	←	Execution Report (ExecType = "B") [Calculated]	

CIV Example 5. Single order for a CIV fund for a known investor/nominee – order modified before execution

A possible flow for an order for a CIV fund for an investor/nominee known to the fund manager, on which the CashOrdQty is modified before execution – is as follows:

BUYSIDE			SELLSIDE
	→	New Order-Single (IntroBroker, ClOrdID, Account & ClientID specified) CashOrdQty = "6,000"	
	←	Execution Report (ExecType = "0" [New] (IntroBroker, ClOrdID, Account & ClientID echoed)	
	→	Order Cancel/Replace Request (IntroBroker, ClOrdID, Account & ClientID specified) CashOrdQty = "7,000"	
	←	Execution Report (ExecType = "5" [Replaced] (IntroBroker, ClOrdID, Account & ClientID echoed)	
			Fund Valuation Point
	←	Execution Report (ExecType = "F") [Trade] (IntroBroker, ClOrdID, Account & ClientID echoed)	
			Commission/ Fee Calc
	←	Execution Report (ExecType = "B") [Calculated] (IntroBroker, ClOrdID, Account & ClientID specified)	

CIV Example 6. Single order for a CIV fund for a new investor/nominee to the fund manager - registration and settlement instructions rejected, then modified & accepted

A possible flow for an order for a CIV fund for an investor/nominee not already known to the fund manager where settlement and registration instructions are supplied, rejected and then corrected after the trade – is as follows:

BUYSIDE		SELLSIDE

© Copyright, 2007-2008, FIX Protocol, Limited

BUYSIDE			SELLSIDE
	→	New Order – Single (IntroBroker & ClOrdID specified, Account, ClientID & RegistID not specified)	
			Fund Valuation Point Commission/ Fee Calc
	←	Execution Report (ExecType = "B") [Calculated] (IntroBroker & ClOrdID echoed)	
	→	Settlement Instruction (SettInstTransType = "N") [New] (SettlInstMode="4") [Specific Order] (IntroBroker & ClOrdID specified)	
	→	Registration Instruction (RegistTransType = "0") [New] (IntroBroker, ClOrdID & RegistID specified)	
			Validate Registration Instruction
	←	Registration Instruction Response (RegistStatus = "H") [Held] (IntroBroker, ClOrdID & RegistID echoed, Account and/or ClientID not returned)	
	←	Registration Instruction Response (RegistStatus = "R") [Rejected] (IntroBroker, ClOrdID & RegistID echoed, Account and/or ClientID not returned)	
	→	Registration Instruction (RegistTransType = "2") [Replace] (IntroBroker, ClOrdID & RegistID specified)	
			Validate Registration Instruction
	←	Registration Instruction Response (RegistStatus = "A") [Accepted] (IntroBroker, ClOrdID echoed, Account and/or ClientID returned)	
	→	Settlement Instruction (SettInstTransType = "R") [Replace] (SettlInstMode="4") [Specific Order] (IntroBroker & ClOrdID specified)	

CIV Example 7. Exchange/switch order between several CIV funds from a single fund manager or via a funds supermarket

A typical flow for an order for a CIV fund for an investor wishing to switch funds between funds from a single fund manager or via a funds supermarket that covers all funds – is as follows:

© Copyright, 2007-2008, FIX Protocol, Limited

BUYSIDE			SELLSIDE
	→	New Order-List (ListId & ListExecInstType specified, e.g. ListExecInstType="3" [Exch/switch - Sell Driven] For each component of exchange/switch: (IntroBroker, ClOrdID, ClientID, Account, Symbol/SecurityId, OrderPercent, Side)	
	← ← ←	For each component of exchange/switch: Execution Report (ExecType = "0" [New] (IntroBroker, ClOrdID, Account & ClientID echoed)	
			Fund Valuation Point
	← ← ←	For each component of exchange/switch: Execution Report (ExecType = "F" [Trade] (IntroBroker, ClOrdID, Account & ClientID echoed)	
			Commission/ Fee Calc
	← ← ←	For each component of exchange/switch: Execution Report (ExecType = "B" [Calculated] (IntroBroker, ClOrdID, Account & ClientID echoed)	

Identifier examples – existing investor & account

CIV Example 8. Order for CIV fund by new or existing investor, routed via a client money/asset holding broker or funds supermarket to fund manager

Typical usage of fields on Order and/or Post-Trade messages would be as follows:

Tag	*Field Name*		*Contents*
453	NoPartyIDs		2
→	448	PartyID	An identifier for the broker or funds supermarket which is recognized by the fund manager
→	447	PartyIDSource	Indicates source of Party identifier used in preceding field
→	452	PartyRole	6 ["Introducing Firm"]
→	448	PartyID	An identifier for the broker or funds supermarket's nominee/custodian company which is recognized by the fund manager

© Copyright, 2007-2008, FIX Protocol, Limited

Tag	*Field Name*		*Contents*
→	447	PartyIDSource	Indicates source of Party identifier used in preceding field, e.g. the Fund manager
→	452	PartyRole	9 [“Fund manager Client ID”]
→	523	PartySubID	An optional sub-identifier for the broker or funds supermarket’s nominee/custodian company which is recognized by the fund manager
11	ClOrdID		Assigned by broker or funds supermarket

CIV Example 9. Order for CIV fund by an institutional investor, routed via a broker to a fund manager – possibly via a hub/exchange

Typical usage of fields on Order and/or Post-Trade messages would be as follows:

Tag	*Field Name*		*Contents*
453	NoPartyIDs		3
→	448	PartyID	An identifier for the broker closest to the investing institution which is recognized by the fund manager
→	447	PartyIDSource	Indicates source of Party identifier used in preceding field
→	452	PartyRole	6 [“Introducing Firm”]
→	448	PartyID	An identifier for hub/exchange (where used) which is recognized by the fund manager
→	447	PartyIDSource	Indicates source of Party identifier used in preceding field
→	452	PartyRole	1 [“Executing Firm”]
→	448	PartyID	An identifier for the investing institution which is recognized by the fund manager
→	447	PartyIDSource	Indicates source of Party identifier used in preceding field, e.g. the Fund manager
→	452	PartyRole	9 [“Fund manager Client ID”]
→	523	PartySubID	An optional sub-identifier for the investor which is recognized by the fund manager
11	ClOrdID		Assigned by investing institution

Identifier examples – new investor and/or account

CIV Example 10. Order for CIV fund by new investor via non-client money/asset holding intermediary to fund manager

© Copyright, 2007-2008, FIX Protocol, Limited

Typical usage of fields on Order and/or Post-Trade messages would be as follows:

Tag	*Field Name*		*Contents*
453	NoPartyIDs		2
→	448	PartyID	An identifier for the broker closest to the investor which is recognized by the fund manager
→	447	PartyIDSource	Indicates source of Party identifier used in preceding field
→	452	PartyRole	6 [“Introducing Firm”]
→	448	PartyID	Not present on the “New Order” message, only on Execution Report(s). An identifier for the investor which is assigned by the fund manager, e.g. after processing a Registration Instruction.
→	447	PartyIDSource	Indicates source of Party identifier used in preceding field, e.g. the Fund manager
→	452	PartyRole	9 [“Fund manager Client ID”]
→	523	PartySubID	Not present on the “New Order” message, only on Execution Report(s). An optional sub-identifier for the investor which is assigned by the fund manager, e.g. after processing a Registration Instruction.
11	ClOrdID		Assigned by intermediary
493	RegistAcctType		An identifier for the type of account required which is recognised by the fund manager
495	TaxAdvantageType		An identifier for the type of tax advantaged account required
492	PaymentMethod		Entered by intermediary (together with Investor’s bank/card details) to show how investor will settle cash with the fund manager

CIV Example 11. Order for CIV fund by new investor, routed via non-client money/asset holding intermediary via a non-aggregating hub/exchange to fund manager

Typical usage of fields on Order and/or Post-Trade messages would be as follows:

Tag	*Field Name*		*Contents*
453	NoPartyIDs		3
→	448	PartyID	An identifier for the broker closest to the investor which is recognized by the fund manager
→	447	PartyIDSource	Indicates source of Party identifier used in preceding field
→	452	PartyRole	6 [“Introducing Firm”]
→	448	PartyID	An identifier for hub/exchange (where used) which is recognized by the fund manager

© Copyright, 2007-2008, FIX Protocol, Limited

Tag	*Field Name*		*Contents*
→	447	PartyIDSource	Indicates source of Party identifier used in preceding field
→	452	PartyRole	1 [“Executing Firm”]
→	448	PartyID	Not present on the “New Order” message, only on Execution Report(s). An identifier for the investor which is assigned by the fund manager, e.g. after processing a Registration Instruction.
→	447	PartyIDSource	Indicates source of Party identifier used in preceding field, e.g. the Fund manager
→	452	PartyRole	9 [“Fund manager Client ID”]
→	523	PartySubID	Not present on the “New Order” message, only on Execution Report(s). An optional sub-identifier for the investor which is assigned by the fund manager, e.g. after processing a Registration Instruction.
11	ClOrdID		Assigned by broker

CIV Example 12. Order for CIV fund by new investor routed via intermediary to a funds supermarket – which places bulk/net orders to the fund manager

Typical usage of fields on Order and/or Post-Trade messages between intermediary and funds supermarket would be as follows:

Tag	*Field Name*		*Contents*
11	ClOrdID		Assigned by intermediary
453	NoPartyIDs		2
→	448	PartyID	An identifier for the intermediary closest to the investor which is recognized by the fund manager
→	447	PartyIDSource	Indicates source of Party identifier used in preceding field
→	452	PartyRole	6 [“Introducing Firm”]
→	448	PartyID	Not present on the “New Order” message, only on Execution Report(s). An identifier for the investor which is assigned by the funds supermarket, e.g. after processing a Registration Instruction.
→	447	PartyIDSource	Indicates source of Party identifier used in preceding field, e.g. the Fund manager
→	452	PartyRole	9 [“Fund manager Client ID”]
→	523	PartySubID	Not present on the “New Order” message, only on Execution Report(s). An optional sub-identifier for the investor which is assigned by the funds supermarket, e.g. after processing a Registration Instruction.

© Copyright, 2007-2008, FIX Protocol, Limited

Typical usage of fields on Order and/or Post-Trade messages between funds supermarket and fund manager for bulk/net orders would be as follows:

Tag	*Field Name*		*Contents*
11	ClOrdID		Assigned by fund supermarket
453	NoPartyIDs		2
→	448	PartyID	An identifier for funds supermarket which is recognized by the fund manager
→	447	PartyIDSource	Indicates source of Party identifier used in preceding field
→	452	PartyRole	1 [“Executing Firm”]
→	448	PartyID	An identifier for the funds supermarket’s nominee/custodian company which is recognized by the fund manager.
→	447	PartyIDSource	Indicates source of Party identifier used in preceding field, e.g. the Fund manager
→	452	PartyRole	9 [“Fund manager Client ID”]
→	523	PartySubID	An optional sub-identifier for the funds supermarket’s nominee/custodian company which is recognized by the fund manager.

Quantity example

CIV Example 13. Exchange/switch order quantities – OrderPercent, Rounding, Sell Driven

Typical use of OrderPercent and Rounding fields on Order and Execution Report messages to and from fund manager or funds supermarket would be as follows:

Investor’s holdings before exchange/switch New Order – List are:

Symbol/SecurityId	**Quantity held**
Fund A	5281
Fund B	2296
Fund C	1833

Exchange/switch order details on the New Order – List are:

Symbol/SecurityId	**Side**	**OrderQty**	**CashOrderQty**	**OrderPercent**

© Copyright, 2007-2008, FIX Protocol, Limited

Fund A	Sell	1281		
Fund B	Sell		£2,000	
Fund C	Sell			100%
Fund X	Buy			20%
Fund Y	Buy			30%
Fund Z	Buy			50%

with : RoundingDirection = 1 [Down]

RoundingModulus = 1

After the Fund Valuation Point the quantities and cash amounts (assuming no commissions, initial or exit charges) are reported on "calculated" Execution Reports are as follows:

Symbol/SecurityId	**Side**	**Price (AvePx)**	**CumQty**	**Cash value**
Fund A	Sell	£5.21	1281	£6,674
Fund B	Sell	£7.28	274	£1,995
Fund C	Sell	£3.27	1833	£5,994
Fund X	Buy	£8.72	336	-£2,930
Fund Y	Buy	£15.00	293	-£4,395
Fund Z	Buy	£1.00	7331	-£7,331

Settlement amount (ContAmtValue) = £6.72 (credit, i.e. excess cash will be paid to Investor)

CIV Example 14. CIV Bulk order – purchase of funds for multiple investors into a designated nominee

Typical use of New Order – List by a broker for purchase of funds for multiple investors into a designated nominee would be to specify ListExecInstType="1" [Immediate], with other fields as follows:

Tag	*Field Name*		*Contents*
11	ClOrdID		Assigned by broker to identify each component within New Order - List. ***As required for each componen.t***
→	448	PartyID	An identifier for the funds supermarket's nominee/custodian company which is recognized by the fund manager. ***Same for each component of order.***

© Copyright, 2007-2008, FIX Protocol, Limited

Tag	*Field Name*		*Contents*
→	447	PartyIDSource	Indicates source of Party identifier used in preceding field, e.g. the Fund manager
→	452	PartyRole	9 [“Fund manager Client ID”]
→	523	PartySubID	An optional sub-identifier for the funds supermarket’s nominee/custodian company which is recognized by the fund manager.
→	448	PartyID	An identifier for the intermediary closest to the investor which is recognized by the fund manager ***Same for each component of order.***
→	447	PartyIDSource	Indicates source of Party identifier used in preceding field
→	452	PartyRole	6 [“Introducing Firm”]
55/ 48	Symbol/SecurityId etc.		Identifier(s) for fund. ***As required for each component.***
54/ 38/ 152	Side/OrderQty/CashOrderQty		Buy/sell & quantity. ***As required for each component.***
513	RegistID		Assigned by broker to identify Registration Instruction for nominee company – if required. ***Same for each component of order.***
494	Designation		Specific registration (“sub-account”) for each component. ***As required for each component.***

plus other New Order – List fields as required.

CIV Example 15. Registration Instruction – Joint Investors

Typical use of the Registration instruction Joint Investors, e.g. husband & wife, with cash distribution split equally between them would be:

Tag	*Field Name*		*Value*
517	OwnershipType		J [“Joint Investors”]
413	NoRegistDtls		2
→	509	RegistDtls	John Smith Esq, 1 Acacia Avenue, Newtown, Countyshire
→	511	RegistEmail	johnsmith99@isp.com
→	522	OwnerType	1 [“Individual Investor”]
→	509	RegistDtls	Mrs Naomi Smith, 1 Acacia Avenue, Newtown, Countyshire
→	511	RegistEmail	Naomismith32@isp.com

© Copyright, 2007-2008, FIX Protocol, Limited

Tag	*Field Name*		*Value*
→	522	OwnerType	1 [“Individual Investor”]
510	NoDistribInsts		2
→	477	DistribPaymentMethod	8 [“Direct Credit”]
→	512	DistribPercentage	50
→	478	CashDistribCurr	GBP
→	498	CashDistribAgentName	Anytown Bank
→	499	CashDistribAgentCode	20-01-00
→	500	CashDistribAgentAcctNumber	23456789
→	501	CashDistribPayRef	Fund income
→	502	CashDistribAgentAcctName	Mr J & Mrs N Smith
→	477	DistribPaymentMethod	5 [“Cheque”]
→	512	DistribPercentage	50
→	478	CashDistribCurr	GBP
→	502	CashDistribAgentAcctName	Mrs Naomi Smith

© Copyright, 2007-2008, FIX Protocol, Limited

CIV Example 16 Registration Instruction – Tenants in Common,

Possible use of the Registration instruction for Tenants in Common, e.g. a club of private investors that reinvest all their income could be:

Tag	*Field Name*		*Contents*
517	OwnershipType		T [“Tenants in Common”]
413	NoRegistDtls		4
→	509	RegistDtls	Frank Jones, 2 South Drive, Anyport, Southshire
→	511	RegistEmail	***fjones@myisp.net***
→	509	RegistDtls	Sally Smith, 192 West Road, Anyport, Southshire
→	511	RegistEmail	ssmith@hotmail.com
→	509	RegistDtls	James Jordan, 88 Lime Tree Avenue, Lower Anyport, Southshire
→	511	RegistEmail	jamesj@mymail.co.uk
→	509	RegistDtls	Anita Robinson, 2 South Drive, Anyport, Southshire
510	NoDistribInsts		1
477	DistribPaymentMethod		12 [“Reinvest in Fund”]

© Copyright, 2007-2008, FIX Protocol, Limited

PRODUCT: LISTED DERIVATIVES (FUTURES & OPTIONS)

Use of CFICode to identify derivatives security

The CFICode (tag 461) is used to identify the type of instrument in FIX. The following is the recommended usage of the CFICode for futures and options. The CFICodes (ISO 10962) shall replace of SecurityType (tag 167) enumerations for futures – "FUT" and options – "OPT". The CFICode for options supports definition of Calls – "C" and Puts – "P" in the second position. The PutOrCall (tag 201) tag is replaced (made obsolete) in FIX 4.3 by the adoption of the CFICode (tag 461).

Single Leg Instruments

		FIX 4.2 Mapping Values	
CFICode[461]	**Description**	**SecurityType[167]**	**PutOrCall[201]**
OCXXXS	Standardized Call Option	OPT	1
OPXXXS	Standardized Put Option	OPT	0
FXXXS	Standardized Future	FUT	na
OCXFXS	Standardized Call Option on a Future	na[1]	1
OPXFXS	Standardized Put Option on a Future	na	0
FFICN	Nonstandard (flex) Financial Future on an index with cash delivery	FUT	na
FCEPN	Nonstandard (flex) Commodity Future on an extraction resource with physical delivery	FUT	na
FXXXN	Nonstandard (flex) future – contract type specified in symbology – not provided in CFICode	FUT	na
OCEFCN	Nonstandard (flex) call option on future with european style expiration and cash delivery	OPT	1
OPAFPN	Nonstandard (flex) put option on future with american style expiration and physical delivery	OPT	0
OPXSPN	Nonstandard (flex) put option on a stock with physical delivery (the expiration style is not specified – so is assumed to default to the market standard for flex options).	OPT	0
OCEICN	Nonstandard (flex) call option on an index with european style expiration and cash delivery	OPT	1

[1] A security type enumeration for an Option on a Future does not currently exist.

© Copyright, 2007-2008, FIX Protocol, Limited

Multileg Instrument Specification

The following use of SecurityType and CFICode are proposed for specifying multileg derivative instruments – such as options strategies or futures spreads.

SecurityType[167]	CFICode[461]	Description
MLEG	FMXXS	Multileg Instrument with futures contract legs CFICode refers to Future – Miscellaneous
MLEG	OMXXXN	Multileg Instrument with option contract legs CFICode refers to Option – Miscellaneous (This would include multileg instruments that include the underlying security).
MLEG	M	Multileg Instrument with legs made up of various types of securities (not primarily a futures or options multileg instrument that contains one or more derivative legs). CFICode refers to M-Miscellaneous

US Listed Options Order Capacity Values

The following are commonly used order capacity codes from the US listed options markets and how they map to FIX 4.3.

Common Listed Option Market Order Capacity Values	OrderCapacity (tag 528)	OrderRestrictions (tag 529)	Other
"B" any account of a broker/dealer, or any account in which a broker or dealer registered or required to be registered with the SEC pursuant to Section 15 under the Act has an interest. This represents any account that is not otherwise an account that falls into any of the below mentioned categories.	**Principal**		
"C" any account in which no member or non-member broker/dealer has an interest.	**Agency**		
"D" any account of a foreign broker/dealer.[2]	**Principal**	**6 - Foreign Entity**	
"F"any firm proprietary account which clears at the Options Clearing Corporation that is not a JBO account.	**Proprietary**		

[2] A foreign broker/dealer is defined as any person or entity that is registered, authorized, or licensed by a foreign governmental agency or foreign regulatory organization (or is required to be registered, authorized, or licensed) to perform the function of a broker or dealer in securities, or both. For purposes of this definition, a broker or dealer may also be a bank.

© Copyright, 2007-2008, FIX Protocol, Limited

Common Listed Option Market Order Capacity Values	OrderCapacity (tag 528)	OrderRestrictions (tag 529)	Other
"M" an account representing a CBOE market-maker.	**Proprietary**	**5-Acting As a specialist or market maker in the security**	
"N" Any options account of a market-maker or specialist of another options exchange ~~who is registered as a market-maker or specialist in the same class of options multiply listed at an away exchange.~~ Sometimes referred to as an order for a "MM or Specialist Away".	**Proprietary**	**5-Acting As a specialist or market maker in the security** **7 - External Market Participant**	
"Y" any options account of a Commodities Trader, Stock Futures Trader or Stock Specialist registered in the underlying security. ~~stock at the primary exchange for trading the stock.~~	**Proprietary**	**8 – Acting as a specialist in the security underlying of a derivative security**	

Proposed option order capacity codes and their FIX 4.3 equivalents

The following are additional codes that are proposed for the listed options markets and how they would map to FIX 4.3.

Proposed Listed Option Market Order Capacity Values	OrderCapacity (tag 528)	OrderRestrictions (tag 529)	Other
"I" Proposed Code used to designate a JBO account <u>which clears Customer at OCC</u>: any joint back office ("JBO") account of a broker/dealer that has a nominal ownership interest in a clearing broker/dealer and receives good faith margin treatment whereby such trade clears in the customer range at OCC. This ownership position allows the JBO clearing firm to finance securities transactions of the JBO participant on a good faith margin basis.	**Agency**		**AccountType (tag 581)=8 Joint Back Office**

© Copyright, 2007-2008, FIX Protocol, Limited

Proposed Listed Option Market Order Capacity Values	OrderCapacity (tag 528)	OrderRestrictions (tag 529)	Other
"J" Proposed Code used to designate a JBO account <u>which clears Firm at OCC</u>: any joint back office ("JBO") account of a broker/dealer that has a nominal ownership interest in a clearing broker/dealer and receives good faith margin treatment whereby such trade clears in the firm range at OCC. This ownership position allows the JBO clearing firm to finance securities transactions of the JBO participant on a good faith margin basis.	**Proprietary**		**AccountType (tag 581)=8 Joint Back Office**
"K" Proposed Code used to designate a JBO account <u>which clears MM at OCC</u>: any joint back office ("JBO") account of a broker/dealer that has a nominal ownership interest in a clearing broker/dealer and receives good faith margin treatment whereby such trade clears in the market-maker range at OCC. This ownership position allows the JBO clearing firm to finance securities transactions of the JBO participant on a good faith margin basis.	**Proprietary**	5-Acting As a specialist or market maker in the security	**AccountType(tag 581)=8 Joint Back Office**
"A" Linkage - Principal acting as agent order ("P/A") order routed through Linkage. (i.e. an order for the principal account of an eligible MM that is authorized to represent customer orders reflecting the terms of related unexecuted customer orders for which the MM is acting as agent).	**Agency**	5-Acting As a specialist or market maker in the security **9 – External Interconnected Market**	
"P"Linkage – Principal order. (i.e. an order for the principal account of an eligible MM which is entered to trade at the NBBO at another exchange and is not a P/A order).	**Principal**	5-Acting As a specialist or market maker in the security **9 – External Interconnected Market**	

© Copyright, 2007-2008, FIX Protocol, Limited

Proposed Listed Option Market Order Capacity Values	OrderCapacity (tag 528)	OrderRestrictions (tag 529)	Other
"S" Linkage – Principal satisfaction order (i.e. an order for the principal account of an eligible market maker sent through the Linkage to satisfy the liability arising from a trade through that was initiated by that market-maker).	**Riskless Principal**	5-Acting As a specialist or market maker in the security **9 – External Interconnected Market**	
"Z" Proposed Code used to designate orders as defined under Filing SR-CBOE-00-62: any non-CBOE member or non-broker/dealer account which typically clears at OCC as customer, but is prohibited from entering orders on RAES (i.e. futures traders, spouses of members, MM's away who are non B/Ds, etc).	**Agency**	**A – Riskless Arbitrage**	

CustomerOrderCapacity(tag 582) Mappings for Futures CTICode

CustOrderCapacity (tag 582)	Description
1	Member trading for their own account
2	Clearing Firm trading for its proprietary account
3	Member trading for another member
4	All other

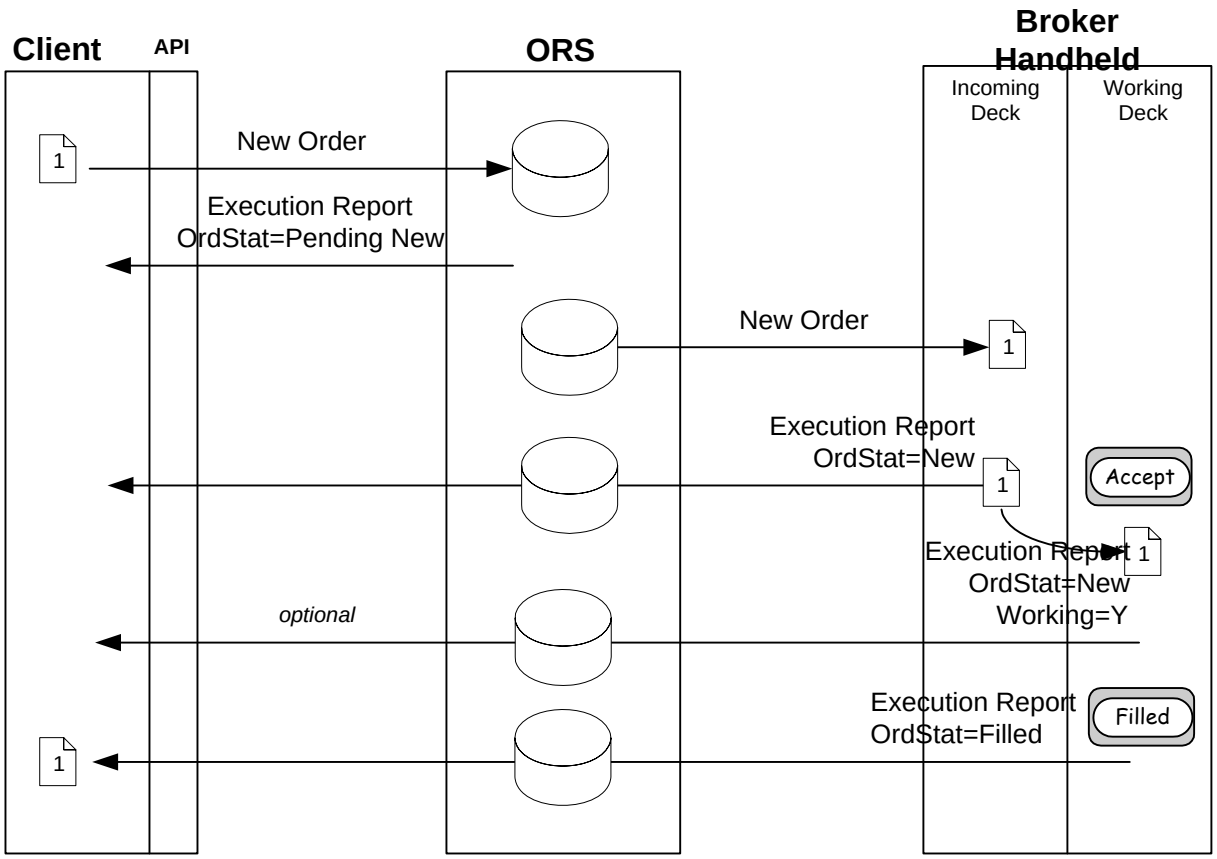
© Copyright, 2007-2008, FIX Protocol, Limited

Negative Prices permitted for futures and options strategies

The AvgPx(tag #6), LastPx(Tag #31), Price(tag #44), StopPx(tag #99), AllocAvgPx(tag #153), DayAvgPx(tag# 426), LegLastPx(tag# 637), UnderlyingLastPx(tag# 651) fields can be negative to support pricing of futures and options strategies, that due to theoretical pricing can result in "buying" a strategy for a negative price (receiving a credit for the strategy) or "selling" a strategy for a price(receiving a debit for the strategy).

Derivatives Markets Order State Transition

Derivatives markets are encouraged to adopt the following order state transition and order state reporting practices for routing orders to the floor.



NOTES:

- The broker is not required to move the order from the incoming deck to the working deck before filling the Order. Therefore, the "Working=Y" might not be received by the client. The Execution Report can be sent by the broker handheld from either the Incoming Deck or the Working Deck.
- The Order can take one or more Fills before the Order is completed, or the Order might only be partially completed by the end of the day.

© Copyright, 2007-2008, FIX Protocol, Limited

Party Roles used for Derivatives Markets

		Futures		Options	
Role	**Description**	**Order**	**Execution**	**Order**	**Execution**
ExecutingFirm	Firm that is executing the trade. Account[1] will be associated with this firm if present. Carries resultant positions of trades at the clearing house – unless GiveupClearingFirm is specified.	Reqd	Reqd	Reqd	Reqd
InitiatingTrader	If this role exists then this PartyID is the trader acronym that is reported to clearing. The Initiating Trader is associated with the ExecutingFirm. For market makers (specialists), the PartySubID for the InitiatingTrader is used for optional joint account identification	Opt	Cond	Opt	Cond
ClientID	Identification of the customer of the order – also known as the correspondent firm in CMS systems. Replaces ClientID[109]	n/a	n/a	Opt	Cond
ExecutingTrader	The trader or broker that actually executes a trade. If no InitiatingTrader role exists on the trade – then the ExecutingTrader is assumed to be associated with the ExecutingFirm For market makers (specialists), the PartySubID for PartyRole=ExecutingTrader can be used for optional joint account identification.	Opt	Reqd	Opt	Cond
OrderOriginator	ID of the party entering the trade into the system (data entry, userid, buy side trader, etc.). Replaces TraderID[466].	Opt	Cond	Opt	Cond

© Copyright, 2007-2008, FIX Protocol, Limited

GiveupClearingFirm	Firm that carries the position that results from a trade against the order. This is the firm to which the trade is given up. The PartySubID will be the account associated with this GiveupClearingFirm. Will be used for CMTA for US listed options.	Opt	Cond	Opt	Cond
CorrespondentClearingFirm	ClearingFirm that is going to carry the position on their books at another clearing house (exchanges). The resultant position does not reside with the market where it is traded – but instead is sent to an alternative market. The PartySubID will be the account associated with the CorrespondentClearingFirm	Opt	Cond	Opt	Cond
ExecutingSystem	System Identifier where execution took place. For instance in some markets there are multiple execution locations – such as an electronic book or automatic execution system. Replaces NYSE ExecutionInformation[9433]	n/a	Cond	n/a	Cond

MAPPING FIX 4.2 to FIX 4.3 Usage for Options Markets

FIX	FIX	Options	
4.2	**4.3**	**Order**	**Execution**
ExecutingBroker[76]	PartyID PartyRole=ExecutingFirm	Reqd	Reqd
Account[1]	Account[1]	Opt	Cond
ClearingFirm[439]	PartyID PartyRole=GiveupClearingFirm	Opt	Cond
ClearingAccount[440]	PartySubID of PartyRole=GiveupClearingFirm	Opt	Cond
Market Maker Sub	PartySubID of	Opt	Cond

© Copyright, 2007-2008, FIX Protocol, Limited

account information (Market Maker Acronym)	PartyRole=ExecutingTrader or PartyRole= InitiatingTrader		
Optional data reported on clearing report	PartySubID of PartyRole=ExecutingFirm	Opt	Cond

General Usage Information – US Futures Markets

There are three business scenarios involving give-ups and allocations within a single firm and across multiple firms in the futures industry.

Scenario 1-Allocate entire trade to multiple accounts within the clearing firm.

All relevant account and allocation information is carried in the allocation block. The total quantity of the order continues to be denoted in the OrderQtyData block. The account field (tag 1) is left blank as the information is fully denoted in the allocation block as outlined in the New Order Single for Corn example in this section. Both the main party block and nested party block within the allocation block are not used to carry allocation information when allocating trades across multiple accounts within the executing firm.

Scenario 2-Giveup entire trade to a single account at another firm

All relevant giveup information is contained in the main party block using PartyID to identify clearing firm (PartyRole=4) and PartyID to identify the carrying firm (PartyRole=14). The clearing firm suspense account is carried in account (tag 1). The carrying firm account number is populated in the PartySubID in the party block iteration when PartyRole=14. See the example contained in the Corn Calendar Multileg Order record.

Scenario 3-Allocate entire trade to multiple accounts across multiple firms

All relevant account and giveup information is carried within the allocation block. The total quantity of the order continues to be denoted in the OrderQtyData block. The quantity to be giveup to the each firm is designated using the AllocQty (tag 80) in the allocation block. The appropriate account at the carrying firm is designated using the AllocAccount (tag 79) in the allocation block. The appropriate carrying firm is designated within the nested party block within the appropriate allocation block using the PartyRole=14.

Execution Time Bracket reporting for US Futures Markets

The TradingSessionSubID (tag 625) is to be used to report execution time bracket codes for the US listed futures markets on the Execution Report.

© Copyright, 2007-2008, FIX Protocol, Limited

Example New Order – Single for Listed Futures Market

The following addresses sending a New Order - Single message into a futures market.

Tags that are not used in the futures and options industries have been omitted from the record. Tags that may be used based on the Exchange, execution medium and product have been included in the record and noted as not applicable ("n/a"). (Examples of such a tag is TradingSessionSubID which is used for floor based trades to carry the required time bracket designation and therefore is not applicable to screen based trading.)

The order created here is to buy 27 December 2001 Wheat at a price of 4.50. The order is being executed and cleared by firm 300. The order is also being allocated to multiple accounts within the executing firm, which is also the clearing firm as reflected in the allocation block. The order is also denoted as part of an average price group by placing a value in ClOrdLinkID field.

Tag	*Example Value*	*Field Name*	*Rqd*	*Comments*
		Standard Header	Y	MsgType = D
11	XXX123	ClOrdID	Y	
583	9876	ClOrdLinkID	N	The executions on this order will be average priced with executions on other orders with the same ClOrdLinkID.
component block <Parties>				
453	2	NoPartyIDs	N	
→ 448	300	***PartyID***	N	Firm executing and clearing the trade
→ 447	D	***PartyIDSource***	N	
→ 452	4	***PartyRole***	N	Firm executing and clearing the trade
→ 523	n/a	***PartySubID***	N	Not used when allocating trade across multiple accounts within the firm
→ 448	Tim1234	***PartyID***	N	
→ 447	D	***PartyIDSource***	N	
→ 452	13	***PartyRole***	N	Order Originator-person who entered the order into a system, if appropriate. Generally, the user id of that person
→ 523	n/a	***PartySubID***	N	
End </Parties>				

© Copyright, 2007-2008, FIX Protocol, Limited

1		1111	Account	N	Not used when allocating trades across multiple account within the firm
581		1	AccountType	N	AKA Origin. Required for futures markets.
591		0	PreallocMethod	N	
78		3	NoAllocs	N	
→	79	123456	AllocAccount	N	
→	467	n/a	IndividualAllocID	N	

Component block <NestedParties>					
→	539	n/a	NoNestedPartyIDs	N	
→ →	524	n/a	***NestedPartyID***	N	
→ →	525	n/a	***NestedPartyIDSource***	N	
→ →	538	n/a	***NestedPartyRole***	N	
→ →	545	n/a	***NestedPartySubID***	N	
End </NestedParties>					

→	80	2	AllocQty	N	
→	79	9876	AllocAccount	N	
→	467	n/a	IndividualAllocID	N	

Component block <NestedParties>					
→	539	n/a	NoNestedPartyIDs	N	
→ →	524	n/a	***NestedPartyID***	N	
→ →	525	n/a	***NestedPartyIDSource***	N	
→ →	538	n/a	***NestedPartyRole***	N	
→ →	545	n/a	***NestedPartySubID***	N	
End </NestedParties>					

→	80	15	AllocQty	N	
→	79	546789	AllocAccount	N	
→	467	n/a	IndividualAllocID	N	

Component block <NestedParties>

→	539	n/a	NoNestedPartyIDs	N	
→ →	524	n/a	***NestedPartyID***	N	
→ →	525	n/a	***NestedPartyIDSource***	N	
→ →	538	n/a	***NestedPartyRole***	N	
→ →	545	n/a	***NestedPartySubID***	N	
End </NestedParties>					

→	80	2	AllocQty	N	
~~63~~			~~SettlmntTyp~~	~~N~~	
~~64~~			~~FutSettDate~~	~~N~~	
635		C	ClearingFeeIndicator	N	
21		3	HandlInst	Y	Floor execution for futures markets should always be a 3
18		n/a	ExecInst	N	
110		n/a	MinQty	N	
111		n/a	MaxFloor	N	
100		XCBT	ExDestination	N	
386		n/a	NoTradingSessions	N	
→	***336***	n/a	***TradingSessionID***	N	
→	***625***	n/a	***TradingSessionSubID***	N	

Component block <Instrument>					
55		W	Symbol	***	ExDestination ticker symbol.
~~65~~			~~SymbolSfx~~	~~N~~	
48		n/a	SecurityID	N	
22		n/a	SecurityIDSource	N	
~~454~~			~~NoSecurityAltID~~	~~N~~	
→	~~***455***~~		~~***SecurityAltID***~~	~~N~~	
→	~~***456***~~		~~***SecurityAltIDSource***~~	~~N~~	
461		F	CFICode	N	
~~167~~			~~SecurityType~~	~~N~~	
200		200112	MaturityMonthYear	N	
541		n/a	MaturityDate	N	
~~470~~			~~CountryOfIssue~~	~~N~~	
~~471~~			~~StateOrProvinceOfIssue~~	~~N~~	

© Copyright, 2007-2008, FIX Protocol, Limited

~~472~~		~~LocaleOfIssue~~	~~N~~	
202	n/a	StrikePrice	N	
206	n/a	OptAttribute	N	
~~231~~		~~ContractMultiplier~~	~~N~~	
207	n/a	SecurityExchange	N	
107	Wheat Future	SecurityDesc	N	
350	n/a	EncodedSecurityDescLen	N	
351	n/a	EncodedSecurityDesc	N	
End </Instrument>				

~~140~~		~~PrevClosePx~~	~~N~~	
54	1	Side	Y	
60	20010806-13:34:29	TransactTime	Y	

Component block <OrderQtyData>				
38	27	OrderQty	N	
152	n/a	CashOrderQty	N	
End </OrderQtyData>				

40	2	OrdType	Y	Limit order.
44	4.500	Price	N	Limit Price of 4.500
99	n/a	StopPx	N	
~~15~~		~~Currency~~	~~N~~	
~~376~~		~~ComplianceID~~	~~N~~	
~~377~~		~~SolicitedFlag~~	~~N~~	
117	n/a	QuoteID	N	
59	0	TimeInForce	N	
168	n/a	EffectiveTime	N	
432	n/a	ExpireDate	N	
126	n/a	ExpireTime	N	
582	4	CustOrderCapacity	N	Also know as Customer Type Indicator (CTI). Required for futures markets.
~~120~~		~~SettlCurrency~~	~~N~~	
58	n/a	Text	N	

© Copyright, 2007-2008, FIX Protocol, Limited

354	n/a	EncodedTextLen	N	
355	n/a	EncodedText	N	
77	n/a	PositionEffect	N	
203	n/a	CoveredOrUncovered	N	
210	n/a	MaxShow	N	
388	n/a	DiscretionInst	N	
389	n/a	DiscretionOffset	N	
		Standard Trailer	Y	

Example New Order – Single for Listed Options Market

The following addresses sending a New Order - Single message into an options market.

Tags that are not used in the futures and options industries are not included in the example.

Tags with strike-through text are not currently used by the industries but may be used in the future.

Tags that have an example value of not applicable (n/a) are used in the industries. Herein, however, the value n/a was assigned for one of two reasons. First, specific futures and options markets may or may not utilize certain tags and, if utilized, their use and valid values would need to be addressed by participants in the particular market.

Second, the order created here is to buy 5 IBM September 2001 call options with a strike price of $100.00 at a price of $5.50. This and other assumptions concerning the order, such as it is not being allocated, result in some tag values being n/a.

Tag	*Example Value*	*Field Name*	*Rqd*	*Comments*
		Standard Header	Y	MsgType = D
11	XXX123	ClOrdID	Y	
583	n/a	ClOrdLinkID	N	

component block <Parties>					
453		5	NoPartyIDs	N	
→	448	PLC	***PartyID***	N	Trader badge
→	447	C	***PartyIDSource***	N	As assigned by exchange or clearing house
→	452	11	***PartyRole***	N	Order Origination Trader (if different from Executing Trader) optional
→	523	n/a	***PartySubID***	N	
→	448	0690	***PartyID***	N	OCC Clearing Firm Number
→	447	C	***PartyIDSource***	N	As assigned by exchange or clearing house
→	452	13	***PartyRole***	N	Order Origination Firm (if different from Executing Firm) optional

© Copyright, 2007-2008, FIX Protocol, Limited

→	523	n/a	***PartySubID***	N	
→	448	SMG	***PartyID***	N	Trader Badge
→	447	**C**	***PartyIDSource***	N	As assigned by exchange or clearing house
→	452	12	***PartyRole***	N	Executing Trader (required)
→	523	n/a	***PartySubID***	N	
→	448	0427	***PartyID***	N	OCC Clearing Firm Number
→	447	C	***PartyIDSource***	N	As assigned by exchange or clearing house
→	452	1	***PartyRole***	N	Executing Firm (required)
→	523	n/a	***PartySubID***	N	
→	448	323	***PartyID***	N	OCC Clearing Firm Number
→	447	C	***PartyIDSource***	N	As assigned by exchange or clearing house
→	452	14	***PartyRole***	N	Giveup Clearing Firm (CMTA) (optional if trade is being given up to another firm)
→	523	n/a	***PartySubID***	N	
End </Parties>					

1		AAA	Account	N	
581		n/a	AccountType	N	
591		n/a	PreallocMethod	N	
78		n/a	NoAllocs	N	
→	***79***	n/a	***AllocAccount***	N	
→	***467***	n/a	***IndividualAllocID***	N	
→	***80***	n/a	***AllocQty***	N	
~~63~~			~~SettlmntTyp~~	~~N~~	
~~64~~			~~FutSettDate~~	~~N~~	
21		2	HandlInst	Y	
18		n/a	ExecInst	N	
110		n/a	MinQty	N	
111		n/a	MaxFloor	N	
100		XCBO	ExDestination	N	
386		n/a	NoTradingSessions	N	
→	***336***	n/a	***TradingSessionID***	N	

© Copyright, 2007-2008, FIX Protocol, Limited

→ ***625***	n/a	***TradingSessionSubID***	N	
54	1	Side	Y	Buying the options.

Component block <Instrument>				
55	IBM	Symbol	***	ExDestination ticker symbol.
~~65~~		~~SymbolSfx~~	~~N~~	
48	n/a	SecurityID	N	
22	n/a	SecurityIDSource	N	
~~454~~		~~NoSecurityAltID~~	~~N~~	
→ ~~***455***~~		~~***SecurityAltID***~~	~~N~~	
→ ~~***456***~~		~~***SecurityAltIDSource***~~	~~N~~	
461	OC	CFICode	N	
~~167~~		~~SecurityType~~	~~N~~	
200	200109	MaturityMonthYear	N	
541	n/a	MaturityDate	N	
~~470~~		~~CountryOfIssue~~	~~N~~	
~~471~~		~~StateOrProvinceOfIssue~~	~~N~~	
~~472~~		~~LocaleOfIssue~~	~~N~~	
202	100.0	StrikePrice	N	
206	n/a	OptAttribute	N	
~~231~~		~~ContractMultiplier~~	~~N~~	
207	n/a	SecurityExchange	N	
107	n/a	SecurityDesc	N	
350	n/a	EncodedSecurityDescLen	N	
351	n/a	EncodedSecurityDesc	N	
End </Instrument>				

~~140~~	~~n/a~~	~~PrevClosePx~~	~~N~~	
60	20010806-13:34:29	TransactTime	Y	

Component block <OrderQtyData>				
38	5	OrderQty	N	
152	n/a	CashOrderQty	N	
End </OrderQtyData>				

© Copyright, 2007-2008, FIX Protocol, Limited

40	2	OrdType	Y	Limit order
44	5.5	Price	N	Buy at price of 5.5
99	n/a	StopPx	N	
~~15~~	~~n/a~~	~~Currency~~	~~N~~	
~~376~~	~~n/a~~	~~ComplianceID~~	~~N~~	
~~377~~	~~n/a~~	~~SolicitedFlag~~	~~N~~	
117	n/a	QuoteID	N	
59	0	TimeInForce	N	
168	n/a	EffectiveTime	N	
432	n/a	ExpireDate	N	
126	n/a	ExpireTime	N	
528	A	OrderCapacity	N	
529	n/a	OrderRestrictions	N	
582	n/a	CustOrderCapacity	N	
~~120~~	~~n/a~~	~~SettlCurrency~~	~~N~~	
58	n/a	Text	N	
354	n/a	EncodedTextLen	N	
355	n/a	EncodedText	N	
77	n/a	OpenClose	N	
203	n/a	CoveredOrUncovered	N	
210	n/a	MaxShow	N	
388	n/a	DiscretionInst	N	
389	n/a	DiscretionOffset	N	
118	n/a	NetMoney	N	
		Standard Trailer	Y	

Example New Order - Multileg for Listed Futures Market (Spread Order)The following addresses sending a New Order – Multileg message into a futures market.

Tags that are not used in the futures and options industries are not included in the example.

Tags with strike-through text are not currently used by the industries but may be used in the future.

Tags that have an example value of not applicable (n/a) are used in the futures industry. Herein, however, the value n/a was assigned for one of two reasons. First, specific futures and options markets may or may not

© Copyright, 2007-2008, FIX Protocol, Limited

utilize certain tags and, if utilized, their use and valid values would need to be addressed by participants in the particular market. (Examples of such tags are MultiLegRptTypeReq [563] and TradingSessionID [336].)

Second, the order created here is to buy 15 May 2002 - July 2002 Corn spreads at a price of –12. Some specifics concerning the order, such as it is not being allocated, result in some tag values being n/a.

The direction of the strategy is indicated by the Side (54) taken. When a strategy is pre-defined by a futures or options market and an inconsistency arises between:

the strategy indicated and the Side, LegSide (624), and/or LegRatioQty (623), or

the Side indicated and any LegSide indicated

the sell-side may either reject the order or accept the order. If the sell-side accepts the order it will be based on the strategy and Side indicated with any inconsistencies in LegSide and/or LegRatioQty being ignored.

The example also has any trade resulting from this order being given up to another firm. The firm being given up to will carry the trade on its books.

Tag	*Example Value*	*Field Name*	*Rqd*	*Comments*
		Standard Header	Y	MsgType = AB
11	1234567897	ClOrdID	Y	
583	n/a	ClOrdLinkID	N	
component block <Parties>				
453	3	NoPartyIDs	N	
→ 448	560	***PartyID***	N	Firm executing and clearing the trade
→ 447	D	***PartyIDSource***	N	
→ 452	4	***PartyRole***	N	
→ 523	n/a	***PartySubID***	N	
→ 448	500	***PartyID***	N	Trade being given up to and carried by this firm
→ 447	D	***PartyIDSource***	N	
→ 452	14	***PartyRole***	N	
→ 523	789567	***PartySubID***	N	Customer account number at carrying firm
→ 448	Tim1234	***PartyID***	N	
→ 447	D	***PartyIDSource***	N	
→ 452	13	***PartyRole***	N	

© Copyright, 2007-2008, FIX Protocol, Limited

→	523	n/a	***PartySubID***	N	
End </Parties>					

1	abcdef	Account	N	Account mnemomic as known by bookkeeping system. In case of giveup specifiied in party block, this account is at the executing firm.
581	1	AccountType	N	Also known as Origin. Required for futures markets.
591	n/a	PreallocMethod	N	
78	n/a	NoAllocs	N	
→ ***79***	n/a	***AllocAccount***	N	
→ ***467***	n/a	***IndividualAllocID***	N	

Component block <NestedParties>					
→	539	n/a	NoNestedPartyIDs	N	
→ →	524	n/a	***NestedPartyID***	N	
→ →	525	n/a	***NestedPartyIDSource***	N	
→ →	538	n/a	***NestedPartyRole***	N	
→ →	545	n/a	***NestedPartySubID***	N	
End </NestedParties>					

→ ***80***	n/a	***AllocQty***	N	
~~63~~		~~SettlmntTyp~~	~~N~~	
~~64~~		~~FutSettDate~~	~~N~~	
635	C	ClearingFeeIndicator	N	
21	3	HandlInst	Y	Floor executions for futures markets should always be "3".
18	n/a	ExecInst	N	
110	n/a	MinQty	N	
111	n/a	MaxFloor	N	
100	XCBT	ExDestination	N	
386	n/a	NoTradingSessions	N	
→ ***336***	n/a	***TradingSessionID***	N	
→ ***625***	n/a	***TradingSessionSubID***	N	
54	1	Side	Y	Buying the strategy.

© Copyright, 2007-2008, FIX Protocol, Limited

Component block <Instrument>					
55		C:CAL	Symbol	***	ExDestination ticker symbol.
~~65~~			~~SymbolSfx~~	~~N~~	
48		n/a	SecurityID	N	
22		n/a	SecurityIDSource	N	
~~454~~			~~NoSecurityAltID~~	~~N~~	
→	~~*455*~~		~~*SecurityAltID*~~	~~N~~	
→	~~*456*~~		~~*SecurityAltIDSource*~~	~~N~~	
461		FM	CFICode	N	
~~167~~			~~SecurityType~~	~~N~~	
200		n/a	MaturityMonthYear	N	
541		n/a	MaturityDate	N	
~~470~~			~~CountryOfIssue~~	~~N~~	
~~471~~			~~StateOrProvinceOfIssue~~	~~N~~	
~~472~~			~~LocaleOfIssue~~	~~N~~	
202		n/a	StrikePrice	N	
206		n/a	OptAttribute	N	
~~231~~		~~n/a~~	~~ContractMultiplier~~	~~N~~	
207		n/a	SecurityExchange	N	
107		n/a	SecurityDesc	N	
350		n/a	EncodedSecurityDescLen	N	
351		n/a	EncodedSecurityDesc	N	
End </Instrument>					

~~140~~		~~n/a~~	~~PrevClosePx~~	~~N~~	
555		2	NoLegs	Y	

Component block <Instrument Leg>					
→	600	C	LegSymbol	***	ExDestination ticker symbol.
→	~~601~~		~~LegSymbolSfx~~	~~N~~	
→	602	n/a	LegSecurityID	N	
→	603	n/a	LegSecurityIDSource	N	
→	~~604~~		~~NoLegSecurityAltID~~	~~N~~	
→	→ ~~*605*~~		~~*LegSecurityAltID*~~	~~N~~	

© Copyright, 2007-2008, FIX Protocol, Limited

→ →	~~606~~		~~LegSecurityAltIDSource~~	~~N~~	
→	608	F	LegCFICode	N	Commodity Future
→	~~609~~		~~LegSecurityType~~	~~N~~	
→	610	200205	LegMaturityMonthYear	N	May 2002 maturity.
→	611	n/a	LegMaturityDate	N	
→	~~596~~		~~LegCountryOfIssue~~	~~N~~	
→	~~597~~		~~LegStateOrProvinceOfIssue~~	~~N~~	
→	~~598~~		~~LegLocaleOfIssue~~	~~N~~	
→	612	n/a	LegStrikePrice	N	
→	613	n/a	LegOptAttribute	N	
→	~~614~~		~~LegContractMultiplier~~	~~N~~	
→	616	n/a	LegSecurityExchange	N	
→	620	Corn Future	LegSecurityDesc	N	
→	621	n/a	EncodedLegSecurityDescLen	N	
→	622	n/a	EncodedLegSecurityDesc	N	
→	623	1	LegRatioQty	N	Equal ratios.
→	624	1	LegSide	N	Buy leg.
→	564	n/a	LegPositionEffect	N	
→	565	n/a	LegCoveredOrUncovered	N	

Component block <NestedParties>					
→	539	n/a	NoNestedPartyIDs	N	
→ →	524	n/a	***NestedPartyID***	N	
→ →	525	n/a	***NestedPartyIDSource***	N	
→ →	538	n/a	***NestedPartyRole***	N	
→ →	545	n/a	***NestedPartySubID***	N	
End </NestedParties>					

→	654	n/a	LegRefID	N	
→	566	n/a	LegPrice	N	
→	~~587~~	~~n/a~~	~~LegSettlmntTyp~~	~~N~~	
→	~~588~~	~~n/a~~	~~LegFutSettDate~~	~~N~~	
→	600	C	LegSymbol	***	
→	~~601~~		~~LegSymbolSfx~~	~~N~~	

© Copyright, 2007-2008, FIX Protocol, Limited

→	602	n/a	LegSecurityID	N	
→	603	n/a	LegSecurityIDSource	N	
→	~~604~~		~~NoLegSecurityAltID~~	~~N~~	
→ →	~~***605***~~		~~***LegSecurityAltID***~~	~~N~~	
→ →	~~***606***~~		~~***LegSecurityAltIDSource***~~	~~N~~	
→	608	F	LegCFICode	N	
→	610	200207	LegMaturityMonthYear	N	July 2002 maturity.
→	611	n/a	LegMaturityDate	N	
→	~~596~~		~~LegCountryOfIssue~~	~~N~~	
→	~~597~~		~~LegStateOrProvinceOfIssue~~	~~N~~	
→	~~598~~		~~LegLocaleOfIssue~~	~~N~~	
→	612	n/a	LegStrikePrice	N	
→	613	n/a	LegOptAttribute	N	
→	~~614~~	~~n/a~~	~~LegContractMultiplier~~	~~N~~	
→	616	n/a	LegSecurityExchange	N	
→	620	Corn Future	LegSecurityDesc	N	
→	621	n/a	EncodedLegSecurityDescLen	N	
→	622	n/a	EncodedLegSecurityDesc	N	
→	623	1	LegRatioQty	N	Equal ratios.
→	624	2	LegSide	N	Sell leg.
→	564	n/a	LegPositionEffect	N	
→	565	n/a	LegCoveredOrUncovered	N	

Component block <NestedParties>					
→	539	n/a	NoNestedPartyIDs	N	
→ →	524	n/a	***NestedPartyID***	N	
→ →	525	n/a	***NestedPartyIDSource***	N	
→ →	538	n/a	***NestedPartyRole***	N	
→ →	545	n/a	***NestedPartySubID***	N	
End </NestedParties>					

→	654	n/a	LegRefID	N	
→	566	n/a	LegPrice	N	
→	~~587~~	~~n/a~~	~~LegSettlmntTyp~~	~~N~~	

© Copyright, 2007-2008, FIX Protocol, Limited

→	~~588~~	~~n/a~~	~~LegFutSettDate~~	~~N~~	
End </Instrument Leg>					

60	20010509-09:20:15	TransactTime	Y	

Component block <OrderQtyData>				
38	15	OrderQty	N	
152	n/a	CashOrderQty	N	
End </OrderQtyData>				

40	2	OrdType	Y	Limit order.
44	-12	Price	N	Buy strategy at negative 12.
99	n/a	StopPx	N	
~~15~~	~~n/a~~	~~Currency~~	~~N~~	
~~376~~	~~n/a~~	~~ComplianceID~~	~~N~~	
~~377~~	~~n/a~~	~~SolicitedFlag~~	~~N~~	
117	n/a	QuoteID	N	
59	0	TimeInForce	N	
168	n/a	EffectiveTime	N	
432	n/a	ExpireDate	N	
126	n/a	ExpireTime	N	
528		OrderCapacity	N	Used for options markets.
529		OrderRestrictions	N	Used for options markets.
582	4	CustOrderCapacity	N	Also know as Customer Type Indicator (CTI). Required for futures markets.
~~120~~	~~n/a~~	~~SettlCurrency~~	~~N~~	
58	n/a	Text	N	
354	n/a	EncodedTextLen	N	
355	n/a	EncodedText	N	
77	n/a	PositionEffect	N	
203	n/a	CoveredOrUncovered	N	
210	n/a	MaxShow	N	
388	n/a	DiscretionInst	N	
389	n/a	DiscretionOffset	N	

© Copyright, 2007-2008, FIX Protocol, Limited

563	n/a	MultiLegRptTypeReq	N	
		Standard Trailer	Y	

Example New Order - Multileg for Listed Futures Market (Butterfly Strategy)

The following addresses sending a New Order – Multileg message into a futures market.

Tags that are not used in the futures and options industries are not included in the example.

Tags with strike-through text are not currently used by the industries but may be used in the future.

Tags that have an example value of not applicable (n/a) are used in the industries. Herein, however, the value n/a was assigned for one of two reasons. First, specific futures and options markets may or may not utilize certain tags and, if utilized, their use and valid values would need to be addressed by participants in the particular market. (Examples of such tags are MultiLegRptTypeReq [563] and TradingSessionID [336].)

Second, the order created here is to buy 10 EuroDollar butterfly spreads at a price of -3.0, and is assumed that it will be productized by the sell-side on its electronic order matching system (ie: trade engine). This and other assumptions concerning the order, such as it is not being allocated, result in some tag values being n/a. (An example is the SecurityID [48] which the buy-side would not know until the sell-side has productized the butterfly.)

The direction of the strategy is indicated by the Side (54) taken. When a strategy is pre-defined by a futures market and an inconsistency arises between:

- the strategy indicated and the Side, LegSide (624), and/or LegRatioQty (623), or
- the Side indicated and any LegSide indicated

the sell-side may either reject the order or accept the order. If the sell-side accepts the order it will be based on the strategy and Side indicated with any inconsistencies in LegSide and/or LegRatioQty being ignored.

Tag	*Example Value*	*Field Name*	*Rqd*	*Comments*
		Standard Header	Y	MsgType = AB
11	05092001-NY-78955	ClOrdID	Y	
583	n/a	ClOrdLinkID	N	
component block <Parties>				
453	2	NoPartyIDs	N	
→ 448	001	***PartyID***	N	

© Copyright, 2007-2008, FIX Protocol, Limited

→	447	D	***PartyIDSource***	N	
→	452	4	***PartyRole***	N	
→	523	n/a	***PartySubID***	N	
→	448	4114Z9871272	***PartyID***	N	
→	447	D	***PartyIDSource***	N	
→	452	13	***PartyRole***	N	
→	523	n/a	***PartySubID***	N	
End </Parties>					

1		Z9871272	Account	N	
581		1	AccountType	N	
591		n/a	PreallocMethod	N	
78		n/a	NoAllocs	N	
→	***79***	n/a	***AllocAccount***	N	
→	***467***	n/a	***IndividualAllocID***	N	

Component block <NestedParties>					
→	539	n/a	NoNestedPartyIDs	N	
→ →	524	n/a	***NestedPartyID***	N	
→ →	525	n/a	***NestedPartyIDSource***	N	
→ →	538	n/a	***NestedPartyRole***	N	
→ →	545	n/a	***NestedPartySubID***	N	
End </NestedParties>					

→	***80***	n/a	***AllocQty***	N	
~~63~~			~~SettlmntTyp~~	~~N~~	
~~64~~			~~FutSettDate~~	~~N~~	
635		C	ClearingFeeIndicator	N	
21		1	HandlInst	Y	
18		n/a	ExecInst	N	
110		n/a	MinQty	N	
111		n/a	MaxFloor	N	
100		XCME	ExDestination	N	
386		n/a	NoTradingSessions	N	

© Copyright, 2007-2008, FIX Protocol, Limited

→	***336***	n/a	***TradingSessionID***	N	
→	***625***	n/a	***TradingSessionSubID***	N	
54		1	Side	Y	

Component block <Instrument>					
55		GE:BF	Symbol	***	
~~65~~			~~SymbolSfx~~	~~N~~	
48		n/a	SecurityID	N	
22		n/a	SecurityIDSource	N	
~~454~~			~~NoSecurityAltID~~	~~N~~	
→	~~***455***~~		~~***SecurityAltID***~~	~~N~~	
→	~~***456***~~		~~***SecurityAltIDSource***~~	~~N~~	
461		FM	CFICode	N	
~~167~~			~~SecurityType~~	~~N~~	
200		n/a	MaturityMonthYear	N	
541		n/a	MaturityDate	N	
~~470~~			~~CountryOfIssue~~	~~N~~	
~~471~~			~~StateOrProvinceOfIssue~~	~~N~~	
~~472~~			~~LocaleOfIssue~~	~~N~~	
202		n/a	StrikePrice	N	
206		n/a	OptAttribute	N	
~~231~~			~~ContractMultiplier~~	~~N~~	
207		n/a	SecurityExchange	N	
107		n/a	SecurityDesc	N	
350		n/a	EncodedSecurityDescLen	N	
351		n/a	EncodedSecurityDesc	N	
End </Instrument>					

~~140~~			~~PrevClosePx~~	~~N~~	
555		3	NoLegs	Y	

Component block <Instrument Leg>					
→	600	GE	LegSymbol	***	
→	~~601~~		~~LegSymbolSfx~~	~~N~~	

© Copyright, 2007-2008, FIX Protocol, Limited

→	602	CME005060001	LegSecurityID	N	
→	603	ISIN	LegSecurityIDSource	N	
→	~~604~~		~~NoLegSecurityAltID~~	~~N~~	
→ →	~~***605***~~		~~***LegSecurityAltID***~~	~~N~~	
→ →	~~***606***~~		~~***LegSecurityAltIDSource***~~	~~N~~	
→	608	F	LegCFICode	N	
→	~~609~~		~~LegSecurityType~~	~~N~~	
→	610	200109	LegMaturityMonthYear	N	
→	611	n/a	LegMaturityDate	N	
→	~~596~~		~~LegCountryOfIssue~~	~~N~~	
→	~~597~~		~~LegStateOrProvinceOfIssue~~	~~N~~	
→	~~598~~		~~LegLocaleOfIssue~~	~~N~~	
→	612	n/a	LegStrikePrice	N	
→	613	n/a	LegOptAttribute	N	
→	~~614~~		~~LegContractMultiplier~~	~~N~~	
→	616	n/a	LegSecurityExchange	N	
→	620	GEU1	LegSecurityDesc	N	
→	621	n/a	EncodedLegSecurityDescLen	N	
→	622	n/a	EncodedLegSecurityDesc	N	
→	623	1	LegRatioQty	N	
→	624	1	LegSide	N	
→	564	n/a	LegPositionEffect	N	
→	565	n/a	LegCoveredOrUncovered	N	

Component block <NestedParties>					
→	539	n/a	NoNestedPartyIDs	N	
→ →	524	n/a	***NestedPartyID***	N	
→ →	525	n/a	***NestedPartyIDSource***	N	
→ →	538	n/a	***NestedPartyRole***	N	
→ →	545	n/a	***NestedPartySubID***	N	
End </NestedParties>					

→	654	n/a	LegRefID	N	
→	566	n/a	LegPrice	N	

© Copyright, 2007-2008, FIX Protocol, Limited

→	~~587~~		~~LegSettlmntTyp~~	~~N~~	
→	~~588~~		~~LegFutSettDate~~	~~N~~	
→	600	GE	LegSymbol	***	
→	~~601~~		~~LegSymbolSfx~~	~~N~~	
→	602	CME005060004	LegSecurityID	N	
→	603	ISIN	LegSecurityIDSource	N	
→	~~604~~		~~NoLegSecurityAltID~~	~~N~~	
→ →	~~***605***~~		~~***LegSecurityAltID***~~	~~N~~	
→ →	~~***606***~~		~~***LegSecurityAltIDSource***~~	~~N~~	
→	608	F	LegCFICode	N	
→	~~609~~		~~LegSecurityType~~	~~N~~	
→	610	200112	LegMaturityMonthYear	N	
→	611	n/a	LegMaturityDate	N	
→	~~596~~		~~LegCountryOfIssue~~	~~N~~	
→	~~597~~		~~LegStateOrProvinceOfIssue~~	~~N~~	
→	~~598~~		~~LegLocaleOfIssue~~	~~N~~	
→	612	n/a	LegStrikePrice	N	
→	613	n/a	LegOptAttribute	N	
→	~~614~~		~~LegContractMultiplier~~	~~N~~	
→	616	n/a	LegSecurityExchange	N	
→	620	GEZ1	LegSecurityDesc	N	
→	621	n/a	EncodedLegSecurityDescLen	N	
→	622	n/a	EncodedLegSecurityDesc	N	
→	623	2	LegRatioQty	N	
→	624	2	LegSide	N	
→	564	n/a	LegPositionEffect	N	
→	565	n/a	LegCoveredOrUncovered	N	

Component block <NestedParties>					
→	539	n/a	NoNestedPartyIDs	N	
→ →	524	n/a	***NestedPartyID***	N	
→ →	525	n/a	***NestedPartyIDSource***	N	
→ →	538	n/a	***NestedPartyRole***	N	
→ →	545	n/a	***NestedPartySubID***	N	

© Copyright, 2007-2008, FIX Protocol, Limited

End </NestedParties>					
→	654	n/a	LegRefID	N	
→	566	n/a	LegPrice	N	
→	~~587~~		~~LegSettlmntTyp~~	~~N~~	
→	~~588~~		~~LegFutSettDate~~	~~N~~	
→	600	GE	LegSymbol	***	
→	~~601~~		~~LegSymbolSfx~~	~~N~~	
→	602	CME005060007	LegSecurityID	N	
→	603	ISIN	LegSecurityIDSource	N	
→	~~604~~		~~NoLegSecurityAltID~~	~~N~~	
→ →	~~***605***~~		~~***LegSecurityAltID***~~	~~N~~	
→ →	~~***606***~~		~~***LegSecurityAltIDSource***~~	~~N~~	
→	608	F	LegCFICode	N	
→	~~609~~		~~LegSecurityType~~	~~N~~	
→	610	200203	LegMaturityMonthYear	N	
→	611	n/a	LegMaturityDate	N	
→	~~596~~		~~LegCountryOfIssue~~	~~N~~	
→	~~597~~		~~LegStateOrProvinceOfIssue~~	~~N~~	
→	~~598~~		~~LegLocaleOfIssue~~	~~N~~	
→	612	n/a	LegStrikePrice	N	
→	613	n/a	LegOptAttribute	N	
→	~~614~~		~~LegContractMultiplier~~	~~N~~	
→	616	n/a	LegSecurityExchange	N	
→	620	GEH2	LegSecurityDesc	N	
→	621	n/a	EncodedLegSecurityDescLen	N	
→	622	n/a	EncodedLegSecurityDesc	N	
→	623	1	LegRatioQty	N	
→	624	1	LegSide	N	
→	564	n/a	LegPositionEffect	N	
→	565	n/a	LegCoveredOrUncovered	N	

Component block <NestedParties>					
→	539	n/a	NoNestedPartyIDs	N	

© Copyright, 2007-2008, FIX Protocol, Limited

→	→	524	n/a	***NestedPartyID***	N	
→	→	525	n/a	***NestedPartyIDSource***	N	
→	→	538	n/a	***NestedPartyRole***	N	
→	→	545	n/a	***NestedPartySubID***	N	
End </NestedParties>						

→	654	n/a	LegRefID	N	
→	566	n/a	LegPrice	N	
→	~~587~~		~~LegSettlmntTyp~~	~~N~~	
→	~~588~~		~~LegFutSettDate~~	~~N~~	
End </Instrument Leg>					

60	20010509-09:20:15	TransactTime	Y	

Component block <OrderQtyData>				
38	10	OrderQty	N	
152	n/a	CashOrderQty	N	
End </OrderQtyData>				

40	2	OrdType	Y	
44	-3.0	Price	N	
99	n/a	StopPx	N	
~~15~~		~~Currency~~	~~N~~	
~~376~~		~~ComplianceID~~	~~N~~	
~~377~~		~~SolicitedFlag~~	~~N~~	
117	n/a	QuoteID	N	
59	0	TimeInForce	N	
168	n/a	EffectiveTime	N	
432	n/a	ExpireDate	N	
126	n/a	ExpireTime	N	
528		OrderCapacity	N	
529		OrderRestrictions	N	
582	4	CustOrderCapacity	N	
~~120~~	~~n/a~~	~~SettlCurrency~~	~~N~~	

© Copyright, 2007-2008, FIX Protocol, Limited

58	n/a	Text	N	
354	n/a	EncodedTextLen	N	
355	n/a	EncodedText	N	
77	n/a	PositionEffect	N	
203	n/a	CoveredOrUncovered	N	
210	n/a	MaxShow	N	
388	n/a	DiscretionInst	N	
389	n/a	DiscretionOffset	N	
563	n/a	MultiLegRptTypeReq	N	
		Standard Trailer	Y	

Example Multlileg Execution Report for Listed Futures Market

Multlileg Execution Report Example for Futures Markets

The following addresses receiving an Execution Report – Multileg message.

Tags that are not used in the futures and options industries are not included in the example.

Tags with strike-through text are not currently used by the industries but may be used in the future.

Tags that have an example value of not applicable (n/a) are used in the industries. Herein, however, the value n/a was assigned for one of two reasons. First, individual futures and options markets may or may not utilize certain tags and, if utilized, their use and valid values would need to be addressed by participants in the particular market.

The execution report references an order to buy 15 July 2001/September 2001 Corn Spreads. The order is a give-up trade being executed and cleared by firm 560 and carried on the books of firm 500. This is the first execution of the order and it is for a total of 5 spreads. The order was executed on the trading floor as atomically and is being reported to the customer atomically via this execution report. The order will also be cleared atomically.

Tag	*Example Values*	*Field Name*	*Rqd*	*Comments*
		Standard Header	Y	MsgType = 8
37	987654	OrderID	Y	
198	n/a	SecondaryOrderID	N	
526	n/a	SecondaryClOrdID	N	
527	n/a	SecondaryExecID	N	
11	123456789	ClOrdID	N	

© Copyright, 2007-2008, FIX Protocol, Limited

41		n/a	OrigClOrdID	N	
583		n/a	ClOrdLinkID	N	

component block <Parties>					
453		3	NoPartyIDs	N	
→	448	560	***PartyID***	N	
→	447	D	***PartyIDSource***	N	
→	452	4	***PartyRole***	N	
→	523	n/a	***PartySubID***	N	
→	448	500	***PartyID***	N	
→	447	D	***PartyIDSource***	N	
→	452	14	***PartyRole***	N	
→	523	789567	***PartySubID***	N	
→	448	tim1234	***PartyID***	N	
→	447	D	*PartyIDSource*	N	
→	452	13	*PartyRole*	N	
→	523	n/a	***PartySubID***	N	
End </Parties>					

382		1	NoContraBrokers	N	
→	375	455	ContraBroker	N	
→	337	ABC	ContraTrader	N	
→	437	5	ContraTradeQty	N	
→	438	20010509-09:22:40	ContraTradeTime	N	
→	655	n/a	ContraLegRefID	N	
66		n/a	ListID	N	
548		n/a	CrossID	N	
551		n/a	OrigCrossID	N	
549		n/a	CrossType	N	
17		X6789	ExecID	Y	
19		n/a	ExecRefID	N	
150		F	ExecType	Y	
39		1	OrdStatus	Y	
636		Y	WorkingIndicator	N	

© Copyright, 2007-2008, FIX Protocol, Limited

103	n/a	OrdRejReason	N	
378	n/a	ExecRestatementReason	N	
1	abcdef	Account	N	
581	1	AccountType	N	
591	n/a	PreallocMethod	N	
~~63~~		~~SettlmntTyp~~	~~N~~	
~~64~~		~~FutSettDate~~	~~N~~	
635	C	ClearingFeeIndicator	N	

Component block <Instrument>				
55	C:CAL	Symbol	***	
~~65~~		~~SymbolSfx~~	~~N~~	
48	n/a	SecurityID	N	
22	n/a	SecurityIDSource	N	
~~454~~		~~NoSecurityAltID~~	~~N~~	
→ ~~***455***~~		~~***SecurityAltID***~~	~~N~~	
→ ~~***456***~~		~~***SecurityAltIDSource***~~	~~N~~	
461	FM	CFICode	N	
~~167~~		~~SecurityType~~	~~N~~	
200	n/a	MaturityMonthYear	N	
541	n/a	MaturityDate	N	
~~470~~		~~CountryOfIssue~~	~~N~~	
~~471~~		~~StateOrProvinceOfIssue~~	~~N~~	
~~472~~		~~LocaleOfIssue~~	~~N~~	
202	n/a	StrikePrice	N	
206	n/a	OptAttribute	N	
~~231~~		~~ContractMultiplier~~	~~N~~	
207	n/a	SecurityExchange	N	
107	n/a	SecurityDesc	N	
350	n/a	EncodedSecurityDescLen	N	
351	n/a	EncodedSecurityDesc	N	
End </Instrument>				

54	1	Side	Y	

555	2	NoLegs	Y	Number of legs. Can be zero – must be provided even if zero	

Component block <Instrument Leg>					
→	600	C	LegSymbol	***	
→	~~601~~		~~LegSymbolSfx~~	~~N~~	
→	602	n/a	LegSecurityID	N	
→	603	n/a	LegSecurityIDSource	N	
→	~~604~~		~~NoLegSecurityAltID~~	~~N~~	
→ →	~~***605***~~		~~***LegSecurityAltID***~~	~~N~~	
→ →	~~***606***~~		~~***LegSecurityAltIDSource***~~	~~N~~	
→	608	F	LegCFICode	N	
→	~~609~~		~~LegSecurityType~~	~~N~~	
→	610	200105	LegMaturityMonthYear	N	
→	611	n/a	LegMaturityDate	N	
→	~~596~~		~~LegCountryOfIssue~~	~~N~~	
→	~~597~~		~~LegStateOrProvinceOfIssue~~	~~N~~	
→	~~598~~		~~LegLocaleOfIssue~~	~~N~~	
→	612	n/a	LegStrikePrice	N	
→	613	n/a	LegOptAttribute	N	
→	~~614~~		~~LegContractMultiplier~~	~~N~~	
→	616	n/a	LegSecurityExchange	N	
→	620	Corn Future	LegSecurityDesc	N	
→	621	n/a	EncodedLegSecurityDescLen	N	
→	622	n/a	EncodedLegSecurityDesc	N	
→	623	1	LegRatioQty	N	
→	624	1	LegSide	N	
→	564	n/a	LegPositionEffect	N	Provide if the PositionEffect for the leg is different from that specified for the overall multileg security
→	565	n/a	LegCoveredOrUncovered	N	Provide if the CoveredOrUncovered for the leg is different from that specified for the overall multileg security.

Component block <NestedParties>					
→	539	n/a	NoNestedPartyIDs	N	
→ →	524	n/a	***NestedPartyID***	N	

© Copyright, 2007-2008, FIX Protocol, Limited

→ →	525	n/a	***NestedPartyIDSource***	N	
→ →	538	n/a	***NestedPartyRole***	N	
→ →	545	n/a	***NestedPartySubID***	N	
End </NestedParties>					

→	654	n/a	LegRefID	N	
→	566	n/a	LegPrice		Provide only if a price was specified for the specific leg. Used for anchoring the overall multileg security price to a specific leg price.
→	637	n/a	LegLastPx		Used to report the execution price assigned to the leg of the multileg instrument
→	~~587~~		~~LegSettlmntTyp~~		
→	~~588~~		~~LegFutSettDate~~		~~Required when SettlmntTyp = 6 (Future) or SettlmntTyp = 8 (Sellers Option)~~
→	600	C	LegSymbol	***	
→	~~601~~		~~LegSymbolSfx~~	~~N~~	
→	602	n/a	LegSecurityID	N	
→	603	n/a	LegSecurityIDSource	N	
→	~~604~~		~~NoLegSecurityAltID~~	~~N~~	
→ →	~~***605***~~		~~***LegSecurityAltID***~~	~~N~~	
→ →	~~***606***~~		~~***LegSecurityAltIDSource***~~	~~N~~	
→	608	F	LegCFICode	N	
→	~~609~~		~~LegSecurityType~~	~~N~~	
→	610	200107	LegMaturityMonthYear	N	
→	611	n/a	LegMaturityDate	N	
→	~~596~~		~~LegCountryOfIssue~~	~~N~~	
→	~~597~~		~~LegStateOrProvinceOfIssue~~	~~N~~	
→	~~598~~		~~LegLocaleOfIssue~~	~~N~~	
→	612	n/a	LegStrikePrice	N	
→	613	n/a	LegOptAttribute	N	
→	~~614~~		~~LegContractMultiplier~~	~~N~~	
→	616	n/a	LegSecurityExchange	N	
→	620	Corn Future	LegSecurityDesc	N	
→	621	n/a	EncodedLegSecurityDescLen	N	
→	622	n/a	EncodedLegSecurityDesc	N	

© Copyright, 2007-2008, FIX Protocol, Limited

→	623	1	LegRatioQty	N	
→	624	2	LegSide	N	
→	564	n/a	LegPositionEffect	N	Provide if the PositionEffect for the leg is different from that specified for the overall multileg security
→	565	n/a	LegCoveredOrUncovered	N	Provide if the CoveredOrUncovered for the leg is different from that specified forthe overall multileg security.

Component block <NestedParties>					
→	539	n/a	NoNestedPartyIDs	N	
→ →	524	n/a	***NestedPartyID***	N	
→ →	525	n/a	***NestedPartyIDSource***	N	
→ →	538	n/a	***NestedPartyRole***	N	
→ →	545	n/a	***NestedPartySubID***	N	
End </NestedParties>					

→	654	n/a	LegRefID	N	
→	566	n/a	LegPrice		Provide only if a price is required for a specific leg. Used for anchoring the overall multileg security price to a specific leg price.
→	637	n/a	LegLastPx		Used to report the execution price assigned to the leg of the multileg instrument
→	~~587~~		~~LegSettlmntTyp~~		
→	~~588~~		~~LegFutSettDate~~		~~Required when SettlmntTyp = 6 (Future) or SettlmntTyp = 8 (Sellers Option)~~
End </Instrument Leg>					

Component block <OrderQtyData>				
38	15	OrderQty	N	
152	n/a	CashOrderQty	N	
End </OrderQtyData>				

40	2	OrdType	N	
44	-12	Price	N	Required if specified on the order
99	n/a	StopPx	N	Required if specified on the order

© Copyright, 2007-2008, FIX Protocol, Limited

388	n/a	DiscretionInst	N	Code to identify the price a DiscretionOffset is related to and should be mathematically added to. Required if DiscretionOffset is specified.
389	n/a	DiscretionOffset	N	Amount (signed) added to the "related to" price specified via DiscretionInst.
~~15~~		~~Currency~~	~~N~~	
~~376~~		~~ComplianceID~~	~~N~~	
~~377~~		~~SolicitedFlag~~	~~N~~	
59	0	TimeInForce	N	Absence of this field indicates Day order
168	n/a	EffectiveTime	N	Time specified on the order at which the order should be considered valid
432	n/a	ExpireDate	N	Conditionally required if TimeInForce = GTD and ExpireTime is not specified.
126	n/a	ExpireTime	N	Conditionally required if TimeInForce = GTD and ExpireDate is not specified.
18	n/a	ExecInst	N	Can contain multiple instructions, space delimited.
528	n/a	OrderCapacity	N	
529	n/a	OrderRestrictions	N	
582	4	CustOrderCapacity	N	
32	5	LastQty	N	
31	-12	LastPx	N	
~~30~~		~~LastMkt~~	~~N~~	
336	n/a	TradingSessionID	N	
625	n/a	TradingSessionSubID	N	Used for time bracket codes for floor trades in the futures markets.
151	10	LeavesQty	Y	
14	5	CumQty	Y	
6	n/a	AvgPx	Y	
424	n/a	DayOrderQty	N	For GT orders on days following the day of the first trade.
425	n/a	DayCumQty	N	For GT orders on days following the day of the first trade.
426	n/a	DayAvgPx	N	For GT orders on days following the day of the first trade.

© Copyright, 2007-2008, FIX Protocol, Limited

75	20010509	TradeDate	N	Used when reporting other than current day trades. For futures markets, used to report current trade date as opposed to current calendar date at time of execution.
60	20010509-09:23:05	TransactTime	N	Time the transaction represented by this ExecutionReport occurred
~~118~~	n/a	~~NetMoney~~	~~N~~	
21	3	HandlInst	N	
110	n/a	MinQty	N	
111	n/a	MaxFloor	N	
77	n/a	PositionEffect	N	
210	n/a	MaxShow		
58	n/a	Text		
354	n/a	EncodedTextLen		Must be set if EncodedText field is specified and must immediately precede it.
355	n/a	EncodedText		Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
442	n/a	MultiLegReportingType	N	Default is a single security if not specified.
		Standard Trailer	Y	

Options Back Office Processing

Background

The Option Clearing Corporation (OCC) initiated an effort to work with the FIX Protocol Organization to enhance FIX as standard message protocol for use in disseminating data to back office organizations such as clearing members, regulatory agencies, and trade sources. The initiative began in earnest at the beginning of 2003. OCC worked to identify gaps in FIX based on existing messages and member requests. The group not only identified missing functionality (primarily in the area of missing fields, component blocks and reports), they pushed to develop a guideline for using FIX for options back office processing. This section contains guidelines for usage of these enhancements to specific post-trade messages FIX for options back office processing.

Position Maintenance Report

PosMaintAction (712) field:

A new enumeration value was added, called "reverse". Reverse differs from a Cancel in that a Reverse would completely back-out the Position Maintenance transaction from the audit trail to make it appear as if the transaction never existed. A Cancel would be the Cancel or Bust of a Position Maintenance transaction but allow for the preservation of the audit trail of the original transaction and the subsequent cancel/bust.

TransactTime (60):

© Copyright, 2007-2008, FIX Protocol, Limited

TransactTime is a **conditionally required** field even though the field is marked as "not required" in this message. This is the time the order request was initiated/released by the trader, trading system, or intermediary.

TransactTime **is not require** when the Position Maintenance Requests are processed in batch and/or the Transaction Time is not available (as in the case of a Clearing Org or other Post-Trade entity).

If PosReqID is not included in the Position Maintenance Report, the TransactTime requirement can be dropped.

Position Report

PosReqType (724):

A new enumeration value called "Settlement Activity" was added to show underlying delivery that resulted from a position.

Trade Capture Report Ack

TradeReportType (865):

A new enumeration value called "Defaulted" was added. A "Defaulted" Trade Report is one that was originally specified to be given up to another party but due to a violation of a give-up condition the transaction was placed into a 'Default' account and not the specified Give-Up account.

Trade Capture Report

TradeReportType (865):

A new enumeration value called "Defaulted" was added. A "Defaulted" Trade Report is one that was originally specified to be given up to another party but due to a violation of a give-up condition the transaction was placed into a 'Default' account and not the specified Give-Up account.

OrderID (37):

Should be conditionally required when Trade Capture Report is used for back office processing.

Security Definition

SecurityReportID (964):

This is the identifier for the Security Definition message in a bulk transfer environment that does not support the request/response model.

It should be noted that in a request/response model the following fields are required: SecurityReqID (320), SecurityResponseID (322), and SecurityResponseType (323).

Security List

SecurityReportID (964):

This is the identifier for the Security List message in a bulk transfer environment that does not support the request/response model.

It should be noted that in a request/response model the following fields are required: SecurityReqID (320), SecurityResponseID (322), and SecurityRequestResult (560).

Parties component block

PartyIDSource (447):

If not specified, the default is the counterparty agreed upon source.

PartySubIDType (803):

If not specified, the default is the counterparty agreed upon type.

© Copyright, 2007-2008, FIX Protocol, Limited

Contrary Intention Report

Contrary Intention Report is used to support the reporting of contrary expiration quantities for Saturday expiring options

Security Definition Update Report

Security Definition Update Report is to support the reporting of updates (Add, Modify, Delete) to a Product Security Masterfile due to Corporate Actions or other business requirements.

SecurityReportID (964):

This is the identifier for the Security Definition Update Report message in a bulk transfer environment that does not support the request/response model.

It should be noted that in a request/response model the following fields are required: SecurityReqID (320), SecurityResponseID (322), and SecurityRequestResult (560).

Security List Update Report

Security List Update Report is to support the reporting of updates (Add, Modify, Delete) to a Contract Security Masterfile due to Corporate actions or other business requirements.

SecurityReportID (964):

This is the identifier for the Security List Update Report message in a bulk transfer environment that does not support the request/response model.

It should be noted that in a request/response model the following fields are required: SecurityReqID (320), SecurityResponseID (322), and SecurityRequestResult (560).

FIA Trade Identification Standards

Background

Trade Identification is a central concept across the listed derivatives post trade space and is essential for efficient and accurate identification of a trade between a Clearing Organization and Firm. This section discusses the standard practice for trade identification between Clearing Organization and Firm as well as defining standard usages for other trade-related ID's.

CME, OCC, NYMEX, NYBOT, and TCC have agreed to the following Trade ID Management Practices within the context of FIA Standards Working Group.

Trade Identification Fields

The Trade Capture Report has become de facto standard for bi-directional reporting of trades between the Clearing System and Firm. The Trade Capture Report, Trade Capture Report Ack, Trade Capture Report Request, and Trade Capture Report Request Ack messages carries the following fields which allows the Firm and the Clearing System to clearly and unambiguously represent the business entity called "Trade" within their respective firms:

1. TradeID – The unique ID assigned to a trade once it enters the Clearing System. This will become the primary ID by which the Clearing Organization and Firm refer to the Trade entity.
2. SecondaryTradeID – Used to carry an internal Clearing System assigned ID which may or may not be reported to the firm.
3. FirmTradeID - The ID assigned to a trade by the Firm to track a trade within the Firm back office system. This ID can be assigned either prior to being submitted for Clearing or after being received from the Clearing System.

© Copyright, 2007-2008, FIX Protocol, Limited

4. SecondaryFirmTradeID – Used to carry an internal back office assigned ID which may or may not be reported to the Clearing System.

A Firm would be able to submit a FirmTradeID on a trade. The Clearing System would have the flexibility to set the TradeID (aka the clearing trade ID) to the value of FirmTradeID or set the TradeID to a completely new clearing trade ID. In both cases, the clearing trade ID would become the primary identifier for that trade.

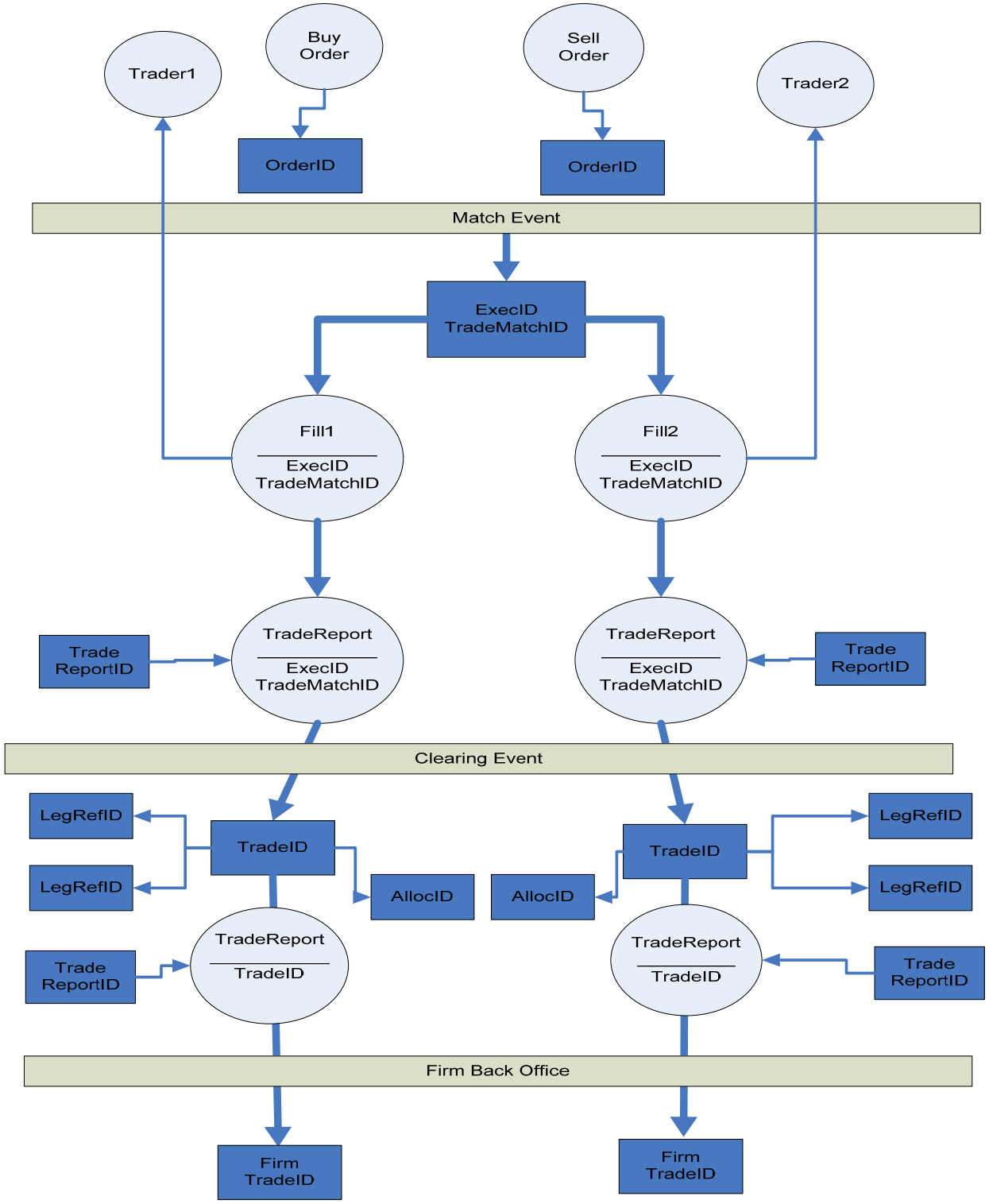
Additionally the TradeID and FirmTradeID fields are available in the AllocationInstruction, AllocationReport and AllocationAlert messages to allow the Firm and Clearing System to reference the trades in an allocation.

Additional Identifier Definitions

Exhibit 1 shows the relationship between the identifiers in the Order/Fill/Cleared Trade life-cycle. The dark blue rectangles represent the ID's that are assigned in a typical trade flow and the relationship between ID's. The identifiers are cumulative and are carried through to the Firm Back Office if so desired.

© Copyright, 2007-2008, FIX Protocol, Limited

Exhibit 1



1. **ExecID** – used to identify the fill event that created the trade. There may be multiple fills per trade and therefore multiple fills with the same exec id. In other words, ExecID has a one-to-many relationship with

© Copyright, 2007-2008, FIX Protocol, Limited

the resulting fills. Since each fill becomes a cleared trade, ExecID also has a one-to-many relationship with TradeCaptureReport.

2. **TradeMatchID** – all purpose internal identifier assigned to fills by the match engine. The TradeMatchID may either be unique to each fill in a match event or common across all fills in a match. In the event that this is the primary ID used to uniquely identify a fill, then ExecID should be used in stead.

3. **TradeReportID** – used to uniquely identify the transaction being used to add, update, or cancel a trade. As required by the specification, TradeReportID is required on the Trade Capture Report and must be unique per message. The Trade Capture Report Ack must echo back the TradeReportID and will not necessarily have a unique ID assigned to it. TradeReportID is optional on Trade Capture Report Request and Trade Capture Report Request Ack.

4. **TradeReportRefID** – used to refer to an original TradeReportID for purposes of update or cancellation. A TradeCaptureReport will specify a TradeReportRefID when it is being used to perform a subsequent update or cancellation.

5. **AllocID** – used to identify the Allocation Group ID to which a trade is being added. A trade may carry allocation information which includes both the Allocation Group as well as the Allocation Instruction for that trade. AllocID is used for both Average Price and Basic Allocations.

6. **IndividualAllocID** – occurs in the Allocation block of the trade and is used to specify the Allocation ID of the allocation to which the trade is being directed.

7. **TradeLinkID** – used to link together a group of trades that make up an average price allocation. TradeLinkID can be used in place of AllocID for average pricing purposes if so desired

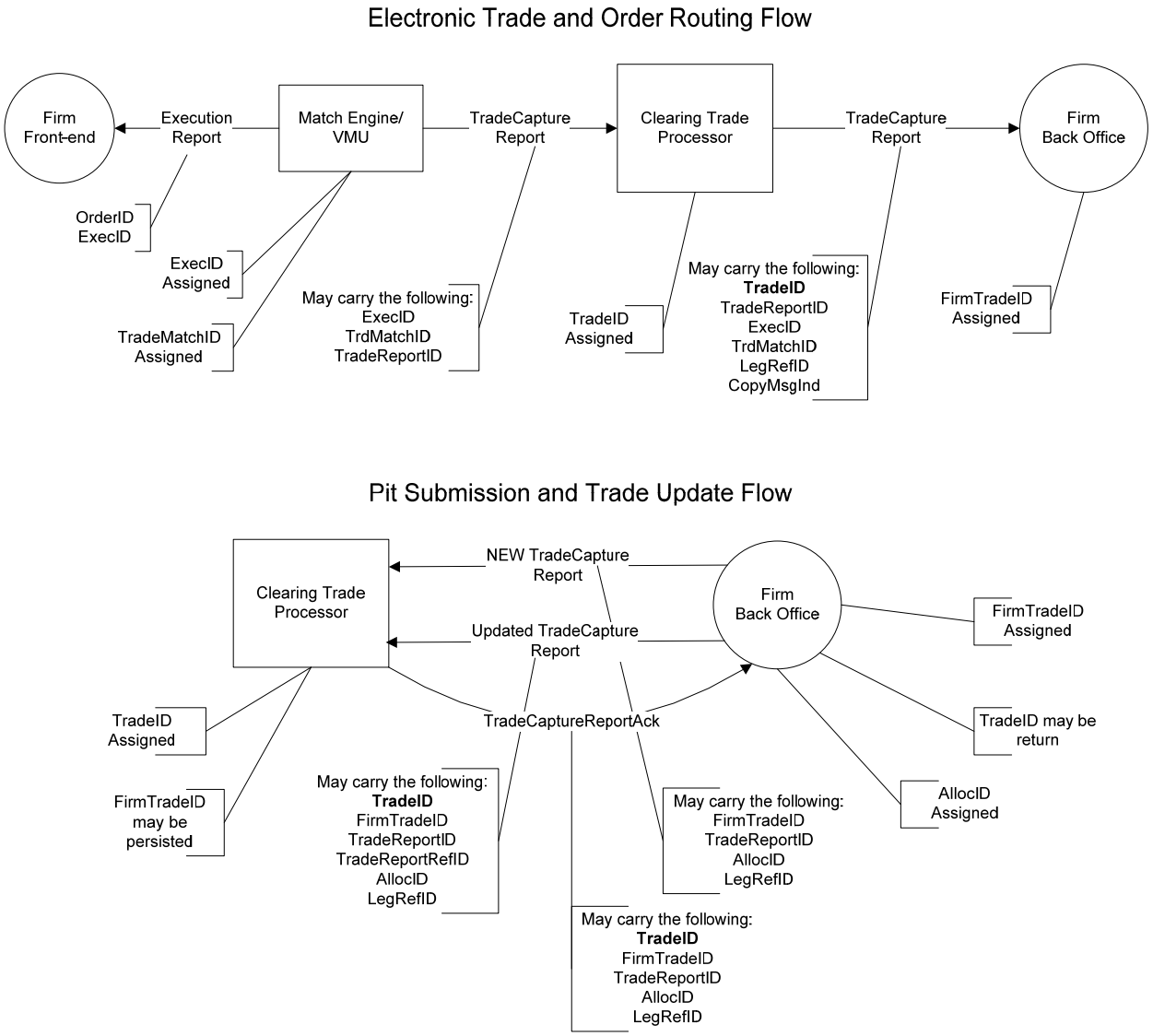
8. **TradeLegRefID** – Used when reporting an individual leg of a multi leg trade. TradeLegRefID references the leg of a multileg instrument (LegRefID) to which this trade refers. Used when MultiLegReportingType = 2 (Single Leg of a Multileg security).

9. **LegRefID** – Used to uniquely identify the leg of a trade when reporting a spread with its associated legs. Note that LegRefID may be unique when paired with TradeID or unique on its own. If the leg is reported separately LegRefID would no longer be used but would be reported in Trade ID. Generally, not used in Clearing as legs are reported individually.

Exhibit 2 illustartes trade identification in the context of electronic trade and order routing flow, and trade reporting flow.

© Copyright, 2007-2008, FIX Protocol, Limited

Exhibit 2


Electronic Trade and Order Routing Flow
Firm Front-end
Execution Report
Match Engine/ VMU
TradeCapture Report
Clearing Trade Processor
TradeCapture Report
Firm Back Office
OrderID
ExecID
ExecID Assigned
TradeMatchID Assigned
May carry the following:
ExecID
TrdMatchID
TradeReportID
TradeID Assigned
May carry the following:
TradeID
TradeReportID
ExecID
TrdMatchID
LegRefID
CopyMsgInd
FirmTradeID Assigned
Pit Submission and Trade Update Flow
Clearing Trade Processor
NEW TradeCapture Report
Updated TradeCapture Report
Firm Back Office
FirmTradeID Assigned
TradeID Assigned
TradeCaptureReportAck
TradeID may be return
FirmTradeID may be persisted
May carry the following:
TradeID
FirmTradeID
TradeReportID
TradeReportRefID
AllocID
LegRefID
May carry the following:
FirmTradeID
TradeReportID
AllocID
LegRefID
AllocID Assigned
May carry the following:
TradeID
FirmTradeID
TradeReportID
AllocID
LegRefID

Trade Identification Usage Table

The table below provides usage guidelines relating to the various identification fields used by the Firm and the Clearing System, detailing which entity or system assigns which identification as the trade moves through the reporting and clearing process.

	Description	Message Type	Trade Source	Sender	Receiver	Trans Type	Copy Message Ind	ExecID	Trade Match ID	Trade Report ID	TradeID [3]	LegRef ID [4]	TrdLeg RefID [5]	Firm Trade ID	AllocID
1	Electronic Trade reported from match engine To Clearing	Trade Capture Report	Electronic	Match Engine or VMU	Clearing Org	New	No	Assigned In Engine	Assigned in Engine	Assigned by Engine	N/A	N/A	N/A	N/A	N/A
2	Cleared Electronic Trade reported from Clearing System to Firm Back Office	Trade Capture Report	Electronic	Clearing System	Clearing Firm	New	Yes	Assigned in Engine	Assigned in Engine	Assigned By Clearing System	Assigned in Clearing System	Assigned in Clearing System	Assigned in Clearing System	Assigned in Firm Back Office	N/A
3	Trade Update sent from Firm Back Office to Clearing System	Trade Capture Report	Electronic	Clearing Firm	Clearing System	Replace	No	N/A	N/A	Assigned by Firm	Returned by Firm	Returned by Firm	Assigned in Clearing System.	Assigned in Firm Back Office	Assigned in Firm Back Office
4	New Trade sent from Firm Back Office to Clearing System	Trade Capture Report	Pit	Firm	Clearing System	New	No	N/A	N/A	Assigned by Firm	N/A	N/A	N/A	Assigned in Firm Back Office	Assigned in Firm Back Office
5	New Trade from Firm is Ack'd back by Clearing System	Trade Capture Report Ack	Pit	Clearing System	Firm	New	No	N/A	N/A	Assigned by Clearing System	Assigned by Clearing System [6]	N/A	N/A	Returned by Clearing System	Returned by Clearing System
6	Firm enters a trade through Clearing System User Interface	Trade Capture Report	Pit	Clearing System	Firm	New	Yes	N/A	N/A	Assigned by Clearing System	Assigned by Clearing System	Assigned by Clearing System	Assigned by Clearing System	Assigned in Firm Back Office	N/A
7	Clearing System	Trade Capture	Pit	Clearing	Firm	Replac	Yes	N/A	N/A	Assigned	Assigned	Assigned	Assigned	Assigne	N/A

[3] Clearing Trade ID

[4] Used for multi-leg trade reporting. Refers to the ID of a Trade Leg as specified in a multi-leg TradeCaptureReport. Not used if trade legs are reported individually

[5] Used for single leg trade reporting. Refers to the LegRefID as specified in the original multi-leg TradeCaptureReport

[6] Clearing System may use FirmTradeID provided by the Firm as the TradeID

© Copyright, 2007-2008, FIX Protocol, Limited

	matches trade and sends report to Firm	Report		System		e				by Clearing System	by Clearing System	by Clearing System	by Clearing System	d in Firm Back Office	

© Copyright, 2007-2008, FIX Protocol, Limited

Collateral Messages for Margin Management

Background

In a Risk-based Margining Model, as used in the listed derivatives industry, the clearing organization sets margin levels based upon the expected volatility of individual contracts with the amount of margin designed to cover the expected one-day price change. In this model, the risk margin calculation is done on a portfolio basis.

In the listed derivatives industry collateral is deposited at the clearing house in order to satisfy clearing margin requirements. The collateral is largely posted on a value basis (market value – haircut). In this model collateral may be actively managed independently from the overlying positions as long as the minimum requirement is met.

Business Workflow

The Clearinghouses assesses clearing margins based upon clearing member's positions. This evaluation produces a margin requirement which the members must meet using accepted forms of margin collateral. Typical forms of margin collateral include cash, letters of credit, government securities, and equity securities.

If a clearing member has a margin shortfall the clearinghouse will immediately draft their settlement cash account. The member may then choose to substitute this cash position with another form of collateral such as a government security. The clearing member would contact their custodian/depository and instruct them to transfer a sufficient quantity of securities to the clearinghouse. The depository would do this via out of band means (non-FIXML). Upon acceptance of the collateral pledge the clearinghouse will deposit the collateral and produce a Collateral Response message for the clearing member documenting the pledge. The clearing member's margin account would then carry an excess balance which the member could reduce by requesting a cash withdrawal. This transaction would also trigger a FIXML Collateral Response message and a transfer of assets, (again out of band).

This margin collateral rebalancing occurs frequently and dynamically through out each business day. At the end of the day the clearinghouse will produce Collateral Report messages detailing the closing collateral inventory positions.

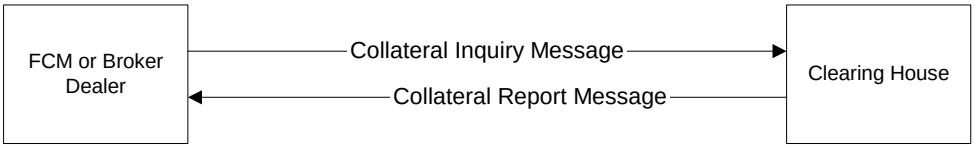
Message flow with a clearinghouse

Listed Derivatives Clearing will only use a part of the existing Collateral Management message flow since it interacts directly with customer's banks rather than the customers themselves. This makes the Collateral Assignment message which is normally sent by the collateral provider to the collateral holder unnecessary.

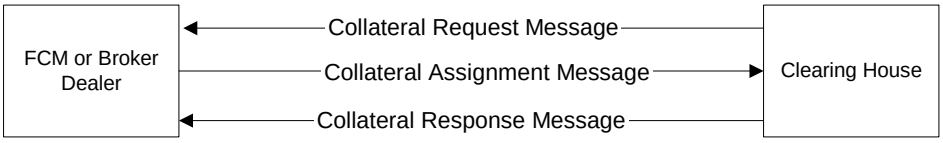
The figure below depict the message flow used by listed derivatives clearing, with comparison to the existing Collateral Management message flow.

© Copyright, 2007-2008, FIX Protocol, Limited

FIXML Specification Messaging Flow

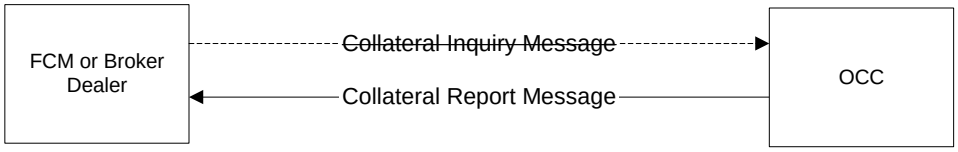


For collateral inventory listing and status, the FIXML specification model shows the Broker Dealer or FCM sending a Collateral Inquiry message to the Clearing House who in turn responds with a Collateral Report message, which lists the collateral inventory on deposit. The Clearing House may also send a Collateral Report message unsolicited.

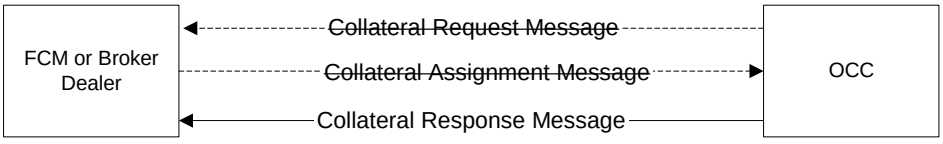


For collateral inventory updates (deposits/withdrawals), the FIXML specification model shows the Clearing House requesting additional collateral using the Collateral Request message. The FCM or Broker Dealer would then respond with a Collateral Assignment message which is used to make an assignment, replenishment or substitution. The Clearing House would then reply with a Collateral Response message which documents the collateral movement. The inventory update may be initiated by the FCM or Broker Dealer, in this case the first message passed between the two parties would be the Collateral Assignment message.

Proposed OCC Messaging Flow



OCC's proposed initial implementation of FIXML collateral messaging for open depository will not support the Collateral Inquiry message. At the end of every business day, OCC will produce Collateral Report messages for all of its participants collateral inventory. In the future if our membership requests the ability to submit Collateral Inquiry messages we will make the necessary changes to support this, but we do not believe that this functionality is desired at the onset.



OCC's proposed initial implementation of FIXML collateral messaging for collateral transactions will not support the Collateral Request and Assignment messages. OCC will produce Collateral Response messages in near real-time for every collateral movement transaction. The inputs and interfaces to OCC's depository are done through many different means, none of which are currently FIXML. OCC does not believe that the capability or willingness to change their collateral interfaces with OCC exists at the various participants we interact with, (many of which are not clearing members). If in the future this situation changes OCC will make the necessary changes to support a model closer to that outlined by the FIXML specification.

© Copyright, 2007-2008, FIX Protocol, Limited

Use of Instrument and UnderlyingInstrument component blocks

For Listed Derivatives Clearing the Instrument and UnderlyingInstrument component blocks in the Collateral Report and Collateral Response messages will be used as follows:

- When reporting about a collateral position specifically assigned to an options position, the position information will be carried in the Instrument block and the collateral information will be in the UnderlyingInstrument block.
- When reporting about a collateral position made on a valued basis there is no overlying position or trade to place in the Instrument block. The Instrument block will be excluded and the collateral will be consistently reported in the UnderlyingInstrument block.

Marginable vs. Valued Collateral

Securities are pledged to the clearinghouse on a Valued or Marginable basis. Collateral types which are accepted on a valued basis include equities, letters of credit, currency, money market mutual funds, and corporate, government and agency debt. Equities, short term treasuries and cash may also be specifically assigned to certain option positions on a marginable basis. When this is done the hedged positions are removed from the margin calculation of a given portfolio.

The simple case is a Valued Security pledge. The clearing member pledges 100 shares of IBM and the clearinghouse gives them some amount of margin credit. (100 shares IBM) x (Share price IBM - 82.12) x (Haircut – 30%) = Collateral Value ($5,748.40).

The more complex case is a Specific Deposit. Depending upon volatility, 1 short IBM call may increase a clearing member's margin requirement by $4,500.00. The clearing member may offset this $4,500.00 by specifically pledging 100 shares of IBM stock to offset the obligation of the short call.

In both of these cases the collateral being pledged is IBM. The CollApplType field (tag 1043) is used to identify whether the collateral being pledge is to be applied specifically against a position or against the entire portfolio on a valued basis.

Covered Spreads and other User Defined Spreads using Security Definition Messages

Covered Spreads are an important subset of User Defined Spreads. At execution, Covered Spreads allow the risk of an option strategy to be offset by taking a position in the underlying instrument. These strategies are referred to as being "delta neutral". A Covered Spread consists of a listed or non-listed option strategy such as a calendar spread with one or more pre-defined underlying instruments specified. For Listed Derivatives, one or more Futures instruments will be used to "cover" the option strategy. The Option Ratio is carried in the option leg to which it applies.

The business rules governing the use of Covered Spreads in Listed Derivatives are as follows:

- An option strategy can only be covered with two futures if there are at least two different option maturities
- No option leg can be specified more than once
- No covering future can be specified more than once
- For covered spreads, the inbound Security Definition ratio can only be between -99.99 to +99.99
- For covered call outright, the inbound Security Definition ratio can only be between 0.01 and +1.00
- For covered put outright, the inbound Security Definition ratio can only be between -1.00 and - 0.01

© Copyright, 2007-2008, FIX Protocol, Limited

Usage examples

Covered Spread Request

A Covered Spread Request consists of a listed or non-listed option strategy such as a calendar spread with one or more pre-defined covering futures specified. The option strategy being covered in this example is a straddle which can be expressed as ST: GEZ5 C9625 P9625. The straddle itself is not explicitly designated – just the legs. Option legs and covering futures are specified in the Instrument Leg repeating group. The Ratio is carried in the option leg to which it applies.

Security Definition Request

Tag	Field Name		Req'd	Value
	Standard Header		Y	MsgType = c (lowercase)
320	SecurityReqID		Y	Unique value assigned by client
321	SecurityRequestType		Y	"1" = Request Security identity for specifications provided – name of security is not provided
component block <Instrument>			N	User Defined Covered Spread specified here
→	55	Symbol	Y	"GE"
→	762	SecuritySubType	Y	Indicates if instrument being defined is a Covered Spread "COV" = Covered Spread
555	NoLegs		Y	Set to "3"
→	**component block <InstrumentLeg>**		N	Straddle Option Leg1
→	602	LegSecurityID	Y	CME111111
→	620	SecurityDesc	Y	GEZ5 C9625
→	624	LegSide	N	"1"=Buy
→	623	LegRatioQty	N	"1"
→	**component block <InstrumentLeg>**		N	Straddle Option Leg2
→	602	LegSecurityID	Y	CME22222
→	620	SecurityDesc	Y	GEZ5 P9625
→	624	LegSide	N	"2"=Sell
→	623	LegRatioQty	N	"1"
→	**component block <InstrumentLeg>**		Y	Covering Future
→	1017	LegOptionRatio	Y	Assume that Leg1 Ratio = .75 and Leg2 Ratio = -.5 LegPositionRatio = Ratio1 and Ratio2 = .25 Total quantity of Futures to buy is: (.25 x Option Strategy Order Qty)
→	602	LegSecurityID	Y	CME333333

© Copyright, 2007-2008, FIX Protocol, Limited

→	620	SecurityDesc	Y	"GEZ5"
→	623	LegRatioQty	N	"1"
→	637	**LegLastPx***	Y	"962500" = Futures Price
827	ExpirationCycle		N	"0" = Expire on trading session close
263	SubscriptionRequestType		N	Not Used
	Standard Trailer		Y	

Covered Spread Response

A Covered Spread Response consists of a listed or non-listed option strategy such as a calendar spread with one or more pre-defined covering futures specified. Option legs and covering futures are specified in the Instrument Leg repeating group. The Price Ratio is carried in the option leg to which it applies.

Security Definition

Tag	Field Name		Req'd	Value
	Standard Header		Y	MsgType = d (lowercase)
320	SecurityReqID		Y	Unique value assigned by client
322	SecurityResponseID		Y	Unique value assigned by host
323	SecurityResponseType		Y	"1" – Accept security proposal as is
component block <Instrument>			N	User Defined Covered Spread specified here
→	55	Symbol	Y	"GE"
→	48	SecurityID	Y	CME444444
	107	SecurityDesc	Y	GE:COV:03
→	762	SecuritySubType	Y	Indicates if instrument being defined is a Covered Spread "COV" = Covered Spread
555	NoLegs		Y	Set to "3"
→	**component block <InstrumentLeg>**		N	Straddle Option Leg1
→	602	LegSecurityID	Y	CME111111
→	620	SecurityDesc	Y	GEZ5 C9625
→	624	LegSide	N	"1"=Buy
→	623	LegRatioQty	N	"1"
→	**component block <InstrumentLeg>**		N	Straddle Option Leg2
→	602	LegSecurityID	Y	CME22222
→	620	SecurityDesc	Y	GEZ5 P9625
→	624	LegSide	N	"2"=Sell

© Copyright, 2007-2008, FIX Protocol, Limited

→	623	LegRatioQty	N	"1"
→	**component block <InstrumentLeg>**		Y	Covering Future
→	1017	LegOptionRatio	Y	Assume that Leg1 Ratio = .75 and Leg2 Ratio = -.5 LegOptionRatio = Ratio1 and Ratio2 = .25 Total quantity of Futures to buy is: (.25 x Option Strategy Order Qty)
→	602	LegSecurityID	Y	CME333333
→	620	SecurityDesc	Y	"GEZ5"
→	623	LegRatioQty	N	"1"
→	637	**LegLastPx***	Y	"962500" = Futures Price
827	ExpirationCycle		N	"0" = Expire on trading session close
263	SubscriptionRequestType		N	Not Used
	Standard Trailer		Y	

Product Reference Usage

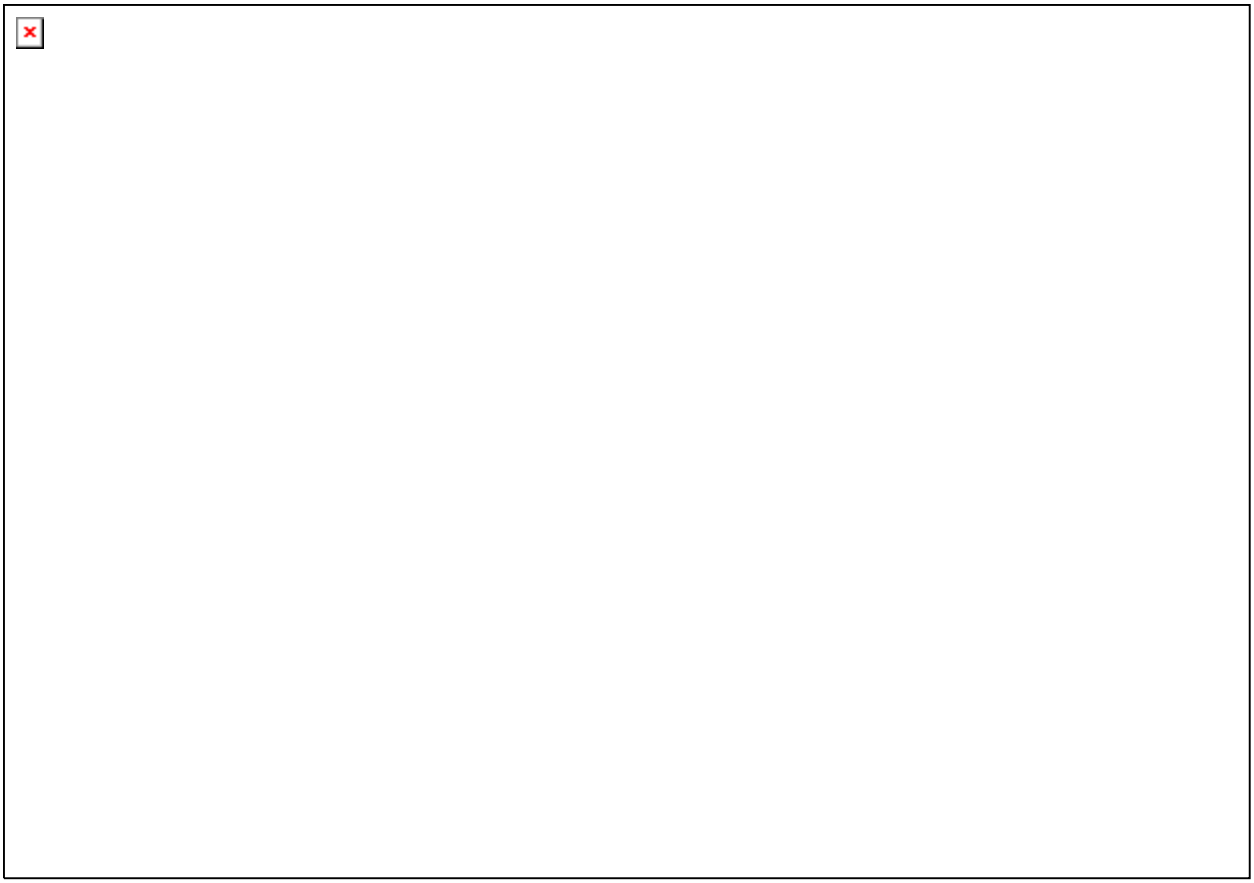
Introduction

Product Reference Data is an essential aspect of securities automation whether the context is electronic trading or clearing. A good product interface reflects the structure of the instruments offered by an organization and provides a thorough definition of each instrument. This product reference model represents financial instruments and the relationship between these instruments for the listed derivatives industry. This product reference model provides support for derivatives trading that may include Futures, Options on Futures and Equity Options. Some coverage is also provided for FX instruments. The model strives to provide comprehensive coverage of all business, economic, and operational characteristics of an instrument.

Business Workflow

Product reference information is generally refreshed on a daily basis in order to convey changes to any properties of an instrument. Users of an exchange or clearing entity service must obtain and load the Product Reference Data. The data can be made available in a variety of ways - static file downloads, request/response mechanism, subscription, or constant circulation over a broadcast feed.

© Copyright, 2007-2008, FIX Protocol, Limited



© Copyright, 2007-2008, FIX Protocol, Limited

Product Reference Model

Product Reference Data is provided as a base set of information that must be established at a customer site prior to conducting business with the entity that is offering those products; usually an exchange or clearing entity although the products may also be offered by a sell-side firm or vendor - or on behalf of an exchange or clearing entity by a sell-side firm.

Users of Product Reference Data should be able to establish a local set of instruments that precisely reflects the products being offered by an exchange or clearing entity. It is important to note that these instruments are also referred to as "contracts" and carry a legal obligation to fulfill the specified terms of the contract. For Listed Derivatives, this involves a broad array of rules that govern where and how the instrument is listed, quoted, traded, cleared, settled and delivered.

In a futures market place, futures instruments are the first order derivative and are based on a "physical" underlying such as corn, treasury notes, dollars on deposit in Europe, or the equity market index. For a futures market, the characteristics of the underlying are usually not elaborated as standalone instruments but may be summarized in the definition of the Futures instrument. Options on Futures are a second order derivative and are based on an underlying Future. A Futures instrument may have several "option classes" or "option families" that use it as an underlying in order to accentuate various aspects of the Future. For example, a Eurodollar Future may have five different Mid-curve Option family in which each Series has its own set of strikes. **A goal of the product reference model is to allow a single message to represent this structure; An Underlying, An Option family, and all strikes for the family.**

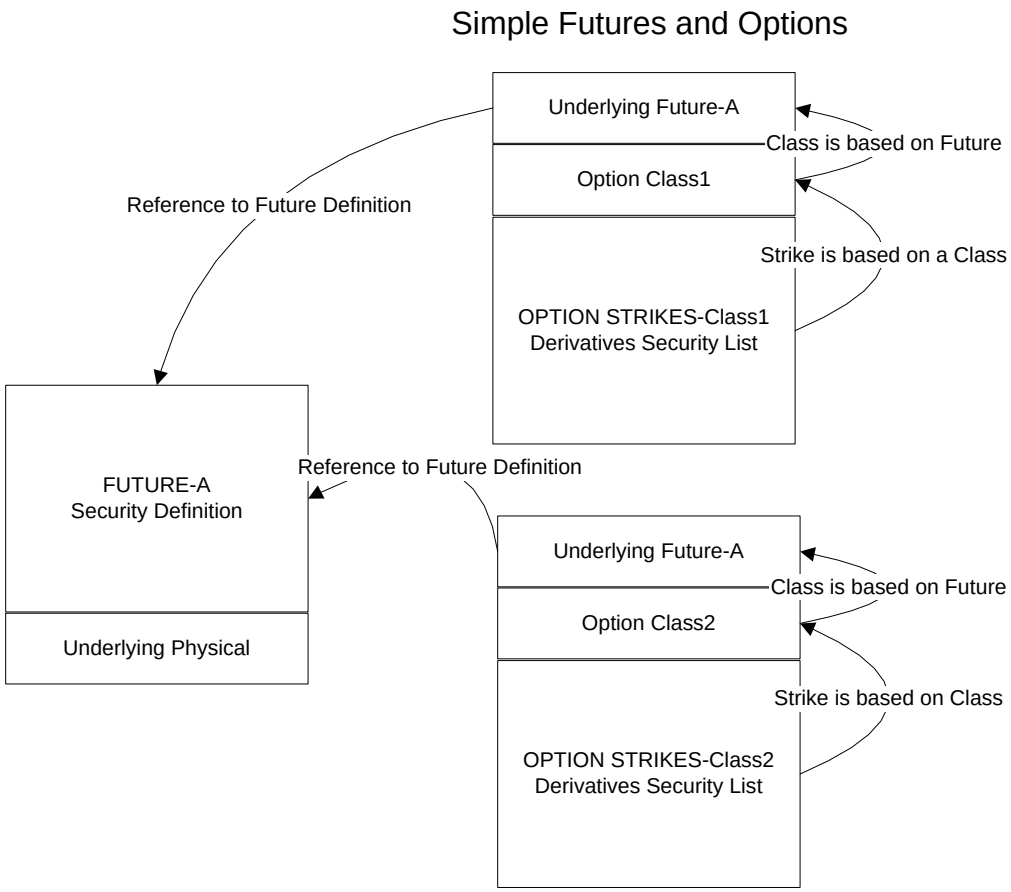
In an equity option market place equity options are first order derivatives based on an underlying stock. In this case it is more common to treat the underlying as a standalone instrument.

In an FX market place, spot can be considered the cash instrument on which forward and swap instruments are based.

Strategies on Futures and Options on Futures are another variation that must be supported by the product model. A strategy may consist of basic Futures legs, basic Options legs, Futures legs and Options legs, Futures strategies and Options strategies. A strategy on a strategy is typically the most complex instrument that must be represented in a Product Reference Model.

The diagram below illustrates the model in which a Futures contract is defined as a standalone instrument using the Security Definition message (Future-A). The Option family is then defined using the Derivatives Security List message (Option family1). The individual option instruments that belong to a given family (or "class") are then elaborated in relation to that family (Option Strikes1). The Option family references the Futures instrument in order to tie back to a specific underlying instrument.

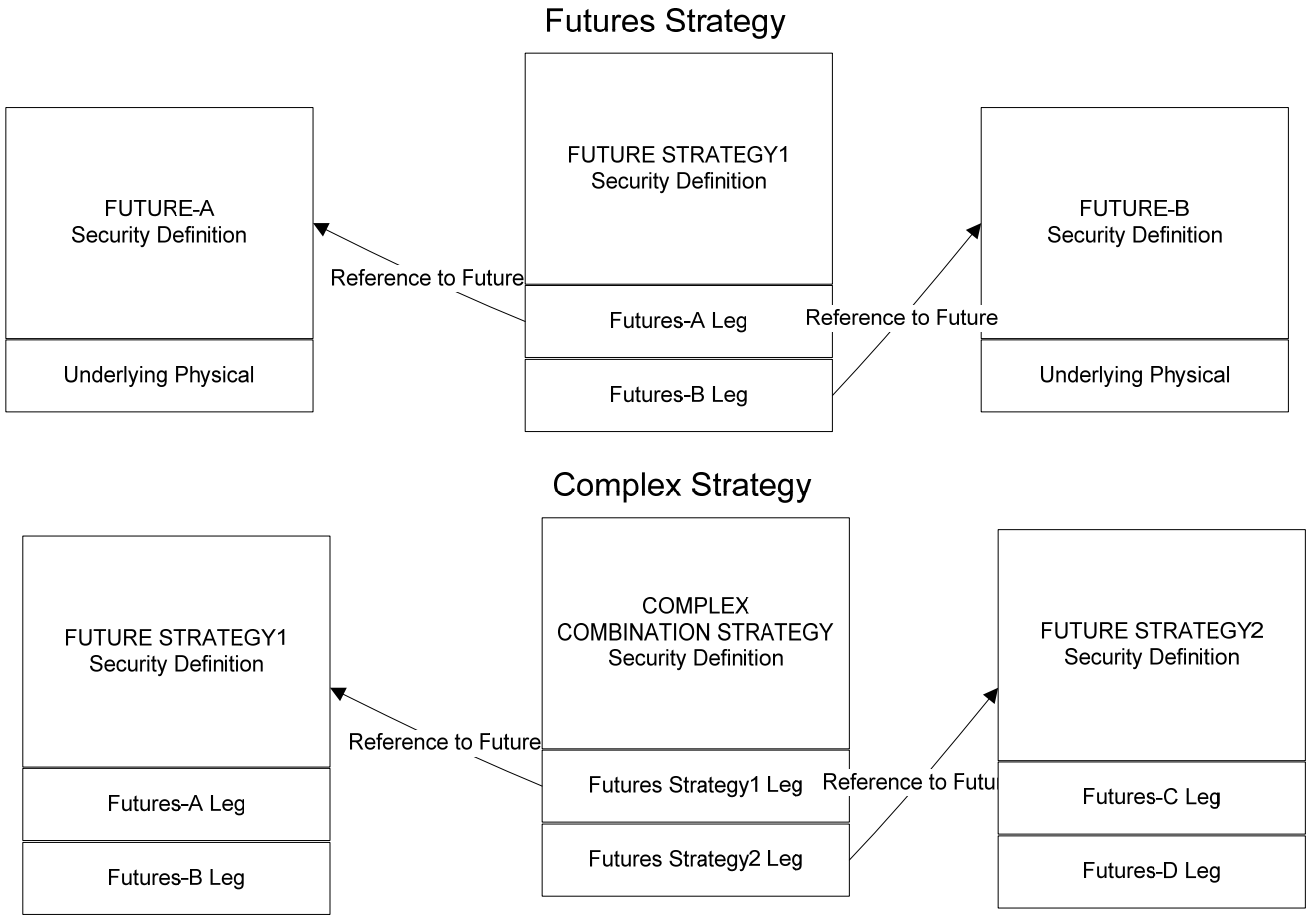
© Copyright, 2007-2008, FIX Protocol, Limited



It is worth noting that options may also be modeled on an individual basis using the Security Definition message or as a part of a list using the Security List message.

The diagrams below illustrate the how strategy instruments are defined. In the Futures Strategy diagram Futures Strategy1 is comprised of Future-A and Future-B. An Option Strategy would be similarly constructed. In the Complex Strategy diagram the Complex Combination Strategy is comprised of Futures Strategy1 and Futures Strategy2. A Complex Option Strategy would be similarly constructed.

© Copyright, 2007-2008, FIX Protocol, Limited



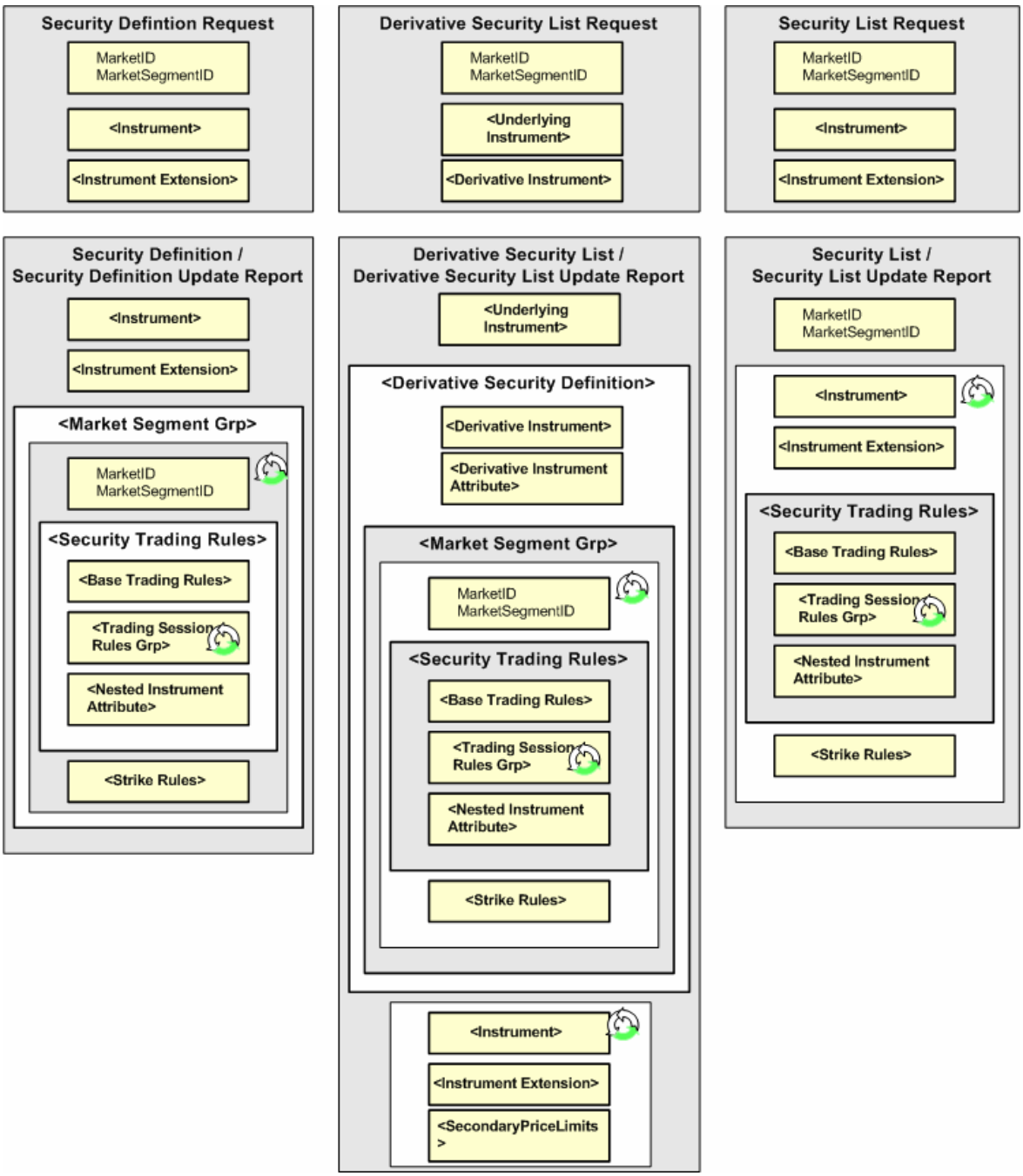
Strike Specification

In general, strikes (option instruments) should be specifically enumerated. This is due to the fact that option strategies are based on specific strikes and need to contain a reference to the strikes that make up the strategy. Otherwise, the strategy itself is defined in a nondeterministic manner.

© Copyright, 2007-2008, FIX Protocol, Limited

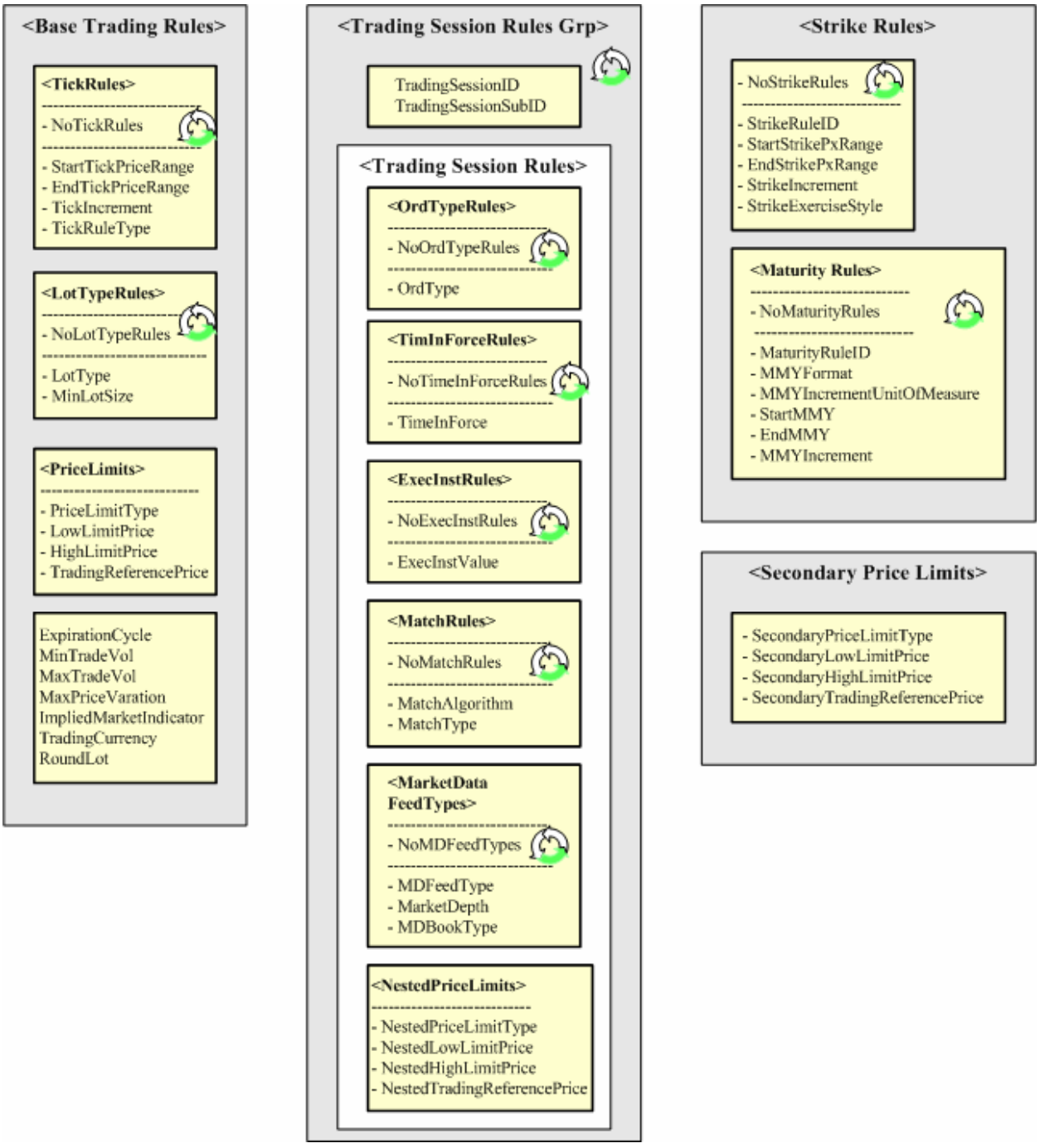
New Security Message Structures

The Security messages are restructured as shown in the diagrams below. The primary features that are being added are the ability to specify trading rules by Market Segment and Trading Session. Requests can be issued using MarketID and MarketSegmentID. Trading rules have been separated into two components; Base Trading Rules which contain the basic set of trading rules for a given Market Segment, and Trading Session Rules Grp which contains trading rules which are specific to a Trading Session. The message structures shown below include the Security Definition set, Derivative Security List set, and Security List set.



Trading Rules Components

The diagrams below illustrate the trading rules component blocks called. The Base Trading Rules component contains trading rules that are specified at the level of Market Segment. The Trading Session Rules Grp component contains trading rules that are specified at the Trading Session level. Each component contains its own set of individual trading rules component.



Security Definition

The Security Definition message is used to define an individual security or set of securities.

Security Definition Usage Guidelines

Recommended Uses:

- Allows a set of individual securities to be requested for a single Market Segment. The Security Definition must be returned for the specified Market Segment
- Allows a request to be made independent of Market Segment. The Security Definition may carry all relevant Market Segments and their corresponding trading rules.
- Allows stand-alone use in which a comprehensive Security Definition is generated for all Market Segments in which that security participates
- If the Market Segment message is in use, Trading rules are to be included only if different than default rules. Otherwise, only the relevant Market Segment identifiers are included

Explanation of Message Structure:

- Security Definition Request carries MarketID + MarketSegmentID at the main level as optional fields.
- Security Definition has a repeating Market Segment Group which allows all Market Segments in which a security participates to be specified in the definition of that security.
- Only a single instance of the group will be used if a specific Market Segment has been specified on a request. Multiple instances will be used if multiple Market Segments have been requested (by not specifying Market Segment on request).
- If Market Segment is not applicable then MarketSegmentID = "[N/A]" (without the quotes) will be used and trading rules will be provided.
- If Market Segment message is to be referenced for the trading rules only the Market Segment identifiers should be provided

Derivative Security List

The Derivative Security List message is used to send a predefined list of securities (usually options) based on a common underlying and option class. It can also be used to send the rules for security creation (usually options) which imply the existence of a set of securities

Derivative Security List Usage Rules

Recommended Uses:

- Allows a set of option classes to be requested for a single Market Segment. The option classes must be returned for the specified Market Segment.
- Allows a request to be made independent of Market Segment. The option classes may carry all relevant Market Segments and their corresponding trading rules
- Allows stand-alone use in which a comprehensive set of option classes are generated for all Market Segments in which those classes participate.
- If the Market Segment message is in use, Trading rules are to be included only if different than default rules. Otherwise, only the relevant Market Segment identifiers are included

***Explanation of** Message Structure*

- Derivative Security List Request will carry MarketID + MarketSegmentID at the main level as optional fields.

© Copyright, 2007-2008, FIX Protocol, Limited

- Derivative Security List will have a repeating Market Segment Group which allows all Market Segments in which a derivative class participates to be specified in the definition of that class.
- Only a single instance of the group will be used if a specific Market Segment has been requested. Multiple instances will be used if multiple Market Segments have been requested (by not specifying Market Segment on request).
- If Market Segment is not applicable then MarketSegmentID = "[N/A]" (without the quotes) will be used and trading rules will be provided.
- If Market Segment message is to be referenced for the trading rules only the Market Segment identifiers should be provided

Security List

The Security List message is used to send a related list of securities in which each security is individually defined within the list

Security List and Security List Request Usage Rules

Recommended Uses:

- Allows a list of securities to be requested and reported for a single Market Segment.
- Allows trading rules to be specified if different from the trading rules specified on the Market Segment message.

Explanation of Message Structure –

- Security List Request and Security List will carry MarketID + MarketSegmentID at the main level as optional fields.
- Security List will carry the SecurityTradingRules (trading rules) and StrikeRules in the instrument repeating group.

Market Segment and Venue

Market Segment and Venue is another important concept. Market Segment is the construct that allows Venue to be represented. Venue is a common term for describing the general means of participating in or accessing a market. Usually, the market is represented in electronic terms, person-to-person on a trading floor or in a privately negotiated fashion. All instruments are defined within the context of a venue in which certain trading characteristics of the instrument can change based on the venue in which they are traded. Likely venues are “Electronic”, “Pit”, and “Ex-Pit” although it is possible that there can be greater granularity in identifying a venue. A Market Segment group block has been added to the Security Trading Definition Block in order to represent the role of the venue in an instrument definition

Message Flows

Request for Instrument Definition

The DerivativeSecurityListRequest can be used to request a snapshot or snapshot supplemented with updates. In static models it is not necessary to subscribe to instrument definitions as the data is made available on a retrieval basis.

© Copyright, 2007-2008, FIX Protocol, Limited

Instrument Definitions Snapshot

The Exchange and Clearing Entity will provide a complete set of instruments at either the beginning or end of the trading session. A set of instrument updates will also be provided. The Security Definition Update Report will be used to update those instruments which may have changed during the course of the business day.

Updates to Instrument Snapshot

An Update Action Code will be provided at the option class level to indicate if the class is being added, modified, or deleted. If deleted, all strikes in that series should be considered deleted. Further an List Update Action Code will be provided at the Derivatives Security level to indicate if an options strike (or other instrument type) is being added, modified, or deleted.

Exchange or Clearing Entity		Participant
Complete Product Reference Dictionary		
Sends • **Security Definition for Futures** • **Derivatives Security List for Options** • **Security Definition for Strategies**	← →	**Sends Request for:** **Security Definition** **Security List** **Derivative Security List** **Receives Product Definitions**
Updates to Product Reference Dictionary		
Sends • **Security Definition Update Report for Futures** • **Derivatives Security List Update Report for Options** • **Security Definition Update Report for Strategies**	→	**Receives Product Definition Updates** **Note: participant must first receive the base set of product reference data in order to apply the updates properly. Updates on their own are not useful.**

Usage exmples

This example shows a Derivative Security List message being used to specify a set of related option instruments. The message structure begins by specifying the underlying. The DerivSecDef includes the option class information which are the common properties shared by the option instruments. DerivSecDef also includes the MktSegGrp block which specifies all the trading rules for the set of options. Finally, the specific options are individually listed using the Instrmt block. Each option must specify the venue on which it is listed historical strikes do not conform to the strike rules.

© Copyright, 2007-2008, FIX Protocol, Limited

Option Class

```
<FIXML>
<DerivSecList=”1234567” Ccy=”USD” BizDt=”2007-06-01”>
  <Undly ID=”ED” Src=”H”             //Underlying for the Series
       MMY=”200706”
       Exch=”CME”
       SecTyp=”FUT”
       CFI=”FFDCSO” >
    <SecAlt ID=”800103” Src=”8”/>       //Underlying Instrument ID (IXM Num)
  </Undly>
  <DerivSecDef>             //Group containing all Series info
    <Deriv ID=”E0” Src=”H”           //series product group
       Sfx=”E0 W1”                //series identifier
       Exch=”CME”
       Desc=”E0 June 2007 W1”        //series name
       MMY=”200706”               //series period code
       SecTyp=”OOF”
       StrkCcy=”USD”
       MatDt=”20070617”           //series expiration
       StrkLstMeth=”2”               //prelisted and user requested
       AsgnMeth=”R”               //random assignment
       ExerStyle=”A”              //American style
       UOM=”Int”
       UOMQty=”1000000”
       Mult=”250”
       PxUOM=”USD” PxUOMQty=”1” PxQteMeth=”Int”   //Price Quote props
       SettlMeth=”C”
       ValueTypCode=”PREM”>
        <SecAlt ID=”E0 W1” Src=”N”/>        //Globex Ticker
        <SecAlt ID=”E0 W1” Src=”O”/>        //Floor Ticker
      <Evnt EventTyp=”5” Dt=”20061013” Tm=”202500000” />  //First day of
  trading
      <Evnt EventTyp=”7” Dt=”20070616” Tm=”202500000” /> //Last day of trading
      <Evnt EventTyp=”16” Dt=”20070616” Tm=”202500000” /> //Position removal
  date
    </Deriv>
      <DerivAttrib>
       <Attrib Typ=”23” Val=””04”/>        //Globex Tick Table
       <Attrib Typ=”27” Val=”3” />         //Fractional Price precision
       <Attrib Typ=”25” Val=”32” />        //Price Denominator
       <Attrib Typ=”26” Val=”4” />         //Price Numerator fraction
       <Attrib Typ=”28” Val=”3” />         //StrikePx Display precision
       <Attrib Typ=”29” Val=”T” />         //Tradable,Non-tradable ind
      </DerivAttrib>
      <MktSegGrp
       MktID=”CME” MktSegID=”Electronic”>
       <StrkRules
         StartStrkPxRng=”90.0”          //starting strike price range
      EndStrkPxRng=”92.5”            //ending strike price range
      StrkIncr=”.010” />          //Strike increment
       <StrkRules
         StartStrkPxRng=”92.505”        //starting strike price range
      EndStrkPxRng=”95.0”            //ending strike price range
      StrkIncr=”.025” />          //strike increment
    <SecTrdgRules
       MaxTrdVol=”1000”        //Maximum order quantity
       MinTrdVol=”1”           //Minimum order quantity
       ImpMktInd=”1”           //Implied market indicator
```

© Copyright, 2007-2008, FIX Protocol, Limited

```
          <TickRules
             StartTickPxRng="90.0"         //starting tick price range
          EndTickPxRng="92.5"         //ending tick price range
          TickIncr=".005"             //Tick increment
             TickRuleTyp="0" />             //Tick Rule Type
           <TickRules
             StartTickPxRng="92.505"       //starting tick price range
          EndTickPxRng="95.0"         //ending tick price range
          TickIncr=".010"             //Tick increment
             TickRuleTyp="0" />             //Tick Rule Type
           <InstrmtAttrib>               //Eligible trade types for venue
                <Attrib Typ="24" Val="1" /> /    /Block eligibility
                <Attrib Typ="24" Val="2" />      //EFP eligibility
              </InstrmtAttrib>
    </SecTrdgRules>
  </MktSegGrp>
  <MktSegGrp
    MktID="CME" MktSeg="Floor">
    <StrkRules
      StartStrkPxRng="90.0"          //starting strike price range
   EndStrkPxRng="92.5"           //ending strike price range
   StrkIncr=".005" />          //Strike increment
    <StrkRules
      StartStrkPxRng="92.505"        //starting strike price range
   EndStrkPxRng="95.0"           //ending strike price range
   StrkIncr=".010" />          //strike increment
 <SecTrdgRules
    MaxTrdVol="1000"        //Maximum order quantity
    MinTrdVol="1"           //Minimum order quantity
    ImpMktInd="1"           //Implied market indicator
          <TickRules
               StartTickPxRng="90.0"       //starting tick price range
            EndTickPxRng="92.5"         //ending tick price range
            TickIncr=".010"           //Tick increment
               TickRuleTyp="0" />          //Tick Rule Type
             <TickRules
               StartTickPxRng="92.505"        //starting tick price range
            EndTickPxRng="95.0"         //ending tick price range
            TickIncr=".010"           //Tick increment
               TickRuleTyp="0" />          //Tick Rule Type
              <InstrmtAttrib>           //Eligible trade types for venue
                <Attrib Typ="24" Val="1" /> /    /Block eligibility
                <Attrib Typ="24" Val="2" />      //EFP eligibility
                </InstrmtAttrib>
            </SecTrdgRules>
 </MktSegGrp>
    </DerivSecDef>
      <Instrmt Sym="GEH7C94675" Desc="E0 Jun07 C 94675""
 PutOrCall="1"              //Call
 StrkPx="94.675"
 <PriceLimit2
    PxLmtTyp="0"            //limits in price terms
    HighLmtPx2="2.50"       //upper price range
 LowLimitPx2="2.00"         //lower price range
 TrdgRefPx2="2.25"/>        //mid range
    <Pty ID="Globex" R="73">      //Venue for which strike is eligible
    <SecAlt ID="700103" Src="8"/>    //Instrument ID (IXM Number)
  </Instrmt>
  <Instrmt Sym="GEH7C94700" Desc="E0 Jun07 C 94700""
 PutOrCall="1"              //Call
```

© Copyright, 2007-2008, FIX Protocol, Limited

```
   StrkPx="94.700"
   <Pty ID="Globex" R="73">         //Venue for which strike is eligible
   <SecAlt ID="700104" Src="8"/>       //Instrument ID (IXM Number)
    </Instrmt>
        <Instrmt Sym="GEH7P94675" Desc="E0 Jun07 P 94675""
   PutOrCall="0"              //Put
   StrkPx="94.675"
   <Pty ID="Globex" R="73">         //Venue for which strike is eligible
   <SecAlt ID="700105" Src="8"/>       //Instrument ID (IXM Number)
    </Instrmt>
    <Instrmt Sym="GEH7P94700" Desc="E0 Jun07 P 94700""
   PutOrCall="0"              //Put
   StrkPx="94.700"
   <Pty ID="Globex" R="73">         //Venue for which strike is eligible
   <SecAlt ID="700106" Src="8"/>       //Instrument ID (IXM Number)
    </Instrmt>
   </DerivSecList>
<FIXML>
```

Outright Future

```
<FIXML>
<SecDef RptID="1234567" Ccy="USD" BizDt="2007-06-01">
  <Instrmt ID="ED" Src="H"          //product code
  MMY="200706"                //period code
  Exch="CME"
  SecTyp="FUT"
  CFI="FFDCSO"             //CFI code – standard/outright
  Desc="ED Jun 2007"          //Clearing alias
  MatDt="20070615"               //settlement date
  Sym="GEHZ"               //Globex symbol
  SecGrp="GE"              //Globex product group
  UOM="Int" UOMQty="1000000"         //unit of measure
  Mult="250"                 //price multiplier
  PxUOM="USD" PxUOMQty="1" PxQteMeth="Int" //Price unit of measure
  SettlMeth="C" ValueTypCode="FUT">
     <SecAlt ID="800103" Src="8"/>        //Instrument ID (IXM Number)
     <SecAlt ID="GE" Src="N"/>       //Globex Ticker
     <SecAlt ID="ED" Src="O"/>       //FloorTicker
   <Evnt EventTyp="5" Dt="20061013" Tm="202500000" />  //First day of trading
   <Evnt EventTyp="7" Dt="20070316" Tm="202500000" /> //Last day of trading
   <Evnt EventTyp="14" Dt="20070316" Tm="202500000" /> //First Intent Date
   <Evnt EventTyp="15" Dt="20070316" Tm="202500000" /> //Last Intent Date
   <Evnt EventTyp="10" Dt="20070316" Tm="202500000" /> //First Delivery Date
   <Evnt EventTyp="11" Dt="20070316" Tm="202500000" /> //Last Delivery Date
  date
   <Evnt EventTyp="12" Dt="20070316" Tm="202500000" /> //Initial inventory due
  date
   <Evnt EventTyp="13" Dt="20070316" Tm="202500000" /> //Final inventory due
  date
   <Evnt EventTyp="16" Dt="20070316" Tm="202500000" /> //Position removal date
  </Instrmt>
   <InstrmtExt>
      <Attrib Typ="23" Val=""04"/>        //Globex Tick Table
      <Attrib Typ="27" Val="3" />         //Fractional Price precision
      <Attrib Typ="25" Val="32" />        //Price Denominator
      <Attrib Typ="26" Val="4" />         //Price Numerator fraction
      <Attrib Typ="29" Val="T" />         //Tradable,Non-tradable ind
```

© Copyright, 2007-2008, FIX Protocol, Limited

```
   </InstrmtExt>
   <MktSegGrp
      MktID="CME" MktSegID="Electronic">
   <SecTrdgRules
      MaxTrdVol="1000"        //Maximum order quantity
      MinTrdVol="1"           //Minimum order quantity
      ImpMktInd="1"           //Implied market indicator
            <PriceLimits
            PxLmtTyp="0"            //limits in price terms
            HighLmtPx="2.50"        //high limit price
      LowLimitPx="2.00"       //low limit price
      TrdgRefPx="2.25" />        //mid price
            <TickRules
                  StartTickPxRng="90.0"   //starting tick price range
                EndTickPxRng="92.5"   //ending tick price range
                TickIncr=".005"    //Tick increment
                   TickRuleTyp="0" />    //Tick Rule Type
            <TickRules
                   StartTickPxRng="92.505"  //starting tick price range
                EndTickPxRng="95.0"   //ending tick price range
                TickIncr=".010"    //Tick increment
                   TickRuleTyp="0" />    //Tick Rule Type
            <InstrmtAttrib Typ="24" Val="1" /> //Block eligibility
            <InstrmtAttrib Typ="24" Val="2" /> //EFP eligibility
      </SecTrdgRules>
  </MktSegGrp>
  <MktSegGrp
      MktID="CME" MktSeg="Floor">
   <SecTrdgRules
      MaxTrdVol="1000"        //Maximum order quantity
      MinTrdVol="1"           //Minimum order quantity
      ImpMktInd="1"           //Implied market indicator
            <PriceLimits
            PxLmtTyp="0"            //limits in price terms
            HighLmtPx="2.50"        //???????????
      LowLimitPx="2.00"       //??????????
      TrdgRefPx="2.25"/>         //??????????
            <TickRules
                StartTickPxRng="90.0"       //starting tick price range
              EndTickPxRng="92.5"         //ending tick price range
              TickIncr=".010"          //Tick increment
                 TickRuleTyp="0" />          //Tick Rule Type
               <TickRules
                 StartTickPxRng="92.505"        //starting tick price range
              EndTickPxRng="95.0"         //ending tick price range
              TickIncr=".010"          //Tick increment
                 TickRuleTyp="0" />          //Tick Rule Type
            <InstrmtAttrib Typ="24" Val="1" /> //Block eligibility
            <InstrmtAttrib Typ="24" Val="2" /> //EFP eligibility
              </SecTrdgRules>
     </MktSegGrp>
</SecDef>
<FIXML>
```

© Copyright, 2007-2008, FIX Protocol, Limited

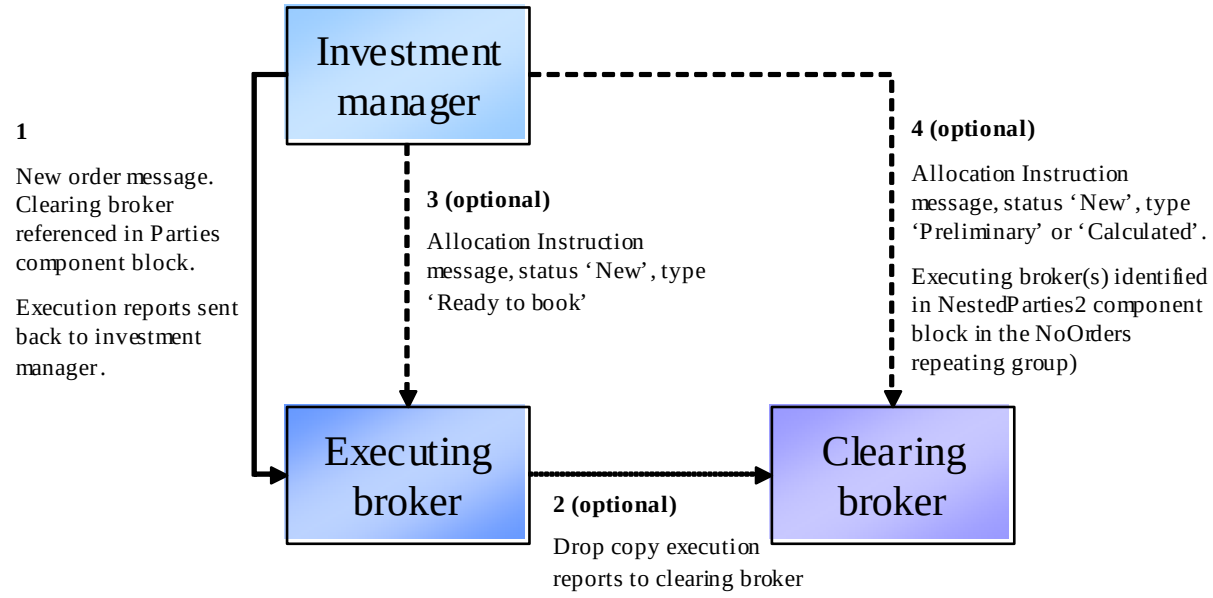
PRODUCT: EQUITIES

Step-Outs and Give-Ups

The new order messages allow a single clearing broker to be identified through use of the Parties component block with PartyRole = 4, Clearing Firm (in the event that the order is to be stepped out to multiple clearing brokers, the NestedParties2 component block in the NoAllocs group should be used, with each entry in the NoAllocs group denoting the quantity to be given up or stepped out to each broker).

The executing broker can optionally send copies of the order executions through to the clearing broker(s) real time using execution report messages. This flow is clearly not relevant in cases where communication to the clearing broker is managed through a central clearing house or similar organisation (e.g. as in the futures markets).

The investment manager provides booking instructions to both the executing and clearing brokers. Where the executing broker does not need to know the details of the underlying funds, a 'ready to book' allocation instruction can be used to tell the executing broker to book the order(s) out and settle against the clearing broker(s). The allocation details themselves are communicated from the investment manager to the clearing broker(s) using an allocation instruction message of type 'Preliminary' or 'Calculated'. This message contains a reference to the Executing Broker in the NestedParties2 field in the NoOrders repeating group (PartyRole = 1, Executing Firm).

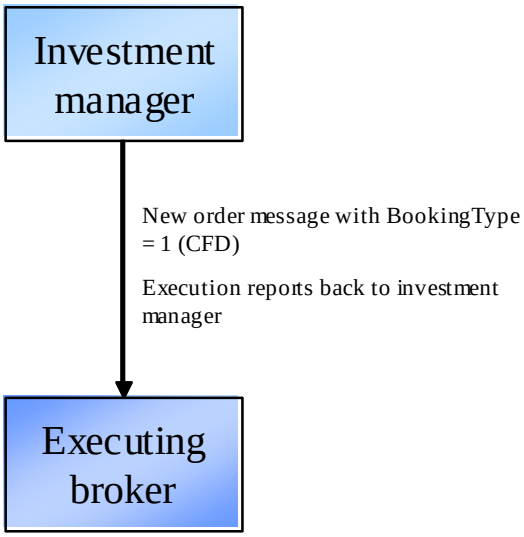


This flow also supports the scenario where the investment manager has a block order which is then sent out (in parts) to a number of executing brokers, all to be settled by the same clearing broker. In this case, each executing broker receives a 'ready to book' allocation from the investment manager for their order(s) and the clearing broker receives a single allocation message for the entire block. This latter message will reference the client order ids on each order (which can be used to match up to the execution reports from the executing brokers) and the executing broker id.

© Copyright, 2007-2008, FIX Protocol, Limited

CFDs

CFD with cash equity hedge executed by same broker as writing the CFD



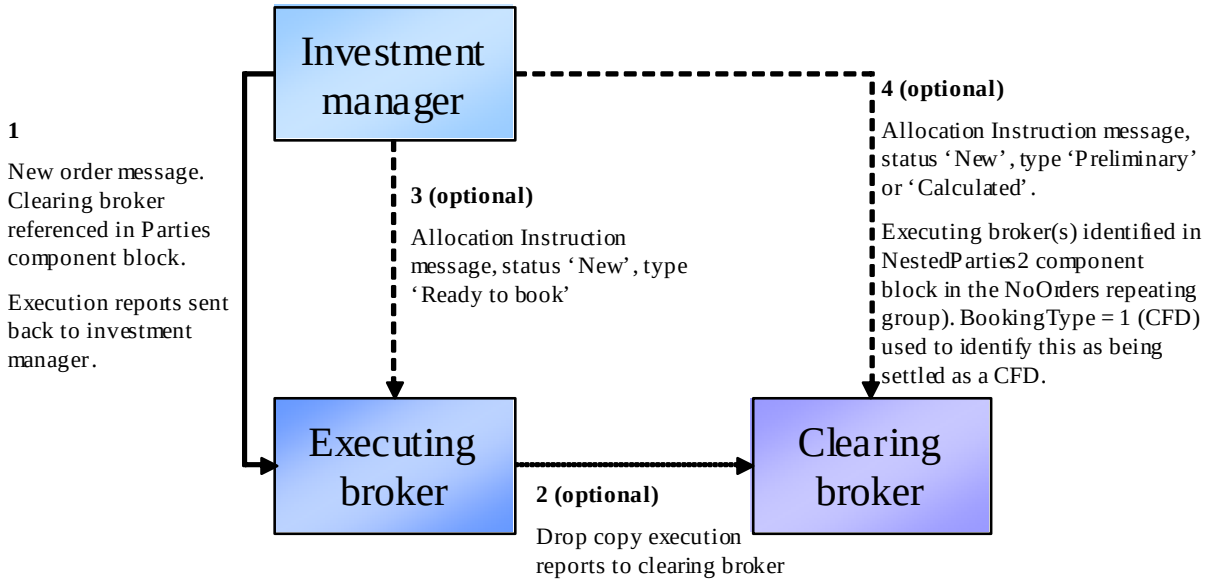
The BookingType field is used on the new order messages to transmit the notification that the order is for a CFD. This information is required at the time of execution as a) the broker may need to invoke separate credit or compliance checks (e.g. different RTL) and b) the broker will need to know to execute a principal cash hedge.

Note the example here could be extended to cover any OTC derivative product where one or more of its cashflows is derived from a cash equity position.

CFD with cash equity hedge executed by different broker from that writing the CFD

Here the clearing broker is writing the CFD and the executing broker is simply executing a cash equity hedge for (and settling with) the clearing broker. The allocation instruction from the investment manager to the clearing broker contains the BookingType field to provide notification that the order is to be booked out as a CFD. BookingType can also optionally be provided on the new order message to the executing broker.

© Copyright, 2007-2008, FIX Protocol, Limited


Investment manager
Executing broker
Clearing broker
1
New order message. Clearing broker referenced in Parties component block.
Execution reports sent back to investment manager.
3 (optional)
Allocation Instruction message, status 'New', type 'Ready to book'
4 (optional)
Allocation Instruction message, status 'New', type 'Preliminary' or 'Calculated'.
Executing broker(s) identified in NestedParties2 component block in the NoOrders repeating group). BookingType = 1 (CFD) used to identify this as being settled as a CFD.
2 (optional)
Drop copy execution reports to clearing broker

Commission Sharing Arrangements

Soft Dollars

Soft dollar programmes are arrangements whereby a proportion of commission on certain trades is not retained by the broker, but set aside for the payment of certain eligible services for the buy side firm sending the orders. FIX supports the handling of such business in two ways:

- Use of the ProcessCode field on the new order messages (new order single and new order list). Takes value '1=soft dollar' for soft dollar trades.
- Use of the ProcessCode field on the FIX allocation instruction or allocation report message. Takes value '1=soft dollar' for soft dollar trades.

The issue with the first approach is that the ProcessCode flag is applied to an order and therefore must be assumed to be associated with every allocated trade belonging to that order which may not necessarily be the case. For this reason, use of ProcessCode on new order messages is not recommended unless the order is pre-trade allocated to a single sub account.

The second approach is recommended as a) it logically forms part of the post trade allocation process, b) existing alternative allocation mechanisms such as Global OASYS block ETC, OASYS Direct and manual (fax etc.) operate in this way.

Directed Brokerage

Directed brokerage (commission recapture) programmes are arrangements whereby a proportion of commission on certain trades is not retained by the broker, but set aside to be paid ultimately to the underlying funds on whose behalf the trades were executed; this may or may not involve an intermediary (e.g. Frank Russell, State Street, Lynch Jones Ryan) who collects payments from the brokers and manages the payment to the end funds.

As with soft dollars, the ProcessCode field (value '6=plan sponsor') is used. In addition, the identity of the scheme administrator must also be identified. Use of the post-trade allocation instruction message is recommended over use of ProcessCode on the new order messages for the same reasons as given for soft dollars above. The NestedParties component block in the NoAllocs repeating group in the allocation instruction message (for post-trade allocation) or new order message (for pre-trade allocation) should be used for identifying the scheme administrator.

The confirmation message contains an optional field SharedCommission which can be used to communicate the amount of commission actually being split out to the intermediary.

© Copyright, 2007-2008, FIX Protocol, Limited

Multi-Day Average Pricing

Introduction

Multi-day average pricing ("warehousing") involves the sellside working a client equity order over a number of days in a similar way to a “good-till” (GT) order, but with the crucial difference that the entire buyside executed quantity is not booked for settlement until the last day of the warehouse period. Given that the sellside will still have to settle its market-side executions, this will involve the funding of buys (sellside receives from the market), and borrowing stock or failing to deliver on sells (sellside delivers to the market). Note that warehousing is not permitted in certain markets.

The flows outlined below and supported in FIX 4.4 are subject to the following constraints:

- Only equities will be warehoused.
- Multi-leg instruments will not be warehoused.
- No special functionality will be provided to cover corporate actions occurring during a warehouse period.
- Only GT orders will carry warehousing instructions on the order message itself (wrong – this covers day orders as well). If the sellside decide to warehouse a day order they will use the FIX allocation message.
- Sellside firms will be responsible for deciding whether or not to accept a warehouse request.

Flow Summary

The following four flows are supported:

	Day orders	**GT orders**
Pre-trade notification	Use 589 DayBookingInst (a new value '2 – accumulate' has been added for this purpose). This is used to signify that the day order should be warehoused **in full** at the end of the day.	Use 427 GTBookingInst, using value '1 – accumulate until filled/expires' or '2 – accumulate until told otherwise'.
Post-trade notification	If the entire order is to be warehoused, use a 'warehouse' allocation instruction message (an Allocation Instruction with AllocTransType = 7 – warehouse) for the portion to be warehoused. If only part of the order is to be warehoused, use a 'warehouse' allocation instruction message for the warehoused portion and book and allocate the rest using a standard allocation instruction message.	As for Day orders.
End of day warehouse recap	At the end of every day where all or part of an order or orders has been warehoused, use an Allocation Report to communicate details of the warehoused portion of the order(s). This message has AllocReportType 5 = Warehouse recap and will communicate the quantity and average price of the warehoused portion of the order(s). For other details relating to the order (e.g. quantity executed that day, quantity remaining at the beginning of that day, the running average price), a 'done for day' execution report should be used. Note trade confirmations will only be generated for any part of the order booked out to a client account (i.e. not warehoused).	As for Day orders.
Warehouse	Reject the warehouse allocation message with an allocation ack with	As for Day orders.

© Copyright, 2007-2008, FIX Protocol, Limited

rejection (pre-trade)	87 AllocStatus '1 – rejected' and 88 AllocRejCode - '13 Warehouse request rejected'. The order will then remain in an unbooked state until it is either booked out manually or a new allocation message is received (and successfully processed).	
Warehouse rejection (post-trade)	As for pre-trade.	As for Day orders.

For all of these flows, either full or partial warehousing is supported (the latter meaning that only part of an order is warehoused, with the balance booked out as normal).

Example Warehouse Flows

These diagrams show a simplified version of the FIX warehousing flows.

Good Till Order – Warehouse Until Filled Using Pre-Trade Booking Instruction

Day 1 – entire part-filled quantity is warehoused

BuySide				SellSide	
	→	New order single *GTBookingInst = 1*	→		1. *Buyside sends new GT order with instruction to warehouse any part-filled quantity until the order fills or expires (i.e. GTBookingInst is 1).*
	←	Execution reports *(new...partial fills)*	←		2. *Sellside accepts the order, then sends 1 or more partial fill execution reports.*
	←	Execution report *(done for day)*	←		3. *Sellside sends a "done-for-the-day" (DFD) execution report when execution completes for the day.*
	←	Allocation report *(AllocReportType = 5)*	←		4. *Sellside sends a warehouse recap allocation report.* *Note a 'warehouse instructon' allocation instruction message from the buyside is not required at this point due to the use of GTBookingInst when placing the order*

Day 2 – further executions; entire part-filled quantity is again warehoused

BuySide				SellSide	
	←	Execution reports *(new...partial fills)*	←		2. *Sellside sends 1 or more partial fill execution reports.*
	←	Execution report *(done for day)*	←		3. *Sellside sends a "done-for-the-day" (DFD) execution report when execution completes for the day.*
	←	Allocation report *(AllocReportType = 5)*	←		4. *Sellside sends a warehouse recap allocation report.* *Note a 'warehouse instructon' allocation*

© Copyright, 2007-2008, FIX Protocol, Limited

				instruction message from the buyside is not required at this point due to the use of GTBookingInst when placing the order

Day 3 – further executions; order is now filled and booked out

BuySide				**SellSide**	
	←	Execution reports *(new...partial fills...fill)*	←		*2. Sellside sends 0 or more partial fill execution reports and a final fill.*
	→	Allocation instruction *AllocTransType 'new'* *AllocType either 'Buyside preliminary' (if without MiscFees) or 'Buyside calculated' (if with)*	→		*4. Buyside provides allocations for entire order quantity*
	←	Allocation Instruction ACK *(AllocStatus 'received')*	←		*5. Sellside acknowledges receipt of the allocation details.*
	←	Allocation Instruction ACK *(AllocStatus 'processed')*	←		*6. Sellside processes and acknowledges allocation details. Confirmation messaging and processing will then take place for the order.*
	←	Allocation report *(AllocReportType = 5)*	←		*7. Sellside sends a warehouse recap allocation report.*

© Copyright, 2007-2008, FIX Protocol, Limited

Good Till Order – Partial Warehousing - Day 1 (some of the part-filled quantity is warehoused; the rest is allocated)

Day 1 – part of the part-filled quantity is warehoused

BuySide				SellSide	
		New order single *GTBookingInst = 1 or 2*			*1. Buyside sends new GT order with instruction to warehouse any part-filled quantity (i.e. GTBookingInst is 1 or 2).* *Should clarify that use of GTBookingInst implies warehouse instructions not required. Should add an example of 'normal' GT (i.e. no GTBookingInst), i.e. post trade instructions*
		Execution reports *(new...partial fills)*			*2. Sellside accepts the order, then sends 1 or more partial fill execution reports.*
		Execution report *(done for day)*			*3. Sellside sends a "done-for-the-day" (DFD) execution report when execution completes for the day.*
		Allocation instruction for non-warehoused portion of the order *AllocTransType 'new'* *AllocType either 'Buyside preliminary' (if without MiscFees) or 'Buyside calculated' (if with)*			*4. (a) Buyside decides to book out a proportion of the part-filled order*
		Allocation Instruction ACK *(AllocStatus 'received')*			*5. (a) Sellside acknowledges receipt of the allocation details.*
		Allocation Instruction ACK *(AllocStatus 'processed')*			*6. (a) Sellside processes and acknowledges allocation details. Confirmation messaging and processing will then take place for the order.*
		Allocation instruction for buyside warehouse notification *AllocTransType 'new'* *AllocType 'warehouse'*			*4. (b) Buyside warehouses the rest of the order.*
		Allocation Instruction ACK for warehouse instruction *(AllocStatus 'received')*			*5. (b) Sellside acknowledges receipt of the warehouse allocation instruction.*
		Allocation Instruction ACK for warehouse instruction *(AllocStatus 'processed')*			*6. (b) Sellside processes and acknowledges allocation details.*
		Allocation report *(AllocReportType = 5)*			*4. Sellside sends a warehouse recap allocation report.*

© Copyright, 2007-2008, FIX Protocol, Limited

	← Allocation report *(AllocReportType = 5)* ←		*7. Sellside sends a warehouse recap allocation report.*

Subsequent days' flows are as in 'Warehouse till filled' scenario above.

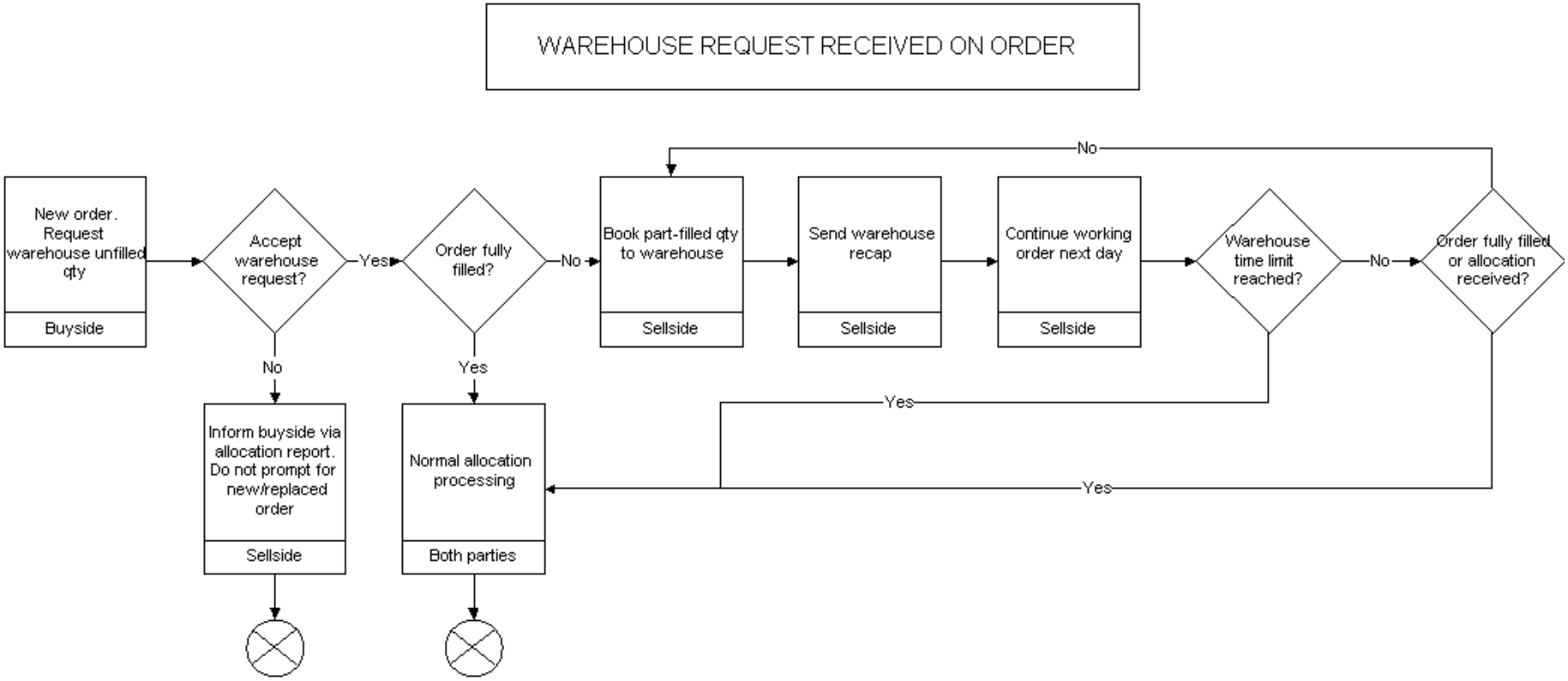
Note the flow is similar when the **entire** order is warehoused – in this case, messages 4a, 5a and 6a are missing.

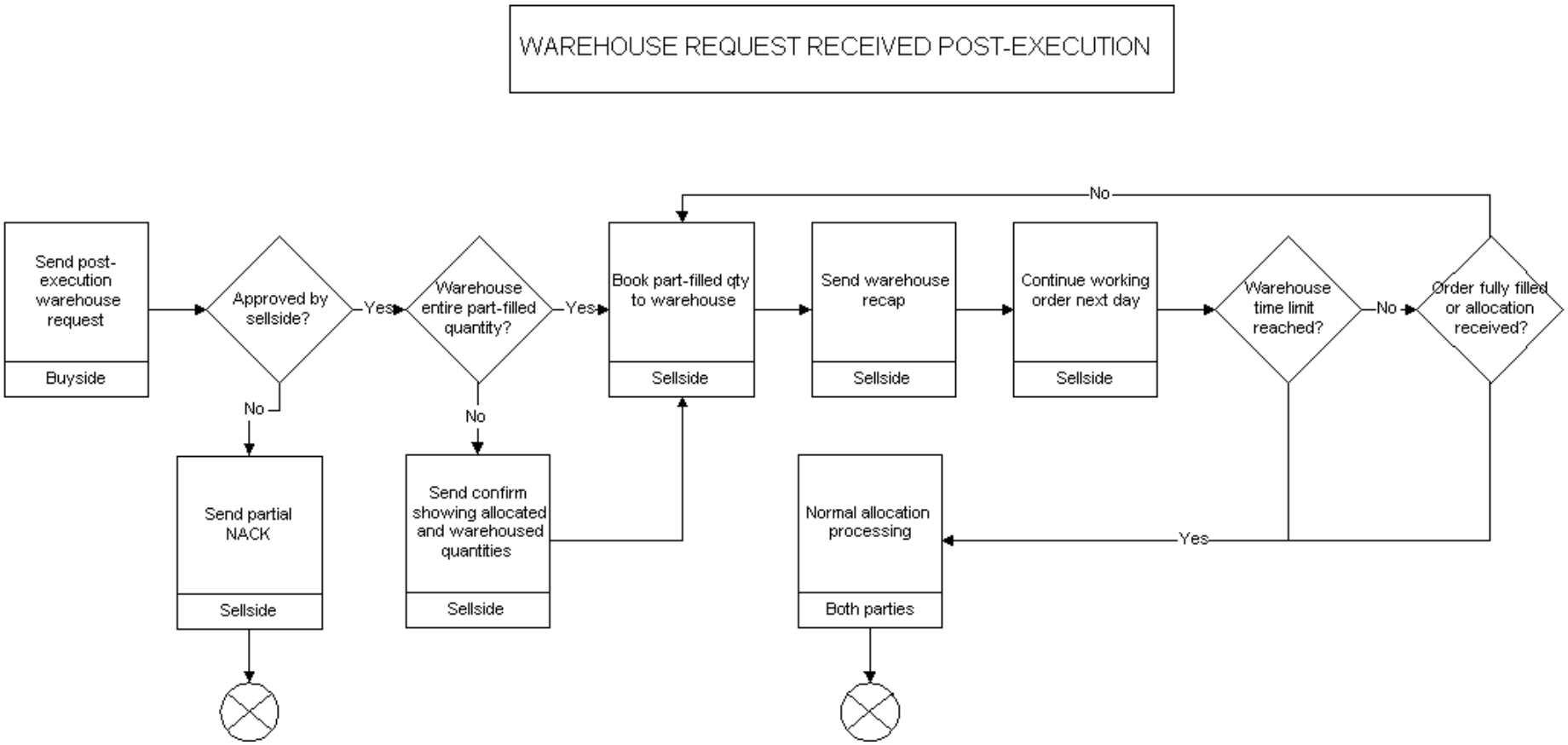
Day Order – Part- or Fully Warehoused

In this case, on day 1 of the order the buyside decides to warehouse a trade after the DFD message has been sent by the sellside. The entire part-filled quantity may be warehoused or a proportion may be allocated to client accounts. The flow is exactly the same as for GT orders as above, apart from the original new order not having any GTBookingInst or DayBookingInst.

© Copyright, 2007-2008, FIX Protocol, Limited

Decision Flows




WAREHOUSE REQUEST RECEIVED POST-EXECUTION
Send post-execution warehouse request
Buyside
Approved by sellside?
Yes
No
Send partial NACK
Sellside
Warehouse entire part-filled quantity?
Yes
No
Send confirm showing allocated and warehoused quantities
Sellside
Book part-filled qty to warehouse
Sellside
Send warehouse recap
Sellside
Continue working order next day
Sellside
Warehouse time limit reached?
No
Yes
Order fully filled or allocation received?
No
Normal allocation processing
Both parties

Regulation SHO - Short-Sell Security Locate

The Security and Exchange Commission (SEC) of the US has issued Regulation SHO which requires that firms conducting short-sell trades must specify the security lending firm in the order. To support the identification of the security lending firm, the PartySubIDType (803) enumeration value of "27" (SecurityLocateID) in conjunction with PartySubID (523) are used within the Parties Component Block to identify the lending firm. The PartySubID would contain the identification of the firm lending the security for the short-sell.

Strategy Parameters for Algorithmic Trading

With the growing number of algorithmic trading strategies being introduced there is the need for the ability to convey additional strategy parameters. The NoStrategyParameters repeating group allows for a more flexible and standardized implementation to support algorithmic trading.

Tag	Field		Type	Description
957	NoStrategyParameters		NumIn Group	Indicates number of strategy parameters
→	958	StrategyParameterName	String	Name of parameter
→	959	StrategyParameterType	Int	Datatype of the parameter. Refer to Appendix-A for a list of valid values.
→	960	StrategyParameterValue	String	Value of the parameter

The NoStrategyParameters repeating group is to be used instead of the deprecate fields TargetStrategyParameters (848) and ParticipationRate (849). This repeating group allows the ability to conveny multiple parameters in an unrestricted manner between the Initiator and Respondent, as long as the StrategyParameterName, StrategyParameterType and StrategyParameterValue ranges are recognized by the Respondent.

For example, a 'VWAP' strategy with specified start time and end time and two additional parameters, participation rate (40%) and aggressiveness (Y) can be represented as follows:

847 (TargetStrategy) = 1 (VWAP)
168 (EffectiveTime) = 20050606-14:00:00
126 (ExpireTime) = 20050606-20:00:00
957 (NoStrategyParameters) = 2
  958 (StrategyParameterName) = ParticipationRate
  959 (StrategyParameterType) = 11 (Percentage)
  960 (StrategyParameterValue) = 0.4
  958 (StrategyParameterName) = Aggressiveness
  959 (StrategyParameterType) = 13 (Boolean)
  960 (StrategyParameterValue) = Y

It should be noted that StrategyParameterType is an enumerated field which may contain the following values (See also Volume 6, Data Dictionary).

1 = Int
2 = Length
3 = NumInGroup
4 = SeqNum
5 = TagNum
6 = Float
7 = Qty
8 = Price
9 = PriceOffset
10 = Amt

© Copyright, 2007-2008, FIX Protocol, Limited

11 = Percentage
12 = Char
13 = Boolean
14 = String
15 = MultipleValueString
16 = Currency
17 = Exchange
18 = Month-Year
19 = UTCTimeStamp
20 = UTCTimeOnly
21 = LocalMktTime
22 = UTCDate
23 = Data

Regulation NMS

Background

The Security and Exchange Commission (SEC) of the US has issued Regulation NMS (Reg NMS) in its final form on June 9, 2005, which is available at ***http://www.sec.gov/rules/final/shtml*** . As it relates to the FIX Protocol this section discusses the support provided by the protocol to be compliant with Reg NMS. The focus will be on identifiers required to assist broker-dealers and trading centers in complying with the Order Protection Rule (Rule 611) and the Sub-Penny Rule (Rule 612, also known as the Minimum Pricing Increment Rule).

Order Protection Rule Compliance

Scope of Order Protection Rule Compliance

The Order Protection Rule applies to Regulation NMS stocks. According to the SEC filing: An "NMS stock" is defined in paragraphs (b)(47) and (b)(46) of Rule 600 as a security, other than an option, for which transaction reports are collected, processed and made available pursuant to an effective national market system plan. This definition effectively covers stocks listed on a national securities exchange and stocks included in either the National Market or SmallCap tiers of Nasdaq. It does not include stocks quoted on the OTC Bulletin Board or elsewhere in the OTC market.

Manual quotations are not protected under the Order Protection Rule. Protected bids and offers are defined as quotations in an NMS stock that are:

- displayed by an automated trading center;
- disseminated pursuant to an effective national market system plan; and
- an automated quotation that is the best bid or best offer of a national securities exchange, the best bid or best offer of The Nasdaq Stock Market, Inc., or the best bid or best offer of a national securities association other than the best bid or best offer of The Nasdaq Stock Market, Inc.

Transactions that are exempted from order protection compliance include the following[7]:

1. The transaction that constituted the trade-through was effected when the trading center displaying the protected quotation that was traded through was experiencing a failure, material delay, or malfunction of its systems or equipment.

   [Referred to as the self-help exemption]

2. The transaction that constituted the trade-through was not a “regular way” contract.

   [Examples of “not a regular way contract” include – next day settlement, same day settlement or sellers option]

3. The transaction that constituted the trade-through was a single-priced opening, reopening, or closing transaction by the trading center.

[7] The exemptions listed are taken directly from the SEC filing with the FIF interpretation of the exemption given in brackets and italics below.

© Copyright, 2007-2008, FIX Protocol, Limited

[The opening process in the OTC market for Nasdaq stocks is different from the listed market. UTP has an official open but CTA does not. While not official, listed markets do open at a single price even if this is not flagged by CTA. FIF will follow up with the Plans to determine if there is an issue.]

4. The transaction that constituted the trade-through was executed at a time when a protected bid was priced higher than a protected offer in the NMS stock.

   [Exemption for trading through in a crossed market]

5. The transaction that constituted the trade-through was the execution of an order identified as an intermarket sweep order.

   [Referred to as the intermarket sweep exemption]

6. The transaction that constituted the trade-through was effected by a trading center that simultaneously routed an intermarket sweep order to execute against the full displayed size of any protected quotation in the NMS stock that was traded through.

   [Exception for a transaction that executes at an inferior from the NBBO because other intermarket sweep orders simultaneously hit protected quotes.]

7. The transaction that constituted the trade-through was the execution of an order at a price that was not based, directly or indirectly, on the quoted price of the NMS stock at the time of execution and for which the material terms were not reasonably determinable at the time the commitment to execute the order was made.

   [Exemption covering executions at a negotiated price, e.g., VWAP orders]

8. The trading center displaying the protected quotation that was traded through had displayed, within one second prior to execution of the transaction that constituted the trade-through, a best bid or best offer, as applicable, for the NMS stock with a price that was equal or inferior to the price of the trade-through transaction.

   [Referred to as the 1 second rule, intended to address flickering quotes.]

9. The transaction that constituted the trade-through was the execution by a trading center of an order for which, at the time of receipt of the order, the trading center had guaranteed an execution at no worse than a specified price (a "stopped order"), where:
   a. The stopped order was for the account of a customer;
   b. The customer agreed to the specified price on an order-by-order basis; and
   c. The price of the trade-through transaction was, for a stopped buy order, lower than the national best bid in the NMS stock at the time of execution or, for a stopped sell order, higher than the national best offer in the NMS stock at the time of execution.

   [Stopped orders are given on the consolidated tape.]

Role of Identifier in Order Protection Compliance

Establishing identifiers is one way in which firms can demonstrate that a quote, order or trade is or is not subject to the Order Protection Rule. Identifiers are not the only way to flag order protection exemptions. Firms can modify existing internal order and trade databases to include exemption information rather than adding flags to inter-firm communication protocols like FIX or consolidated tapes like UTP and CTA. This document will focus on those identifiers that would be needed in communication between counterparties.

- **Quote Identifiers:** Quote Identifiers would be added to market data feeds that provide quote information via FIX, proprietary protocols, or through consolidated feeds like CQS, UQDF.
- **Trade Identifiers:** Trade identifiers would be added to market data feeds that provide trade information via FIX, proprietary protocols, or through consolidated feeds like CTS and UTDF.

© Copyright, 2007-2008, FIX Protocol, Limited

- **Order Identifiers:** Incoming orders to trading centers would use Order Identifiers to indicate how an order should be handled. Order identifiers would be added to protocols for electronic trade communication including FIX, CMS and other proprietary protocols used by trading centers. Instituting appropriate order identifiers is the responsibility of each trading center but could be coordinated across industry participants for ease of implementation.

  Additionally, outgoing execution reports could echo the order identifier. While not mandated by Regulation NMS, execution reports echoing exemptions designated on the order would be useful for evaluating execution quality.

FIX Role in Order Protection Compliance

As it related to Reg NMS, FIX support for order protection compliance focuses on the following identifiers:

- Order Identifiers (for electronic trade communication)
  - Intermarket Sweep Order Identifiers (for orders and execution reports)
  - Single Execution Requested for block trade
- Quote Identifiers (for market data feeds)
  - Manual Quote Identifiers
- Trade Identifiers (for market data feeds)
  - Manual Trade Identifiers
  - Intermarket Sweep Trade Identifiers

At this time FIX does not address the other Order Protection Rule exemptions.

Intermarket Sweep Order Identifier

According to the SEC filing: Intermarket sweep order means a limit order for an NMS stock that meets the following requirements: (i) When routed to a trading center, the limit order is identified as an intermarket sweep order; and (ii) Simultaneously with the routing of the limit order identified as an intermarket sweep order, one or more additional limit orders, as necessary, are routed to execute against the full displayed size of any protected bid, in the case of a limit order to sell, or the full displayed size of any protected offer, in the case of a limit order to buy, for the NMS stock with a price that is superior to the limit price of the limit order identified as an intermarket sweep order. These additional routed orders also must be marked as intermarket sweep orders.

An intermarket sweep order functions like an Immediate or Cancel limit order (or other order type and time in force), but it indicates that the firm sending the order has taken responsibility for price protection, and the firm receiving the order should execute it immediately, if possible, without concern for price protection of other markets.

As such the ExecInst field (tag 18) now includes a new value which would be used for order handling and could be echoed on the execution report for this order.

ExecInst (tag 18)

- value "f" (lowercase F) to designate an "intermarket sweep" order

The Execution Reports do not need to identify intermarket sweep trades in the scenario where an incoming order was executed against an intermarket sweep order since the original incoming order had not been designated as an intermarket sweep order.

Quote & Trade Identifiers

Reg NMS differentiates between fast quotes, which are executed automatically, and slow quotes which are executed manually. Reg NMS affords certain price protections to fast quotes that are not available to slow quotes.

To differentiate between slow quotes, trades resulting from slow quotes, and trades resulting from intermarket sweep orders in market data feeds the following fields and the associated new values can be used for this purpose:

© Copyright, 2007-2008, FIX Protocol, Limited

QuoteCondition (tag 276)

- value "L" (captial L) to designate a manual or slow quote

TradeCondition (tag 277)

- value "Y" (capital y) to designate a trade resulting from a manual or slow quote
- value "Z" (capital z) to designate a trade resulting from an intermarket sweep

Interoperability with Other Standards

Currently the CTA[8] and UTP[9] Plans have outlined flags 'A', 'B', and 'H' as follows:

Quote Condition Code	Current Definition	New Definition
A	Depth on Ask	Manual Ask, Automatic Bid
B	Depth on Bid	Manual Bid, Automatic Ask
H	Depth on Bid and Ask	Manual Bid and Ask

The basic data element in the CTA and UTP Plans is a two-sided quote, while the FIX Protocol represents a bid and ask pair as two distinct one-sided data elements. So these three values can map to QuoteCondition(276) = L on the respective bid or ask Market Data Entries.

Additionally, the CTA Plan has redefined sales condition 'F' to reflect that an order was executed as an intermarket sweep order.[10] This can map to a Market Data Entry representing the trade and having TradeCondition(277) = Z

Sub-penny Rule Compliance

Scope of Sub-penny Rule Complaince

According to the SEC filing: "New Rule 612 prohibits an exchange, association, vendor, ATS, or broker-dealer from accepting, ranking, or displaying an order, quotation, or indication of interest in an NMS stock priced in a sub-penny increment (except for an order, quotation, or indication of interest priced less than $1.00 per share, in which case the price may not extend beyond four decimal places)."

FIX Role in Sub-penny Rule Compliance

To allow firms the ability to specify unambiguously to their counterparties that the message in question was rejected due to an invalid price increment, the following fields in the appropriate message type can be used, along with the associated new values, for this purpose:

CxlRejReason (tag 102)

- value "18" to indicate an invalid price increment

OrdRejReason (tag 103)

- value "18" to indicate an invalid price increment

BusinessRejectReason (tag380)

- value "18" to indicate an invalid price increment

[8] For full details on CTA quote conditions, see ***http://www.nysedata.com/announce.asp?id=41***

[9] For full details on UTP quote conditions, see ***http://www.nasdaqtrader.com/trader/news/2005/utpvendoralerts/uva2005-036.stm***

[10] For full details on the F sales condition, see ***http://www.nysedata.com/announce.asp?id=66***

© Copyright, 2007-2008, FIX Protocol, Limited

OATS Phase 3 Requirements

Background

On September 28, 2005, the SEC approved rule filing ***SR-NASD-00-23*** relating to the OATS rules. As approved, the amendments (1) implement the OATS requirements for manual orders (OATS Phase III); (2) provide that members are required to capture and report both the time the order is received by the member from the customer and the time the order is received by the member's trading desk or trading department, if those times are different; (3) exclude certain members from the definition of "Reporting Member" for those orders that meet specified conditions and are recorded and reported to OATS by another member; and (4) permit NASD to grant exemptive relief from the OATS reporting requirements in certain circumstances to members that meet specified criteria.

Meeting OATS 3 Requirements using FIX

The following table summarizes the OATS Phase 3 requirements and how each is supported by FIX.

	Requirement Nickname	Requirement Description	FIX Mapping (vs. FIX 4.4)
1	Manual Order Indicator	Indicates whether the order was initially received by the broker manually (vs. electronically)	ManualOrderIndicator (1028)
2	Order Received Timestamp	Indicates the time broker first received the order from the customer. Requirement to record if > 1 second delay before order is entered into electronic system.	TrdRegTimestamp (769) within <TrdRegTimestamp> component block repeating group in conjunction with: TrdRegTimestampType (770) using value of "4 = Broker Receipt"
3	Customer Directed Order	Indicates whether the customer 'directed' the order to a specific execution venue.	CustDirectedOrder (1029)
4	Received Department ID	The department or desk within a firm that receives an order. Either the Receiving Terminal ID or the Receiving Department ID must be provided when an order is received directly from a customer. The member firm must maintain a list of the department identifiers and provide them on request to NASD. Codes must be unique within a firm, regardless of locations in which it operates. (This information is on an OATS "NW" record)	ReceivedDeptID (1030)
5	Customer Special Order Handling Instructions	Codes (24 handling codes with max of 5 codes on any one order) denoting additional order instructions that serve to qualify	CustOrderHandlingInst (1031) in conjunction with: OrderHandlingInstSource (1032) with value

© Copyright, 2007-2008, FIX Protocol, Limited

		the pricing, quantity, execution timing, or execution method of an order specified by the customer. For PEG, this includes Contingent and/or Hedged type orders.	of "1 = NASD OATS"
6	Received By Desk ID	The desk or department within a firm that receives an order. The member firm must maintain a list of department identifiers and provide them on request to NASD. Codes must be unique within a firm, regardless of locations in which it operates. Requirement to capture per Desk if order routes through multiple Desks.	TrdRegTimestampOrigin (771) within <TrdRegTimestamp> component block repeating group in conjunction with: TrdRegTimestampType (770) using value of "6 = Desk Receipt"
7	Desk Type Code	Indicates the type of Desk or Department at which the order was received. The OATS Phase III appendix lists 11 codes. Requirement to capture per Desk if order routes through multiple Desks.	DeskType (1033) within <TrdRegTimestamp> component block repeating group in conjunction with: DeskTypeSource (1034) with value of "1 = NASD OATS"
8	Desk Received Timestamp	The time the desk received the order. Requirement to capture per Desk if order routes through multiple Desks.	TrdRegTimestamp (769) within <TrdRegTimestamp> component block repeating group in conjunction with: TrdRegTimestampType (770) using value of "6 = Desk Receipt"
9	Desk Special Order Handling Instructions	Codes (24 handling codes with max of 5 codes on any one order) denoting additional order instructions that serve to qualify the pricing, quantity, execution timing, or execution method of an order transmitted to another Desk or Department within a firm. For PEG, this includes Contingent and/or Hedged type orders. Requirement to capture per Desk if order routes through multiple Desks.	DeskOrderHandlingInst (1035) within <TrdRegTimestamp> component block repeating group in conjunction with: OrderHandlingInstSource (1032) with value of "1 = NASD OATS"

TrdRegTimestamp Usage Example for OATS 3

Below is an exmple of the TrdRegTimestamp component block with the OATS Phase 3 fields included.

© Copyright, 2007-2008, FIX Protocol, Limited

768	NoTrdRegTimestamps		N	"NoDesks"
→	***769***	***TrdRegTimestamp***	N	Required if NoTrdRegTimestamps > 0 "Receive Time"
→	***770***	***TrdRegTimestampType***	N	Required if NoTrdRegTimestamps > 0 Traded / Regulatory timestamp type. Valid values: 1 = Execution Time 2 = Time In 3 = Time Out 4 = Broker Receipt ["OrderReceivedTimestamp"] 5 = Broker Execution 6 = Desk Receipt
→	***771***	***TrdRegTimestampOrigin***	N	"DeskID"
→	***1033***	***DeskType***	N	For DeskTypeSource = 1 (NASD OATS), valid values are: A = Agency AR =Arbitrage D = Derivatives IN = International IS = Institutional O = Other PF = Preferred Trading PR = Proprietary PT = Program Trading S = Sales T = Trading
→	***1034***	***DeskTypeSource***	N	valid values: 1 = NASD OATS

© Copyright, 2007-2008, FIX Protocol, Limited

→	*1035*	***DeskOrderHandlingInst***	N	For DeskTypeSource = 1 (NASD OATS), valid values are: ADD = Add-on Order AON = All or None CNH = Cash Not Held DIR = Directed Order E.W = Exchange for Physical Transaction FOK = Fill or Kill IO = Imbalance Only IOC = Immediate or Cancel LOO = Limit on Open LOC = Limit on Close MAO = Market at Open MAC = Market at Close MOO = Market on Open MOC = Market on Close MQT = Minimum Quantity NH = Not Held OVD = Over the Day PEG = Pegged RSV = Reserve Size Order S.W = Stop Stock Transaction SCL = Scale TMO = Time Order TS = Trailing Stop WRK = Work

Here is a comprehensive example of the use of the OATS Phase 3 fields.

A customer order was received by telephone at 2:00:07 PM EST to be executed as a "Not Held" order. The order was passed to the New York Institutional desk, which received it at 2:00:42 PM EST, and a trader chose to route it as an "All or None" order. The resulting order could appear as follows:

```
ManualOrderIndicator(1028)=Y              //received manually
CustDirectedOrder(1029)=N                 //customer did not direct it to an execution venue
CustOrderHandlingInst(1031)=NH            //Not Held, customer's handling instruction
OrderHandlingInstSource(1032)=1           //NASD OATS is handling instruction source
NoTrdRegTimestamps(768)=2
TrdRegTimestamp(769)=20061209-19:00:07    //Broker Receipt time in UTC
TrdRegTimestampType(770)=4                //Broker Receipt
TrdRegTimestamp(769)=20061209-19:00:42    //Desk Receipt time in UTC
TrdRegTimestampType(770)=6                //Desk Receipt
TrdRegTimestampOrigin(771)=NYINST         //Desk ID
DeskType(1033)=IS                         //Instutional desk
```

© Copyright, 2007-2008, FIX Protocol, Limited

```
DeskTypeSource(1034)=1                    //NASD OATS source code
DeskOrderHandlingInst(1035)=NH AON        //Desk's order handling inst, "not
                                          held" and "AON"
```

External Order Routing Control

Over the past decade, the securities industry has experienced a growing trend towards decentralization of liquidity. Within the United States, the landscape for equities has evolved into competing Exchanges, ECNs, ATSs, etc., each maintaining their own decentralized pool of liquidity.

With the trend towards decentralization came a need for access to liquidity between markets and guarantees of price protection. Linkages between markets developed to meet business needs, as a result of market regulation, and, most recently, as a result of government mandated price protection through Regulation NMS. With these developments, the landscape for equities has further evolved towards decentralized pools of liquidity that are interconnected. Orders routed to one market might find no match, but might be routed by that market to another market where a match at a better price exists.

Whether orders will, by default, be eligible for external routing is outside the scope of the FIX Protocol specification. This is determined by the rules and business practices of the market in question. These flags allow customers to override the market's defaults. Further, markets may decline to allow users to override their defaults for some or all order types, time in force values, etc.

The ExecInst (18) values to support external order routing control are:

- value "g" (lowercase G) - allows the customer to inform an Exchange, ECN, ATS, etc. that an order may be routed to another market
- value "h" (lowercase H) - allows the customer to inform an Exchange, ECN, ATS, etc. that an order may not be routed to another market. In this case, an order that locks or crosses the market but which has no match within the Exchange, ECN, or ATS that received the order may reject the order.

© Copyright, 2007-2008, FIX Protocol, Limited

PRODUCT: FIXED INCOME (FI)

Introduction

This section and the enhancements to the protocol has been the result of the joint effort between the BMA and FPL's Global Fixed Income Committee (formerly Fixed Income Working Group). This Appendix summarizes how FIX messages can be used to support FI trading activities – offerings, negotiated trade/bid or offer request, my bid/offer order, order initiation and execution, and allocation – for the following fixed income asset classes:

- US Treasury Bonds
- US Corporate Bonds
- Municipal Securities
- Agency Securities
- To-Be-Announced (TBA) Mortgage Backed Securities
- Euro Sovereign Bonds
- Euro Corporate Bonds
- US and European Commercial Paper
- Repurchase Agreements (Repos) and Related Securities Lending Activities

The usage models are described as between two counterparties, an Initiator and a Respondent – see the Glossary in Volume 1 for definitions of these roles.

Note that this documentation should be used as a starting point and serves merely to provide guidance in the reader's FIX for FI implementation.

Message Dialog

In FI the trading dialog typically starts in one of two ways: 1) one party sending out offerings to their clients and their clients responding to the offerings, or 2) an interested party initiating an inquiry or a bid/offer request. Once the dialog is initiated a trade could be consummated. The allocation of the trade could be conducted "pre-trade" or "post-trade" directly with the trading counterparty. Third party post-trade reporting using FIX messages is also illustrated.

The diagrams below attempts to illustrate the various dialogs that can happen to facilitate an FI trade and the message flows to use depending on the initiation point of the dialog. Note that the diagrams will also show, via the green colored circles, the next step in the message dialog and do not show error conditions (i.e. one party receiving an unknown CUSIP) that can occur during the dialog.

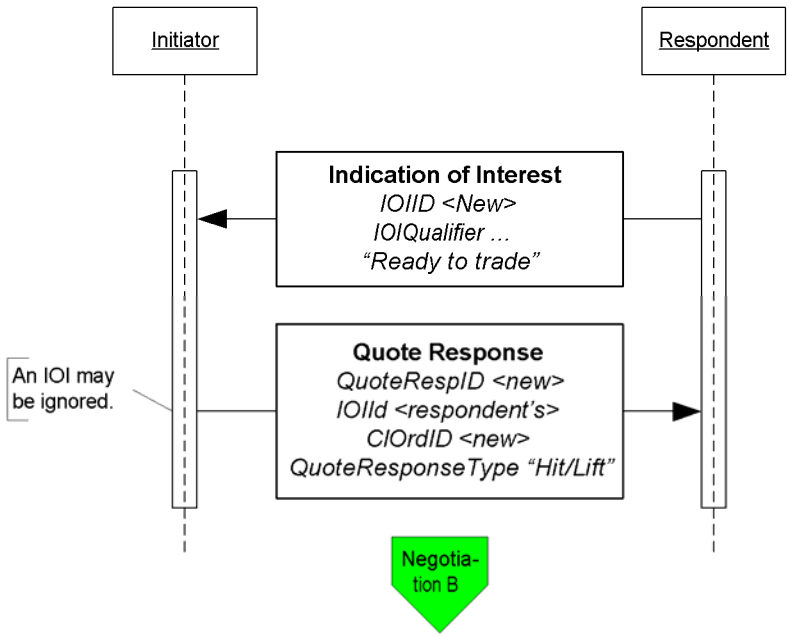
© Copyright, 2007-2008, FIX Protocol, Limited

Indication of Interest (Offerings)

Offerings are communicated using the Indication Of Interest (IOI) message type. The recipient of the offerings can elect to ignore the IOI messages or respond to specific IOI messages via the use the Quote Response message type.

Offerings can be sent by the Respondent to an Initiator on a continuous basis as long as the Initiator wants to receive them. The Initiator has the option to ignore the messages sent by not responding or to respond to an offering of interest by sending a Quote Response message back to the Respondent to either "hit" or "lift" the offering. Figure 1 below illustrates the message flow. The Respondent will pickup on the message dialog flow at "B" in the Negotiated Trade diagram (see next section).

Figure 1: Indication of Interest/Offerings



Click ***here*** to go to "B"

© Copyright, 2007-2008, FIX Protocol, Limited

Negotiated Trade /Inquiry/Bid or Offer Request

A negotiated trade dialog can be initiated not only via the offerings or IOIs as indicated above, but also via a "my bid or offer", an inquiry/bid or offer request, both using a Quote Request message type. The difference between a "my bid/offer" message and an inquiry/bid or offer request message is that in a "my bid/offer" the Initiator sends a Quote Request message with a "my bid/offer" price set for the security in question. The Respondent would respond with a Quote message. The rest of the dialog would follow the dialog described below and it is illustrated in the "My bid/offer" diagram below.

An inquiry, bid or offer request/wanted begins with a Quote Request from the Initiator. It is possible for the Respondent to send an unsolicited Quote message to their counterparty to initiate the negotiated trade dialog, however, this arrangement should be bilaterally agreed upon by the counterparties involved.

In the negotiation dialog, the Initiator would send a Quote Request message to the Respondent to inquire about a certain security, inquire for a list of securities that meet certain stipulations and parameters, request a bid or offer or request a quote on a certain security. Should the Respondent choose not to provide a quote a Quote Request Reject can be sent with the appropriate reject reason code set. At this point the current dialog would terminate. Alternatively the Respondent can respond to the Quote Request with a Quote message. The Quote message would provide the pricing levels for the securities requested by the Initiator.

The Initiator will respond to the Quote from the Respondent via the use of the Quote Response message type. The Quote Response message type can be used to end the dialog, "hit/lift" the Quote, or counter the Quote. A "hit/lift" response from the Initiator indicates to the Respondent that the Initiator agrees with the price level and the quantity, and want to complete a trade. On the other hand, if the Initiator responded with a counter offer then the negotiation can continue until one party decides to terminate the dialog or a trade is completed.

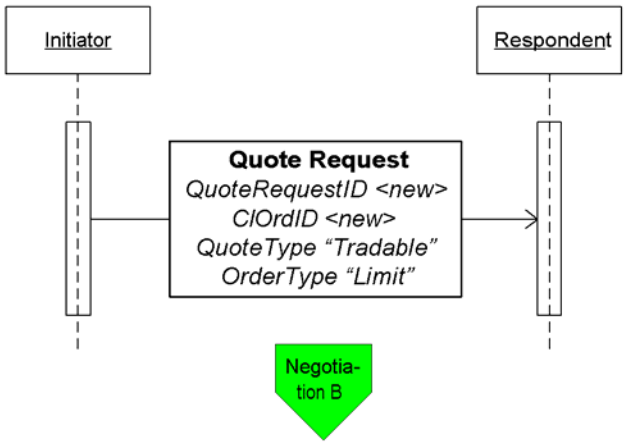
To a "hit/lift" or counter message from the Initiator, the Respondent can respond with another "counter" message using the Quote message type, end the negotiation dialog with a Quote Status Report, or give the Initiator an Execution Report message indicating that the trade has been completed. This Execution Report message may or may not include calculations for information such as accrued interest, gross trade amount, etc.

Lastly, if the Initiator deems that there are discrepancies in the Execution Report message received from the Respondent, the Initiator may use the Don't Know Trade (a.k.a. DK Trade) message type to "reject" the trade information. Resolving the error or discrepancies would be done manually and is currently out of scope for the suggested use of the protocol.

The diagram, Negotiated Trade, on the following page illustrates this flow with some additional details of what values within certain fields can be used.

© Copyright, 2007-2008, FIX Protocol, Limited

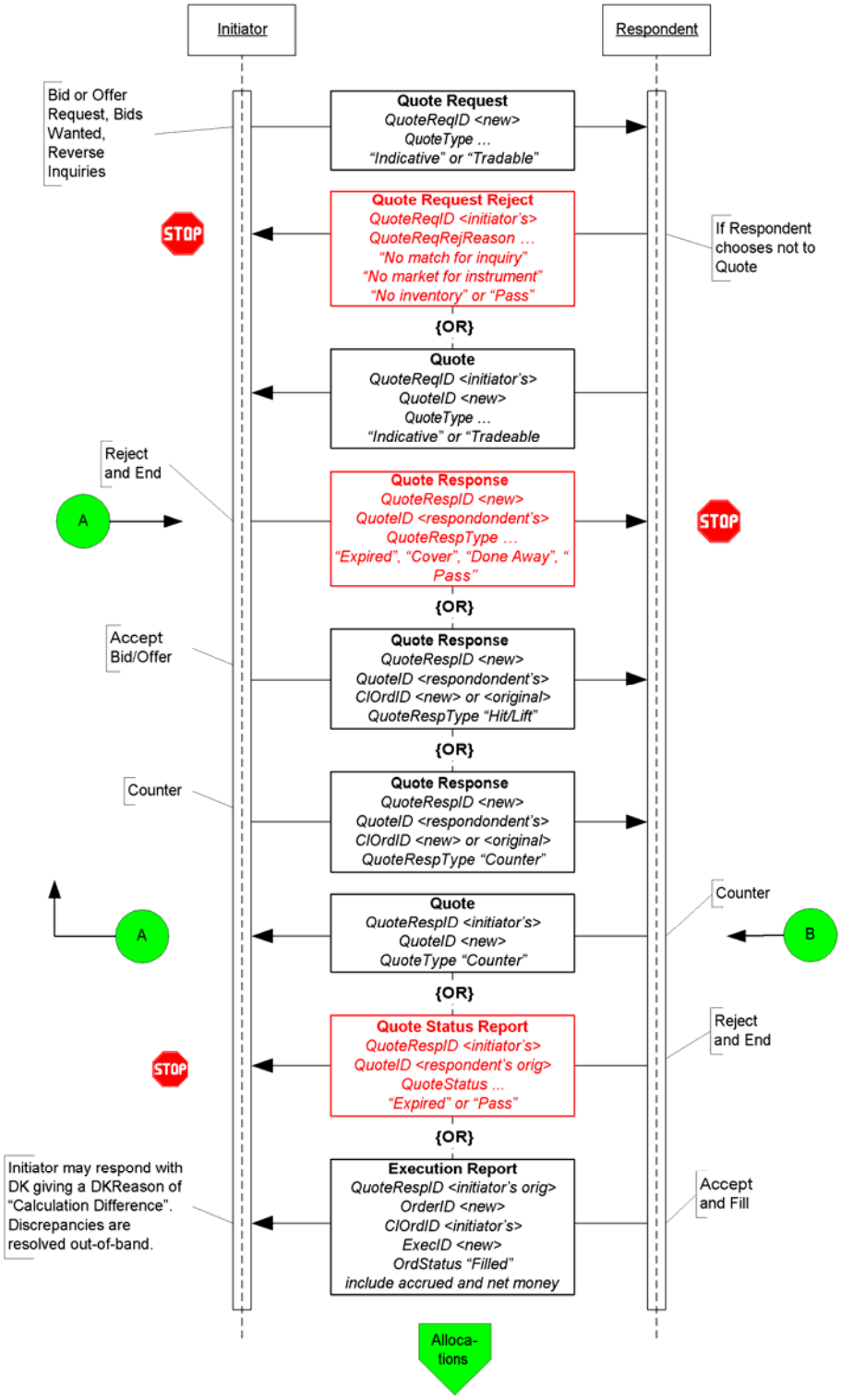
Figure 2: My Bid/Offer



Click ***here*** to go to "B"

© Copyright, 2007-2008, FIX Protocol, Limited

Figure 3: Negotiated Trade/Bid or Offer Request



Click ***here*** to go to "Allocations"

© Copyright, 2007-2008, FIX Protocol, Limited

Out-of-Band Negotiated Order

A trade that is negotiated "out-of-band" is a trade negotiated through other means such as verbally on the phone or via an alternate trading system platform. In this dialog it is assume that the Respondent is able to send the completed trade information electronically using the FIX protocol. The initiation of the order placed by the Initiator could be through the New Order message type or through other means (i.e. verbally or via an alternate trading system platform) agreed upon between the counterparties.

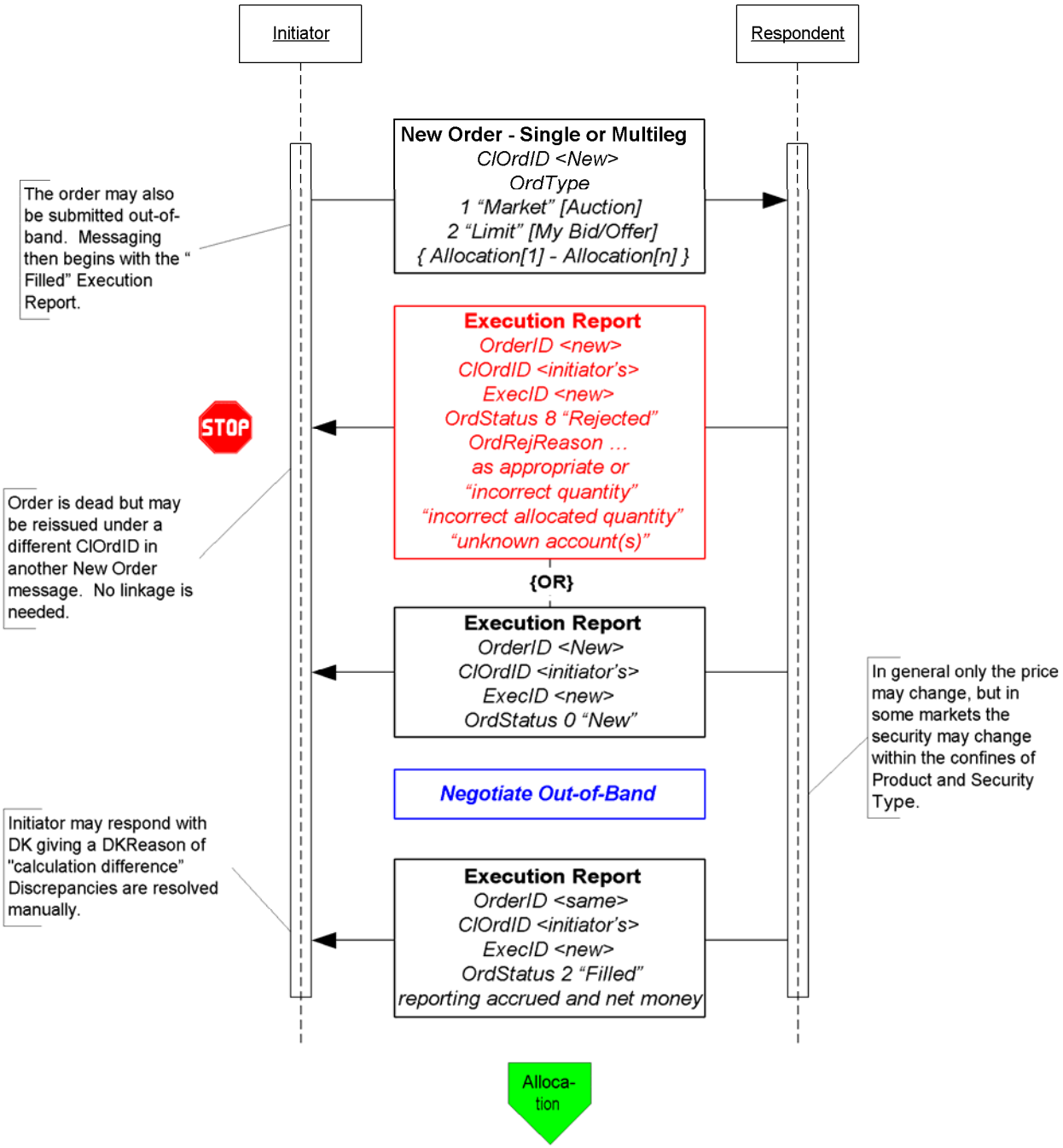
When an order is placed by the Initiator using the New Order message type the Respondent could either accept the order or reject the other using the Execution Report message type. If the order is reject the dialog ends. If the order is accepted the negotiation can begin out-of-band or "offline". When the negotiation is completed and the terms of the trade are agreed upon the Respondent would send the Initiator an Execution Report message to confirm that the trade has been completed. The terms of the trade are reiterated in the Execution Report message.

In the event that the Initiator deems that there are discrepancies in the Execution Report message received from the Respondent, the Initiator may use the Don't Know Trade (a.k.a. DK Trade) message type to "reject" the trade information. Resolving the error or discrepancies would be done manually and is currently out of scope for the suggested use of the protocol.

The diagram on the following page illustrates this dialog.

© Copyright, 2007-2008, FIX Protocol, Limited

Figure 4: Out-of-Band Negotiated Trade



Click ***here*** to go to "Allocations"

© Copyright, 2007-2008, FIX Protocol, Limited

Allocation Instructions

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

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

For the Initiator options 2 and 3 represents the same message flow. The main difference is when the Allocation message is sent – in option 2 it is sent prior to the trade being completed and in option 3 it is sent after the trade has been completed.

Once the trade is completed and the Respondent is ready to act on the allocation instructions, assuming no errors in the allocation instructions from the Initiator, the message flow for the Respondent is the same regardless of which option is used by the Initiator to communicate those allocation instructions.

Note that these options work for Fixed Income because of FI's simple trading practices – there is no concept of "done for day", one set of allocations is applied to a single order usually filled in a single execution.

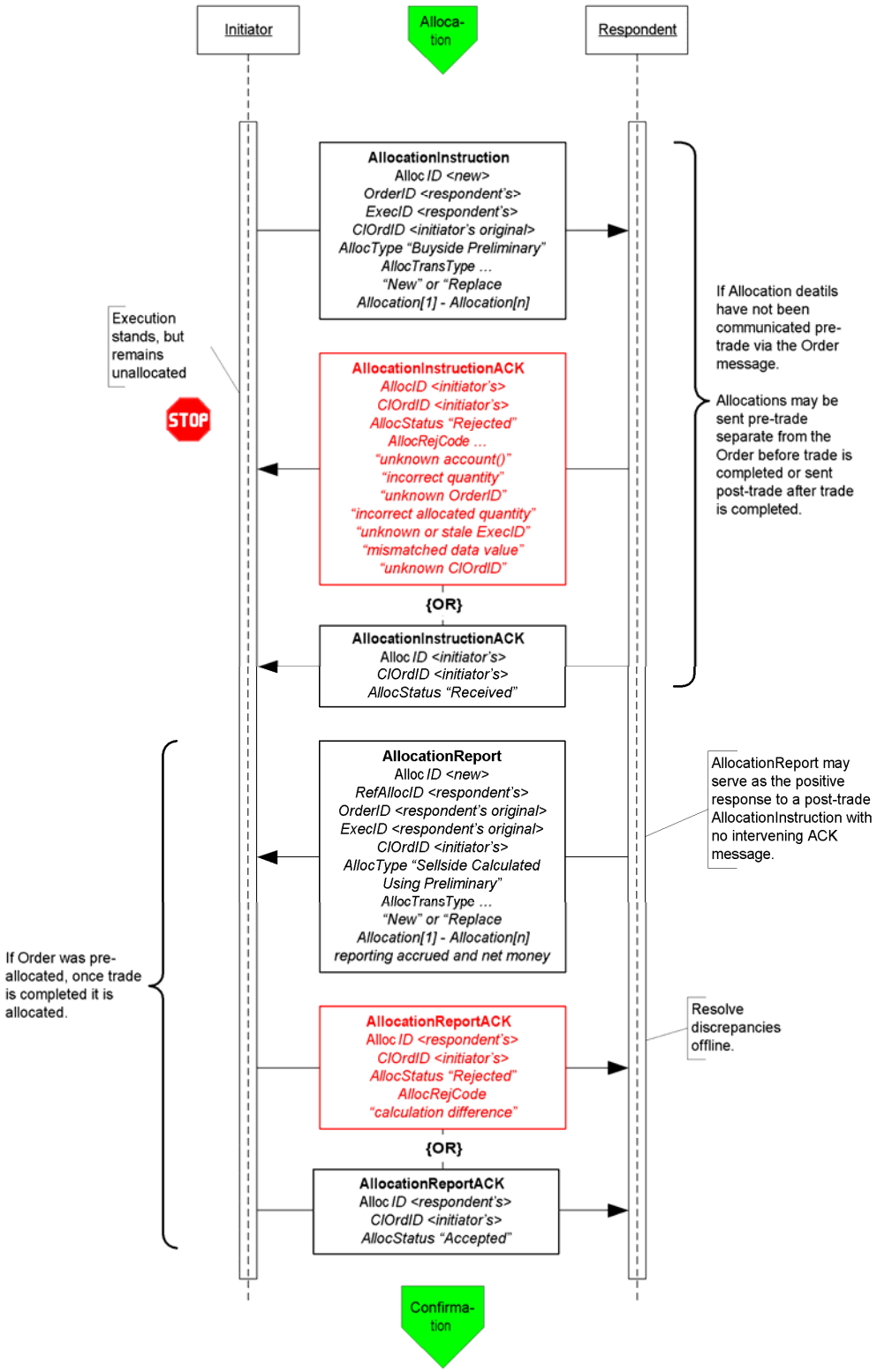
In the Pre-allocated Order scenario the Initiator would send a New Order message that includes the allocation information needed by the Respondent to allocate the trade once the trade is completed. Note, however, that if even one account cannot be identified, or the quantity of one allocation instance does not meet minimum quantity/minimum increment rules for the instrument, or the sum of allocated quantities does not equal the block trade quantity, the **entire** request must be rejected. If erroneous allocations are sent via the New Order message, the entire New Order message is rejected using the Execution Report message with the appropriate reject code.

If the pre-allocated Order is accepted and filled, the Respondent communicates that information to the Initiator using the Execution Report message type, setting all the appropriate status values per standard protocol usage.

At this point in the message flow the Respondent would begin to allocate the trade according to the allocation instructions already provided in the New Order message and communicating that information back to the Respondent according to the message flow shown in Figure 5, starting with the AllocationReport.

© Copyright, 2007-2008, FIX Protocol, Limited

Figure 5: Allocations



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

© Copyright, 2007-2008, FIX Protocol, Limited

In the Pre-trade allocation scenario the Initiator would send the allocation instructions, after placing the order but before the Execution Report message indicated that the trade is completed, to the Respondent using the AllocationInstruction message type. On the other hand, in the Post-trade allocation scenario the Initiator would send the allocation instructions to the Respondent after receiving the Execution Report message indicated that the trade is completed – again using the AllocationInstruction message type.

Before accepting the request the Respondent should determine that all accounts are known, the quantity of each allocation instance meets minimum quantity/minimum increment rules for the instrument and the sum of allocated quantities equals the block trade quantity. If any error is found the Respondent must reject the entire Allocation using the AllocationInstructionAck message with the appropriate reject reason code. In this event, whether the trade that has been completed or is pending completion, the order is a still a live order. A rejection of the AllocationInstruction message does not equate to a reject of the order placed in this case. The Initiator can send a new AllocationInstruction message with the correct instructions and information to the Respondent.

If the Respondent accepts the AllocationInstruction, the message flow would continue as shown in Figure 5 with the Respondent sending the AllocationReport message to communicate the sub-account level calculations for net monies and accrued interest if appropriate. At this stage the Initiator still has the option to reject the validated/calculated allocation message due to differences in calculations of net money, gross amounts, etc., for each of the allocated sub-accounts. Alternatively the Initiator can acknowledge back to the Respondent that the validated/calculated message is accepted. Both the Initiator's response is communicated via the use of the AllocationReportAck message type.

© Copyright, 2007-2008, FIX Protocol, Limited

Figure 6: Confirmation and Affirmation

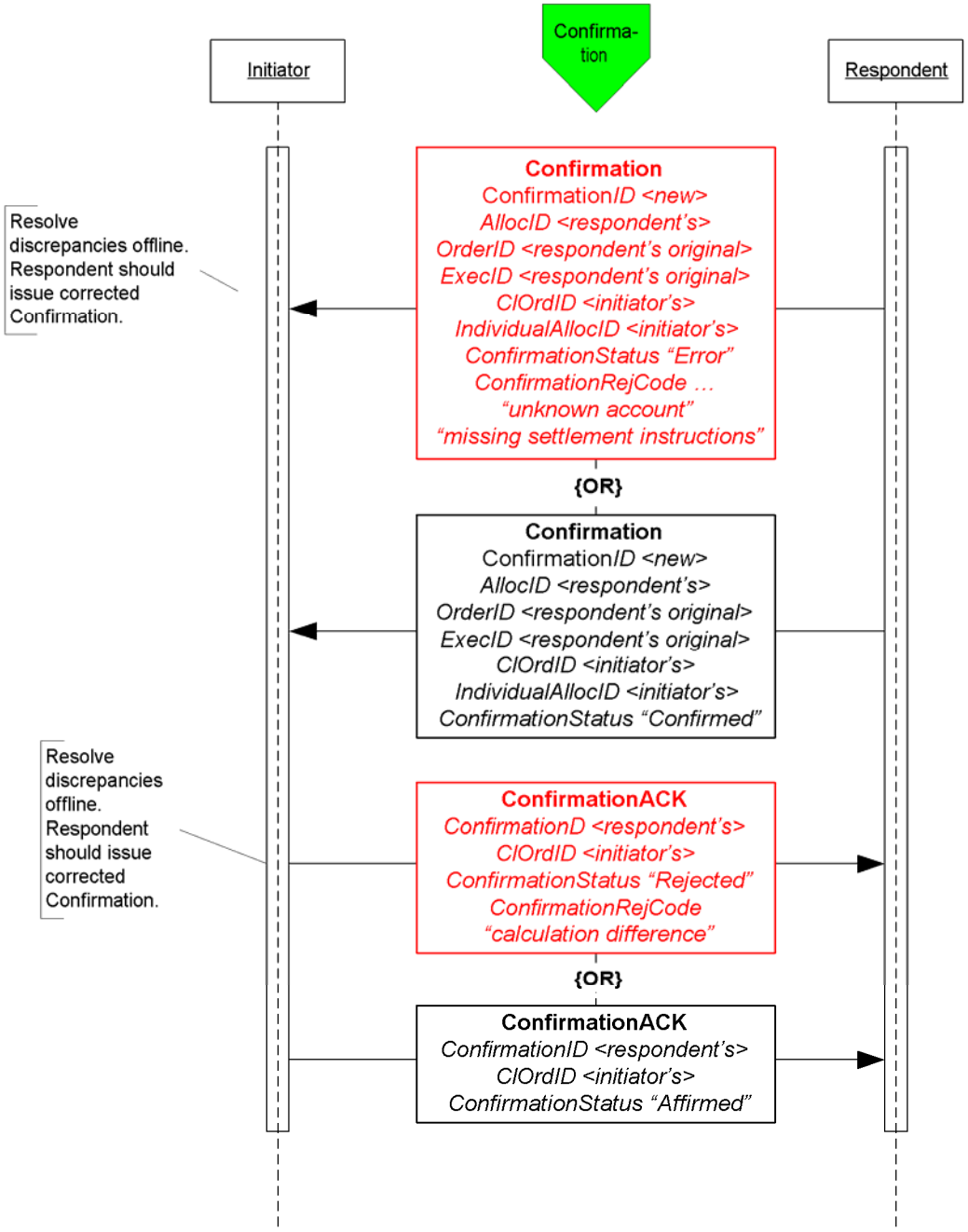


Figure 6 illustrates the message flow of the confirmation process for each of the allocated account instance (the sub-accounts in the AllocationInstruction message) the Respondent would use once the allocation calculations have been confirmed by the Initiator.

The Confirmation message is an optional message that the Respondent can use to report back, confirms or raise an exception of the booking/confirm status of each of the allocation instances in the trade. When the "confirmed"

© Copyright, 2007-2008, FIX Protocol, Limited

status is reported to the Initiator it indicates that that piece of the allocated trade is ready to settle. Each Confirmation message will report the details of a single "ticket", therefore the account names, fees, net money and settlement information are reported using fields designated for single account trades.

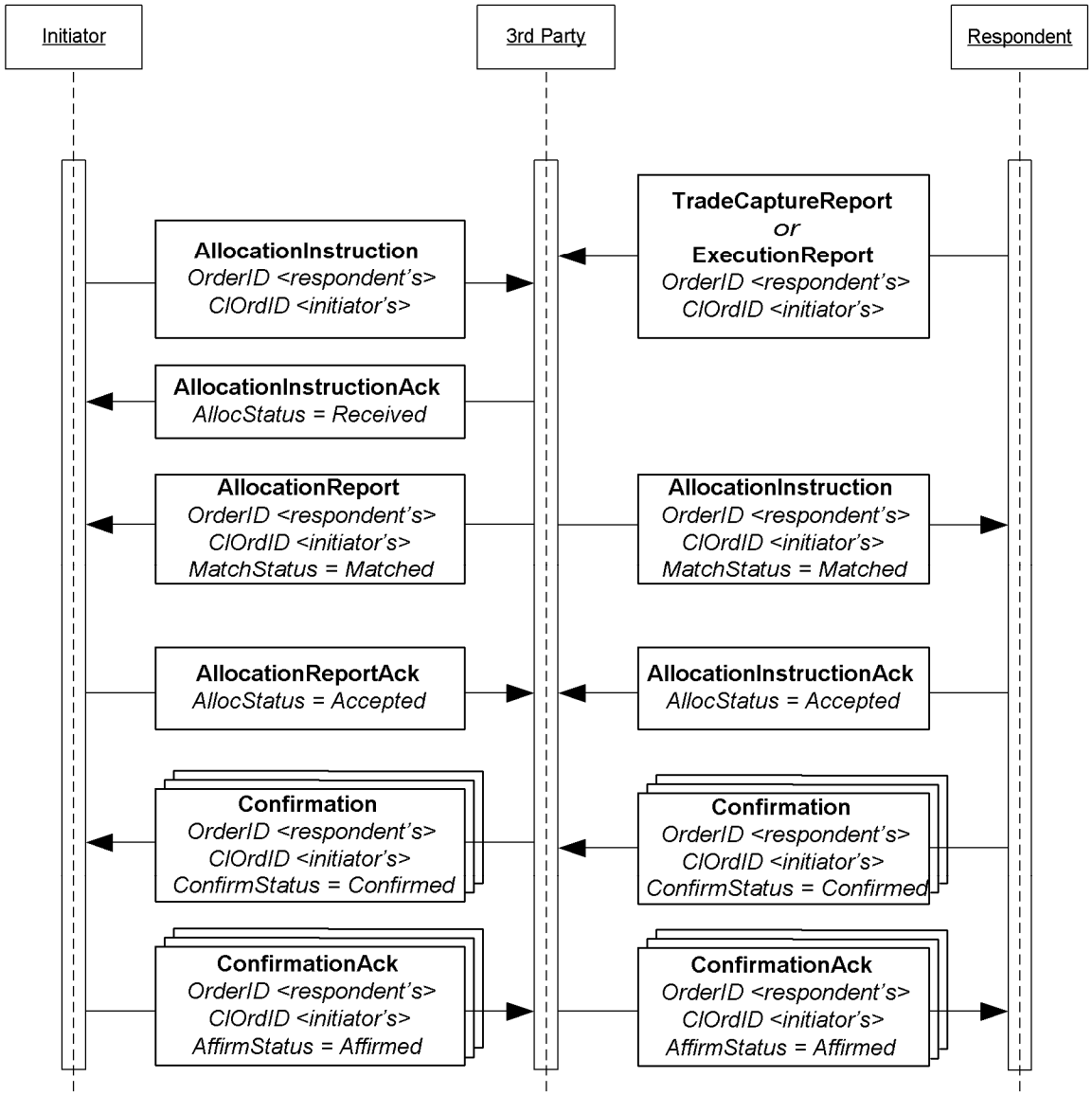
Once the "confirmed" is received from the Respondent the Initiator has the final say by sending the ConfirmationAck message with the "affirmed" status. However, should the Initiator disagree with the Respondent's "confirm" the Initiator can send a reject using the ConfirmationAck message with a status of "rejected" and provide a reason for rejection.

Post Trade Reporting to a 3rd Party or Virtual Matching Utility

Figure 7 illustrates the messages needed by the Initiator and the Respondent to send trade notices to a 3rd party or VMU for trade matching.

© Copyright, 2007-2008, FIX Protocol, Limited

Figure 7: Post Trade 3rd Party or VMU Trade Reporting



The Allocation Instruction message type is used by the Initiator to report one or more orders and block trades along with associated allocations to a 3rd party or VMU for trade matching.

The Respondent will use the Trade Capture Report, or an Execution Report depending on the 3rd party's requirements, message type to report trades to a 3rd party. This notice of execution will be for block level trades.

© Copyright, 2007-2008, FIX Protocol, Limited

Message Usage Details

This section provides some details to the usage of specific fields within messages. These usage guidelines are a supplement to the usage already described in the main volumes of the specification. The usage guidelines discusses requirements for FI that are required by the baseline protocol or will make clarifications specific to FI usage.

General Usage Rules

1. PriceType field must be present when the following price fields are used in any message: Price, BidPx, OfferPx, MktBidPx, MktOfferPx, MidPx.
2. AvgPx field is **usually** expressed as "percent of par". Where it is not, such as in certain Confirmation scenarios, AvgParPx and LastParPx have been added for communicating the percent-of-par price that will drive settlement calculated from the negotiated price.
3. LegPriceType must be present when LegBidPx or LegOfferPx is used in the NoLegs repeating block of any message that contains this repeating block.
4. In all trade and post-trade messages where price information is being communicated, a limit or execution price is **always** conveyed in Price or LastPx, respectively, with PriceType set appropriately. Depending on market convention for a particular asset class other fields may be used to supplement the quote or execution price such as YieldData component block and/or SpreadOrBenchmark component block. Yield and Spread should communicate only derived information, never the negotiated price.
5. All FIX messages identified for use in FI trading except New Order Single support both single instrument trades "outrights" and trades of two instruments – one to be sold and the other to be bought as a replacement. In the US the latter are often called "swaps", in other regions they are "switches", and two-instrument trades involving the sale and purchase of futures contracts with different contract settlement months are called "rolls". The NoLegs repeating block is used to identify and link the two sides of the trade. LegSwapType can be used instead of LegQty on one side of the trade to instruct the Respondent to calculate the LegQty based on the opposite leg's quantity. To submit a new order for a swap or roll use New Order Multileg instead of New Order Single.
6. LastPxPar conditionally required in the Execution Report, Allocation, and TradeCaptureReport messages when LastPx is expressed with a PriceType other than "percent of par" (i.e. when LastPx is expressed as "discount" or "yield" PriceType then LastPxPar must be used to express the price in "percent of par" equivalent.)
7. When SettlType is not "regular" then SettlType must to be specified. SettlType "future" requires a value for SettlDate.

Indication Of Interest

An IOI must specify price information by using either one of the set of price information fields (see General Usage Rules section)

Either the IOIQty or the NoLegs repeating block is required.. If the NoLegs repeating block is used, put "0" (zero) in the IOIQty field. IOIQty is required and used for offerings of single instruments. The NoLegs repeating block is used for multilegs (swaps/switches/rolls). In FI's use there would only be two legs – a buy leg and a sell leg.

ValidUntilTime is where the IOI sender can specify the "firm time" of the offering.

Quote Request

In this message the Initiator can specify what form the quote should be in by using the QuotePriceType field.

The ClOrdID field has been added to this message allowing the Initiator to assign a ClOrdID when requesting for quotes that are of QuoteType "Tradable" and OrdType of "Limit".

© Copyright, 2007-2008, FIX Protocol, Limited

To submit a "my bid/offer" quote request the Initiator will need to specify QuoteType of "Tradable" and OrdType of "Limit". Pricing information must be specified using either one of the set of price information fields (see General Usage Rules section)

ValidUntilTime – used by the Initiator to indicate the period of time the resulting Quote must be valid for

ExpireTime – used by the Initiator to indicate the period of time when this quote request expires

OrderQtyData component block – required when QuoteType is "Tradeable"

Quote Response

Initiator will use the QuoteRespType field to indicate what type of response this is, i.e. "hit/lift", "counter", etc.

IOIid is required if the Quote Response is used to respond to an IOI (offering) message, the field would contain the ID of the IOI message.

Fields required when QuoteRespType is "hit/lift" or "counter quote": OrderQtyData component block, Side, ValidUntilTime, ClOrdID (see paragraph below), and either one of the set of price information fields (see General Usage Rules section).

In the initial use of the "hit/lift" QuoteRespType, the Initiator is required to assign a ClOrdID. This ClOrdID will be reused throughout the negotiation process, including in the "counter", until the negotiation ends in either a fill or the negotiation dialog is terminated by either party.

In a "counter quote" to a Quote, only a limited set of data elements can change depending on the security type. Price can be expected to change, but also Instrument being quoted can change in some markets as well as Stipulations and ClearingCode within the Parties component block.

In a "counter quote" with a "my price" set, OrdType must be "Limit" and either one of the set of price information fields (see General Usage Rules section).

Quote

Fields required when QuoteType is "counter" or "Tradeable": OrderQtyData component block, Side, ValidUntilTime, and either one of the set of price information fields (see General Usage Rules section).

New Order - Single

For OrdType only the following enumeration are applicable: 1 (market), 2 (limit), D (previously quoted), E (previously indicated).

For OrdType of "limit" either one of the set of price information fields (see General Usage Rules section) is required.

TradeDate is required and is set by the Initiator.

HandlInst is required by the protocol but is not a required field for FI. However, for the purposes of being compliant to the protocol the counterparties should bilaterally agree on the value to use.

New Order - Multileg

TradeOriginationDate is used for municipal new issue market. Specifies the date in which agreement in principal between counterparties, prior to actual TradeDate.

TradeDate is required and is specified by Initiator.

© Copyright, 2007-2008, FIX Protocol, Limited

For the Multileg Order, if the following fields are not applicable to all legs of the trade then the NestedParties component block associated with each leg within the NoLegs repeating block will be used: Account, AccountType, NoAllocs repeating block, SettlType, and SettlDate.

Execution Report

This message should always use SettlType "future" with a value for SettlDate.

Stipulations component block information must be reiterated and echo back by the Respondent if Initiator had provided information in the Stipulations component block.

For multilegs only use the NoLegs blocks of the Execution Report message for swaps/switches/rolls when OrdStatus is "new". The partial fill or fill (OrdStatus) Execution Report for each of the legs will be reported separated and execution price for each leg is conveyed in LastPx, AvgPx and LastPxPar, if applicable.

The following fields are required when OrdStatus is "partial", "filled" or "calculated": PriceType, Price

The following fields are required when ExecType is "trade" or "trade correct": LastQty, LastPx, AvgPx, LastPxPar (when conditionally applicable)

The following fields are required when OrdStatus is "filled" or "calculated" AND if NumDaysInterest is populated and not zero: AccruedInterestRate, AccruedInterestAmt

GrossTradeAmt and NetMoney is required when OrdStatus is "filled" or "calculated".

NumDaysInterest is required where applicable based on security type and when OrdStatus is "filled" or "calculated".

InterestAtMaturity is required in lieu of AccruedInterestAmt for security types that pay lump-sum at maturity.

Allocation Instruction

PreviouslyReported, ReversalIndicator and MatchType is conditionally required when Initiator is sending the Allocation Instruction message to a 3rd party or VMU.

This message should always use SettlType "future" with a value for SettlDate.

GrossTradeAmt – Initiators are required to send this information when sending Allocation post-trade.

For Financing Trades Use QtyType and ContractMultiplier if necessary to identify how quantities are to be expressed and specify in OrderQty the block cash amount to be allocated and in AllocQty the cash amount to be assigned to each fund.

Allocation Report

Respondents are required to send this information when reporting the Allocation back with calculations.

NetMoney is required from Respondents when reporting the Allocation back with calculations.

NumDaysInterest, AccruedInterestAmt and AccruedInterestRate is required from Respondents when reporting the Allocation back with calculations for security types where this information can be derived or is available.

InterestAtMaturity is required in lieu of AccruedInterestAmt for security types that pay lump-sum at maturity.

AllocNetMoney is required from Respondents when reporting the Allocation back with calculations.

AllocAccruedInterestAmt is required, if the value is not zero, from Respondents when reporting the Allocation back with calculations. AllocAccruedInterestAmt should be calculated and rounded appropriately for each allocation instance. This means that the sum of AllocAccruedInterestAmt will not always match AccruedInterestAmt.

© Copyright, 2007-2008, FIX Protocol, Limited

AllocInterestAtMaturity is required, if value is not zero, from Respondents when reporting the Allocation back with calculations. AllocInterestAtMaturity is required in lieu of AllocAccruedInterestAmt for security types that pay lump-sum at maturity. Similar to AccruedInterestAmt, the sum of AllocInterestAtMaturity will not always match InterestAtMaturity.

For Financing Trades use the same quantity rules as given for the Allocation Instruction above.

Trade Capture Report

This message should always use SettlType “future” with a value for SettlDate.

Parties component block is required.

GrossTradeAmt and NetMoney are required.

NumDaysInterest is required where information is applicable.

AccruedInterestRate is required if NumDaysInterest is used and is not zero.

AccruedInterestAmt is required is required for security types that trade with accrued interest.

InterestAtMaturity is required in lieu of AccruedInterestAmt for security types that pay lump-sum at maturity.

Instrument component block

Symbol – use “[N/A]” when there are no applicable symbol. For corporate bonds the symbol or ticker for the company issuing the security can be used in this field.

SecurityID and SecurityIDSource are both required.

SecurityType is required

Factor is conditionally required when it is not equal to one (1) for MBA, TIPS, ABS.

OrderQtyData component block

OrderQty is to be expressed as par amount.

Repurchase Agreements (Repo) and Collateral Management

Repo Terminology

The following table maps Repurchase Agreements and Security Lending terminology to FIX data elements with additional usage explaination specific to repos and security lending.

Element	Description	FIX fields	Usage
Accrued interest	Start accrued interest calculated using the day count method appropriate to the underlying security	AccruedInterestAmt	
Allocating entity	The party responsible for assigning specific securities and amounts to the trade	<Parties>	PartyRole 39 = Allocating Entity

© Copyright, 2007-2008, FIX Protocol, Limited

Element	Description	FIX fields	Usage
Call or put dates	Dates on which the seller or buyer may liquidate the position	<Instrument> NoEvents (group) EventType EventDate EventPx EventText	
Cash amount	Amount of currency	StartCash	
Cash outstanding	The current balance of the cash debt	CashOutstanding	
Clean price	Spot price of the security without accrued interest	<UnderlyingInstrument> UnderlyingPx	
Collateral assignment reason	The reason for an initial assignment or subsequent substitution of collateral for a financing deal	CollAsgnReason 0 = Initial 1 = Scheduled 2 = Time Warning 3 = Margin Deficiency 4 = Margin Excess 5 = Forward Collateral Demand 6 = Event of default 7 = Adverse tax event	
Collateral value	Repo value times the inverse of haircut, also known as the “all in” price	TotalNetValue	At the initial collateral assignment TotalNetValue is the sum of (UnderlyingStartValue * (1-haircut)). In a collateral substitution TotalNetValue is the sum of (UnderlyingCurrentValue * (1-haircut)).
Contract currency	The base agreement currency, not necessarily the same as the payment currency	<FinancingDetails> AgreementCurrency	
Currency of payments	Currency in which payments are to be made	Currency	
Day count	Method for calculating accrued interest – 30/360, actual/360, actual/actual, actual/365, 30/365.		Not supported directly in the protocol – understood in the context of the underlying security type and master agreement

© Copyright, 2007-2008, FIX Protocol, Limited

Element	Description	FIX fields	Usage
Delivery	Delivery or custody of underlying securities	<FinancingDetails> DeliveryType	DeliveryType 0 = "Versus. Payment": Deliver (if Sell) or Receive (if Buy) vs. (Against) Payment 1 = "Free": Deliver (if Sell) or Receive (if Buy) Free 2 = Tri-Party 3 = Hold In Custody
Dirty price	Spot price of the security including accrued interest	<UnderlyingInstrument> UnderlyingDirtyPrice	
End consideration	Total cash returned at the end of the term	EndCash	
End date	Close date, date of the return of the securities for monay, "off" date	<FinancingDetails> EndDate	
Face or cash fill	In collateral assignment and substitution dictates whether the quantity of the replacement security is to be based on par-for-par (face) or value-for-value (cash).	<Stipulations>	StipulationType=FILL StipulationValue=<face or cash>
Flex schedule	Single maturity but moneygiver's cash may be returned most often on a predetermined paydown schedule	<FinancingDetails> TerminationType <Stipulations>	StipulationType=PAYFREQ StipulationValue= <dates>
Forward accrued interest	End accrued interest calculated using the day count method appropriate to the underlying security	EndAccruedInterestAmt	
Forward price	Price agreed to on the end leg of the transaction – will vary for indexed bonds	Price2	Denominated in the same type as Price
Frequency of substitutions	Maximum frequency – monthly, semi-annually, annually	<Stipulations>	StipulationType=SUBSFREQ StiuplationValue=<frequency>, e.g. M

© Copyright, 2007-2008, FIX Protocol, Limited

Element	Description	FIX fields	Usage
General collateral	Securities collateralizing a repurchase agreement described generally (treasuries, corporates) rather than specifically by identifier.	<Instrument> <UnderlyingInstrument> UnderlyingSecurityType TREASURY PROVINCE AGENCY MORTGAGE CP CORP EQUITIES SUPRA CASH	Product=FINANCING SecurityType=REPO SecuritySubType=GENERAL UnderlyingSecurityType=TREASURY *If bonds of a particular issuer or country are wanted and UnderlyingSecurityType is not granular enough, include UnderlyingIssuer, UnderlyingCountryOfIssue, UnderlyingProgram, UnderlyingRegType, and/or <UnderlyingStipulations>* *Examples:* SecurityType=REPO UnderlyingSecurityType=MORTGAGE UnderlyingIssuer=GNMA SecurityType=REPO UnderlyingSecurityType=AGENCY UnderlyingIssuer=CA Housing Trust UnderlyingCountryOfIssue=CA SecurityType=REPO UnderlyingSecurityType=CORP UnderlyingNoStipulations=1 UnderlyingStipulationType=RATING UnderlyingStipulationValue=>bbb-
Haircut	Reduction in market value taken on assigned securities in calculating their collateral value – based on market volatility and credit.	<UnderlyingStipulations>	UnderlyingStipType=HAIRCUT UnderlyingStipValue=<percent>
Largest piece	Maximum size of securities acceptable in the transaction	<Stipulations>	StipulationType=MAXDNOM StiuplationValue=<size>
Lookback days	Number of business days prior to floating rate reset date when the benchmark price will be captured and used to determine the new rate upon reset	<Stipulations>	StipulationType=LOOKBACK StiuplationValue=<number of days>

© Copyright, 2007-2008, FIX Protocol, Limited

Element	Description	FIX fields	Usage
Margin	The fraction of the cash consideration that must be collateralized, expressed as a percent. A MarginRatio of 102% indicates that the value of the collateral (after deducting for "haircut") must exceed the cash consideration by 2%.	<FinancingDetails> MarginRatio	
Margin excess	The amount by which the total net value of collateral times margin ratio exceeds cash outstanding	MarginExcess	
Market value	Dirty price times nominal amount	*not supported directly – see Repo value*	
Master agreement	The name of the standard master agreement forming the basis of the financing relationship	<FinancingDetails> AgreementDesc AgreementID AgreementDate Current list of master agreements, amendments and annexes: MRA 1996 Repurchase Agreement MRA 1996 Repurchase Agreement – Annex I 1997 (for FASB 125 compliance) MRA 1996 Repurchase Agreement – Amended 1997 for FASB 125 MRA 1996 International Transaction (Annex III) MRA 1996 Agency Transaction (Annex IV) MRA 1996 Forward Transaction (Annex V) MRA 1996 Buy/Sell Back Transaction (Annex VI) MRA 1996 Equity Securities Transaction (Annex VIII, Feb 1998) MRA 1996 Japanese Financial Institutions Transaction (Annex IX, Aug 2002) MRA 1987 Repurchase Agreement MRA 1987 Repurchase Agreement – Amended 1997 for FASB 125 GMRA 2000 Repurchase Agreement GMRA 2000 Agency Transaction GMRA 2000 Bills Transaction (U.K.) GMRA 2000 Forward Transaction GMRA 2000 Buy/Sell Back Transaction GMRA 2000 Equities Transaction	

© Copyright, 2007-2008, FIX Protocol, Limited

Element	Description	FIX fields	Usage
		GMRA 2000 Canadian Transaction GMRA 2000 Italian Transaction GMRA 2000 Japanese Transaction GMRA 2000 Netherlands Transaction GMRA 1995 Repurchase Agreement GMRA 1995 Buy/Sell Back Transaction GMRA 1995 Agency Transaction GMRA 1995 Repurchase Agreement – Amended for GMRA 2000 Conformance GMRA 1995 Buy/Sell Back Transaction – Amended for GMRA 2000 Conformance GMRA 1995 Agency Transaction – Amended for GMRA 2000 Conformance GMRA 1995 Forward Transaction (as enabled by Amendment for GMRA 2000 conformance) GMRA 1992 Repurchase Agreement MSLA 2000 Securities Loan MSLA 2000 Agency Transaction (Annex I) MSLA 2000 Term Loan MSLA 1993 Securities Loan MSLA 1993 Agency Transaction MSLA 1993 Securities Loan – Amended 1998	
Maturity type – fixed or open	Open (term is indefinite and may be terminated by either party on demand) or Fixed (pre-determined, may be overnight or from one day to five years). Termination prior to maturity is open to negotiation.	<FinancingDetails> TerminationType 1 = Overnight 2 = Term 3 = Flexible 4 = Open	
Maximum pieces	Maximum number of pieces acceptable in the transaction	<Stipulations>	StipulationType=PMAX StiuplationValue=<count>
Minimum pieces	Minimum number of pieces acceptable in the transaction	<Stipulations>	StipulationType=PMIN StiuplationValue=<count>
Number of substitutions	Number of substitutions permitted	<Stipulations>	StipulationType=MAXSUBS StiuplationValue=<count>
Other dynamic stipulations		<Stipulations>	StipulationType=TEXT StiuplationValue=<text>
Par quantity	Face or nominal value of securities	<UnderlyingInstrument> UnderlyingQty	

© Copyright, 2007-2008, FIX Protocol, Limited

Element	Description	FIX fields	Usage
Payment calendar	Schedule of dates based on frequency of interest payments	<Stipulations>	StipulationType=PAYFREQ StipulationValue= <dates>
Payment interval	Payment interval, i.e. 3 months, 6 months, etc.	<Stipulations>	StipulationType=PAYFREQ StipulationValue=<interval> e.g. 3M
Percent of variance	Maximum variance allowable in the value of replacement securities	<Stipulations>	StipulationType=TRDVAR StiuplationValue=<count>
Rate reset calendar	Schedule of dates based on frequency	<Stipulations>	StipulationType=PRICEFREQ StipulationValue=<dates>
Rate reset interval	Reset interval, i.e. 3 months, 6 months, etc.	<Stipulations>	StipulationType=PRICEFREQ StipulationValue=<frequency> e.g. 6M
Rate type	How the yield paid on the cash investment is to be calculated	PriceType 9 [yield = Fixed Rate] 6 [spread = Floating Rate] <BenchmarkCurveData>	
Repo rate	The fixed yield or yield spread paid on the cash investment	Price	expressed in yield or spread to benchmark
Repo value	Market value rounded using the appropriate market practice convention of the security in the repo market.	<UnderlyingInstrument> UnderlyingStartValue UnderlyingCurrentValue UnderlyingEndValue	These fields are the repo value (rounded market value) of each piece of collateral at the start, current and end of the deal. Haircut is not factored in these values. The respondent is free to populate these fields as needed based on the purpose of the current message, but we recommend UnderlyingStartValue on initial assignment and UnderlyingCurrentValue on substitution since TotalNetValue is conditionalized on these actions.
Securities lending fee	Used in lieu of interest rate of Fee-based transactions	MiscFeeType MiscFeeAmt	MiscFeeType 13 = Securities Lending
Security rating range	Minimum acceptable rating on any securities involved in the transaction	<Stipulations>	StipulationType=RATING StiuplationValue=<source / range>
Smallest piece	Minimum size of securities acceptable in the transaction	<Stipulations>	StipulationType=MINDNOM StiuplationValue=<size>
Spot price	Price for the start leg of the transaction	Price PriceType 1 = Percentage 2 = Per unit 3 = Fixed amount	

© Copyright, 2007-2008, FIX Protocol, Limited

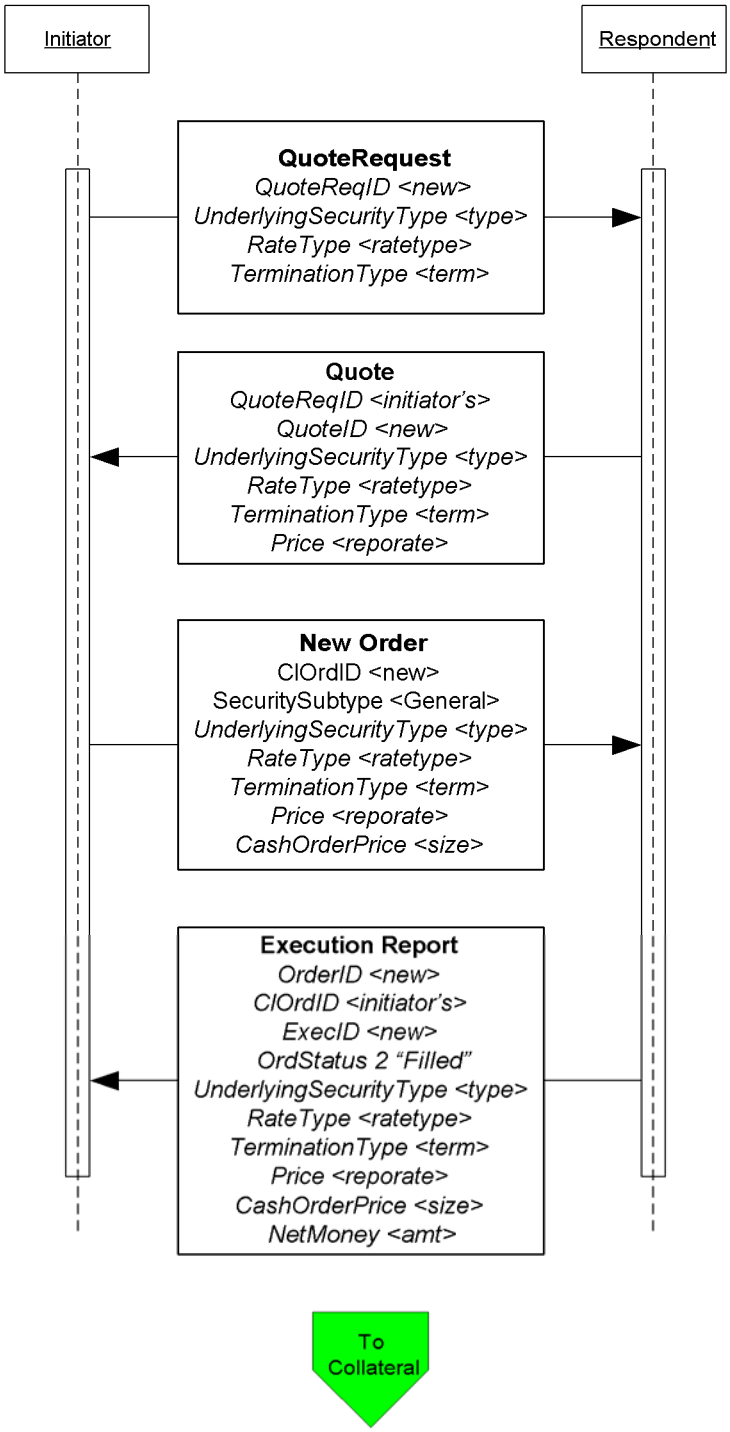
Element	Description	FIX fields	Usage
Start consideration	Total cash remitted at the beginning of the term	StartCash	
Start date	Settlement date for "on" date or "start leg"	<FinancingDetails> StartDate	
Trade date	Date of trade agreement	TradeDate	
Type of financing deal	Attributes of the financing arrangement – Repo, Reverse Repo, Sell/Buy, Buy/Sell, Fee-based Loan, Fee-based Borrow, Loan vs. Cash, Borrow vs. Cash, Fee-based Loan vs. Cash, Fee-based Borrow vs. Cash, Master Forward Sell/Buy, Master Forward Buy/Sell, Sec Lend, Sec Borrow, Borrow Pledge *Often combined with* Overnight, Term, Flexible, Open	<Instrument> SecurityType REPO – repurchase agreement FORWARD – forward BUYSELL – buy/sellback or sell/buyback SECLOAN – securities loan SECPLEDGE – securities pledge Side <FinancingDetails> TerminationType StartDate EndDate <UnderlyingInstrument>	Product=FINANCING SecurityType=REPO SecuritySubType=GENERAL Side=<buy, sell, lend, borrow> TerminationType=<type> StartDate=<start> EndDate=<end> UnderlyingSecurityType=<type> AgreementDesc=<master agreement>

Collateral Management

The following diagrams illustrates an example flow for collateral management once a repo or financing deal has been completed. Figures 8 to 11 shows an example for 2-party model and Figure 12 shows an example for 3-party model.

© Copyright, 2007-2008, FIX Protocol, Limited

Figure 8: Example flow of Repo Trade



Click ***here*** to go to "Collateral Assignment"

© Copyright, 2007-2008, FIX Protocol, Limited

Figure 9: Example flow for Collateral Assignment

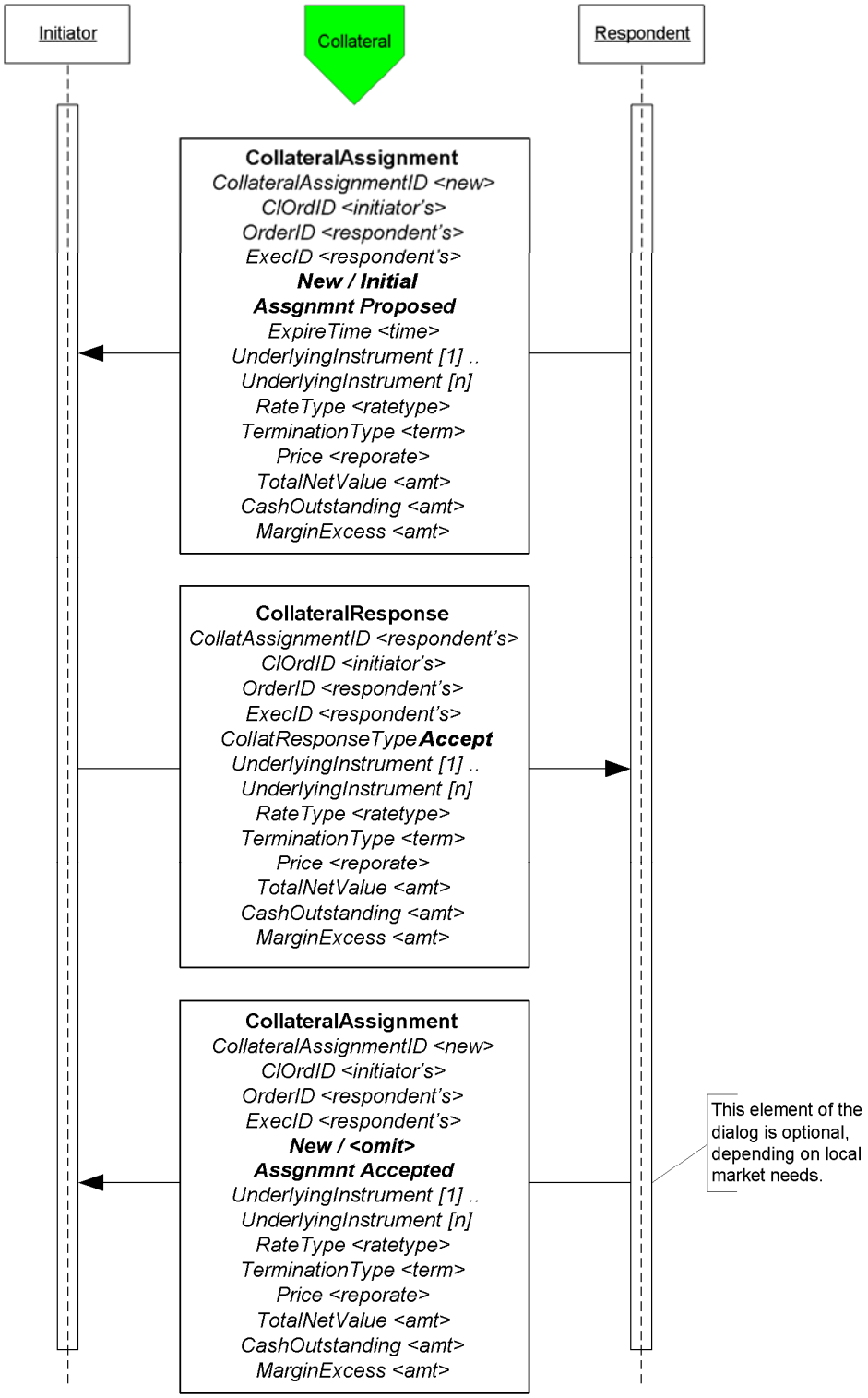
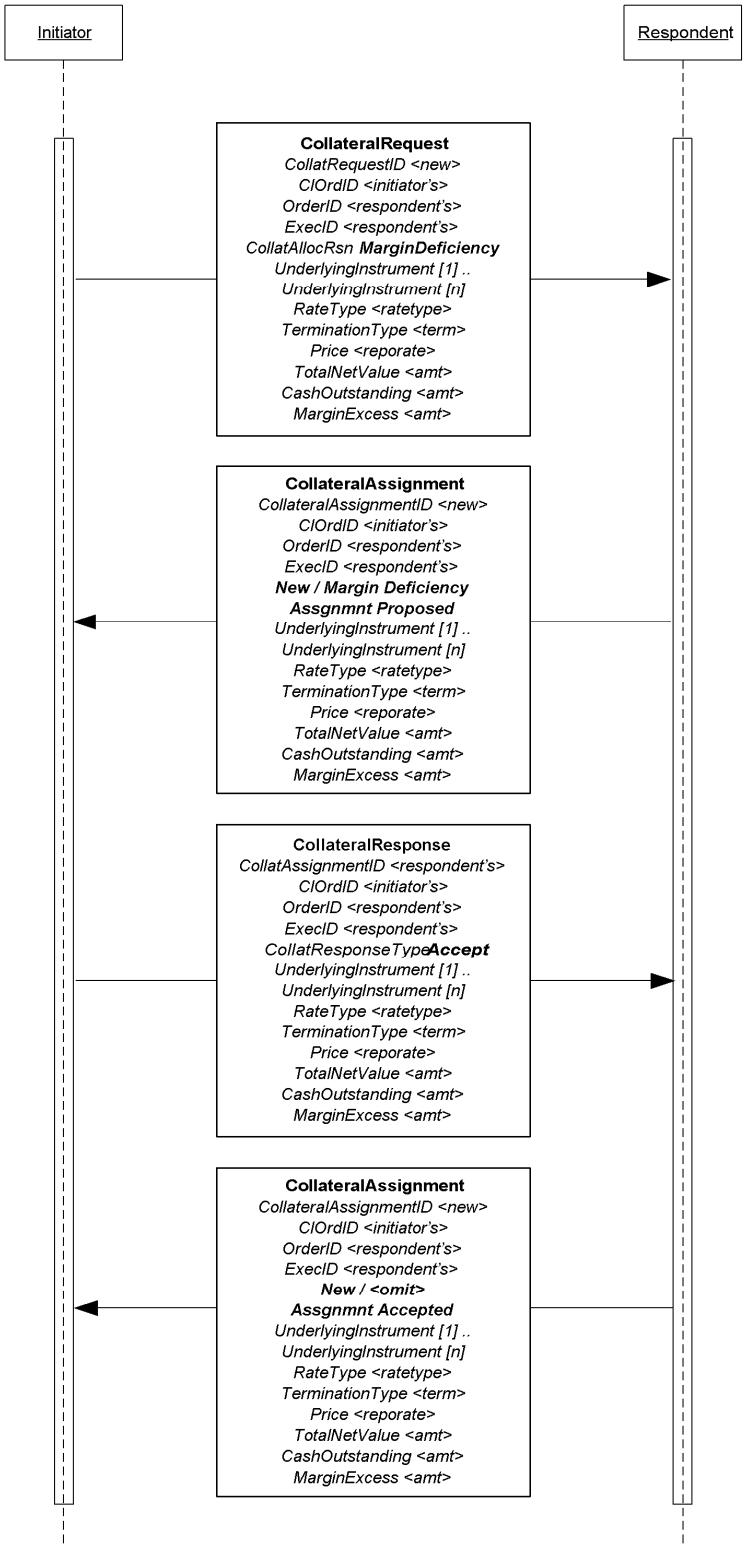
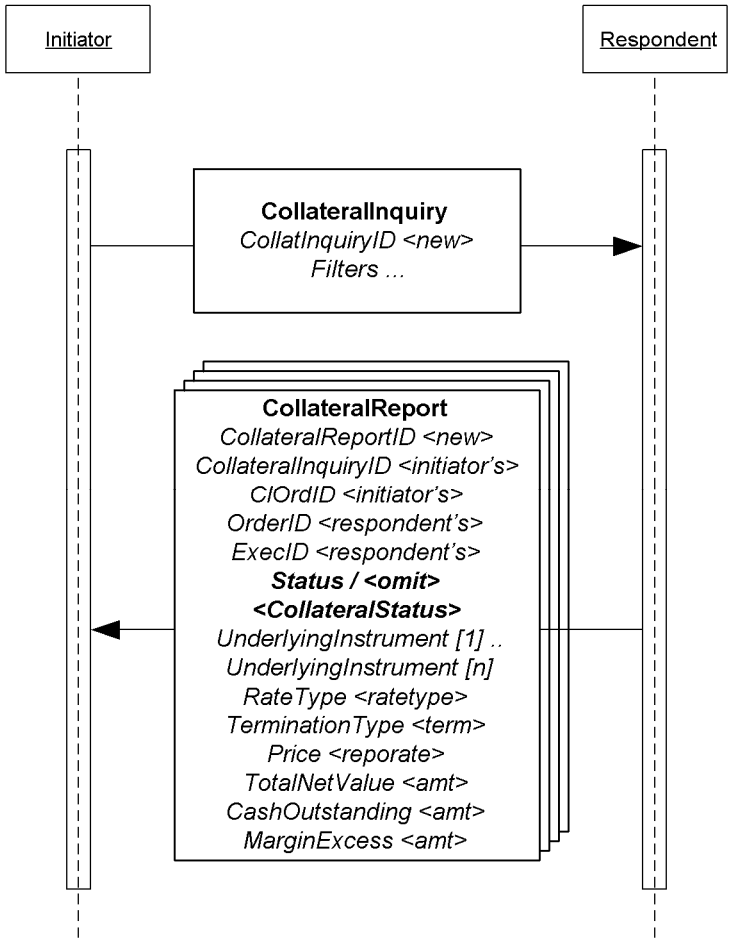


Figure 10: Example use of Collateral Request



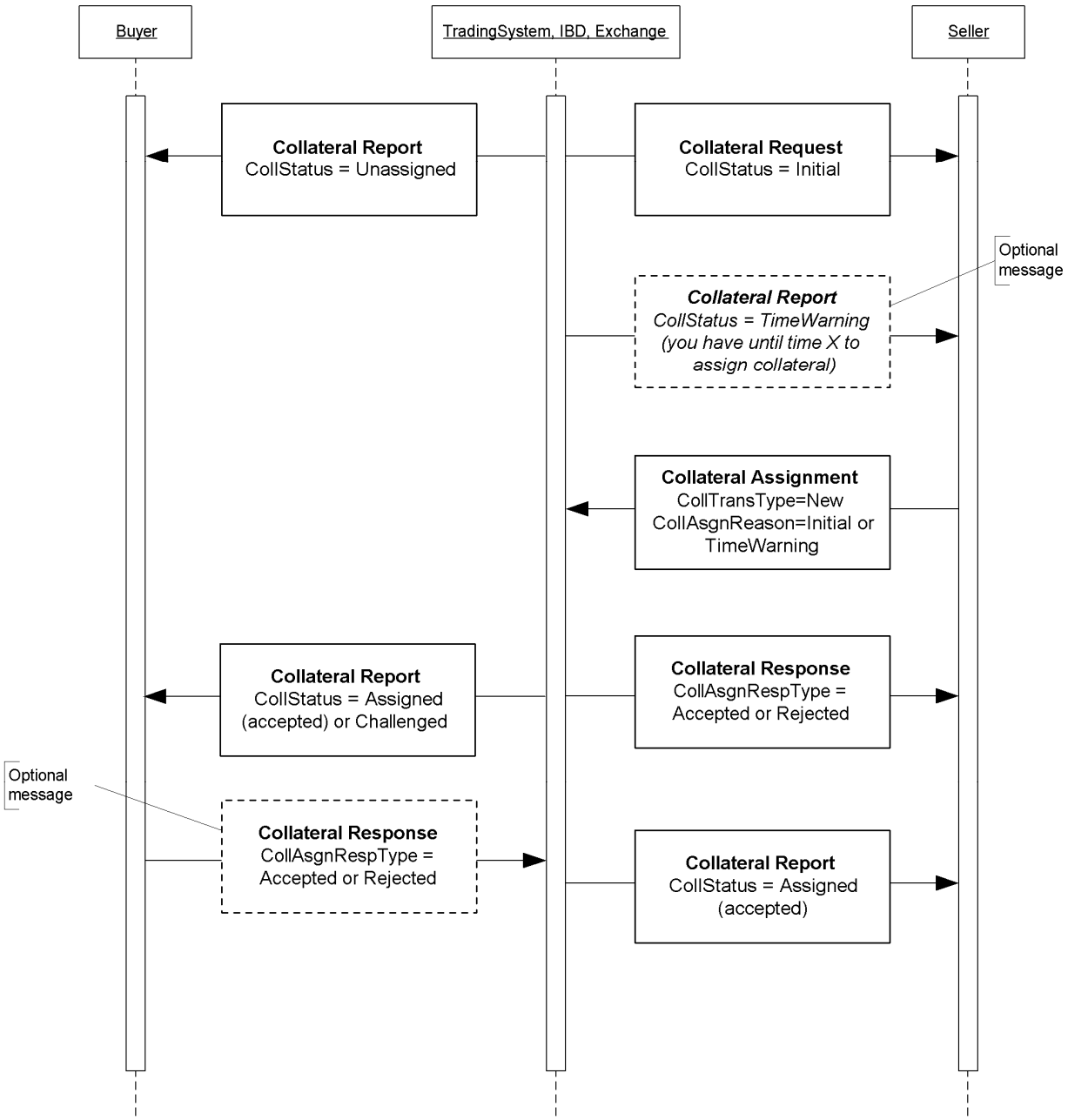
© Copyright, 2007-2008, FIX Protocol, Limited

Figure 11: Collateral Inquiry



© Copyright, 2007-2008, FIX Protocol, Limited

Figure 12: 3-Party Collateral flow



Identifying Euro Issuers

Euro CountryOfIssue Codes:

Use ISO codes in CountryOfIssue to identify the issuing country for non-US Governments. Omit CountryOfIssue or use a value of 'XS' when the issuer is a supra-national agency, e.g. the first nine entries in the table below.

Euro Issuer Values:

The list below are used in the Issuer (106) field to further identify the issuer for securities such as EUSUPRA, EUSOV and PFAND (see data dictionary entry to SecurityType (167) in Volume 6. The abbreviations are from Bloomberg.

*Credit / Sovereign issued in any currency.

COE	Council of Europe
DTA	Deutsche Ausgleichsbank
EBRD	European Bank for Reconstruction and Development
EIB	European Investment Bank
HESLAN	Hessen
KFW	Kreditanstalt fuer Wiederaufbau
LANREN	Landwirtschaftliche Rentenbank
NORWES	Nord-Rhein-Westfalen NRW
SACHAN	Sachsen-Anhalt

RATB	Austrian Treasury Bill
RAGB	Austrian Government Bond
AOBL	Austrian Bundesobligation (OBL)
RABSS	Austrian Bundesschatzscheine
AUST	Austrian Government International Bond*
RAGBS	RAGB Coupon Strip (Austrian)
RAGBR	RAGB Principal Strip (Austrian)
RAMTB	Austria Medium Term Bill

BGTB	Belgian Treasury Bill
BGB	Belgian Government Bond
BELG	Belgian Government International Bond
OLOS	Belgian Strip
OLOR	Belgian Principal Strip

© Copyright, 2007-2008, FIX Protocol, Limited

DGTB	Danish Treasury Bill
DGB	Danish Government Bond
DENK	Danish Government International Bond* (DKK)
RFTB	Finnish Treasury Bill
RFGB	Finnish Government Bond
FINL	Finnish Government International Bond*
FNHF	Finnish Housing Bond
BTF	BTF - French Fixed-Rate Short Term Discount Treasury Bills
BTNS	BTAN - French Fixed-Rate Treasury Notes
FRTR	OAT - French Treasury Bonds
FRTRR	OAT - French Treasury Bonds Principal STRIPS
FRTRS	OAT - French Treasury Bonds Coupon STRIPS
CADES	Social Security Debt Repayment Fund (French)*
BUBILL	German Treasury Bill
DBSB	German Federal Treasury Bill (rarely used puttable & DM Ccy)
BKO	German Two Year Notes
FSDB	German Financing Treasury Notes (DM Ccy)
DBR	German Government Bond
DBRR	German Government Bond Principal STRIPS
DBRS	German Government Bond Coupon STRIPS
OBL	German Five Year Bonds
DBRUF	German Unity Fund DBR – S (only 2)
BKOUF	German Unity Fund – BKO (None)
DBP	German Federal Post -- BUNDESPOST
DBB	German Federal Railroad --BUNDESBAHN
THA	Treuhand Agency Bonds
TOBL	Treuhand Agency Obligations – All matured
ENTFND	German Retribution Fund – Only 2 sinking funds
GERP	European Recovery Program Special Funds (German only 2)
BUKASS	Bundeskassenscheine – 1 matured
GTB	Hellenic Republic Treasury Bill
GGB	Hellenic Republic Government Bond

© Copyright, 2007-2008, FIX Protocol, Limited

GREECE	Hellenic Republic Government International Bond*
GGBSTP	Hellenic Republic Government Bond Coupon STRIPS
GGBRES	Hellenic Republic Government Bond Residual STRIPS
IRISH	Irish Government Bond
IRELND	Irish Government International Bond*
BOTS	Italian Treasury Bill
BTPS	Italian Government Bond
CCTS	Italian Treasury Certificate
ICTZ	Italian Zero Coupon Bonds
CTES	Italian Government Bonds Issued in EUR –Matured
CTOS	Italian Government Bonds with Put Option – All matured
ITALY	Italian International Bonds*
BTPSS	Italian Government Bond Coupon STRIPS
BTPSR	Italian Government Bond Residual STRIPS
LGB	Luxembourgeois Government Bond
NETHER	Dutch Government Bond
NETHRR	Dutch Principal Strip
NETHRS	Dutch Strip
DTB	Dutch Treasury Certificate
NBC	Dutch Bank Certificate – All matured
NGTB	Norwegian Treasury Bill
NGB	Norwegian Government Bond
NORWAY	Norwegian Government International Bond* (NOK)
PORTB	Portuguese Treasury Bills
PGB	Portuguese Government Bond
PORTUG	Portuguese Government International Bond*
SPGB	Spanish Government Bond
SPGBS	Spanish Government Bond Coupon Strips
SPGBR	Spanish Government Bond Principal Strips

© Copyright, 2007-2008, FIX Protocol, Limited

SPAIN Spanish Government International Bond*

SGLT Spanish Letras del Tesoro

SWTB Swedish Treasury Bill

SGB Swedish Government Bond

SWED Swedish Government International Bond* (SEK)

SGBS Swedish Government Bond Coupon Strip

SGBR Swedish Government Bond Residual Strip

SWISTB Swiss Treasury Bill

SWISS Swiss Government Bond

GENTB Geneva Treasury Bill (CHF)

UKTB United Kingdom GBP/EUR Treasury Bill

UKT United Kingdom Gilt Bond

UKTS United Kingdom Gilt Bond Coupon STRIPS

UKTR United Kingdom Gilt Bond Residual STRIPS

UKIN United Kingdom International Bond*

BOE Bank of England EUR Bill

BOEN Bank of England EUR Note

Example usage of FI specific component blocks

Example usage of BenchmarkCurve fields

Note: the following is a **subset** of possible value combinations.

Description/ Common Name	BenchmarkCurveCurrency	BenchmarkCurveName	BenchmarkCurvePoint
Curve	USD	Treasury	INTERPOLATED
5 Year	USD	Treasury	5Y
Old 5 Year	USD	Treasury	5Y-OLD
10 Year	USD	Treasury	10Y
Old 10 Year	USD	Treasury	10Y-OLD
30 Year	USD	Treasury	30Y
Old 30 Year	USD	Treasury	30Y-OLD
3 Month LIBOR	USD	LIBOR	3M
6 Month LIBOR	USD	LIBOR	6M
Canadian	CAD	Treasury	INTERPOLATED

© Copyright, 2007-2008, FIX Protocol, Limited

UK Curve	GBP	Treasury	INTERPOLATED
ECU/EURO	EUR	Treasury	INTERPOLATED
US Swap	USD	SWAP	INTERPOLATED
Euro Swap	EUR	SWAP	INTERPOLATED
EDFS	EUR	FutureSWAP	INTERPOLATED
German Bund	DEM	Treasury	INTERPOLATED
US MuniAAA	USD	MuniAAA	10Y
US T point	USD	Treasury	2/2031 5 3/8 (combination of maturity and coupon)

Example usage of Stipulation fields

NoStipulations	StipulationType	StipulationValue	Description of the Stipulation
4	WALA	>=60	Weighted average loan age Less than or equal to 60 months
	TRDVAR	.0025	Trade variance .25%
	PSA	.25	Prepayment speed 25%
	GEOG	ORANGE OR CONTRACOSTA	Geographics Orange OR Contra Costa Counties

© Copyright, 2007-2008, FIX Protocol, Limited

PRODUCT: FOREIGN EXCHANGE

Introduction

This section of the FIX Protocol specification describes how FIX messages can be used to support FX trading activities - executable streaming prices, request for quotes, order initiation and execution. This body of work is the effort of the Global Foreign Exchange Committee (formed in the summer of 2005) and its sub-committees. The GFXC will continue to enhance this section of the specification as new FX-related functionality is supported by FIX.

The FX asset types support by FIX are:

- Spot
- Forwards (outrights)
- FX Swaps
- Vanilla FX OTC Spot Options (post-trade TradeCaptureReport and TradeCaptureReportAck messages only)

The objective of this section is to serve as a starting point and provide guidance to the reader in their implementation of FIX for Foreign Exchange trading. Note that discussions around FX accommodation trades (i.e. indicating that an FX trade be conducted as part of a transaction in a foreign security) are currently not covered in this section.

Message Dialog

In FX the trading dialog typically starts with a request for quote by the customer or a request for streaming prices by the customer. Once the customer receives the rate and quantity desired for the currency pair they wish to deal in the dealer offering the rate will contacted and a trade could be consummated.

The discussed usages of FIX for FX trading focused on the interactions between the customer and the bank or dealer, and illustrated in the diagrams in the following sections[11].

Price Discovery

In FX price discovery there are two main ways in which customers receive prices from their bank or dealer. One is through a request for quote (via phone or electronically) and the second is through a price stream - the latter is typically in electronic form.

In FIX a distinction is made between the two types of price discovery methods. The Quote message set is used to support "one-off" quote requests. The Market Data message set is used to support requests for indicative and executable price streams for FX asset types that do not require negotiation. It should also be noted that the Quote message set will also support "one-off" quote requests that may be "hit" with an order message without any negotiation.[12]

[11] Further enhancements will be made to the protocol to better support FX dealings through 3rd party electronic trading platforms and exhcnages.

[12] Negotiation in FX is currently not covered although the protocol supports such interaction.

© Copyright, 2007-2008, FIX Protocol, Limited

Quoting Message Dialog

The quote/order usage model of the Quote message set shown in Figure 1 is a straightforward request for a "one-off" quote that is then "hit"[13]. The dialog flow is described below.

A. The Initiator or customer requests a quote from the Respondent or dealer. The Respondent responds with either:

   1. a quote by sending the Quote message
   2. decline to provide a quote, reject the request, or indicate that a quote cannot be provided by sending a Quote Request Reject message with an appropriate reject reason. Although the Respondent can let the request time out if an expiration time is specified, it is best practice to explicitly respond with a reject if the dealer does not want to provide or cannot provide a quote.

B. The Initiator then responds with either:

   1. a New Order to accept the quote provided
   2. do nothing and just let the quote expire. The Initiator is not obligated to explicitly indicate to the dealer that they do not wish to act on the quote provided.
   3. a QuoteResponse message to explicitly indicate to the Respondent that the Initiator has either "done away" or "pass" on the quote, or has "expired" when the Quote was received after the ExpireTime in the Quote Request

C. If the Initiator places an order the Respondent responds with an acknowledgement that the order has been received and either:

   1. fills the order by sending an Execution Report
   2. rejects the order. This meets the plain English document requirement that the Respondent reserves the right to reject an order that is placed against a quote.

Additionally in a "one-off" quote request, the dealer may update the Quote as long as the Quote has not expired. The updated Quote may contain a new expiration time or preserve the existing expiration time. The updated Quote is the only live quote, thus rendering the original quote obsolete or canceled. The dealer may also cancel or "withdraw" a live quote prior to its expiration via the use of the Quote Cancel message. Once the most current live Quote has expired or canceled/withdrawn the dealer may not update or "replace" it. It would be up to the customer to issue a new quote request. The dealer may only update quotes corresponding to a Quote Request before the expiration time of the request, indicated in ExpireTime field.

It should be noted that the Quote Request and Quote message interaction is used only for short-lived RFQs and requests for a single rate quote. A "Short-lived" RFQ is defined as a request that has a very short life span, mimicking a rate request that would be made over the phone - typically not longer than 1 or 2 minutes.

[13] It should be noted that in this model the New Order message (rather than the QuoteResponse message) is used to "hit" the tradeable Quote when the Negotiation model is not supported. The QuoteResponse message is used to "hit" a tradeable Quote when the Negotiation model is being supported.

© Copyright, 2007-2008, FIX Protocol, Limited

Figure 1: "One-off" Quote Message Flow

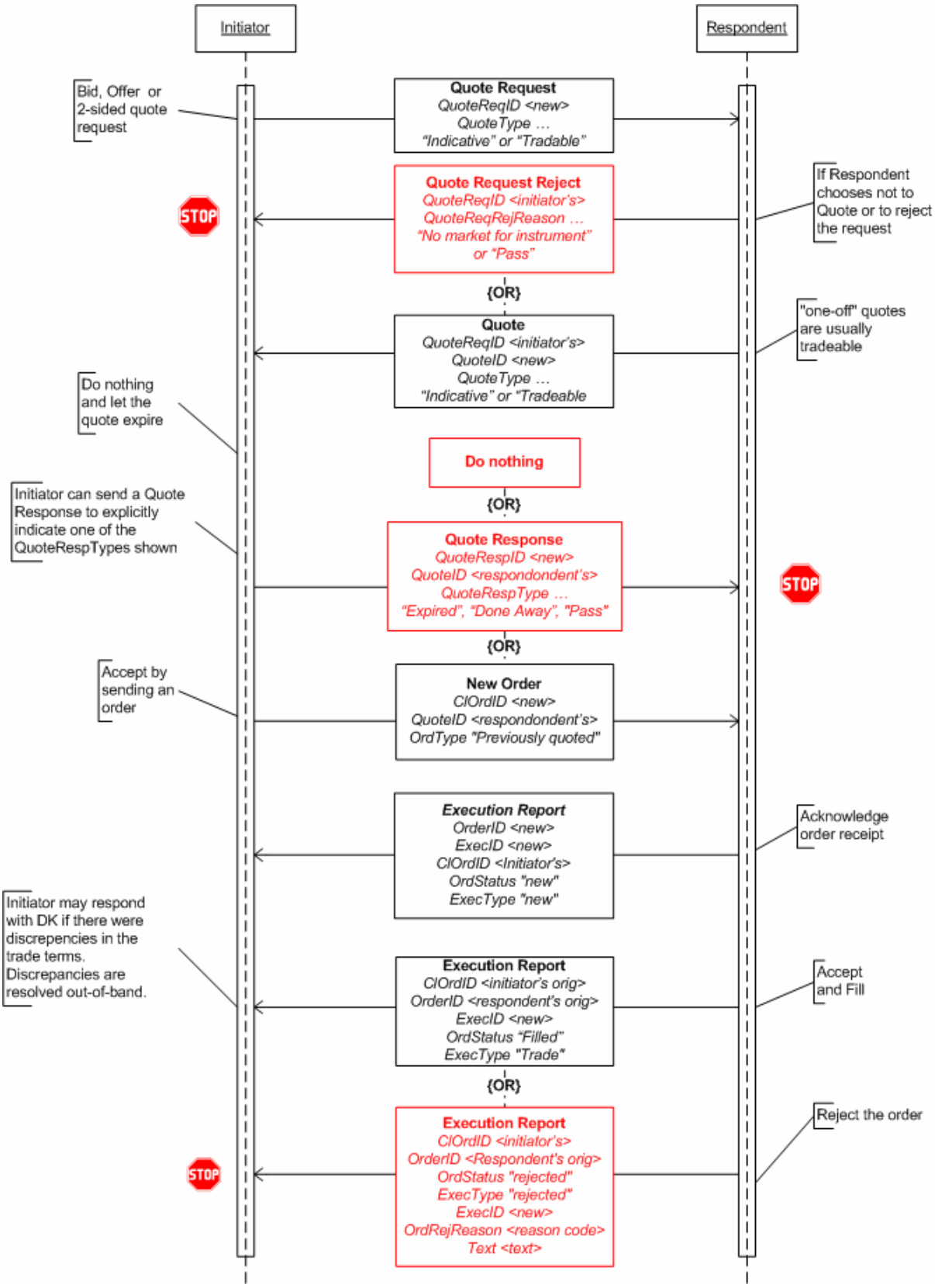
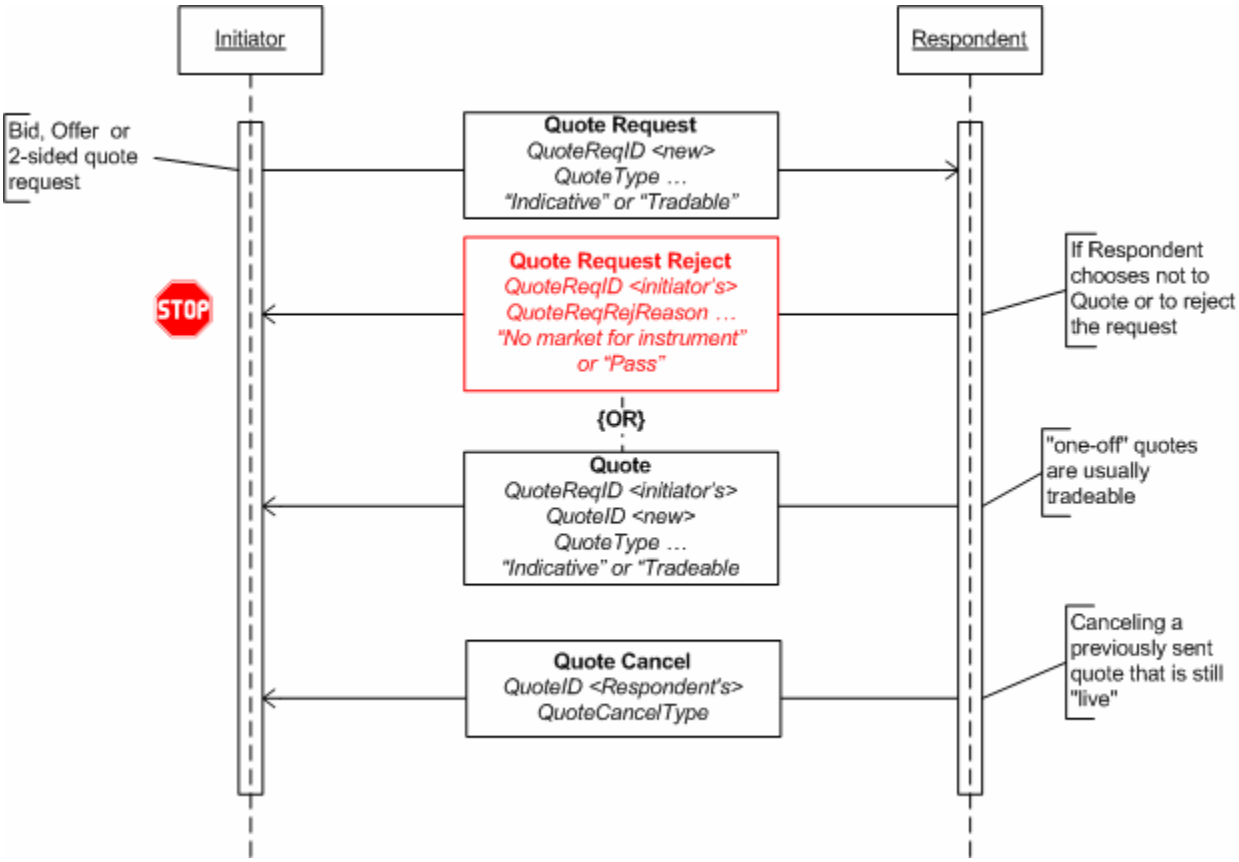


Figure 2: Quote Cancel Message Flow



Streaming Prices Message Dialog

The Market Data messages are used for price stream subscriptions. The message set supports both a subscription request submission via FIX and out-of-band (the latter usage is currently out of scope of this document). There are three general platform models within FX that would stream prices: a) Single Bank; b) "exchange" (e.g. HotSpotFXi); c) multi-bank portals (e.g. FXall).

The dialog flow shown in Figure 3 illustrates at a high level the use of Market Data Request, Market Data Snapshot, Market Data Incremental Refresh, and Market Data Request Reject messages. The dialog flow corresponding to Figure 3 is described below.

Additionally the scope for streaming prices is initially spot prices, while forwards can be accommodated but currently not widely sent in a streaming price feed.

A. The Initiator or customer requests a price stream from the Respondent or dealer by sending a Market Data Request message indicating a subscription with incremental refresh is requested. The Respondent responds with either:

   1. a (or multiple) price stream by sending a Market Data Snapshot message to provide the initial snapshot followed by Market Data Incremental Refresh messages to provide updates.

   2. decline to provide a price stream, reject the request, or indicate that a quote cannot be provided by sending a Market Data Request Reject message with an appropriate reject reason

B. The Initiator can respond by:

© Copyright, 2007-2008, FIX Protocol, Limited

1. doing nothing if the prices do not interest them
2. place an order against a market data entry that interests them at the same price and amount level indicated in the market data entry

C. If an order is placed the Respondent will respond with the Execution Report to acknowledge receipt of the order, followed by a fill (or partial if allowed) or a rejection if the quote can no longer be honored.

D. At any time the Initiator can stop the price stream subscription by sending a Market Data Request message to "unsubscribe"

The Market Data messages will also support a full refresh subscription model. The differences would be in the request and the response would be only Market Data Snapshot messages. For "exchange" style aggregators this may be the Market Data usage model for them.

A price stream can be either executable/tradeable or indicative, where all the quotes in the particular price stream are either tradeable or indicative, not both. The Initiator may explicitly specify in the Market Data Request message that the request is for a tradeable or indicative price stream. If this is not specified the Respondent should assume the request is for indicative price stream unless other arrangements are made bilaterally (e.g. via customer profile configuration).

Trading platforms may require the Initiator to indicate which dealer's prices should be included in the price stream. Alternatively, trading platforms may provide configurable customer profiles where defaulted dealers' prices will always be provided in a price stream unless specifically indicated by the Initiator. Trading platforms may respond with a single consolidated price stream with prices from all requested dealers or open up individual price streams for each dealer, although the latter model is not the preferred model.

Vector Prices

Vector prices are described as price bands for FX rates for a specifically requested currency pair and tenor. The Initiator would submit a request to the Respondent with at least the currency pair and the required tenor. The Respondent who would supports displaying of vector prices may respond with the price bands. Each band would be for an "up to" amount and the price for the band. It is similar to displaying the "depth of book", however, the main difference between other asset types and FX is that the "book" is not swept when an order is received from a customer.

For example: customer submits a request for a price stream for 6-month EUR/USD. The dealer may elect to provide price bands by showing 3 price bands for the requested currency and tenor, for example:

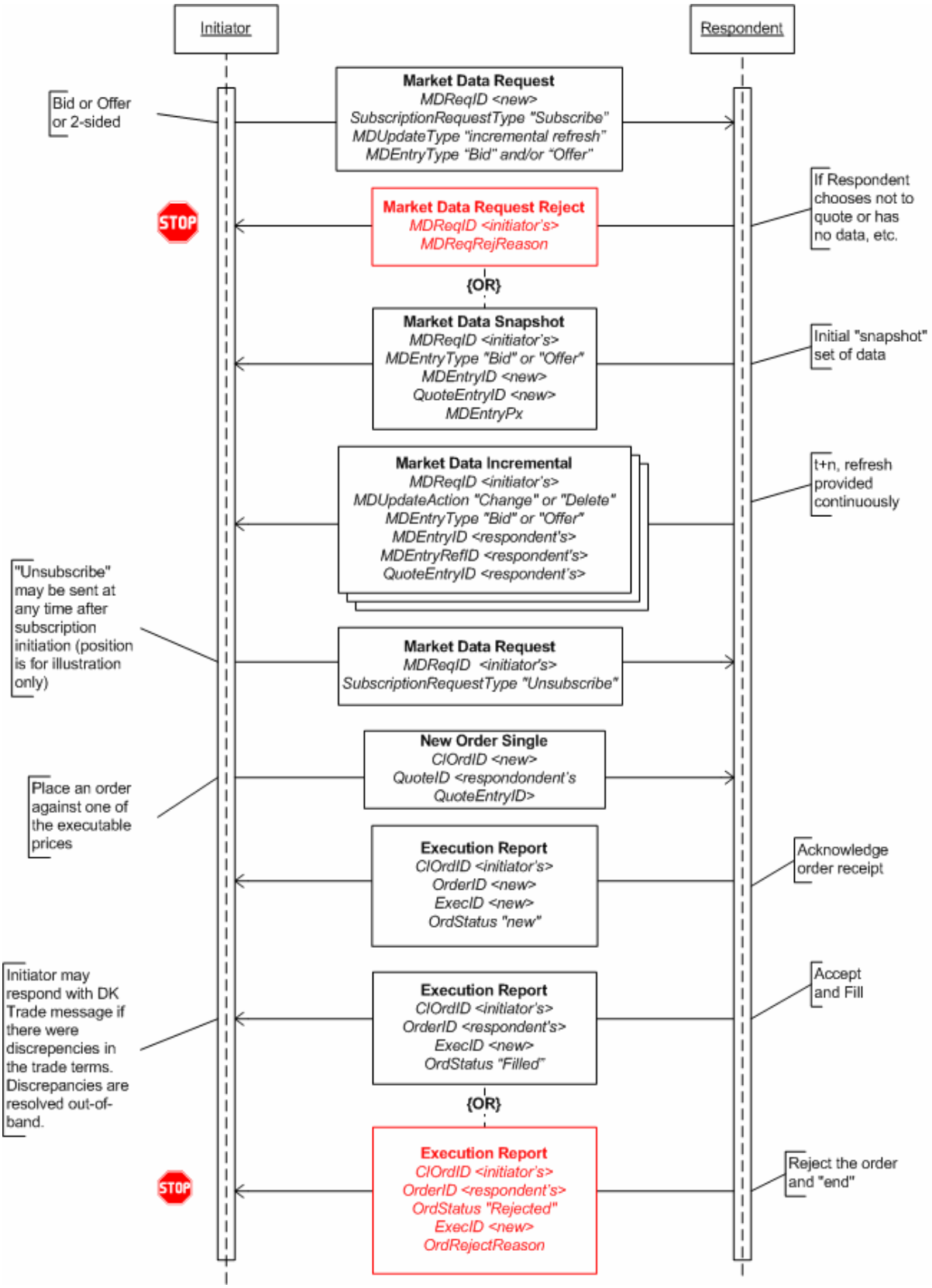
Band 1:	EUR 5,000,000	1.2510 / 1.2512	(bid/offer)
Band 2:	EUR 10,000,000	1.2510 / 1.2513	
Band 3:	EUR 25,000,000	1.2509 / 1.2514	

Each band's size is an "up-to" amount. When a customer places an order, for example, for EUR 7,000,000 then the entire amount will be filled at the price from Band 2 in the example above, not from a combination of Band 1 and Band 2, which would be a "sweep".

Vector prices or price bands are implicitly supported using the Market Data messages. Each market data entry has an identifier, MDEntryID, thus a vector price would be represented by the number of MDEntryIDs needed. In the example above, the Market Data message would identify each price band with its own MDEntryID, as well as QuoteEntryID, for the stated same currency pair and from the same dealer. The difference in the entries would be the amount and rate(s).

© Copyright, 2007-2008, FIX Protocol, Limited

Figure 3: Streaming Price Message Flow



Example Scenarios

The following are example scenarios of how the Market Data Request can be used when sending a request to different types of Respondent and what can be expected as a response.

1. Single Bank

In this scenario the Respondent is a single bank. The customer would likely be requesting prices for one or more currency pairs and may request a specific quantity for each pair. The quantity in the request will be treated as an "up to" quantity.

Market Data Request:

- Customer requests prices for a currency pair
- Customer may also request a specific quantity
- The target bank is implicit

The Respondent may provide prices at different quantity levels if the customer did not request a specific quantity. The bank also may specify that the prices are indicative or tradeable if the customer did not explicitly request tradeable prices.

Market Data messages:

- Bank may provide prices at different quantities if the Customer does not request a specific quantity
- In this case, aggregated and non-aggregated "book" produces the same results
- Prices may be indicative or tradeable

MarketDepth="full" or MarketDepth="top of book". AggregatedBook would not be applicable as a the results would the same as a non-aggregated book.

2. "Exchange" platform (e.g. HotSpotFXi)

The customer would most likely be requesting prices for one or more currency pairs and not likely to include a quantity. In an exchange style platform the request would not be directed at any particular dealer as that information may not be known.

Market Data Request:

- Customer requests prices for a currency pair
- Customer will not specify the quantity
- Request is not targeted at specific banks providing prices

The Responding platform most likely would respond with an aggregated view of the market with full market depth. In other words, showing aggregated quantity for a given bid/ask and multiple bid/ask levels. However, it should be noted that some platforms may choose to only show top of book.

Market Data messages:

- Will be full depth and aggregated to be anonymous
- Prices will be tradeable

3. Multi-bank Portal (e.g. FXall)

On a multi-bank portal the customer would most likely be requesting prices for one or more currency pairs that may be targeted at specific bank(s). As with the single bank scenario, a quantity may optionally be specified.

Market Data Request:

- Customer will request prices for a currency pair
- Customer may request a specific quantity

© Copyright, 2007-2008, FIX Protocol, Limited

- Customer may specify the target bank(s) in the request

The Respondent would likely respond with a non-aggregated view of the market showing full depth of book so that each bank's quotes can be discretely identified. The dealers may also specify whether the prices are indicative or tradeable if the customer did not explicitly request tradeable prices.

Market Data messages:

- Will be full depth and not aggregated, so different bank's quotes can be identified
- Prices may be indicative or tradeable

General Order and Execution Handling

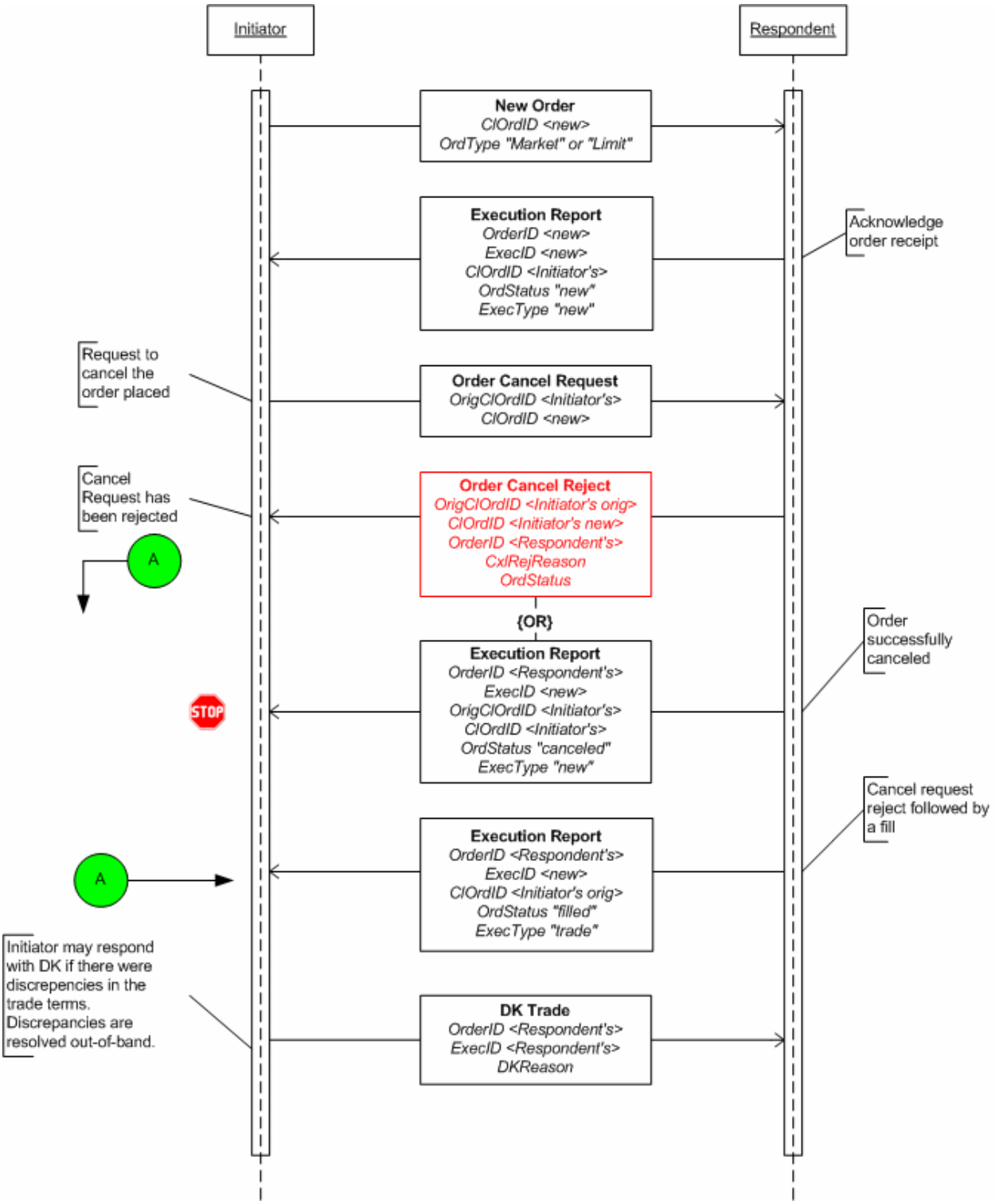
The order and execution message set are shown in both Figure 1 and Figure 3 as the quote or streaming price is "hit" an order is created. The high level description of these steps are described in the sections discussing proposed usage for Quote/Order message usage and Market Data message usage models.

Orders that have been placed may be canceled or replaced. To affect this, the Initiator will send a Cancel Request or Cancel/Replace Request to the Respondent. The Respondent will either accept or reject the request depending on whether the order was filled or not. Figure 4 illustrates this Cancel Request flow. The Cancel/Replace Request flow is very similar and is illustrated in Figure 5.

In the flow of order and execution handling, it is also possible for the Initiator to "reject" an Execution Report sent by the Respondent. This is done using the Don't Know Trade (DK Trade) message. It should be noted that historically DK Trade is used when the order was affected outside of FIX (e.g. directly on a portal's GUI interface) but an Execution Report is received via FIX, allowing the Initiator to disagree with the trade terms as there may be an error in the trade terms. However, DK Trade can also be used in the flow where an order was affected via FIX. Figure 4 illustrates this latter flow.

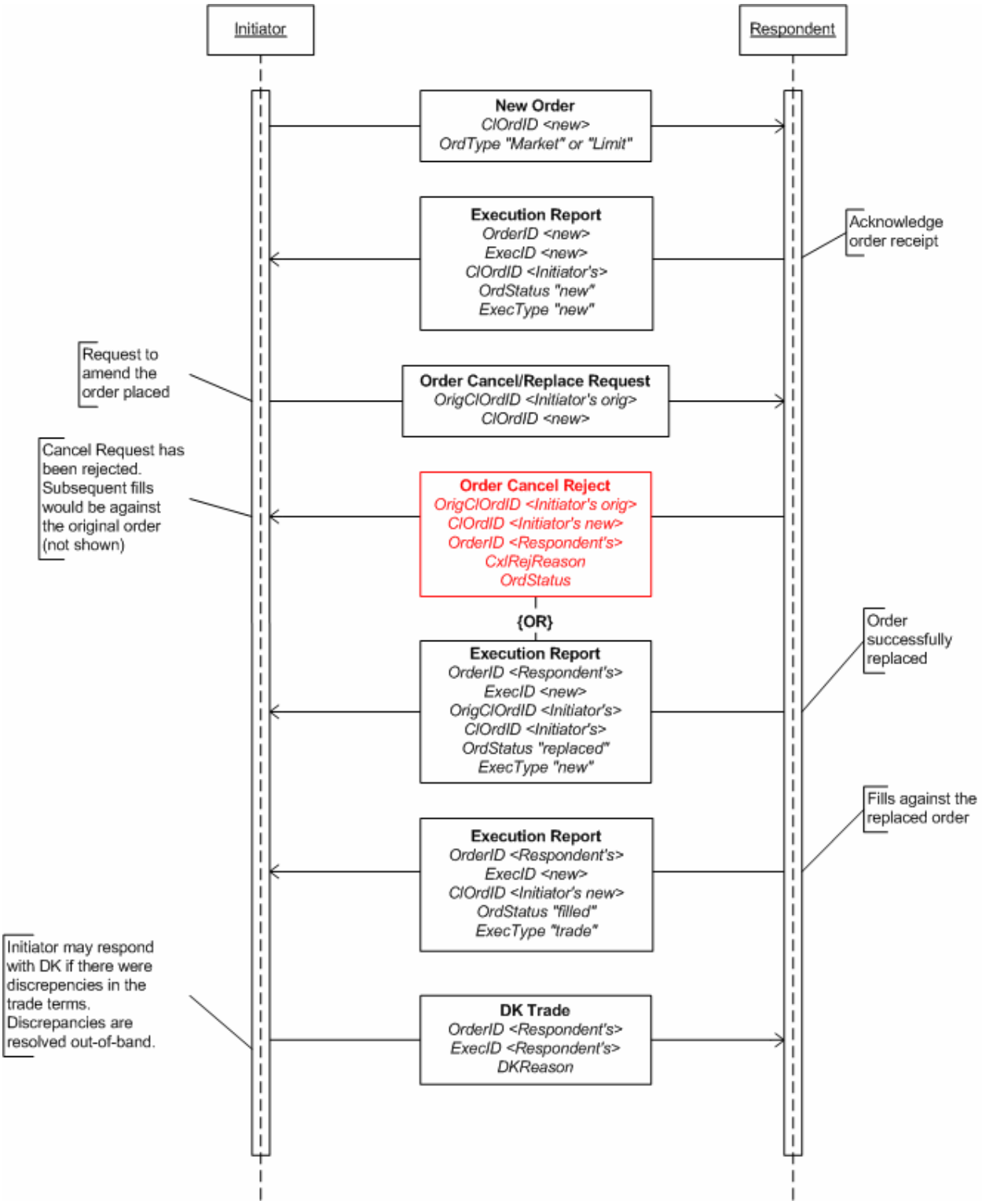
© Copyright, 2007-2008, FIX Protocol, Limited

Figure 4: Order, Execution, Cancel Request Flow



© Copyright, 2007-2008, FIX Protocol, Limited

Figure 5: Order, Execution, Cancel/Replace Flow



FX Settlement Obligation

FX OTC Settlement requires that settlement information be reported to counterparties such that they are aware of their obligations at both an individual trade level as well as an aggregated level. Trade level settlement information includes the receiving accounts of both the central counterparty as well as the firm counterparty. Settlement reporting for trades is considered as being reported at a gross level.

Settlement reporting can also be aggregated or netted across a series of trades for a given currency pair and value date. The Settlement Obligation Report is used to inform counterparties of their settlement obligation and provides information on where each party is to receive payment. Settlement account information may include the bank at which each party holds a specific currency as well as CLS bank information. A Settlement Obligation Report can be sent at various points throughout the settlement day with a status of preliminary or final.

Central Counterparty Workflow

In a central counterparty workflow the Settlement Obligation Report will be sent out from the central counterparty responsible for guaranteeing the FX OTC deal to each of the counterparties involved in the deal. Reporting may take place intra-day or end-of-day with the reports being flagged accordingly. Settlement Obligation Reports can report settlement as either net or gross. Settlement Obligation Reports will be reported by value date.

FX OTC trades with settlement information will be sent on a real time basis as the trade is executed, received, and posted by the central counterparty. Trades carrying settlement information are considered to be specifying gross settlement. FX OTC trades will be reported as they occur – sometimes significantly in advance of the value date.

Individual FX OTC trades are aggregated by value date into a Settlement Obligation Report which can be sent to parties involved in the deal on an intra-day or end-of-day basis.

Instrument: EUR/USD

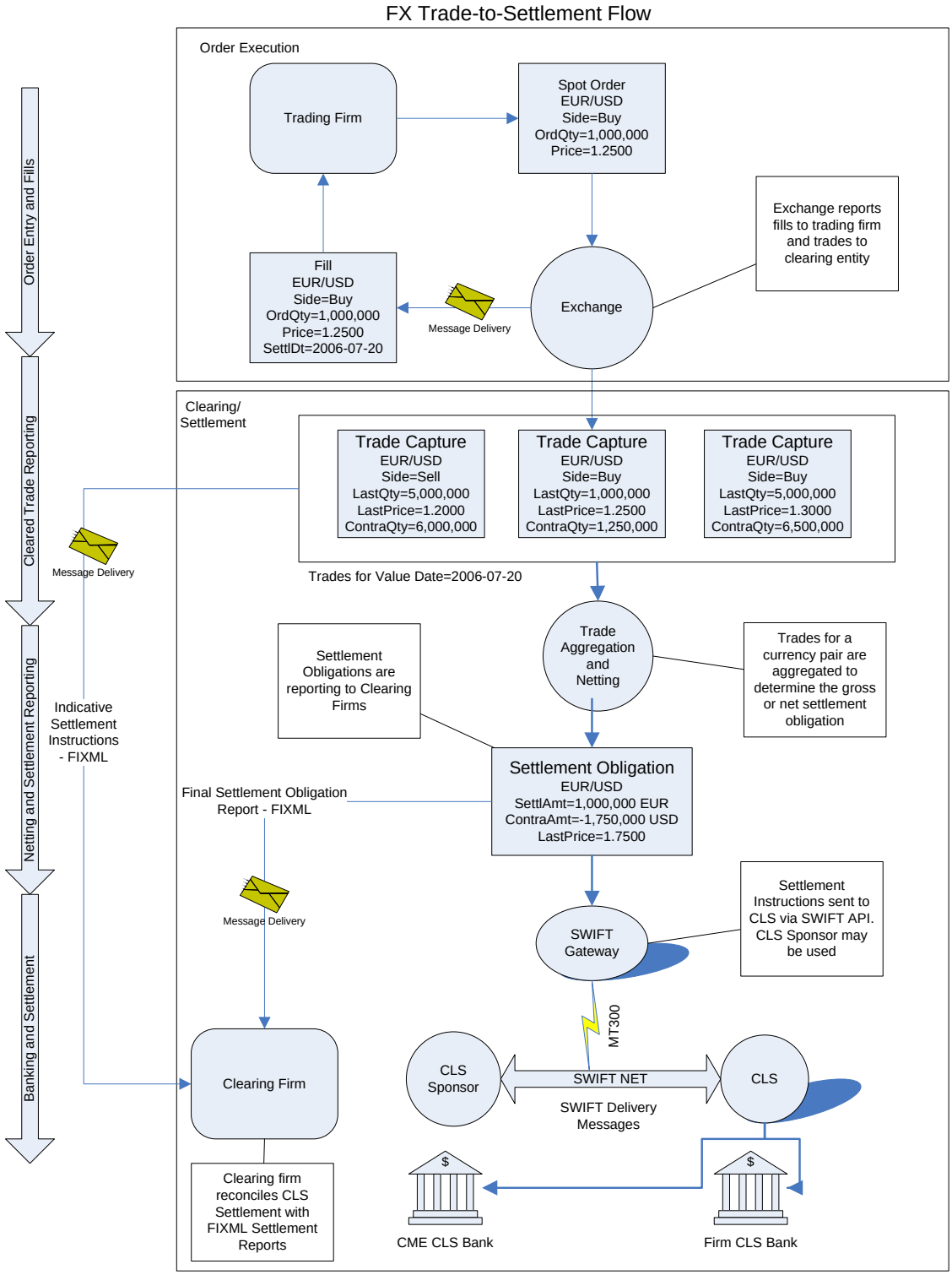
Trade1: Buy 1M EUR @ 1.25 → Sell 1.25M USD

Trade2: Sell 1M EUR @ 1.35 → Buy 1.35M USD

Aggregated: Receive .1M USD, Deliver 0 EUR

© Copyright, 2007-2008, FIX Protocol, Limited

Figure 6: FX Trade-to-Settlement flow



Usage Notes

This section discusses the detailed usage of specific fields within the messages and some specific FX asset type usages as well. These usage notes were the compilation of discussions that occured within the GFXC's Technical sub-committee during its review of the FIX protocol. These notes should be considered the starting point for implementation FIX for FX.

General Usage Notes

This section lists usage notes that are common requirements among the messages used for FX trading.

1. FX symbology is defined in the Electronic Broking Services, Ltd. (see http://www.ebs.com) format of "CCY1/CCY2", where CCY1 and CCY2 are ISO currency codes. This is read as "currency 2 per currency 1". FX symbology is carried in the Symbol (55) field.

2. Currency (15) field denotes the dealt currency. This field is mandatory for FX trading interactions in all messages types.

3. SettlType (63) was enhanced as of version FIX 5.0 to allow proper expressions of standard tenors. It should be noted that for FX tenors expressed using Dx, Mx, Wx, and Yx values do not denote business days, but calendar days. Usage of SettlType values are as follows:

   0 = Regular / FX Spot settlement (T+1 or T+2 depending on currency). "Regular" is defined as the default Spot settlement period for the dealt currency.

   1 = Cash / TOD (T+0)

   2 = Next Day (T+1) / TOM (T+1)

   B = Broken date - for FX expressing non-standard tenor, SettlDate (64) must be specified

   C = FX Spot Next settlement (Spot+1, aka "next day")

   Dx = FX tenor expression for "days", e.g. "D5" for 5-days, where "x" is any integer > 0

   Mx = FX tenor expression for "months", e.g. "M3" for 3-months, where "x" is any integer > 0

   Wx = FX tenor expression for "weeks", e.g. "W13" for 13-weeks, where "x" is any integer > 0

   Yx = FX tenor expression for "years", e.g. "Y1" for 1-year, where "x" is any integer > 0

4. "Settlement currency" for FX trading is defined as the "counter currency" of the transaction. SettlCurrency (120) is optional except in the cases where the transaction is to be settled in a third currency that is different from the currencies identified in the pair. When it is not sent, by default it means the opposite currency in the pair from the dealt currency, as identified in Currency (15) field.

   For non-NDF deals (FX swaps, spot and forward) the term "settlement currency" can only mean one thing: the currency that is on the opposite from the dealt currency (expressed in FIX using Currency (15) field). For example: Symbol is EUR/USD, and the dealt is EUR then SettlCurrency is USD.

   For NDF deals the term "settlement currency" could be either the dealt currency or the "counter currency" or a third currency. For example: In a USD/KRW NDF deal where the dealt currency is KRW, the settlement currency is USD, if the dealt currency is USD then the settlement currency can also be USD. In a GBP/KRW NDF deal where the deal typically settles in a third currency, USD in this case, then the settlement currency is USD. (NDFs will be discussed in detail in the Phase 2 gap analysis).

   For FX OTC Spot Options, the settlement currency can refer to either the counter currency or the currency of the option premium (or premia). However, for the purposes of FIX usage, it will refer to the currency of the option premium.

5. CFI Code (ISO 10962) is encouraged as a means to differentiate between the different FX asset types. The following are CFI Codes for the FX asset types currently supported in FIX (based on ISO 10962:2006)

© Copyright, 2007-2008, FIX Protocol, Limited

FX Spot: RCSXXX (was MRCSXX)
FX Forward: FFCPNO
FX Swap: FFCPNW
NDF: FFCNNO

6. Either SettlType (63) or SettleDate (64) is required in Initiator sent messages (e.g. Quote Request, Market Data Request, and Order messages) except as specified in the section "SettlDate and SettlType Required Usage Exception" below.

7. The following fields' value are to be expressed in decimal form. For example, 61.99 points is expressed and sent as 0.006199.
   BidSwapPoints
   OfferSwapPoints
   LegBidForwardPoints
   LegOfferForwardPoints
   SwapPoints
   LastSwapPoints
   LegLastForwardPoints
   MDForwardPoints
   LastForwardPoints
   MDEntryForwardPoints

Quote Request

The Quote Request message is sent by the Initiator to request a "one-off" quote for a specific currency pair with specific tenor or settlement date (a.k.a. value date). The type of request, QuoteRequestType, should also be specified to indicate whether the request is for an indicative quote or a tradeable/executable quote. For the most part FX requests would be executable.

Note that the Quote Request does allow the Initiator to send, in a single request, multiple currency pairs to the Respondent. When multiple currency pairs are requested, the Respondent will send multiple Quote messages in response, this is because the Quote message can only provide a quote for a single currency pair. However, it should be noted that the Quote messages for the different currency pairs would reference the same quote request identifier (QuoteReqID).

For FX the Quote Request message may not be used to send a "can you meet this rate" type of request, therefore order-related fields in the Quote Request message such as ClOrdID (11), Price (44), and Price2 (640) are not used in FX.

ExpireTime Usage

When requesting a short-lived Quote Request the ExpireTime must be specified. The ExpireTime is set by the Initiator to indicate when the request expires and quotes corresponding to the request will not be accepted and should not be sent after that time. Updates to the quotes may be sent within that ExpireTime period.

A Quote Request message that does not contain an ExpireTime will result in one and only one Quote message from the Respondent (if the Respondent chooses to respond with a rate). There will be no updates to this quote. This type of request may be viewed as "one-offs".

Field Usage Notes

1. Either the SettlType (63) or SettlDate (64) must be specified in the Quote Request to specify the tenor or value date (respectively). See "SettlDate and SettlType Required Usage Exception" section below on exceptions to this requirement for certain groups of FIX users.

2. QuoteType (537) values applicable for FX are
   0 - Indicative quote
   1 - Tradeable quote

© Copyright, 2007-2008, FIX Protocol, Limited

Absence of this field implies a request for an indicative quote.

3. Side (54) indicates from the Initiator's perspective whether the request is for a buy or a sell. Absence of this field indicates a request for a two-sided quote. For FX Swaps, if requesting a 1-sided quote, the value "B" (as defined) should be used - the side for each leg (LegSide) would be defined in NoLegs repeating group.
4. OrderQty (38) is required for FX. Specified the exact amount of the dealt currency to be transacted if the rate is acceptable.
5. Currency (15) is required and specifies the dealt currency of OrderQty (38). For FX Swaps (using NoLegs repeating group) this denotes the dealt currency of the swap.
6. ExpireTime (126) is required for short-lived requests. See "ExpireTime" usage above.
7. The minimum required fields in a request are:
   - FX Spot: the currency pair (Symbol), side (Side), amount (OrderQty), settle date (SettlDate) or tenor (SettlType), dealt currency (Currency)
   - FX Forward: the currency pair (Symbol), side (Side), amount (OrderQty), settle date (SettlDate) or tenor (SettlType), dealt currency (Currency)
   - FX Swap: the currency pair (Symbol), side (Side), near and far amounts (LegQty), near and far settle date (LegSettlDate) or near and far tenor (LegSettlType), dealt currency (Currency)
8. NoLegs repeating group is used to define an FX Swap.

Quote Response

The QuoteResponse message can be used by the Initiator in a "one-off" quoting flow to explicitly tell the Respondent that the Initiator is "passing" on the quote, has "done away" or if the Quote was received after the Quote Request's ExpireTime ("expired").

1. QuoteID (117) is required for FX when responding to a Quote. This is the Respondent's QuoteID.
2. Current values to use for FX in the QuoteRespType (694) field are:
   3 - Expired
   5 - Done away
   6 - Pass
3. Fields from the Quote message should be echoed back in the Quote Response.

Quote

The Quote message is used by the Respondent to respond to a Quote Request message. If the Quote Request contains multiple currency pairs in the request, the Respondent will send a quote message for each currency pair. Each of these quote messages will have its own unique QuoteID while referencing the same QuoteReqID supplied in the Quote Request message.

The Quote may be updated by the Respondent as long as the original Quote has not expired. The initial Quote in response to a Quote Request would reference the QuoteReqID along with the time which is it valid until. The update will be implied by reference to the same QuoteRequestID. The Respondent may, at some time before the Quote expires, update the Quote by sending a Quote with a new QuoteID, referencing the same QuoteReqID. The ValidUntilTime may be a new time or the same time as the replaced Quote. At any one time there can only be one live quote for a QuoteReqID for a given currency pair (there may be multiple currency pairs associated with the same QuoteReqID since the request may contain more than one currency pair). The ValidUntilTime of the Quote cannot be later than the ExpireTime specified in the QuoteRequest.

Definition of ValidUntilTime in FX Quote

© Copyright, 2007-2008, FIX Protocol, Limited

ValidUntilTime is defined as the time that the quote expires and as the time value that cannot extend past the ExpiryTime on the QuoteRequest. The committee also agreed that as a matter of policy there should be only one live quote for a QuoteRequestID for a given currency pair from a given quote provider at any one time. It would be up to implementations to decide whether to honor a hit on a quote that has expired.

Field Usage Notes

1. OrderQty is required for "tradeable" quote and optional for "indicative". For FX Swap, OrderQty is not required, even when QuoteType = tradeable, as the amounts are indicated in LegQty.
2. SettlType and SettlDate are required in Quote message, except as specified in the "SettlDate and SettlType Required Usage Exception" section below.
3. LegRefID is required for FX Swap.
4. BidPx and OfferPx expresses the "all-in" rate. For single-sided quote, either BidPx or OfferPx is required. For two-sided quote, both BidPx and OfferPx is required. For FX Swaps these are not required.
5. MinBidSize can be used to specify the minimum or floor amount to qualify for the FX rate specified in BidPx.
6. BidSize always represents that maximum or ceiling or "up to" amount for the FX rate specified in BidPx.
7. MinOfferSize can be used to specify the minimum or floor amount to qualify for the FX rate specified in OfferPx.
8. OfferSize always represents the maximum or ceiling or "up to" amount for the FX rate specified in OfferPx.
9. ValidUntilTime is required for FX. See usage noted in "Definition of ValidUntilTime in FX Quote" section above.
10. BidSpotRate can be used to specify the bid spot rate. For forward bid quotes, if BidPx is specified, either BidSpotRate or BidForwardPoints should be specified.
11. OfferSpotRate can be used to specify the offer spot rate. For forward offer quotes, if OfferPx is specified, either OfferSpotRate or OfferForwardPoints should be specified.
12. BidForwardPoints is the bid forward points added to BidSpotRate. This may be a negative value. For forward bid quotes, if BidPx is specified, either BidSpotRate or BidForwardPoints should be specified.
13. OfferForwardPoints is the offer forward points added to OfferSpotRate. This may be negative value. For forward offer quotes, if OfferPx is specified, either OfferSpotRate or OfferForwardPoints should be specified.
14. BidSwapPoints and OfferSwapPoints are the swap points of an FX Swap quote.

Quote Request Reject

The Quote Request Reject is used by the Respondent to reject only the Quote Request message from the Initiator. A reject reason must be supplied. At a minimum the required fields of this message type are required in FX. The Respondent may choose to provide more information by "echoing back" the data from the message that is being rejected.

Quote Cancel

The Quote Cancel message is used by the Respondent to cancel a previously sent Quote message that is still "live" (i.e. not expired). Once a Quote has been canceled the Quote Request that initiated the chain would also be considered "dead", in other words no further quotes will be provided against that request. At any given time,

© Copyright, 2007-2008, FIX Protocol, Limited

there should only be one "live" quote for the corresponding Quote Request for the specific currency pair (Quote Request may contain more than one currency pair).

Field Usage Notes

1. QuoteID is conditionally required when QuoteCancelType is "5" (cancel quote specified in QuoteID)
2. Symbol is conditionally required when QuoteCancelType is "1" (cancel for symbol) or "5" (cancel quote specified)

Market Data Request

The Market Data Request message is used by the Initiator to initiate a streaming price feed. The MDReqID would be the stream ID. If the Initiator wishes to co-mingle different currency pairs into a single stream the Initiator should expect to receive multiple Market Data Snapshot messages as the initial response, one Market Data Snapshot message for each currency pair requested. The updates would be provided in a single Market Data Incremental Refresh message.

Field Usage Notes

1. SettleType and SettlDate fields are optional in a Market Data Request message. If a request is sent without either the SettlType or SettlDate specified, the Initiator could receive a significance amount of information. The Respondent may respond with a price stream of all available tenors for the currency pair.
2. MarketDepth is used by the Initiator to request depth of book or "vector prices" by specifying "full book" or best bid/offer by specifying "top of book". However, if the Respondent does not support the type of request the Respondent should reject the request via the Market Data Request Reject message.
3. AggregatedBook field is optionally used by the Initiator to request that only the aggregated entries be sent or not. Again, if the Respondent does not support the type of request then a Market Data Request Reject message should be sent.

   The combination of MarketDepth and AggregatedBook in the request would result in different content in the response.
4. MDEntryType, for FX, only the values "0" (Bid) or "1" (Offer) would be used.
5. MDQuoteType is used to specify whether the request is for indicative or tradeable, or both, streaming prices. Absence of this field provides the Respondent with the option whether to provide indicative and/or tradeable prices.
6. MDEntrySize is optionally used by the Initiator to specify a ceiling or "up to" quantity. The Respondent is free to provide prices for amounts up to the quantity specified by the Initiator in this field. If MDEntrySize is not specified then the market data response may contain prices for all quantity levels available for the requested currency pair.

Market Data Snapshot/Full Refresh

This message type is used by the Respondent to provide the initial or starting snapshot of a price stream for the currency pair requested. If the request contained multiple currency pairs then each pair will receive its own Market Data Snapshot to start, however, it must be noted that each Market Data Snapshot message will have the same MDReqID (the stream ID) but with price data for different currency pairs.

Field Usage Notes

1. MDReqID is required when responding to a Market Data Request message
2. MDEntryType for FX streaming prices is either "0" for Bid or "1" for Offer

© Copyright, 2007-2008, FIX Protocol, Limited

3. MDEntryID is required and is a unique reference assigned by the Respondent for this instance of the market data entry.
4. MDEntryPx is required for FX. This specifies the "all in" or "outright" rate (spot rate + foward points).
5. MDEntrySize specifies the amount being for the bid/offer. This provides an "up to" or ceiling amount for the quoted rate.
6. ExpireTime in this message allows the Respondent to specify when the price will expire
7. MinQty is optionally used by the Respondent to specify the minimum quantity of an order to qualify for the rate quoted
8. QuoteEntryID is required and is a unique quote entry identifier as assigned by the Respondent.
9. MDQuoteType indicates whether the price is indicative or executable. Abscence of this field indicates the price is indicative.
10. MDEntrySpotRate is used for specifying the spot rate. It is recommended that either spot rate or forward points be specified for FX forwards.
11. MDEntryFowardPoints is used for specifying the forward points. This may be a negative value. It is recommended that either spot rate or forward points be specified for FX forwards.
12. SettlDate and SettlType is required.
13. The Parties component block is optionally used by multi-bank portals to identify the banks that are providing the rate information. PartyRole is required and in this case the role should be set to "executing broker".

Market Data Incremental Refresh

The Market Data Increment Refresh message is used to send price updates to the Initiator once a snapshot has been sent. For the most part the MDUpdateAction would be either "change" or "delete", however, a new price quote can be supplied in this message with an MDUpdateAction of "new". In this latter scenario, MDEntryType and a unique MDEntryID must be specified.

MDEntryID and QuoteEntryID Usage in Refreshes

The Market Data Incremental Refresh message is used when there is a change to the data of a previously sent MDEntryID or a deletion of a previously sent entry. MDEntryIDs must be unique for the day and each live entry must have an MDEntryID.

In an MDUpdateAction of "delete" the ID of the entry being deleted must be specified in MDEntryID. This signifies to the recipient that this is the entry to be removed.

In an MDUpdateAction of "change" there are two methods that can be used to refer to the entry being changed:

1. Refer to the MDEntryID being changed in the MDEntryID field itself. This also means that the ID will not change and remains a "live" ID.
2. Refer to the MDEntryID of the entry being changed in the MDEntryRefID and provide a new unique MDEntryID. This new MDEntryID would be the "live" ID. This would be the preferred method for firms that wish to maintain an audit trail of the changes to their pricing feed.

For changes/updates the Market Data Incremental Refresh message would contain only the MDEntryID of the entry being changed and only the data elements that are being changed.

For delete, the Respondent may send just the ID for the entry to be deleted in MDEntryID. This would minimize bandwidth usage.

© Copyright, 2007-2008, FIX Protocol, Limited

QuoteEntryID is required and on a change/update this would be a new and unique ID as assigned by the Respondent.

Field Usage Notes

1. MDReqID is required when responding to a Market Data Request message
2. MDEntryType for FX streaming prices is either "0" for Bid or "1" for Offer
3. MDEntryID is required and is a unique reference assigned by the Respondent for this instance of the market data entry.
4. MDEntryPx is required for FX. This specifies the "all in" or "outright" rate (spot rate + foward points).
5. MDEntrySize specifies the amount being for the bid/offer. This provides an "up to" or ceiling amount for the quoted rate.
6. ExpireTime in this message allows the Respondent to specify when the price will expire
7. MinQty is optionally used by the Respondent to specify the minimum quantity in an order to qualify for the rate quoted
8. MDQuoteType indicates whether the price is indicative or executable. Abscence of this field indicates the price is indicative.
9. MDEntrySpotRate is used for specifying the spot rate. It is recommended that either spot rate or forward points be specified for FX forwards.
10. MDEntryFowardPoints is used for specifying the forward points. This may be a negative value. It is recommended that either spot rate or forward points be specified for FX forwards.
11. SettlDate and SettlType is required.
12. The Parties component block is optionally used by multi-bank portals to identify the dealers that are providing the rate information. PartyRole is required and in this case the role should be set to "executing broker".

Market Data Request Reject

The Market Data Request Reject message is used by the Respondent to reject the request from the Initiator. A reject reason must be supplied.

Field Usage Notes

1. MDReqRejReason is required.

New Order - Single

The New Order - Single message is used by the Initiator to place an order with the dealer. When an order is initiated as a result of a request for quote or streaming price, the QuoteID in the New Order - Single message is used to reference the price quote in both cases. The QuoteID would contain the dealer provided QuoteID from the Quote message or the QuoteEntryID from the MarketDataSnapshot or MarketDataIncrement messages. The OrdType (40) would specify "previously quoted".

Field Usage Notes

1. Either SettlType or SettlDate must be specified, except as specified in the section SettlDate and SettlType Required Usage Exception" below. If order is a result of a quote or streaming price quote these values should be the same as in quote or streaming price quote.

© Copyright, 2007-2008, FIX Protocol, Limited

2. OrderQty must match the amount in the Quote message if the order is a result of a quote. If the order is against a streamed price the OrderQty of the order can be less than or equal to the quantity shown in the streamed price quote (MDEntrySize).
3. Price and StopPx, used when placing a "limit" or "stop/stop loss" order respectively, would contain the "all-in" rate.
4. QuoteID is conditionally required when the order is in response to a Quote or Market Data message. Contains the QuoteID from the Quote message or the QuoteEntryID from the market data message.
5. SettlCurrency is used only to denote a third currency to be used for settlement (i.e. not one of the currencies in the currency pair specified in Symbol). See also description in General Usage Notes.
6. Parties component block is conditionally required when an order is sent via a multi-bank portal. It is used to identify the executing broker.

New Order - Multileg

The New Order - Multileg message is used by the Initiator to place a multilegged order with the dealer - typically an FX Swap.

Field Usage Notes

1. For FX Swaps the Side field would carry the value "B" (as defined"). The LegSide will identify which is the buy leg and which is the sell leg.
2. Symbol would specify the currenty pair in the swap.
3. SwapPoints is optionally used to express the differential between the far leg and the near leg.
4. NoLegs, required, for FX Swaps there would only be 2 legs.
5. LegSymbol, required, is the currency pair in the swap, same as Symbol
6. LegCurrency, required, is the dealt currency of the leg and denotes the currency denomination of LegQty.
7. LegSide, required, denotes the side of the leg
8. LegQty, required, the amount of this leg denomincated in the currency specified in LegCurrency.
9. LegRefID is required. If the order is a result of a Quote message the value needs to match the LegRefID of the quote message. If the order is a result of an "out of band" quote, the Initiator is required to assign a unique identifier for each leg.
10. LegPrice is conditionally required when OrdType is "previously quoted". This is the "all in" rate for this leg as specified in the quote.
11. Either LegSettlType or LegSettlDate should be specified. If the order is a result of a quote this should be the same as in quote.
12. OrdType, if the FX Swap order is a result of a quote the OrdType = D (previously quoted). OrdType G = Forex-swap may also be used.
13. QuoteID is required for FX Swap when the order is a result of a Quote message. Contains the QuoteID from the Quote message.
14. SettlCurrency is used only to denote a third currency to be used for settlement (i.e. not one of the currencies in the currency pair specified in Symbol). See also description in General Usage Notes.
15. Parties component block is conditionally required when an order is sent via a multi-bank portal. It is used to identify the executing broker.

© Copyright, 2007-2008, FIX Protocol, Limited

Execution Report

The Execution Report message is used by the Respondent to respond to an order (New Order - Single and New Order - Multileg). The Execution Report message has several "modes" and provides information on the status of the order.

When reporting an execution or trade (partial or full fill) the LastPx (31) field is used to specify the "all in" rate for the partial or full fill. It is considered best practice that the spot rate and forward points used to arrive at the "all in" rate be specified in LastSpotRate (194) and LastForwardPoints (195) when appropriate. For example, if the fill is for a forward then both the LastSpotRate and LastForwardPoints should be specified.

Field Usage Notes

1. CalculatedCcyLastQty (1056) is used to express the contra amount or "contra order quantity" that was executed. This is the quantity of the other side of the currency pair (from the dealt currency as expressed in Currency (15)) and can be derived from LastQty (32) and LastPx (31).
2. LastQty expresses the quantity of the traded currency, as specified by Currency (150). For FX Future if LastQty is expressed in terms of contracts ContractMultiplier (231) is conditionally required.
3. Parties component block is conditionally required when an execution is sent by a bank via a multi-bank portal. It is used to identify the executing broker.
4. LefRegID is required when using a single Execution Report message to report on both legs of an FX Swap.
5. SettlType is optional but should be specified for spot and outright FX forward trades. For FX Swaps the LegSettlType should be used instead.
6. SettlDate is required for FX spot and forward. Banks/dealers must specify the value date for spot and outright FX forward trades. For FX Swap trades refer to the LegSettlDate.
7. OrderQty is required for FX spot and outright forward trades. For FX Swaps it is not required. See LegQty.
8. OrdType is required if specified on the order.
9. LastSwapPoints is optionally used when ExecType = Trade or Trade Correct and it is a FX Swap trade. used to express the swap points for the swap trade event.
10. LastPx is the "all in" price of the trade. Conditionally required when ExecType = Trade or Trade Correct and the trade is for FX spot and forwards. Not required for FX Swap even when ExecType = Trade or TradeCorrect as there is no "all in" rate that applies to both legs of a FX Swap.
11. For FX forward trades, either LastSpotRate or LastForwardPoints should be specified. These would be the spot rate or forward points used in the "all in" price for the fill.
12. AvgPx is the "all in" price
13. TradeDate and TransactTime are required
14. GrossTradeAmt can be used for FX Future to express the notional value of a trade when LastQty and other quantity fields are expressed in terms of number of contracts - in which case ContractMultiplier (231) is conditionally required.
15. ContractMultiplier (231) is conditionally required when quantities are expressed in terms of number of contracts for FX Futures.
16. For FX Swaps, LegSymbol, required, is the currency pair in the swap, same as Symbol
17. For FX Swaps, LegCurrency, required, is the dealt currency of the leg and denotes the currency denomination of LegQty.
18. For FX Swaps, LegSide, required, denotes the side of the leg

© Copyright, 2007-2008, FIX Protocol, Limited

19. For FX Swaps, LegQty, required, the amount of this leg denomincated in the currency specified in LegCurrency.
20. For FX Swaps, LegSettlType is optional
21. For FX Swaps, LegSettlDate is required. Expresses the value date on thsi leg of the swap.

FX OTC Spot Option

Broker dealers and futures commission merchants (FCMs) requested the ability to report trades and positions for vanilla FX OTC Spot Options to support multiple asset middle and back office processing.

The following table specifies which fields should be specified to identify an FX OTC Spot Options contract.

<table>
<tr><th>FIX Tag #</th><th>FieldName</th><th>Usage</th><th>Expected Value</th></tr>
<tr><td>55</td><td>Symbol</td><td>Symbol (ISO3) - Ccy1/Ccy2 (ISO3) – following market conventions</td><td></td></tr>
<tr><td>541</td><td>MaturityDate</td><td>Option maturity date – which is the date at which the underlying spot price is used to determine value of option at expiration. The exact fixing time is specified in MaturityTime.</td><td></td></tr>
<tr><td>1079</td><td>MaturityTime</td><td>Time and Timezone of the price fixing for the options expiration</td><td></td></tr>
<tr><td>461</td><td>CFI Code
<table>
<tr><th>Position</th><th>Attribute name</th><th>Usage</th><th>Values</th></tr>
<tr><td>1</td><td>Asset Class</td><td>Option</td><td>“O”</td></tr>
<tr><td>2</td><td>Put or Call</td><td>Put or Call value</td><td>“C” or “P”</td></tr>
<tr><td>3</td><td>Underlying Asset Class</td><td>“C” – Currency</td><td>“C”</td></tr>
<tr><td>4</td><td>Delivery Style</td><td>Use “P” for physical delivery is full amount of currency is being delivered
Use “C” for Cash delivery is settlement is being netted</td><td>“P” or “C”</td></tr>
<tr><td>5</td><td>Product standardization</td><td>“N” – Non-standard (OTC)</td><td>“N”</td></tr>
<tr><td>6</td><td>Exercise Style</td><td>“A” – American, “U”-European, “B” – Bermuda</td><td>“A”, “U”, “B”</td></tr>
</table></td><td></td><td></td></tr>
<tr><td>202</td><td>StrikePrice</td><td>The spot price at which the option will be valued and possibly exercised.</td><td></td></tr>
<tr><td>947</td><td>StrikeCurrency</td><td>Currency the strike price is denominated in</td><td></td></tr>
<tr><td>107</td><td>SecurityDesc</td><td>Optional description of the option contract</td><td></td></tr>
<tr><td>231</td><td>ContractMultiplier</td><td>Specifies the ratio or multiply factor to convert from "nominal" units (e.g. contracts) to total units (e.g. shares) (e.g. 1.0, 100, 1000, etc). For FX options post trade – recommend that contract multiplier be set to 1 and that all currency amounts be in their full denominated value (as opposed to millions for instance).</td><td>1</td></tr>
</table>

© Copyright, 2007-2008, FIX Protocol, Limited

FIX Tag #	FieldName	Usage	Expected Value
167	SecurityType	Security Type "FOR" – foreign exchange	"FOR"
469	Product	Product class (derived from Bloomberg yellow key) "4" – foreign exchange	"4"
63	SettlType	Not used for FX OTC Spot Option - use explicit dates instead	
64	SettleDate	Settlement date for the option trade premium not the exercise spot transaction	
987	UnderlyingSettlementDate	Settlement date for the spot trade if the option is exercised, usually MaturityDate(tag 541) + 2 business days – following normal FX settlement conventions	
15	Currency	Dealt Currency – either Ccy1 or Ccy2 This is the currency which is being called or put based upon CFICode position to "C"-Call, "P"-Put	
120	SettlCurrency	Settlement Currency – will usually be same as Currency	
54	Side	Transaction Side (1-Buy or 2-Sell)	
32	LastQty	Transaction amount (in measured currency) of the dealt currency (the one that matches the Put or Call flag)	
1056	CalculatedCurrencyLastQty	Quantity of the contra currency, LastQty * LastPx	
31	LastPx	Premium price for the option	

Example:

The following example if for an EUR/USD FX OTC Spot Option 6 month contact settling on a standard date that will deliver EUR if the contract is in the money as of the contract expiration date.

55=EUR/USD	// Symbol - CCY pair
541=20060927	// MaturityDate – fixing date for maturity (expiration) of option contract
1056=16:00:00-5	// MaturityTime – fixing time for maturity of option contract
641= OCCPNU	// CFICode
202=0.8223	// StrikePrice
947=EUR	// StrikeCurrency
107=EUR 6 month call option	// SecurityDescription
231=1	// Contract Multiplier
167=FOR	// SecurityType - foreign exchange
640=4	// Product – Foreign Currency
15=EUR	// Currency - dealt currency
63=M6	// SettlType – 6 month tenor
64=20060331	// SettleDate – for option premium payment (fixing)
120=EUR	// SettleCurrency

© Copyright, 2007-2008, FIX Protocol, Limited

987=20060402	// Underlying SettlementDate – actual pay date for option premium payment
54=1	// Side - Buy
75=20060331	// Trade Date – the date the option trade occurred – may be different from Settlement date
32=1000000.00	// LastQty – quantiy of USD in the event the option is exercised
1056=822300.00	// CalculatedCurrencyLastQty – Quantity of EUR in the event the // option is exercised.
31=30697.63	// LastPx – option price

SettlDate and SettlType Required Usage Exception

The usage of the SettlDate and SettlType (used to express FX tenors) is generally required when implementing FIX for FX. However, for certain types of community of users within FX, these fields are not required. As such, where SettlDate and SettlType is specified as "either SettlDate or SettlType is required" in request type messages (e.g. Quote Request, Market Data Request, New Order messages) or where required in response messages (e.g. Market Data Increment Refresh, Market Data Snapshot Full Refresh, Execution Report, Quote) it would not be required in implementations for the exception user group. The exception user group is the retail user community and trading platforms utilized by this community.

In the retail FX trading space, the customers would open accounts by depositing cash that acts as margin for their trading activities. These deposits are held in one of a few possible account currencies. As the customers trades, instead of generating cash balances in the currencies traded, they generate profit and loss in their account only. The P/L is realized when the "position" is closed (for example, Sell 1M EUR/USD and then Buy back 1M EUR/USD). In retail FX trading, positions are not settled, instead the positions are continuously rolled over on a daily basis, debiting or crediting the customer's account with interest in the account currency. Positions remain open until the reverse transaction is made by the customer to close the position.

For example a customer has a USD-based account and an initial cash deposit of $100,000 was made into the account.

- At 11:00 EST the customer Sells 1M EUR/USD at 1.2300, therefore establishing an open position
- At 17:00 EST the EUR/USD rate is 1.2200
- At 17:00 EST, the customer's account would be credited with $70 as an interest payment for being short EUR/USD. The account balance is now $100,070
- The account has generated a "floating P/L" in the account currency, USD, since the position has not been "closed" by the customer.
- The position is still open (the customer still has not bought back the 1M EUR/USD to close the position), so a distinction between Balance and Equity is made where Account Equity = Balance + Floating P/L. In this example, the customer's Account Equity would be $110,070
- The customer's Account Equity will change in real-time as the EUR/USD rate changes. If the customer decides at 17:15 EST to "close" the position at the EUR rate of 1.2205, then the customer's Account Balance would be $109,570.

As can been seen by this FX retail trading example, that there is no settlement or cash delivery. If the customer had decided not the buy back the 1M EUR/USD the position would remain open indefinitely. The position would roll over daily, charging/crediting the appropriate interest payment daily.

Message Samples

© Copyright, 2007-2008, FIX Protocol, Limited

These sample FIX message usage servers only to illustrate usgae of key fields in the different message types in the context of FX. Data used are fictional. Only relevent fields from the header and message body are shown (i.e. some message header and trailer fields are not shown).

Quote Request for FX Swap using NoLegs repeating group

The sample Quote Request messages shows examples for Spot/Forward (1M) and Forward/Forward (1M/3M).

Spot/Forward (1M)

35=R	//MsgType - Quote Request message type
49=ABC_AM	//SenderCompID - sending client
56=SSBFX	//TargetcompID - target bank
131=Req123	//QuoteReqID - uniquely assigned by client, format is arbitrary
146=1	//NoRelatedSym
55=EUR/USD	// [1] Symbol
15=EUR	// [1] Currency - dealt currency
555=2	// [1] NoLegs
600=EUR/USD	// [1] LegSymbol
687=1000000	// [1] LegQty - amount of near leg
587=0	// [1] LegSettlType - Regular/Spot ((T+1) or (T+2))
588=20060130	// [1] LegSettlDate - value date of near leg
654=A0001	// [1] LegRefID
600=EUR/USD	// [2] LegSymbol
687=1000000	// [2] LegQty - amount of far leg
587=6	// [2] LegSettlType - future
588=20060228	// [2] LegSettlDate - value date of far leg
654=A0002	// [2] LegRefID
126=20060127-14:35:42	// [1] ExpireTime - time the request expires

Forward/Forward (1M/3M)

35=R	//MsgType - Quote Request message type
49=ABC_AM	//SenderCompID - sending client
56=SSBFX	//TargetcompID - target bank
131=Req456	//QuoteReqID - uniquely assigned by client, format is arbitrary
146=1	//NoRelatedSym
55=EUR/USD	// [1] Symbol
15=EUR	// [1] Currency - dealt currency
555=2	// [1] NoLegs
600=EUR/USD	// [1] LegSymbol
687=1000000	// [1] LegQty - amount of near leg
587=6	// [1] LegSettlType - future
588=20060228	// [1] LegSettlDate - value date of near leg
654=B0001	// [1] LegRefID
600=EUR/USD	// [2] LegSymbol
687=1000000	// [2] LegQty - amount of far leg
587=6	// [2] LegSettlType - future
588=20060428	// [2] LegSettlDate - value date of far leg
654=B0002	// [2] LegRefID
126=20060127-14:47:23	// [1] ExpireTime - time the request expires

Note that in the examples, Side is not specified, thus the request is for a 2-sided quote.

Quote for FX Swap using NoLegs repeating group

Examples shows a quote for a Spot/Forward (1M) and a Forward/Forward (1M/3M)

© Copyright, 2007-2008, FIX Protocol, Limited

Spot/Forward (1M) quote

35=S	//MsgType - Quote message type
49=ABC_AM	//SenderCompID - sending client
56=SSBFX	//TargetcompID - target bank
131=Req123	//QuoteReqID - uniquely assigned by client, format is arbitrary
117=QT123	//QuoteID
537=1	//QuoteType - Tradeable
55=EUR/USD	//Symbol
15=EUR	//Currency - dealt currency
555=2	//NoLegs
600=EUR/USD	// [1] LegSymbol
687=1000000	// [1] LegQty - amount of near leg
587=0	// [1] LegSettlType - Regular/Spot ((T+1) or (T+2))
588=20060130	// [1] LegSettlDate - value date of near leg
681=1.2214	// [1] LegBixPx - near leg bid rate
684=1.2214	// [1] LegOfferPx - near leg offer rate
654=A0001	// [1] LegRefID
1067 = 0	// [1] LegBidForwardPoints - near leg
1068 = 0	// [1] LegOfferForwardPoints - near leg
600=EUR/USD	// [2] LegSymbol
687=1000000	// [2] LegQty - amount of far leg
587=6	// [2] LegSettlType - future
588=20060228	// [2] LegSettlDate - value date of far leg
681=1.223448	// [1] LegBixPx - far leg bid rate
684=1.223475	// [1] LegOfferPx - far leg offer rate
654=A0002	// [2] LegRefID
1067 = 0.002048	// [1] LegBidForwardPoints - far leg
1068 = 0.002075	// [1] LegOfferForwardPoints - far leg
188=1.2214	//BidSpotRate
190=1.2214	//OfferSpotRate
1065=0.002048	//BidSwapPoints
1066=0.002075	//OfferSwapPoints

Forward/Forward (1M/3M) quote

35=S	//MsgType - Quote message type
49=ABC_AM	//SenderCompID - sending client
56=SSBFX	//TargetcompID - target bank
131=Req456	//QuoteReqID - uniquely assigned by client, format is arbitrary
117=QT456	//QuoteID
537=1	//QuoteType - Tradeable
55=EUR/USD	//Symbol
15=EUR	//Currency - dealt currency
555=2	//NoLegs
600=EUR/USD	// [1] LegSymbol
687=1000000	// [1] LegQty - amount of near leg
587=6	// [1] LegSettlType - future
588=20060228	// [1] LegSettlDate - value date of near leg
681=1.223448	// [1] LegBixPx - near leg bid rate
684=1.223475	// [1] LegOfferPx - near leg offer rate
654=B0001	// [1] LegRefID
1067 = 0.002048	// [1] LegBidForwardPoints - near leg
1068 = 0.002075	// [1] LegOfferForwardPoints - near leg
600=EUR/USD	// [2] LegSymbol
687=1000000	// [2] LegQty - amount of far leg

© Copyright, 2007-2008, FIX Protocol, Limited

587=6	// [2] LegSettlType - future
588=20060428	// [2] LegSettlDate - value date of far leg
681=1.227599	// [1] LegBixPx - far leg bid rate
684=1.227641	// [1] LegOfferPx - far leg offer rate
654=B0002	// [2] LegRefID
1067 = 0.006199	// [1] LegBidForwardPoints - far leg
1068 = 0.006241	// [1] LegOfferForwardPoints - far leg
188=1.2214	//BidSpotRate
190=1.2214	//OfferSpotRate
1065 = 0.004124	//BidSwapPoints
1066 = 0.004193	//OfferSwapPoints

Single Bank Market Data Request

The sample set of market data messages below illustrates a price stream request directly from a bank. It also illustrates the responses back from the bank.

Market Data Request: client requests best bid/offer tradeable prices for a 1-month forwards of a currency pair with a specified quantity in this example. The target bank is implicit in that it is assumed there is a direct FIX session between the client and the bank.

Market Data response messages: bank may provide prices at different quantities if client did not request a specific quantity. In this example, a quantity was specified and this is the ceiling amount. Prices may be indicative or tradeable, in this example the request was for tradeable only.

Market Data Request

35=V	//MsgType - Market Data Request message type
49=ABC_AM	//SenderCompID - sending client
56=SSBFX	//TargetcompID - target bank
262=20050922.09:30:59.1	//MDReqID - uniquely assigned by client, format is arbitrary
263=1	//SubscriptionRequestType - snapshot+update
264=1	//MarketDepth - top of book
267=2	//NoMDEntryTypes
269=0	// [1] MDEntryType - bid
269=1	// [2] MDEntryType - offer
146=1	//NoRelatedSymbol - number of CCY pairs
55=EUR/USD	// [1] Symbol - CCY pair
167=FOR	// [1] SecurityType - foreign exchange
15=USD	// [1] Currency - dealt currency
537=1	// [1] QuoteType - tradeable
63=M1	// [1] SettlType - 1-month tenor
38=20000000	// [1] OrderQty - 20 million

Market Data Snapshot

35=W	//MsgType - Market Data Snapshot message type
49=SSBFX	//SenderCompID - sending bank
56=ABC_AM	//TargetCompID - target client
262=20050922.09:30:59.1	//MDReqID - uniquely assigned by client, format is arbitrary
55=EUR/USD	//Symbol - CCY pair
167=FOR	//SecurityType - foreign exchange
268=2	//NoMDEntries - number of MD entries
269=0	// [1] MDEntryType - Bid
278=EED02091-47AD-4EDD-A0AA-	// [1] MDEntryID - unique entry identifier assigned by the bank.

© Copyright, 2007-2008, FIX Protocol, Limited

0B2D9D1B9B0F	Format and scheme is arbitrary
270=1.2141	// [1] MDEntryPx - all-in bid price/rate
15=USD	// [1] Currency - dealt currency
271=20000000	// [1] MDEntrySize - amount
299=FFA23081-51ED-78CE-B9AF-8F3D4B89D012	// [1] QuoteEntryID - unique quote identifier assigned by the bank. format and scheme is arbitrary
63=M1	// [1] SettlType - 1-month tenor
64=20051020	// [1] SettlDate - value date 1 month out
269=1	// [2] MDEntryType - Offer
278=FB5F1910-F110-11d2-BB9E-00C04F795683	// [2] MDEntryID - unique entry identifier assigned by the bank
270=1.2145	// [2] MDEntryPx - all-in offer price/rate
15=USD	// [2] Currency - dealt currency
271=20000000	// [2] MDEntrySize - amount
299=92780B25-18CC-41C8-B9BE-3C9C571A8263	// [2] QuoteEntryID - unique quote identifier assigned by the bank
63=M1	// [2] SettlType - 1-month tenor
64=20051020	// [2] SettlDate - value date 1 month out

Market Data Incremental Refresh - the bank updates the bid side

35=X	//MsgType - Market Data Incremental Refresh message type
49=SSBFX	//SenderCompID - sending bank
56=ABC_AM	//TargetCompID - target client
262=20050922.09:30:59.1	//MDReqID - uniquely assigned by client, format is arbitrary
268=1	//NoMDEntries - number of MD entries
279=1	// [1] MDUpdateAction - change/update
278=<new unique entry ID>	// [1] MDEntryID - new unique entry ID assigned by the bank
280=EED02091-47AD-4EDD-A0AA-0B2D9D1B9B0F	// [1] MDEntryRefID - referencing the entry to be changed/updated
270=1.2139	// [1] MDEntryPx - all-in bid price/rate
299=AAC02189-DF23-11FB-F135-4C0D4A83D238	// [1] QuoteEntryID - unique quote identifier assigned by the bank

"Exchange" Market Data Request

The sample set of market data messages below illustrates a price stream request from a customer to an exchange-style FX platform. It also illustrates the responses back from the exchange.

Market Data Request: client requests "aggregated full book" tradeable Spot prices for a currency pair with no specified quantity in this example.

Market Data response messages: exchange provides aggregated amounts at each bid/ask price points at different quantities. In this example the request was for tradeable only.

Market Data Request

35=V	//MsgType - Market Data Request message type
49=ABC_AM	//SenderCompID - sending client
56=FXEXCHANGE	//TargetcompID - target exchange
262=20050921.09:30:59.1	//MDReqID - uniquely assigned by client, format is arbitrary
263=1	//SubscriptionRequestType - snapshot+update
264=0	//MarketDepth - full book
266=Y	//AggregatedBook - one entry per side per price
267=1	//NoMDEntryTypes

© Copyright, 2007-2008, FIX Protocol, Limited

269=1	// [1] MDEntryType - offer
146=1	//NoRelatedSymbol - number of CCY pairs
55=EUR/USD	// [1] Symbol - CCY pair
167=FOR	// [1] SecurityType - foreign exchange
15=USD	// [1] Currency - dealt currency
537=1	// [1] QuoteType - tradeable
63=0	// [1] SettlType - spot

Market Data Snapshot

35=W	//MsgType - Market Data Snapshot message type
49=FXEXCHANGE	//SenderCompID - sending exchange
56=ABC_AM	//TargetCompID - target client
262=20050921.09:30:59.1	//MDReqID - uniquely assigned by client, format is arbitrary
55=EUR/USD	//Symbol - CCY pair
167=FOR	//SecurityType - foreign exchange
268=3	//NoMDEntries - number of MD entries
269=1	// [1] MDEntryType - Offer
278=EED02091-47AD-4EDD-A0AA-0B2D9D1B9B0F	// [1] MDEntryID - unique entry identifier assigned by the exchange. Format and scheme is arbitrary
270=1.2144	// [1] MDEntryPx - all-in offer price/rate
15=USD	// [1] Currency - dealt currency
271=10000000	// [1] MDEntrySize - amount
299=FFA23081-51ED-78CE-B9AF-8F3D4B89D012	// [1] QuoteEntryID - unique quote identifier assigned by the exchange. format and scheme is arbitrary
63=0	// [1] SettlType - Spot
64=20050923	// [1] SettlDate - value date for spot settle
269=1	// [2] MDEntryType - Offer
278=FB5F1910-F110-11d2-BB9E-00C04F795683	// [2] MDEntryID - unique entry identifier assigned by the exchange
270=1.2145	// [2] MDEntryPx - all-in offer price/rate
15=USD	// [2] Currency - dealt currency
271=50000000	// [2] MDEntrySize - amount
299=92780B25-18CC-41C8-B9BE-3C9C571A8263	// [2] QuoteEntryID - unique quote identifier assigned by the exchange
63=0	// [2] SettlType - Spot
64=20050923	// [2] SettlDate - value date for spot settle
269=1	// [3] MDEntryType - Offer
278=<unique id>	// [3] MDEntryID - unique entry identifier assigned by the exchange
270=1.2146	// [3] MDEntryPx - all-in offer price/rate
15=USD	// [3] Currency - dealt currency
271=15000000	// [3] MDEntrySize - amount
299=<unique id>	// [3] QuoteEntryID - unique quote identifier assigned by the exchange
63=0	// [3] SettlType - Spot
64=20050923	// [3] SettlDate - value date for spot settle

Note that in the above Market Data Snapshot message the exchange assigned the unique MDEntryID and QuoteEntryID

Market Data Incremental Refresh - the exchange updates the one of the offers

© Copyright, 2007-2008, FIX Protocol, Limited

35=X	//MsgType - Market Data Incremental Refresh message type
49=FXEXCHANGE	//SenderCompID - sending bank
56=ABC_AM	//TargetCompID - target client
262=20050921.09:30:59.1	//MDReqID - uniquely assigned by client, format is arbitrary
268=1	//NoMDEntries - number of MD entries
279=1	// [1] MDUpdateAction - change/update
278=<new unique entry ID>	// [1] MDEntryID - new unique entry ID assigned by the bank
280= FB5F1910-F110-11d2-BB9E-00C04F795683	// [1] MDEntryRefID - referencing the entry to be changed/updated
271=40000000	// [1] MDEntrySize - amount dropped
299=AAC02189-DF23-11FB-F135-4C0D4A83D238	// [1] QuoteEntryID - new unique quote identifier assigned by the bank

FX Swap Multi-legged Order

New Order - Multileg examples for Spot/Forward (1M) and Forward/Forward (1M/3M) FX Swap.

Spot/Forward (1M) order

35=AB	//MsgType - New Order Multileg message type
49=ABC_AM	//SenderCompID - sending client
56=SSBFX	//TargetcompID - target bank
11=ORD123	//ClOrdID - uniquely assigned by client, format is arbitrary
117=QT123	//QuoteID - references the quote
54=B	//Side - as defined in NoLegs
55=EUR/USD	//Symbol
15=EUR	//Currency - dealt currency
555=2	//NoLegs
600=EUR/USD	// [1] LegSymbol
556=EUR	// [1] LegCurrency
624=1	// [1]LegSide - Buy
687=1000000	// [1] LegQty - amount of near leg
654=A0001	// [1] LegRefID
566=1.2214	// [1] LegPrice - near leg
587=0	// [1] LegSettlType - Regular/Spot ((T+1) / (T+2))
588=20060130	// [1] LegSettlDate - value date of near leg
600=EUR/USD	// [2] LegSymbol
556=EUR	// [1] LegCurrency
624=2	// [1]LegSide - Sell
687=1000000	// [2] LegQty - amount of far leg
654=A0002	// [2] LegRefID
566=1.223448	// [1] LegPrice - far leg
587=6	// [2] LegSettlType - future
588=20060228	// [2] LegSettlDate - value date of far leg
40=D	//OrdType - previously quoted

Forward/Forward (1M/3M) order

35=AB	//MsgType - New Order Multileg message type
49=ABC_AM	//SenderCompID - sending client
56=SSBFX	//TargetcompID - target bank
11=ORD456	//ClOrdID - uniquely assigned by client, format is arbitrary
117=QT456	//QuoteID - references the quote
54=B	//Side - as defined in NoLegs
55=EUR/USD	//Symbol

© Copyright, 2007-2008, FIX Protocol, Limited

15=EUR	//Currency - dealt currency
555=2	//NoLegs
600=EUR/USD	// [1] LegSymbol
556=EUR	// [1] LegCurrency
624=1	// [1]LegSide - Buy
687=1000000	// [1] LegQty - amount of near leg
654=B0001	// [1] LegRefID
566=1.223475	// [1] LegPrice - near leg
587=6	// [1] LegSettlType - future
588=20060228	// [1] LegSettlDate - value date of near leg
600=EUR/USD	// [2] LegSymbol
556=EUR	// [1] LegCurrency
624=2	// [1]LegSide - Sell
687=1000000	// [2] LegQty - amount of far leg
654=B0002	// [2] LegRefID
566=1.227599	// [1] LegPrice - far leg
587=6	// [2] LegSettlType - future
588=20060428	// [2] LegSettlDate - value date of far leg
40=D	//OrdType - previously quoted

Execution Report for FX Swap Multi-legged Order

The examples below shows execution reports for Spot/Forward (1M) and Forward/Forward (1M/3M) orders.

Execution report for Spot/Forward (1M) order

35=8	//MsgType - Execution Report message type
49=ABC_AM	//SenderCompID - sending client
56=SSBFX	//TargetcompID - target bank
37=ER123	//OrderID - uniquely assigned by broker
11=ORD123	//ClOrdID - uniquely assigned by client, format is arbitrary
17=ER123-1	//ExecID
150=F	//ExecType - Trade
39=2	//OrdStatus - filled
54=B	//Side - as defined in NoLegs
40=D	//OrdType - previously quoted
55=EUR/USD	//Symbol
15=EUR	//Currency - dealt currency
1071 =0.002048	//LastSwapPoints
194=1.2214	//LastSpotRate
555=2	//NoLegs
600=EUR/USD	// [1] LegSymbol
556=EUR	// [1] LegCurrency
624=1	// [1]LegSide - Buy
687=1000000	// [1] LegQty - amount of near leg
654=A0001	// [1] LegRefID
587=0	// [1] LegSettlType - Regular/Spot ((T+1) / (T+2))
588=20060130	// [1] LegSettlDate - value date of near leg
637=1.2214	// [1] LegLastPx - near leg
675=USD	// [1] LegSettlCurrency - near leg
1073 = 0	// [1] LegLastForwardPoints - near leg
1074 = 1221400	// [1] LegCalculatedCcyLastQty
600=EUR/USD	// [2] LegSymbol
556=EUR	// [1] LegCurrency
624=2	// [1]LegSide - Sell
687=1000000	// [2] LegQty - amount of far leg

© Copyright, 2007-2008, FIX Protocol, Limited

654=A0002	// [2] LegRefID
587=6	// [2] LegSettlType - future
588=20060228	// [2] LegSettlDate - value date of far leg
637=1.223448	// [1] LegLastPx - far leg
675=USD	// [1] LegSettlCurrency - far leg
1073 = 0.002048	// [1] LegLastForwardPoints - far leg
1074 = 1223488	// [1] LegCalculatedCcyLastQty

Execution report for Forward/Forward (1M/3M) order

35=8	//MsgType - Execution Report message type
49=ABC_AM	//SenderCompID - sending client
56=SSBFX	//TargetcompID - target bank
37=ER456	//OrderID - uniquely assigned by broker
11=ORD456	//ClOrdID - uniquely assigned by client, format is arbitrary
17=ER456-1	//ExecID
150=F	//ExecType - Trade
39=2	//OrdStatus - filled
54=B	//Side - as defined in NoLegs
40=D	//OrdType - previously quoted
55=EUR/USD	//Symbol
15=EUR	//Currency - dealt currency
1071 = 0.004124	//LastSwapPoints
194=1.2214	//LastSpotRate
555=2	//NoLegs
600=EUR/USD	// [1] LegSymbol
556=EUR	// [1] LegCurrency
624=1	// [1]LegSide - Buy
687=1000000	// [1] LegQty - amount of near leg
654=B0001	// [1] LegRefID
587=6	// [1] LegSettlType - future
588=20060228	// [1] LegSettlDate - value date of near leg
637=1.223475	// [1] LegLastPx - near leg
675=USD	// [1] LegSettlCurrency - near leg
1073 = 0.002075	// [1] LegLastForwardPoints - near leg
1074 = 1223475	// [1] LegCalculatedCcyLastQty
600=EUR/USD	// [2] LegSymbol
556=EUR	// [1] LegCurrency
624=2	// [1]LegSide - Sell
687=1000000	// [2] LegQty - amount of far leg
654=B0002	// [2] LegRefID
587=6	// [2] LegSettlType - future
588=20060428	// [2] LegSettlDate - value date of far leg
637=1.227599	// [1] LegLastPx - far leg
675=USD	// [1] LegSettlCurrency - far leg
1073 = 0.006199	// [1] LegLastForwardPoints - far leg
1074 = 1227599	// [1] LegCalculatedCcyLastQty

Settlement Obligation Report

The example below shows an example of the Settlement Obligation Report from a central counterparty.

35=BQ	//MsgType - Settlement Obligation Report
49=CCP	//SenderCompID - the central counterparty
56=ClientA	//TargetCompID - the receiver/client of the CCP

© Copyright, 2007-2008, FIX Protocol, Limited

715=20060929	//ClearingBusinessDate
1160=123456	//SettlObligMsgID
1159=2	//SettlObligMode - Final
1153=3	//SettlmentCycleNo
60=20060929-16:45:15.006	//TransactTime
1165=1	//NoSettlOblig - 1 instance
430=1	// [1] NetGrossInd - Net
1161=7654321	// [1] SettlObligID
1162=N	// [1] SettlObligTransType - New
1157=1000000	// [1] CcyAmt - net flow of currency 1 (positive means firm receiving)
119=-1200000	// [1] SettlCurrAmt - net flow of currency 2 (negative means firm paying)
15=EUR	// [1] Currency - currency 1, the "dealt" currency
120=USD	// [1] SettlCurrency - currency 2, the "contra" currency
155=1.2	// [1] SettlCurrFxRate - rate
64=20061002	// [1] SettlDate
207=FXM	// [1] SecurityExchange
48=EURUSD	// [1] SecurityID
22=8	// [1] SecurityIDSource - Exchange Symbol
200=20061002	// [1] MaturityMonthYear
461=FFCPNO	// [1] CFICode
167=FOR	// [1] SecurityType - foreign exchange contract
453=6	// [1] NoParties - 6 instances
448=CME	// [1.1] PartyID
447=H	// [1.1] PartyIDSource - CSD participant/member code
452=21	// [1.1] PartyRole - clearning organization
448=350	// [1.2] PartyID
447=H	// [1.2] PartyIDSource - CSD participant/member code
452=4	// [1.2] PartyRole - clearing firm
448=CME	// [1.3] PartyID
447=H	// [1.3] PartyIDSource - CSD participant/member code
452=22	// [1.3] PartyRole - exchange
448=350	// [1.4] PartyID
447=H	// [1.4] PartyIDSource - CSD participant/member code
452=1	// [1.4] PartyRole - executing firm
448=350	// [1.5] PartyID
447=H	// [1.5] PartyIDSource - CSD participant/member code
452=38	// [1.5] PartyRole - position account
802=1	// [1.5] NoPartySubIDs
523=1	// [1.5.1] PartySubID
803=26	// [1.5.1] PartySubIDType - position account type
448=CUST1234	// [1.6] PartyID
447=H	// [1.6] PartyIDSource - CSD participant/member code
452=24	// [1.6] PartyRole - customer account
1158=2	// [1] NoSettlDetails
1164=1	// [1.1] SettlObligSource - broker's instructions
781=1	// [1.1.] NoSettlPartyIds
782= XABC12345XXX	// [1.1.1] SettlPartyID
783=B	// [1.1.1] SettlPartyIDSource - BIC
784=10	// [1.1.1] SettlPartyRole - settlement location
801=1	// [1.1.1] NoSettlPartySubIDs
785= Acct12345	// [1.1.1.1] SettlPartySubIDs
786=15	// [1.1.1.1] SettlPartySubIDType - cash account number
1164=2	// [1.2] SettlObligSource - institutions instructions

© Copyright, 2007-2008, FIX Protocol, Limited

781=1	// [1.2] NoSettlPartyIds
782= XDEF5678XXX	// [1.2.1] SettlPartyID
783=B	// [1.2.1] SettlPartyIDSource - BIC
784=10	// [1.2.1] SettlPartyRole - settlement location
801=1	// [1.2.1] NoSettlPartySubIDs
785= Acct56789	// [1.2.1.1] SettlPartySubIDs
786=15	// [1.2.1.1] SettlPartySubIDType - cash account number

© Copyright, 2007-2008, FIX Protocol, Limited

USER GROUP: EXCHANGES AND MARKETS

Introduction

This section addresses issues and requirements that are specific to environments of exchanges and similar marketplaces where many parties interact with a central system, and provides guidance for using the FIX Protocol in these environments. The behaviour can differ from the typical buy-side/sell-side environments and is described here where applicable . The content is produced by the former Exchange and ECN Working Group which has been superseded by the Global Exchanges and Markets Committee.

The content of this section supplements the content in the other volumes of the FIX Protocol specification where appropriate. Additionally the usage notes in this section may describe additional requirements above the base requirements of the FIX Protocol that is recommended for use by Exchanges.

Order State Change Matrices for Exchanges

This section addresses issues with order state changes in an exchanges or marketplace environment. These supplement the Order State Change Matrices in Volume 4 of the FIX Protocol specification and are documented as specific to exchanges and marketplaces. The titles and references have been chosen in accordance with the existing matrices in Volume 4. These specific cases supercedes the ones in Volume 4 when implementing the FIX Protocol for exchanges and centralized marketplaces. Order State Changes matrices as documented in Volume 4 that are not mentioned in this section applies to the exchange and centralized marketplace environment. Please also refer to the Order State Change Matrices defined in *Volume 4 – FIX Application Messages: Orders and Executions (Trade).*

A Vanilla

Ref	Description
A.1.a	Filled order after order rests on book
A.1.b	Part-filled day order after order rests on book, done for day
A.1.c	Order filled upon hitting the book
A.1.d	Order partially filled upon hitting the book

I TimeInForce

Ref	Description
I.1.a	Fill or Kill order cannot be filled
I.1.b	Immediate or Cancel order that cannot be immediately hit

Applicability of scenarios depicted in Volume 4 for electronic exchange/ECN environments:

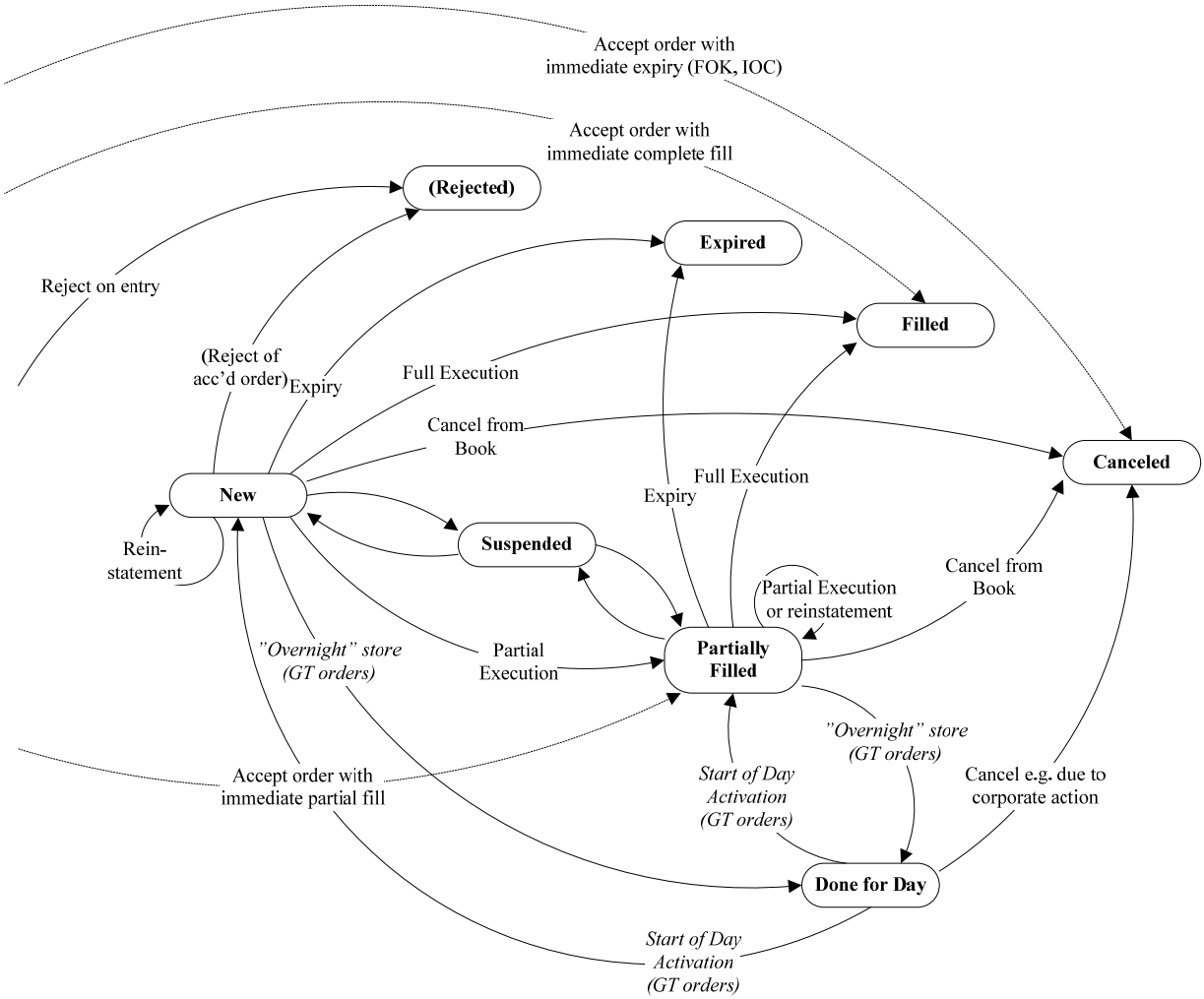
- Filled and Canceled are considered to be terminal states of an order, i.e. a state transition from Filled or Canceled to Partially Filled or Pending Replace should be avoided
- Pending order states requiring additional messages should be avoided in the interest of performance

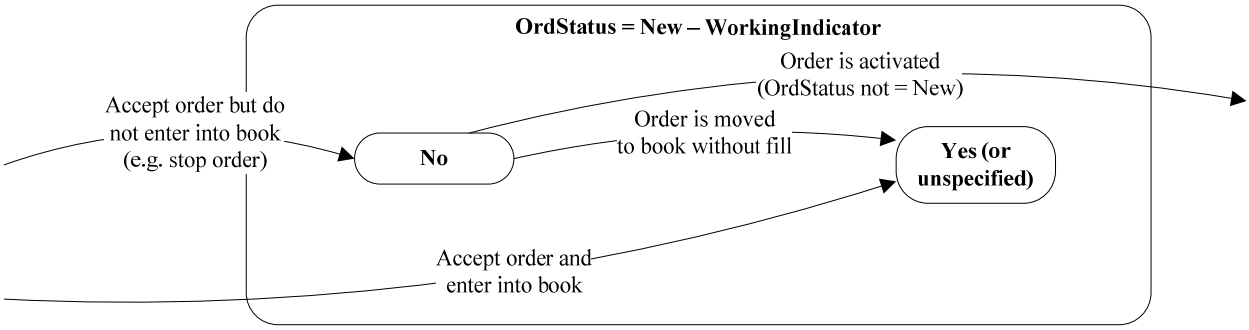
The ExecType is used to identify the purpose of the execution report message. The value of ExecType will typically be New to convey the fact that a new order has been received and processed. However, the value of OrdStatus in

© Copyright, 2007-2008, FIX Protocol, Limited

this initial response may not necessarily be New as the order might have been executed immediately. The initial value of OrdStatus can therefore also be Partially Filled or Filled. It can even be Canceled if the order has time in force values such as Fill or Kill and Immediate or Cancel and the order could not be executed immediately (and in its entirety in case of Fill or Kill).

The following diagram illustrates the complete set of state transitions recommended for electronic exchange/ECN environments. The dotted lines lead to initial order states other than New and apply to cases where an order does not simply rest on the order book after having been accepted by the exchange/ECN. It is a possibility aimed at increasing performance by reducing the overall number of "Execution Report" messages that need to be provided and processed. Message flows with explicit messages to convey the order state New are equally possible.




OrdStatus = New – WorkingIndicator
Order is activated
(OrdStatus not = New)
Accept order but do
not enter into book
(e.g. stop order)
No
Order is moved
to book without fill
Yes (or
unspecified)
Accept order and
enter into book

A Vanilla

A.1.a - Filled order after order rests on book

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**Exec Type**	**OrdStatus**	**Order Qty**	**Cum Qty**	**Leaves Qty**	**Last Qty**	**Comment**
1	New Order(X)				10000				
2		*Execution(X)*	*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by salesthe exchange*
2		Execution(X)	New	New	10000	0	10000	0	
3		*Execution(X)*	*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by trader/exchange*
3		Execution(X)	Trade	Partially Filled	10000	2000	8000	2000	Execution of 2000
4		Execution(X)	Trade	Partially Filled	10000	3000	7000	1000	Execution of 1000
5		Execution(X)	Trade	Filled	10000	10000	0	7000	Execution of 7000

© Copyright, 2007-2008, FIX Protocol, Limited

A.1.b – Part-filled day order after order rests on book, done for day

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**Exec Type**	**OrdStatus**	**Order Qty**	**Cum Qty**	**Leaves Qty**	**Last Qty**	**Comment**
1	New Order(X)				10000				
2		*Execution(X)*	*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by the exchange*
2		Execution(X)	New	New	10000	0	10000	0	
3		Execution(X)	Trade	Partially Filled	10000	2000	8000	2000	Execution of 2000
4		Execution(X)	Trade	Partially Filled	10000	3000	7000	1000	Execution of 1000
5		Execution(X)	Done for Day	Done for Day	10000	3000	0	0	Assuming day order. See other examples which cover GT orders

© Copyright, 2007-2008, FIX Protocol, Limited

A.1.c – Order filled upon hitting the book

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**Exec Type**	**OrdStatus**	**Order Qty**	**Cum Qty**	**Leaves Qty**	**Last Qty**	**Comment**
1	New Order(X)				10000				
2		*Execution(X)*	*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by the exchange*
2		Execution(X)	Trade	Filled	10000	10000	0	10000	Immediate execution of 10000

A.1.d – Order partially filled upon hitting the book

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**Exec Type**	**OrdStatus**	**Order Qty**	**Cum Qty**	**Leaves Qty**	**Last Qty**	**Comment**
1	New Order(X)				10000				
2		*Execution(X)*	*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by the exchange*
2		Execution(X)	Trade	Partially Filled	10000	7000	3000	7000	Immediate execution of 7000

© Copyright, 2007-2008, FIX Protocol, Limited

I TimeInForce

I.1.a – Fill or Kill order cannot be filled

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**Exec Type**	**OrdStatus**	**Order Qty**	**Cum Qty**	**Leaves Qty**	**Last Qty**	**Comment**
1	New Order(X)				10000				Order is FOK
2		*Execution(X)*	*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by sell-side (broker,the exchange, ECN)*
2		Execution(X)	New	New	10000	0	10000	0	If messages are not bundled
3		Execution(X)	Canceled	Canceled	10000	0	0	0	If order cannot be immediately filled
4	New Order(Y)				10000				Order is FOK
5		*Execution(Y)*	*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by sell-side (broker, exchange, ECN)*
6		Execution(Y)	Canceled	Canceled	10000	0	0	0	If message bundling is being used and order cannot be immediately filled

I.1.b – Immediate or Cancel order that cannot be immediately hit completely

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**Exec Type**	**OrdStatus**	**Order Qty**	**Cum Qty**	**Leaves Qty**	**Last Qty**	**Comment**
1	New Order(X)				10000				Order is IOC
2		*Execution(X)*	*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by sell-side (broker,the exchange, ECN)*
2		Execution(X)	New	New	10000	0	10000	0	If messages are not bundled
3		Execution(X)	Trade	Partially Filled	10000	1000	9000	1000	Execution for 1000
4		Execution(X)	Canceled	Canceled	10000	1000	0	0	If order cannot be immediately hit completely
5	New Order(Y)				10000				Order is IOC
6		*Execution(Y)*	*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by sell-side (broker,the exchange, ECN)*
6		Execution(Y)	Trade	Canceled	10000	1000	9000	1000	If message bundling is being used and execution of 1000 occurs immediately upon hitting the book

© Copyright, 2007-2008, FIX Protocol, Limited

Continuous Market Maker Quoting

This section discusses the use of Quote and Mass Quote related messages for continuous quoting by a centralised market.

Quote Identifiers

Quote Entity Identifer

Every individual quote needs a unique identifier. The identifier should refer back to investor and his wish to invest (in the quote case, the quote issuer). In the case of continuous market maker quotes, the quote identifier is preferably static (as the decision to quote do not really change). An important aspect of this identifier is that it can be used on trades as a reference back to the quote (i.e. in Execution Reports and Trade Capture Reports). A further aspect is that the cumulative quantity and other similar properties of the Execution Report are maintained throughout the lifetime of the QuoteID (in practice normally terminated and restarted each day). Note that in the case of Quote Negotiation where the Quote is used to reply to Quote Requests, the quote identifier could have more of an order characteristic, and thereby be assigned a new value for every quote request received.

Quote identifiers are supported through:

- QuoteEntryID (299) in Mass Quote messages
- QuoteID (117) field in single Quote messages

It is recommended that the QuoteEntryID (299) and QuoteID (117) remains static when quote updates are done - in practice the quote identifier value does not change. In cases where a quote issuer is allowed to have multiple simultaneous quotes in the market, the quote identifier identifies which one of those should be updated.

The scope within which the quote identifier is unique varies and details of the identifier model should be bilaterally agreed. Recognized models:

1. ID is unique in context of quote issuer (market maker + Quote ID). Includes model where the ID is unique in itself but embeds the market maker identifier.
2. Separate ID not used at all (a quote is identified by market maker + Security)
3. ID unique in context of quote issuer and security (market maker + Security + Quote ID). This means the ID is always “1” with the following exceptions:
    - In cases where multiple quotes are allowed in a single security
    - When quotes are used in quote negotiations (multiple parallel negotiations in same security).
4. ID unique per market maker + QuoteID + QuoteSetID.

For continuous Market Maker quotes, a marketplace can assign the quote identifier values the Market Maker should use, or leave that to the Market Maker. The former is preferred in cases where the marketplace wants a globally unique quote entry index. This behavior is very similar to the marketplace assigned OrderIDs used in order routing - but assignment is not done through interactive responses, it is done by bilateral agreement/contract.

Quote Message Identifier

In cases where participants need an audit trail for quote messages, the Quote and Mass Quote messages need a quote-issuer assigned identifier. The identifier must be unique for every submission of a quote (whether the quote is new, updated or canceled). The primary usage is to serve as a message identifier. Users that so wish can secondarily use the field as a revision count – which, together with the Quote identifier, could be unique. The message identifier is relayed back on outbound messages produced as a result of the Quote. The preferred message identifiers are:

© Copyright, 2007-2008, FIX Protocol, Limited

- QuoteID (117) in the Mass Quote message
- QuoteMsgID (1166) for single Quote message

Quote referencing in Market Data, Executions (fills), etc.

Execution Reports, Trade Capture Reports, Market Data and other messages produced as a consequence of a quote may need to refer to the quote. Some markets and users require that the exact revision of the quote is relayed (i.e. the identifier of the message that last updated the quote).

The recommended approach is to:

- Use the ClOrdID (11) for the Quote entity identifier
- Use the SecondaryClOrdID (526) field for the Quote message identifier in cases where it needs to be relayed
- In cases where the OrderID (37) field has no other usage, set it to "[N/A]" or the Quote entity identifier

Details of identifier usage should be bilaterally agreed.

Use of Quote Identifiers

Inbound quote messages can have two identifiers:

- A message identifier that is used to identify each inbound message uniquely. This message identifier has a purpose similar to the ClOrdID used for orders.
- An entity identifier that is used to identify each quote entry over time. This identifier has a purpose similar to the OrderID used for orders, but it should be noted that the receiver of a quote does not assign this identifier - it is expected to be entered by the quote issuer.

The following table illustrates the use of the Quote identifiers in various messages in quote workflows:

Table 1 - Quote Identifiers

Message Type	Identifier Type	Field	Comment
Quote	Message identifier	QuoteMsgID	
	Entity identifier	QuoteID	
Mass Quote	Message identifier	QuoteID	Note that the QuoteID is the message identifier in Mass Quote messages!
	Entity identifier	QuoteEntryID	
Quote Cancel	Message identifier	QuoteMsgID	
	Entity Identiifier	QuoteID	
Quote Status Report	Message identifier	QuoteMsgID	Note that in the case a Mass Quote was the source of the individual quote – the QuoteID goes into the QuoteMsgID
	Entity identifier	QuoteID	Note that in the case a Mass Quote was the source of the individual quote – the QuoteEntryID goes into the QuoteID

© Copyright, 2007-2008, FIX Protocol, Limited

Message Type	Identifier Type	Field	Comment
Mass Quote Acknowledgement	Message identifier	QuoteID	Note that in the case of a Quote Cancel - the QuoteMsgID of the Quote Cancel go into the QuoteID. Note that in the case of a Quote Status Request – the QuoteStatusReqID go into the QuoteID
	Entity identifier	QuoteEntryID	Note that in the case a (Single) Quote was the source of the individual quote – the QuoteID goes into the QuoteEntryID
Other messages	Message identifier	ClOrdID	Note that the message identifier of the original Quote/Mass Quote message goes into the ClOrdID for reference
	Entity identifier	SecondaryClOrdID	Note that the entity identifier of the original Quote/Mass Quote message goes into the SecondaryClOrdID for reference

Use of the QuoteSetID Field

There is arguably no value in using the QuoteSetID (302) for any business purposes. The field is mandatory in Mass Quote messages, but reasonably only as it is a delimiter for the Quote Set repeating group. As the QuoteEntryID can be used to uniquely identify the entries, the QuoteSetID can be regarded as a simple sequence number that has no persistence outside the message instance.

Pre FIX 5.0 SP1 versions of FIX assigned the QuoteSetID more importance, the QuoteEntryID for example was defined in the Mass Quote message as:

- "Uniquely identifies the quote as part of a QuoteSet. Must be used if NoQuoteEntries is used"

That definition is considered contrary to the practices used in many implementations. The role of the QuoteSetID (302) is now deemphasized allowing for bilateral agreement of quote identification schemes.

Quote Acknowledgement and Status

Some markets require that every incoming quote message should have an ack (or reject). This is obvious in cases where marketplace identifiers need to be relayed as they are used in subsequent messages as references to the quote. Quote Ack messages can also be used to limit the re-quoting speed, e.g. to support rules such as "Market Maker is allowed to send one quote update before ack on first quote is received, not more". An ack message is preferably very lean and should contain minimal information.

Reject messages should also be as lean as possible. In the case of a Mass Quote:

© Copyright, 2007-2008, FIX Protocol, Limited

- If the whole message is rejected, only the root part of the Mass Quote is relayed back, i.e. the QuoteID
- If individual Quote Entries are rejected, the root + the individual rejected entries are relayed back, not the accepted entries.

The marketplace by bilateral agreement, or the user by providing the QuoteResponseLevel (301) field, should be able to indicate what quote responses are to be disseminated. Alternatives include:

- **No acks at all**. Meaning neither Mass Quote Acknowledgement nor Quote Status Reports are produced for solicted actions.
- **Rejects only**. Meaning no positive ack's is produced for solicited actions.
- **Summary Acknowledgement.** Meaning e.g. a Mass Quote Acknowledgement is produced for mass Quote Cancels. The Mass Quote Ack will in those cases show the total number of canceled quotes (per underlying), not the individual quotes. A Mass Quote Ack without specifying the quote entries could also be relayed as an ack to Mass Quote messages.
- **Ack each**. Same as Summary Acknowledgement except that Quote Status Reports are produced for every individual quote entry fill.

As practices vary, FIX does not require a specific default for the QuoteResponseLevel (301) field. This is especially relevant for markets not supporting the field as part of messages (instead using a standard behavior).

Parties can bilaterally agree whether to relay unsolicited Quote Status Reports produced as a result of (example events):

- The quote being exhausted and the marketplace encouraging the quote issuer to re-quote
- Other similar "warnings" to the quote issuer. Note that "locked" and "crossed" market warning are available in the QuoteStatus (297) field.
- Marketplace initiated quote modifications, for example a quote modified based on "out" or replenishment parameters which automatically inserts a new quote (with wider spread) if the quoted quantity is filled.

A reject message should contain a reject reason. Reject reasons may need to include "Quote Locked - Unable to Update/Cancel" to cover for the Quote Cancel case where a quote is locked. For example, a quote may be locked for execution in another marketplace.

The Quote Status of "Active" can be used as a reply to queries where the quote is active in the market. In cases where the Mass Quote Acknowledgement is used to respond to a Quote Status request, the Quote Status needs to be relayed per individual quote entry.

On cancellation, the QuoteCancelType (298) can be echoed in responses and the single Quote Status of "Canceled" be used.

© Copyright, 2007-2008, FIX Protocol, Limited

Quote Status Usage Table

The following table shows the recommended use of the QuoteStatus field in the Quote Status Report and the Mass Quote Acknowledgment messages. The list of statuses shown includes the main values.

Request	Response		
	(Single) Quote Quote Status Report	Set of Quotes Mass Quote Acknowledgement	
	Quote Status	Quote Status	Quote Entry Status
Quote Status Request	• Rejected • Active • Canceled • Expired • Removed from Market • Quote Not Found	• Query • Rejected	• Active • Canceled • Expired • Removed from Market • Quote Not Found
Quote	• Accepted • Rejected • Canceled (if both sides = 0)	N/A	N/A
Mass Quote	N/A	• Accepted • Rejected	• Accepted • Rejected • Canceled (if both sides = 0)
Quote Cancel	• Canceled • Rejected	• Accepted • Rejected	• Canceled • Rejected (if "locked")
Unsolicited	• Removed from Market • Unsolicited Quote Replenishment	• Removed from Market • Unsolicited Quote Replenishment	• Removed from Market • Unsolicited Quote Replenishment

Reporting a Mass Cancel

Some marketplaces avoid sending out each individual quote entry on mass cancel. Those markets use an aggregated message instead, showing the underlying for which cancel have been done and the number of quotes removed. This action is used together with the QuoteResponseLevel (301) = Summary Acknowledgement. Example:

- Canceled Quotes
  - IBM – 200
  - APL – 300
  - DELL - 500

When acknowledging a mass cancel (using the Mass Quote Acknowledgement) the number of totally canceled quotes per Quote Set should be relayed. The requirement can be generalized to relay also the total number of

© Copyright, 2007-2008, FIX Protocol, Limited

accepted quotes and rejected quotes respectively. The following fields are used to relay the number of affected quotes:

- the number of canceled quote entries, TotNoCxldQuotes (1168)
- the number of accepted quote entries, TotNoAccQuotes (1169)
- the number of rejected quote entires, TotNoRejQuotes (1170)

The usage of the Mass Quote Acknowledgement message in response to a mass Quote Cancel should be bilaterally agreed as it means the requestor may have to receive a Quote Status report in the case of a single quote cancel or a Mass Quote Ack in the case multiple quotes where canceled.

Quote Cancel Scope

Market makers often supplies quotes based on a trading desk or some more virtual unit, e.g. in the case where the quoting obligations moves across geography depending on business hours in different parts of the world. At the same time the firm might be organized so various units quotes in various sets of security. One unit is then not allowed to cancel the quotes of another unit. High speed quoting may also require multiple sessions (connections) between each market maker unit and the marketplace. An implication of all this is that quotes are often not “owned” by individual traders or FIX sessions, but rather by that organizational unit.

Pre FIX 5.0 SP1 specification stated that Quote Cancels applied to quotes made by the "current user" (which could be interpreted as a FIX session). The concept of a "quote issuer" (a “unit”) is now introduced - i.e. a concept represented by the <Parties> component.

As an example, a Quote Cancel can limit the scope of a mass cancel by specifying the following PartyRoles:

- PartyRole = “1” – Executing Firm = Market Maker firm
- PartyRole = ”58” – Executing Unit = Trading desk
- PartyRole = “12” – Executing Trader = Individual trader

What party roles and scope limitations are available is bilaterally agreed.

Workflows

Introduction

The following rules are used as the basis for the workflows:

1. A Mass Quote always results in a Mass Quote Acknowledgement unless QuoteResponseLevel (301) = “0” (No Acknowledgement) has been specified. A Mass Quote’s should not result in multiple Quote Status Reports in response.
    a. The only exception to this rule occurs if restatements are needed due to automatic quantity refreshes or similar. In this case, the QuoteID (117) of that Quote Status Report would carry the QuoteEntryID (299) of the previously submitted Mass Quote.
2. A Quote Cancel can result in a Mass Quote Acknowledgement under the following conditions:
    a. Multiple quotes are affected, i.e. the QuoteCancelType (298) is set to “1” (Cancel for Symbol[s]), “2” (Cancel for Security Type[s]), “3” (Cancel for Underlying Symbol) or “4” (Cancel all Quotes)
    b. QuoteResponseLevel (301) has been set to “2” (Acknowledge each quote messages)

    If both conditions are not met then the Quote Status Report should be used. An exception to the rule would be a bilateral agreement to always do one or the other.
3. A Quote Status Request can result in a Mass Quote Acknowledgement under the following conditions:

© Copyright, 2007-2008, FIX Protocol, Limited

a. Multiple quotes are affected. This means that QuoteID (117) should not be provided and <UndInstrmtGrp> or other filters are specified, meaning e.g. that all strikes in a series should be returned. Since this is a query it is assumed that any qualified quote will be reported.

If the condition is not met then the Quote Status Report should be used. An exception to the rule would be a bilateral agreement to always do one or the other.

The below table defines what messages can be used to relay request responses and unsolicited actions back to the quote issuer. The following abbreviations are used in the Comment column to clarify the mapping between the response and the message of the quote origination:

- Q = Quote
- MQ = Mass Quote
- QSR = Quote Status report
- MQA = Mass Quote Acknowledgement
- QSRq = Quote Status Request

Table 2 - Quote Response messages

Incoming request	Action	Outgoing Response	Comment
(Single) Quote	New	Quote Status Report	QSR.QuoteID := Q.QuoteID QSR.QuoteMsgID := Q.QuoteMsgID
Mass Quote	New	Mass Quote Acknowledgement	MQA.QuoteID := MQ.QuoteID MQA.QuoteEntryID := MQ.QuoteEntryID
(Single) Quote	Update	Quote Status Report	QSR.QuoteID := QuoteID QSR.QuoteMsgID := Q.QuoteMsgID
Mass Quote	Update	Mass Quote Acknowledgement	MQA.QuoteID := MQ.QuoteID MQA.QuoteEntryID := MQ.QuoteEntryID
(Single) Quote	Cancel (Size&Qty=0)	Quote Status Report	QSR.QuoteID := Q.QuoteID QSR.QuoteMsgID := Q.QuoteMsgID
Mass Quote	Cancel (Size&Qty=0)	Mass Quote Acknowledgement	MQA.QuoteID := MQ.QuoteID MQA.QuoteEntryID := MQ.QuoteEntryID
Quote Cancel	Cancel	Quote Status Report	Canceling single quote and (subject to bilateral agreement) when cancelling multiple quotes QSR.QuoteID := Q.QuoteID or MQ.QuoteEntryID QSR.QuoteMsgID := QC.QuoteMsgID
		Mass Quote Acknowledgement	Canceling multiple quotes (subject to bilateral agreement) MQA.QuoteID := QC.QuoteMsgID MQA.QuoteEntryID := Q.QuoteID or MQ.QuoteEntryID
N/A	Unsolicited	Quote Status Report	Used for unsolicited replenishment of exhausted

© Copyright, 2007-2008, FIX Protocol, Limited

	State Change		quote size (subject to bilateral agreement) QSR.QuoteID := Q.QuoteID or MQ.QuoteEntryID QSR.QuoteMsgID := Q.QuoteMsgID or MQ.QuoteID
Quote Status Request	Query	Quote Status Report	Querying for single quotes and (subject to bilateral agreement) when querying for multiple quotes QSR.QuoteID := Q.QuoteID or MQ.QuoteEntryID QSR.QuoteMsgID := QSRq.QuoteStatusReqID
		Mass Quote Acknowledgement	Querying for multiple quotes (subject to bilateral agreement) MQA.QuoteID := QSRq.QuoteStatusReqID MQA.QuoteEntryID := Q.QuoteID or MQ.QuoteEntryID
N/A	Fills	Execution Report (and Trade Capture Reports)	SecondaryClOrdID := Q.QuoteID or MQ.QuoteEntryID ClOrdID := Q.QuoteMsgID or MQ.QuoteID

© Copyright, 2007-2008, FIX Protocol, Limited

Single Quote Message Scenarios

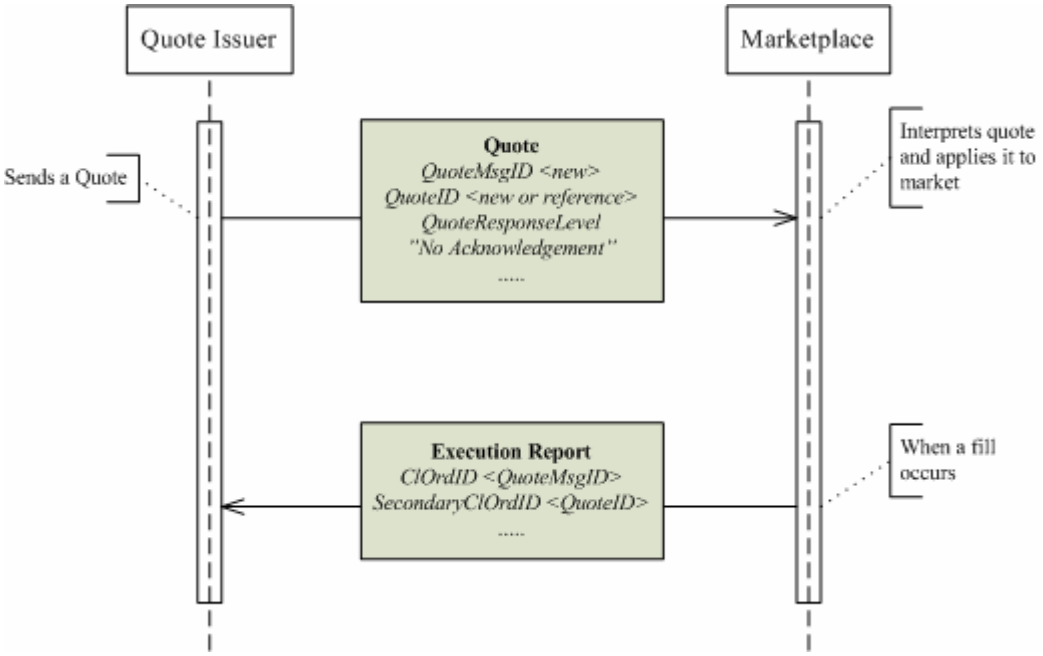
The (Single) Quote message supports:

- Adding individual quotes (if there was no previous quote in the market)
- Updating individual quotes (if there already was a quote in the market)
- Withdrawing (cancelling) individual quotes - if the bid / offer prices and sizes are set to zero in the message

Single Quote with QuoteResponseLevel = 0 (No Ack)

In the first example a Quote is sent from the quote issuer to the marketplace. The quote has the QuoteResponseLevel = 0 or omitted. The marketplace does not acknowledge the receipt of the quote. If the quote is later hit, resulting in a trade, an Execution Report is sent to the first party. The following Figure 7 depicts the workflow.

Figure 7: Single Quote with QuoteResponseLevel=0



Note that:

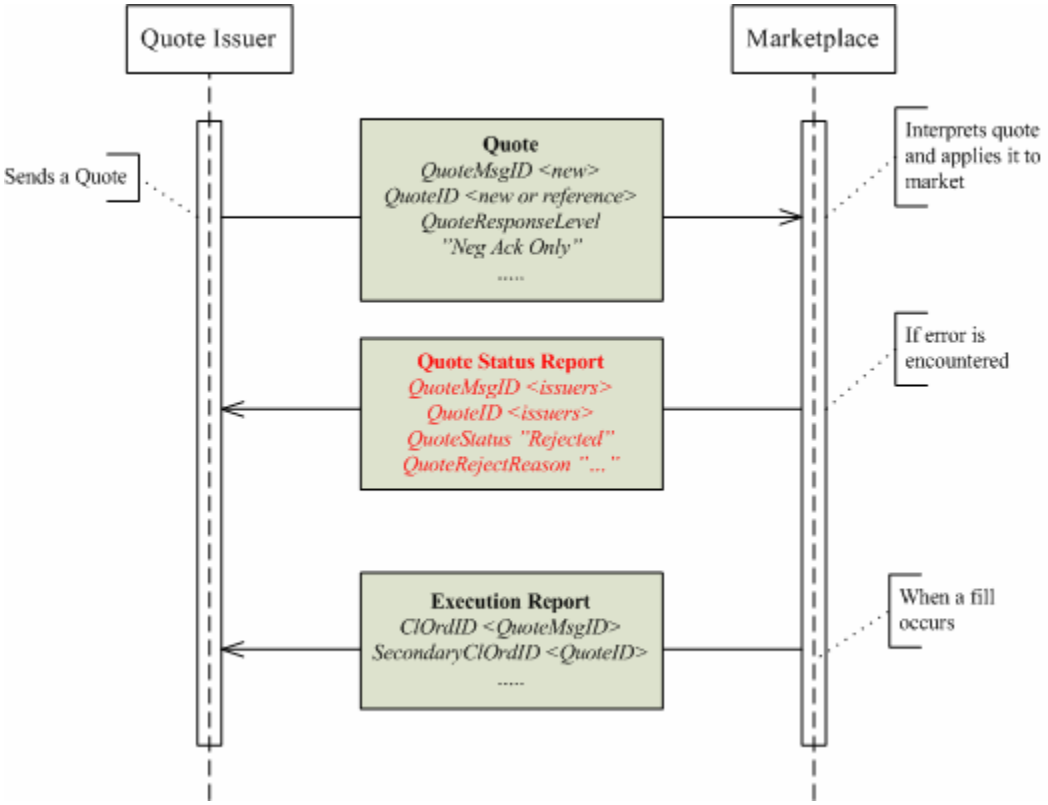
- The QuoteMsgID (if used) is renewed for every message sent.
- The QuoteID will contain a new value when a quote is first inserted and that id is then referenced for subsequent updates. The same id can be reused in cases where both sides of the quote are cancelled or exhausted – so a quote issuer can assign a static QuoteID to every quote responsibility (security or options series).

© Copyright, 2007-2008, FIX Protocol, Limited

Single Quote with QuoteResponseLevel = 1 (Negative Ack only)

In the second example, illustrated in Figure 8, a Quote is again sent from the quote issuer to the marketplace. The quote has the QuoteResponseLevel = 1. The marketplace only acknowledges the quote if there is an error. If the marketplace encounters an error while processing the quote, a Quote Status Report message is sent with the QuoteRejectReason set to the error encountered.

Figure 8: Single Quote with QuoteResponseLevel=1

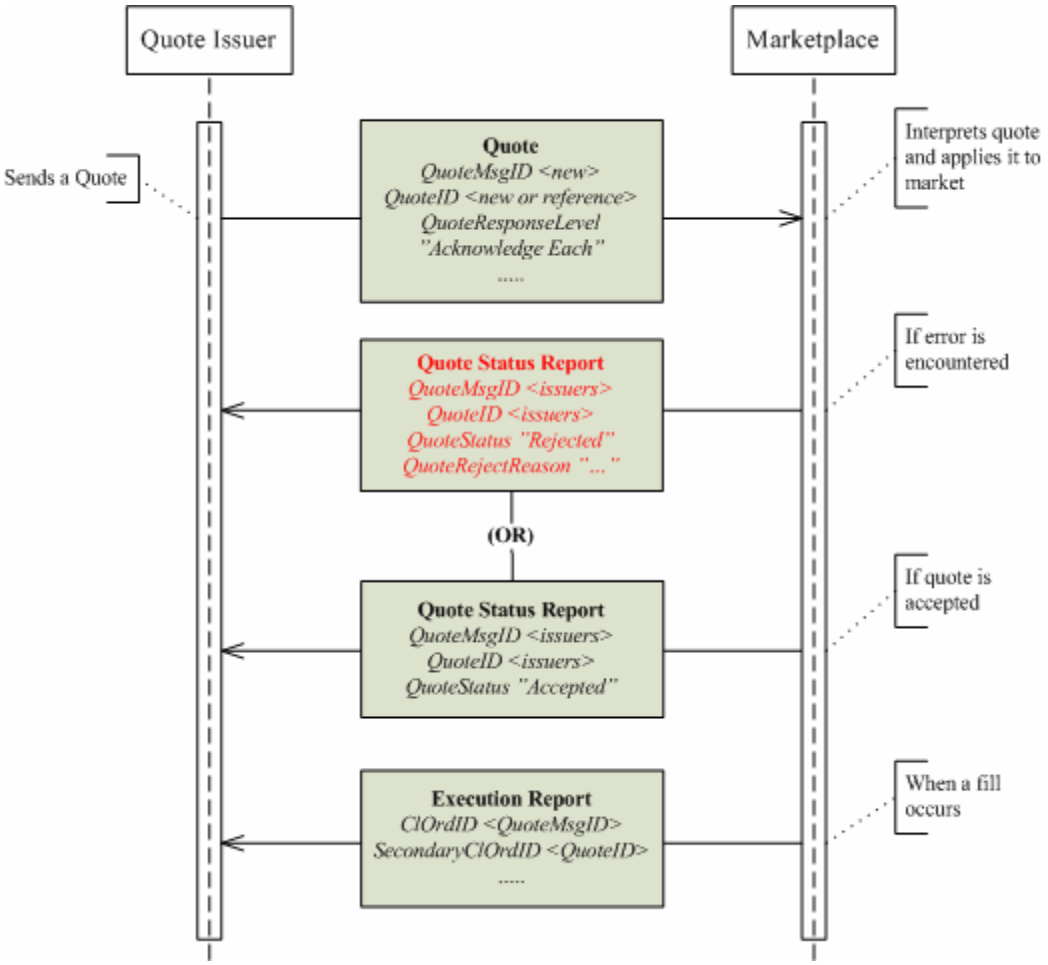


© Copyright, 2007-2008, FIX Protocol, Limited

Single Quote with QuoteResponseLevel = 2 (Ack each)

In the third example, shown in , a (Single) Quote is sent from the quote issuer to the marketplace. The quote has the QuoteResponseLevel = 2. The marketplace acknowledges each quote.

Figure 9: Single Quote with QuoteResponseLevel= 2

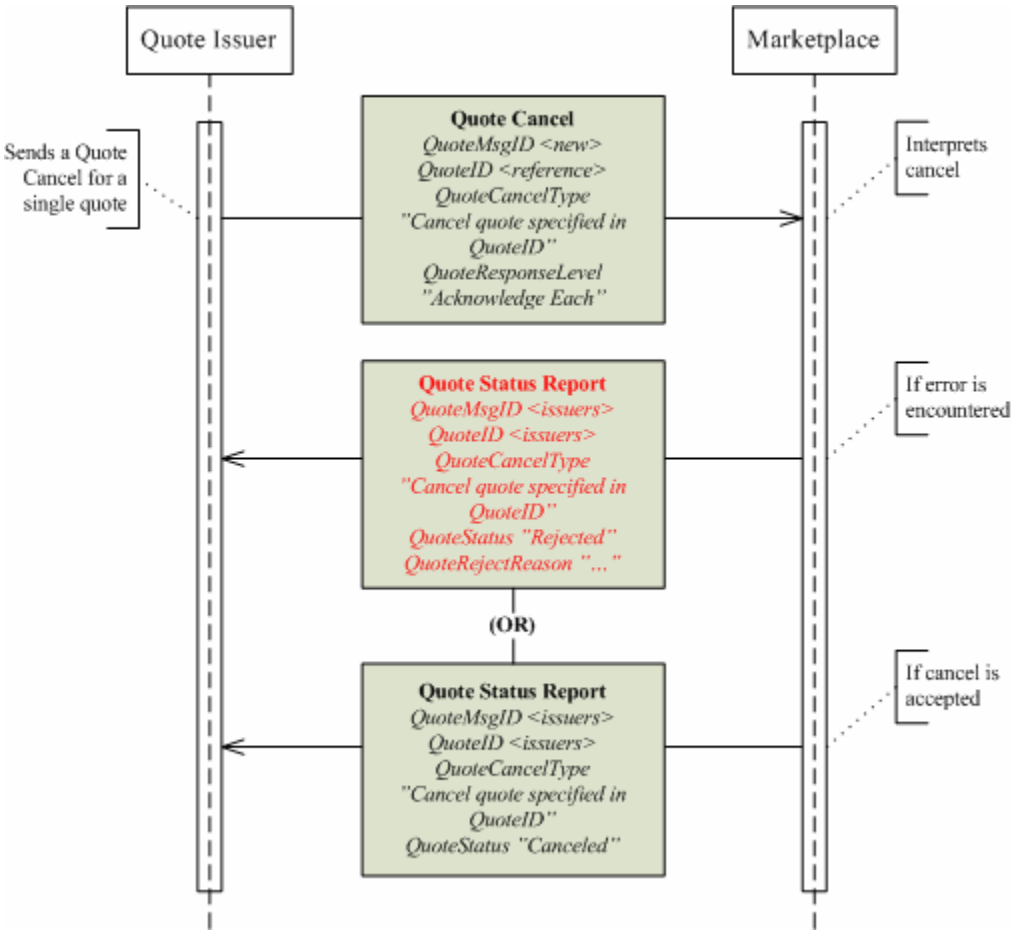


© Copyright, 2007-2008, FIX Protocol, Limited

Single Quote Cancel

Figure 10 shows an example of a Quote Cancel identifying a single quote to be cancelled is sent.

Figure 10: Single Quote Cancel

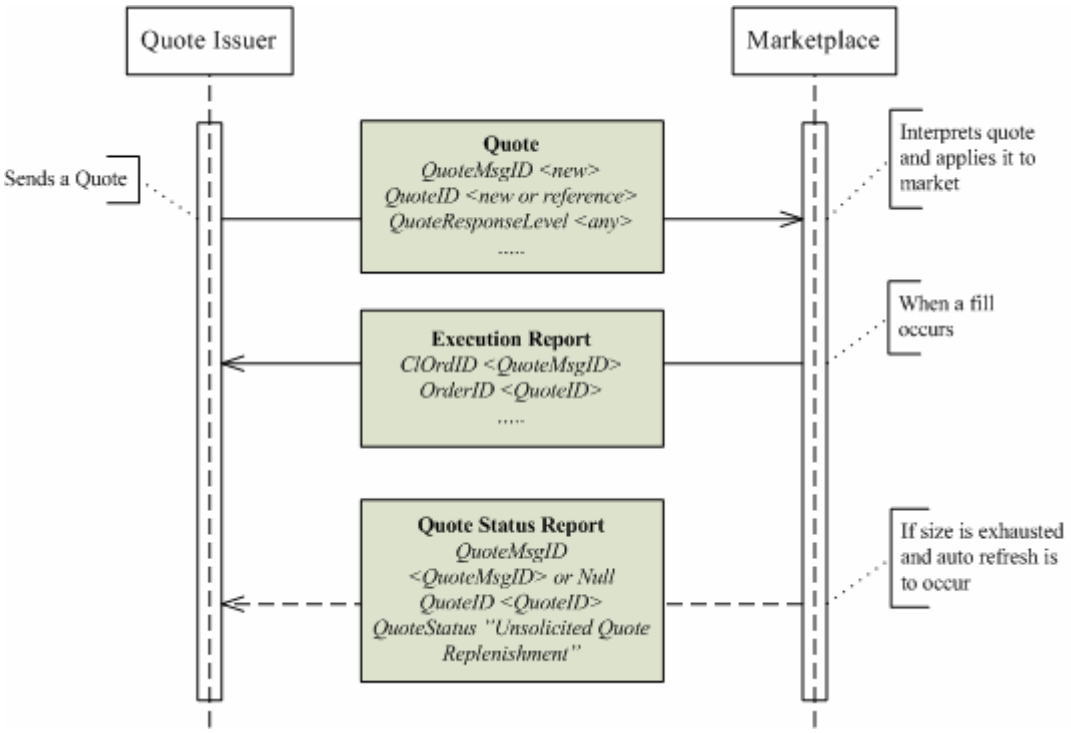


© Copyright, 2007-2008, FIX Protocol, Limited

Unsolicited Actions - Single Quote Restatement

Some marketplaces, when the quote size is exhausted, support the automatic replenishment with a pre-defined quantity (and moving the price). In such cases a restatement of the quote is appropriate.

Figure 11: Single Quote Restatement

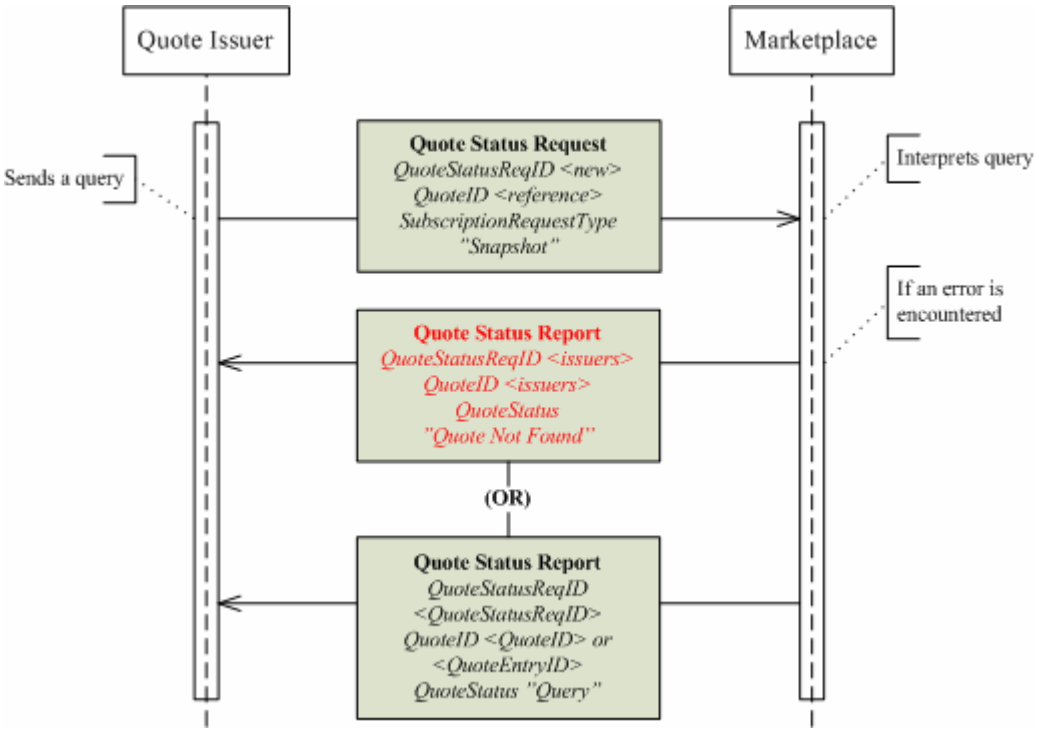


© Copyright, 2007-2008, FIX Protocol, Limited

Query for Single Quote

Figure 12 shows a Quote Status Request identifying a single quote is sent.

Figure 12: Query for Single Quote



© Copyright, 2007-2008, FIX Protocol, Limited

Mass Quote Messaging Scenarios

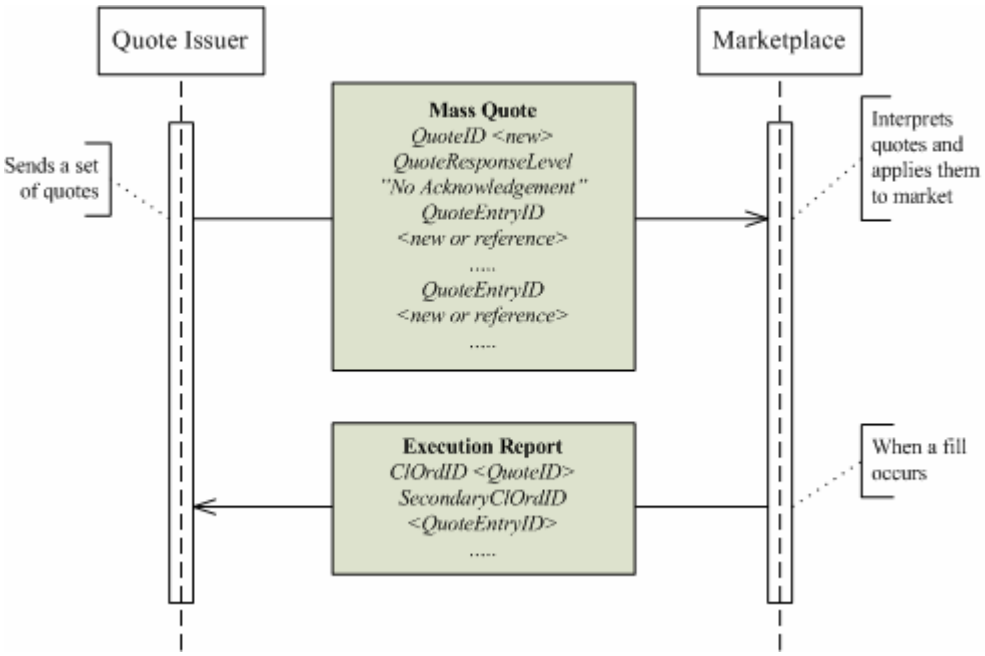
The Mass Quote message supports:

- Adding individual quotes (if there was no previous quote in the market)
- Updating individual quotes (if there already was a quote in the market)
- Withdrawing (cancelling) individual quotes – if the bid / offer prices and sizes are set to zero in the message

Mass Quote with QuoteResponseLevel = 0 (No Ack)

In the first example a Mass Quote is sent from the quote issuer to the marketplace. The quote has the QuoteResponseLevel (301) set to 0 or omitted. The marketplace does not acknowledge the quote. If the quote is later hit, resulting in a trade, an Execution Report is sent to the first party.

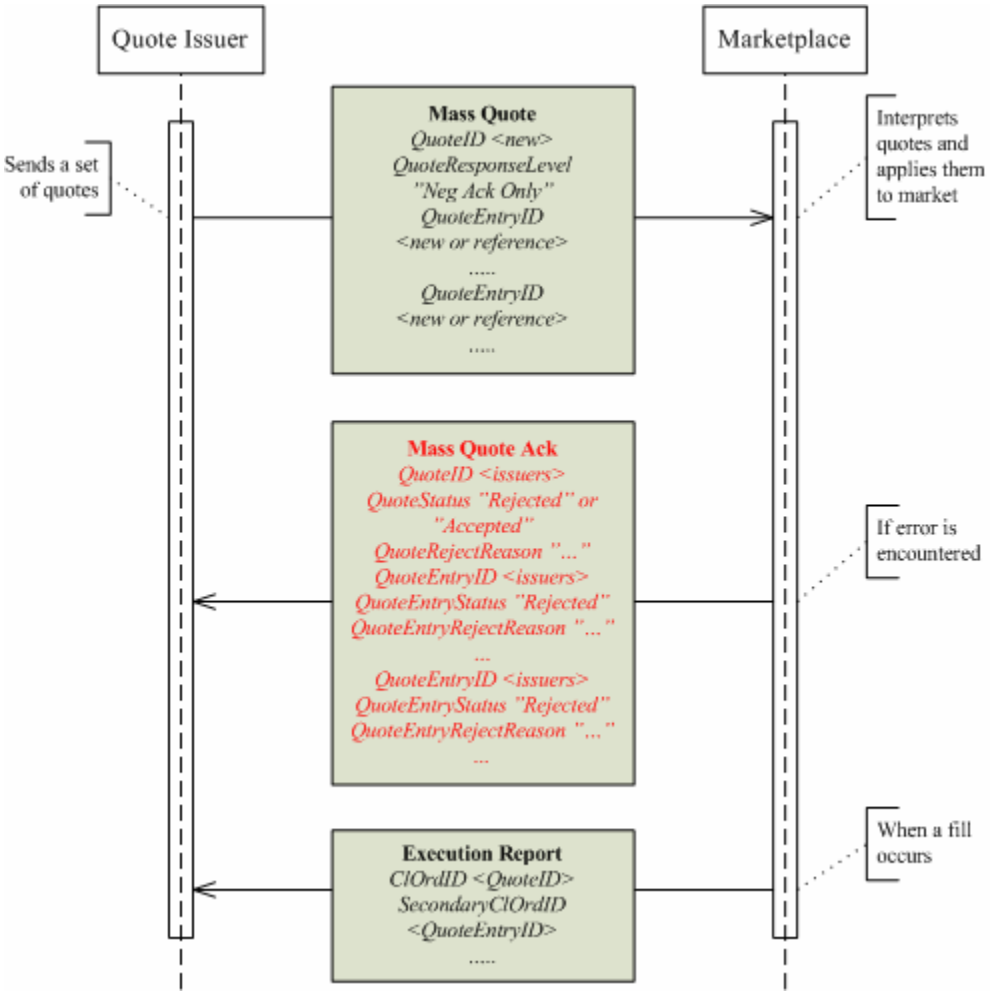
Figure 13: Mass Quote with QuoteResponseLevel=0



Mass Quote with QuoteResponseLevel = 1 (Negative Ack only)

In the second example a Mass Quote is sent from the market maker to the marketplace. The quote has the QuoteResponseLevel (301) set to 1. The marketplace only acknowledges the quote if there is an error. If the marketplace encounters an error while processing the quote, a Mass Quote Acknowledgement message is sent with the QuoteRejectReason set to the error encountered.

Figure 14: Mass Quote with QuoteResponseLevel=1

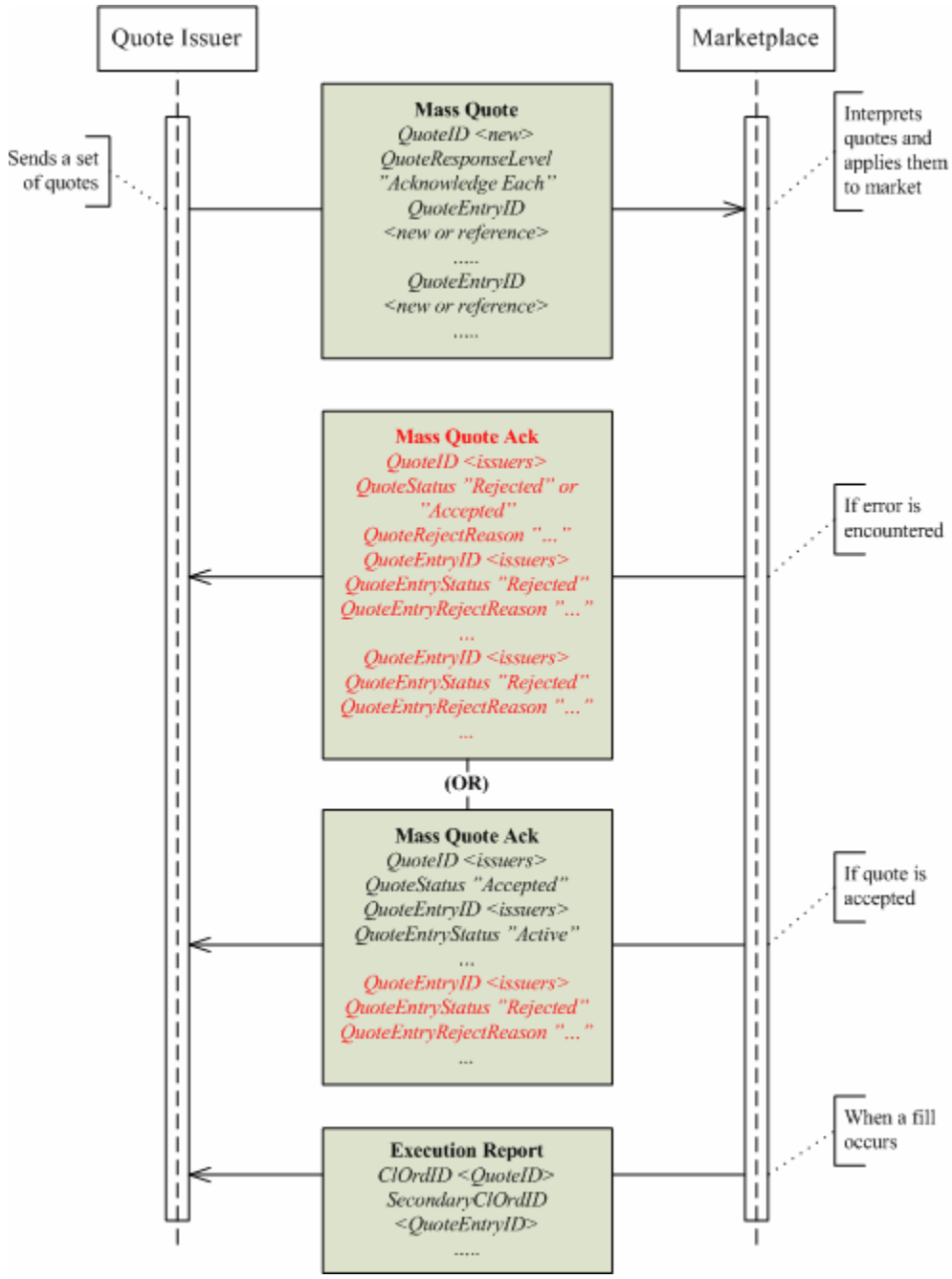


© Copyright, 2007-2008, FIX Protocol, Limited

Mass Quote with QuoteResponseLevel = 2 (Ack Each)

In the third example Mass Quote is sent from the market maker to the marketplace. The quote has the QuoteResponseLevel (301) set to 2. The marketplace acknowledges each quote. Note that the whole message can be accepted together with any number of quote entries, while individual quote entries are rejected.

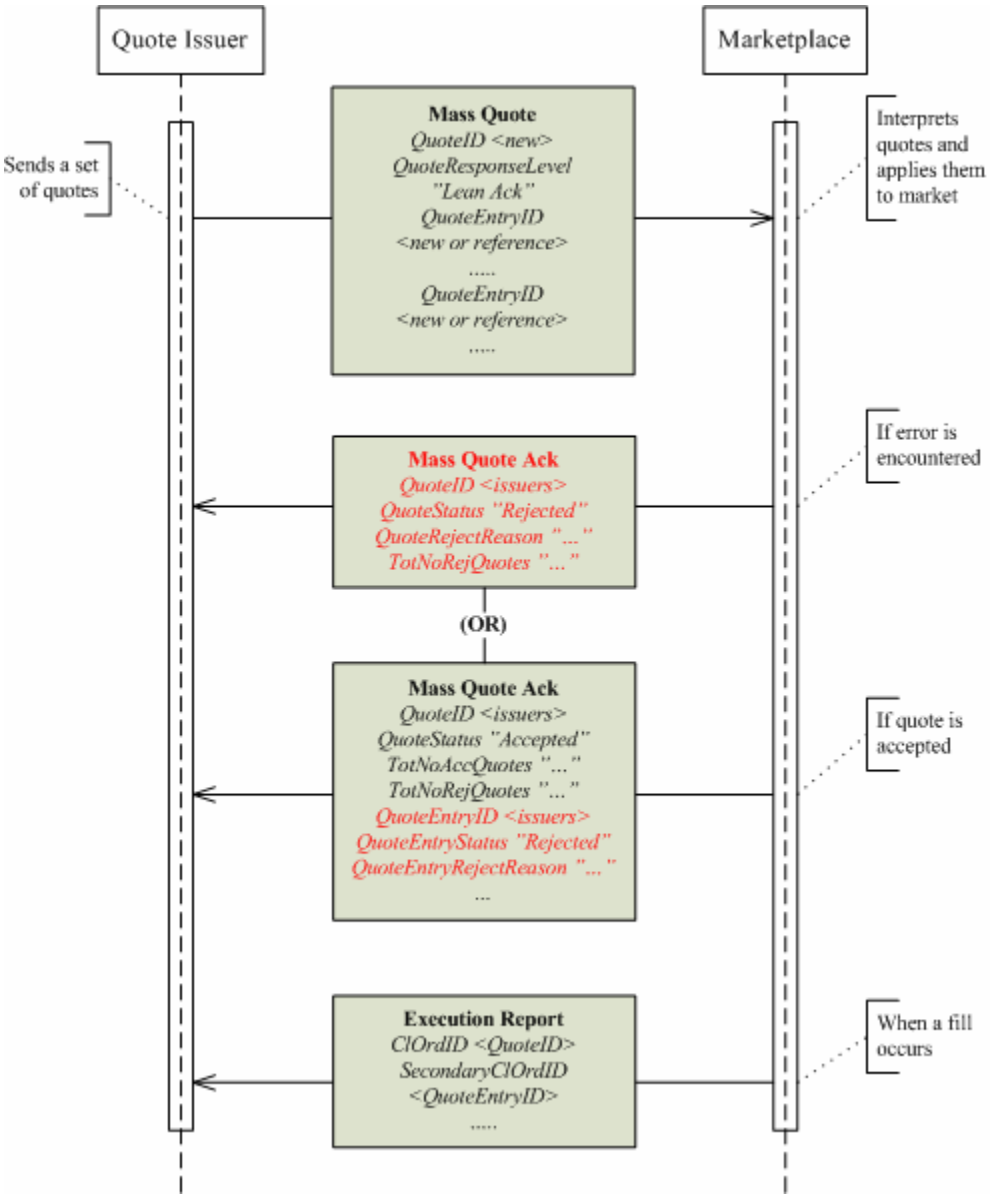
Figure 15: Mass Quote with QuoteResponseLevel=2



Mass Quote with QuoteResponseLevel = 3 (Summary Ack)

An alternative is to use the Summary Acknowledgement:

Figure 16: Mass Quote with QuoteResponseLevel=3

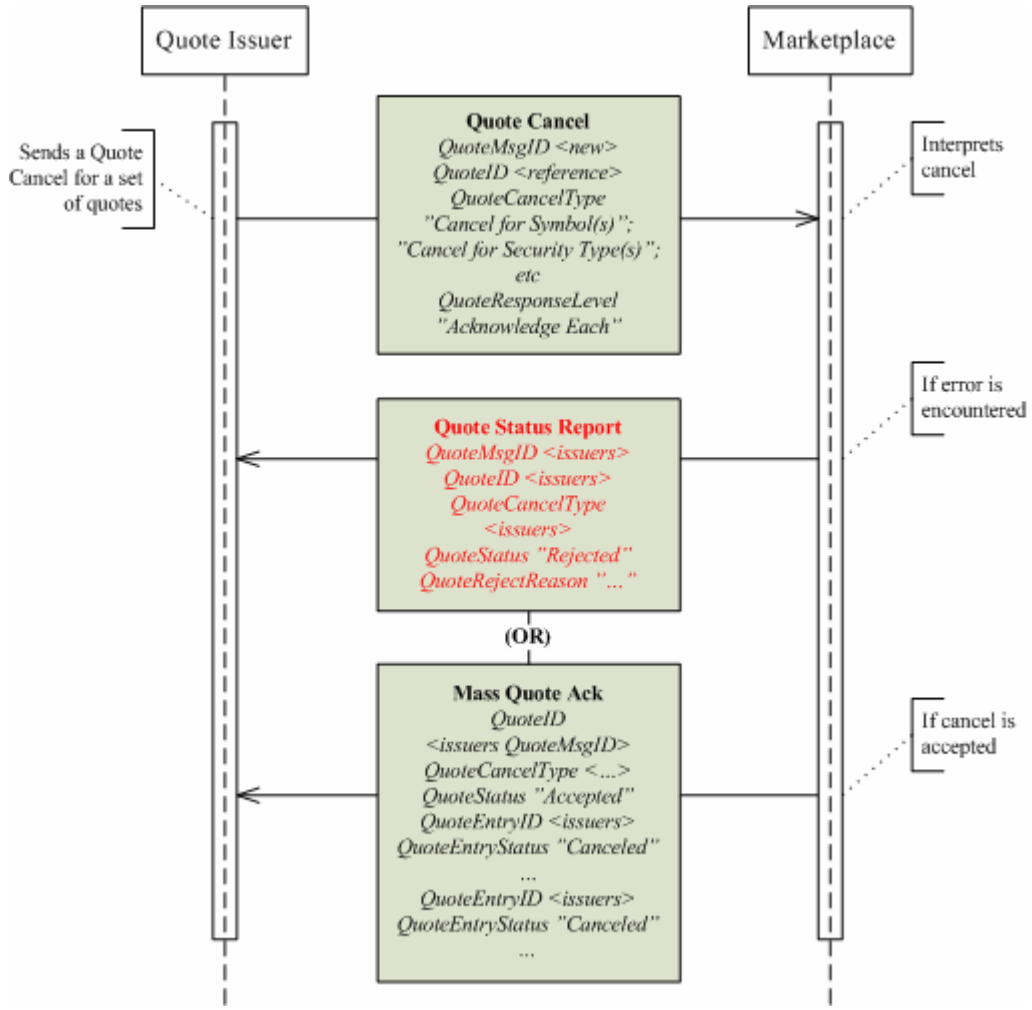


© Copyright, 2007-2008, FIX Protocol, Limited

Mass Quote Cancel

In this example a Quote Cancel identifying multiple quotes is sent.

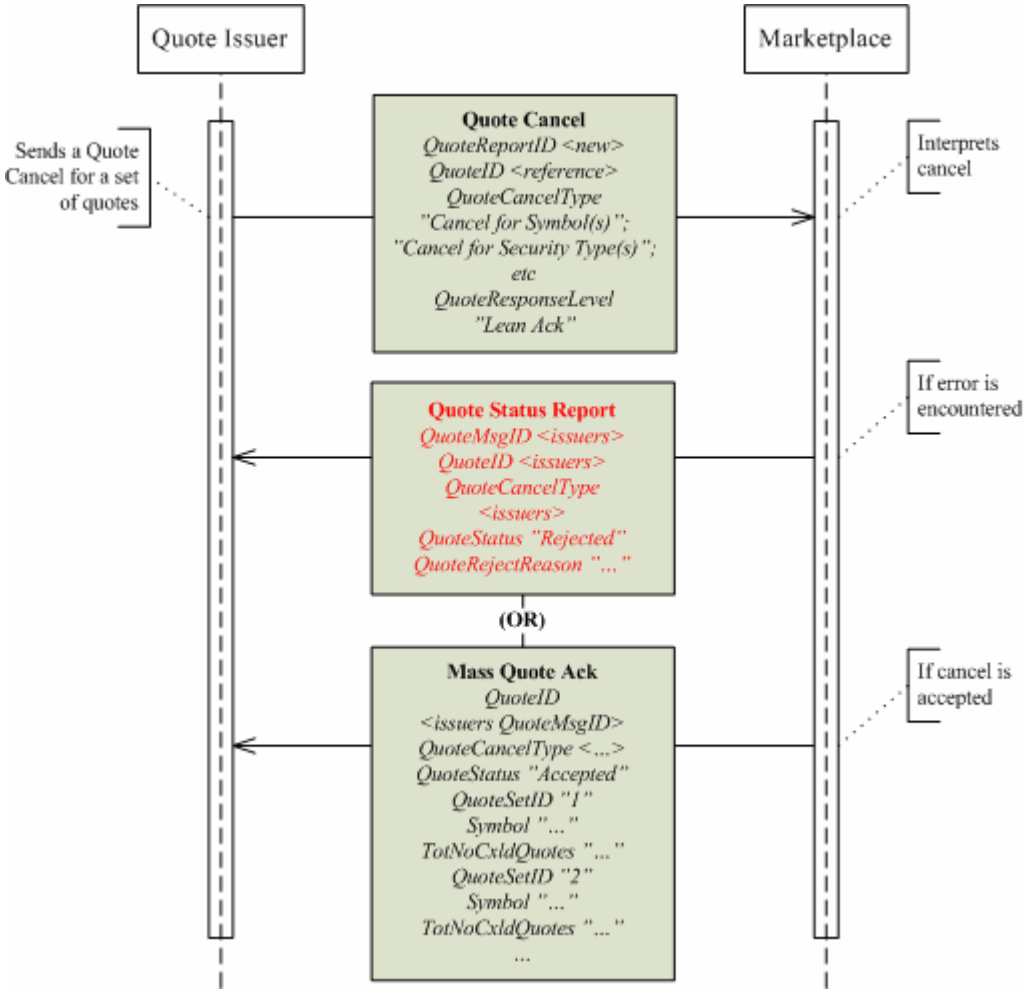
Figure 17: Mass Quote Cancel



© Copyright, 2007-2008, FIX Protocol, Limited

The mass cancel can also use the Summary Acknowledgement QuoteResponseLevel as shown below:

Figure 18: Mass Quote Cancel with QuoteResponseLevel=3

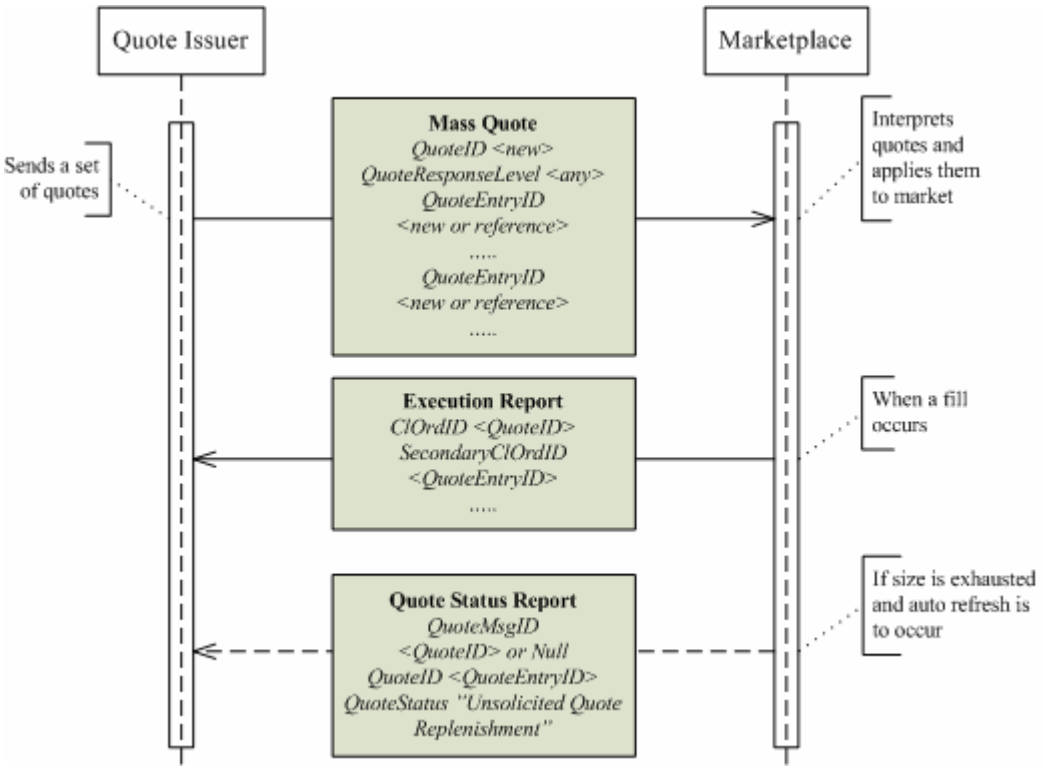


© Copyright, 2007-2008, FIX Protocol, Limited

Unsolicited Actions - Mass Quote Restatement

Some marketplaces, when the quote size is exhausted, support the automatic replenishment with a pre-defined quantity (and moving the price). In such cases a restatement of the quote is appropriate.

Figure 19: Mass Quote Restatement

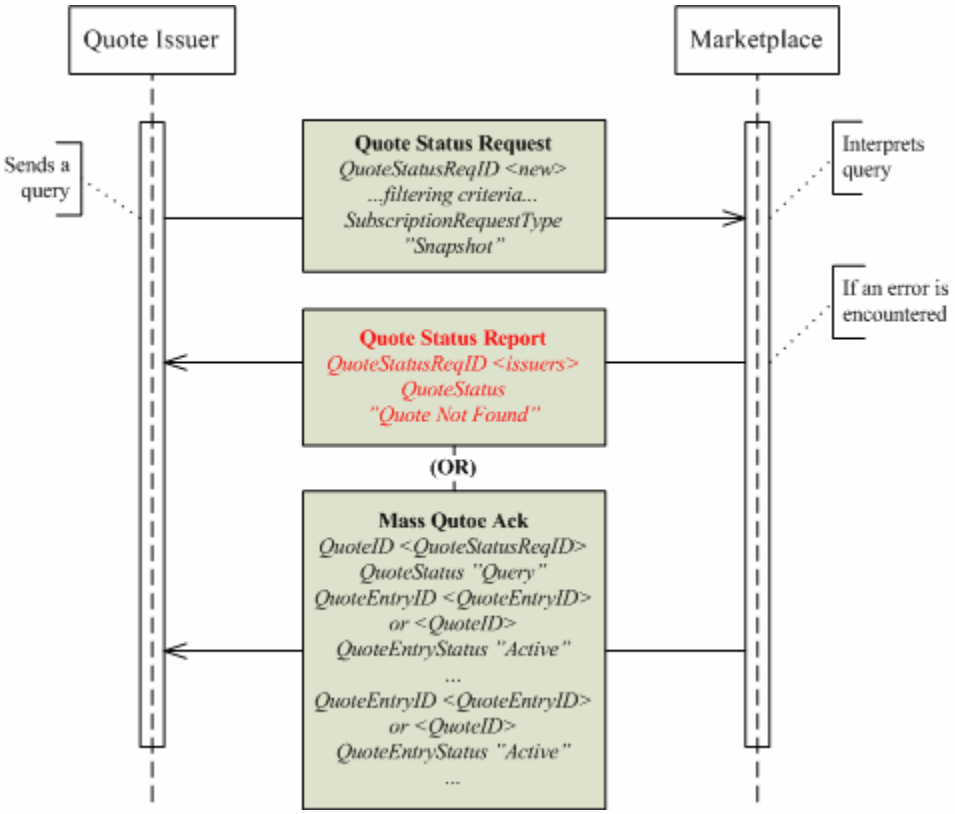


© Copyright, 2007-2008, FIX Protocol, Limited

Query for Mass Quote

In the last example a Quote Status Request identifying a set of quotes is sent.

Figure 20: Mass Quote Query



Quote Negotiation

Introduction

Some marketplaces choose to provide services helping market participants privately negotiate trades. Quote Requests are commonly used for this purpose. Marketplaces may also use Quote Requests in order to solicit tradable quotes in securities that do no have continuous prices, high liquidity or lean spreads.

The support provided by FIX 5.0, exemplified in the Fixed Income and Foreign Exchange appendixes of volume 7 are generally very comprehensive. Marketplaces may, however, choose to expand the use of those models to a three party environment using a marketplace as an intermediary.

This part of the document does not include the indicative negotiation model where:

- A reply with a directed Order (previously indicated/quoted) and the Market Maker may accept or decline.
- The quote may be in the market or a response to a Quote Request.

Usage Notes

Request for a Public or Private Quote

Some markets allow private quote negotiations where that the market maker responding to a Quote Request provides a Quote for the requestor only (a Private Quote, also known as Directed Quote). Markets may also allow the requestor to explicitly state whether he expects a directed or a public response. A public response is a normal quote relayed as market data to all eligible parties and can be hit by other orders and/or quotes according to normal market rules, whereas a directed Quote is visible to and can be hit by the requestor only.

Unless bilaterally agreed, the Quote Request specifies whether the request is private or public by using the PrivateQuote (1171) indicator:

- “Y” = The negotiation is private, therefore the Quote should only be published to the requestor
- “N” (default value) = The negotiation is public,therefore the Quote should be published as market data, viewable by market data subscribers

Directed Quote Requests

An intermediary like a marketplace could allow Quote Request initiators to direct the request to a single or specific group of counterparties. Examples include:

- All eligible market participants. This instructs the marketplace to route the Quote Request to all market participants eligible for such, including, but not limited to, market makers.
- Specified market participants. This implies that a list of participants is provided and that the marketplace should route the request to them only.
- All eligible market makers in the respective securities. This instructs the marketplace to route the Quote Request to all market makers eligible for Quote Requests.
- The primary market maker(s). Instructs the marketplace to route to specialists / primary market makers / designated market makers (or whatever other similar terminology is used).

When the quote request is directed, the marketplace can choose to push the messages to the relevant actors whether they subscribe (RFQ Request) to Quote Requests or not.

In order to support directed quote request, the RespondentType (1172) indicator must be specified

The RootParties component block can be used to list named receivers of the Quote Request. The following PartyRoles are relevant:

- 17 = Contra Firm

© Copyright, 2007-2008, FIX Protocol, Limited

- 37 = Contra Trader

Pre-trade Anonymity

A marketplace could allow the initiator of a quote negotiation to be anonymous when the Quote Request is sent to the respondent(s). This may be handy when there is a risk the respondent may respond with a worse quote upon knowing who the initiator is. Note, however, that the rest of the negotiation process does not support anonymity.

In order to support anonymous quote requests, the PreTradeAnonymity (1091) indicator is used.

Minimum Executable Quantity

When a Respondent quotes a price to specific counterparties, the Respondent may choose to provide a better price under the condition that a certain size is filled. In a Quote Negotiation situation, parties may need to indicate a minimum execution quantity in order to solicit relevant prices and, in the case of the respondent, avoid getting hit on lower than expected quantity. In cases where the Three-Party Matching model (decribed below) is used, a minimum quantity is especially relevant.

The MinQty (110) field is used to specify a minimum executable quantity in a quote negotiation.

Finalising a Quote Negotiation

The Three-Party Matching Negotiation Model

When the marketplace acts as an intermediary in a private quote negotiation, it may want to control when the trade is created. It would regard a trade to occur when both parties have issued matching firm commitments, for example:

- One party issues a firm quote, the other a “Hit / Lift” Quote Response (both within time limits provided in the negotiation process and specifying matching conditions).
- One party issues a firm “Counter” Quote Response and the other a “Hit / Lift” one (both within time limits provided in the negotiation process and specifying matching conditions).

The marketplace would also automatically terminate the negotiation when specified time limits expires.

Any usage of marketplace matching must be bilaterally agreed.

The Trade-Reporting Model

In the trade reporting model the negotiation is finalized by the parties moving to the privately negotiated trade workflows. When the parties through exchanging Quote Requests, Quote’s and Quote Responses have agreed on the terms, they conclude the trade by exchanging Trade Capture Reports exactly as when reporting any other privately negotiated trade.

Refer to TRADE CAPTURE (“STREETSIDE”) REPORTING in volume 5 of the specification.

Other Models

Other models can also be used such as the ones described in the FIXED INCOME and FOREIGN EXCHANGE sections of this volume.

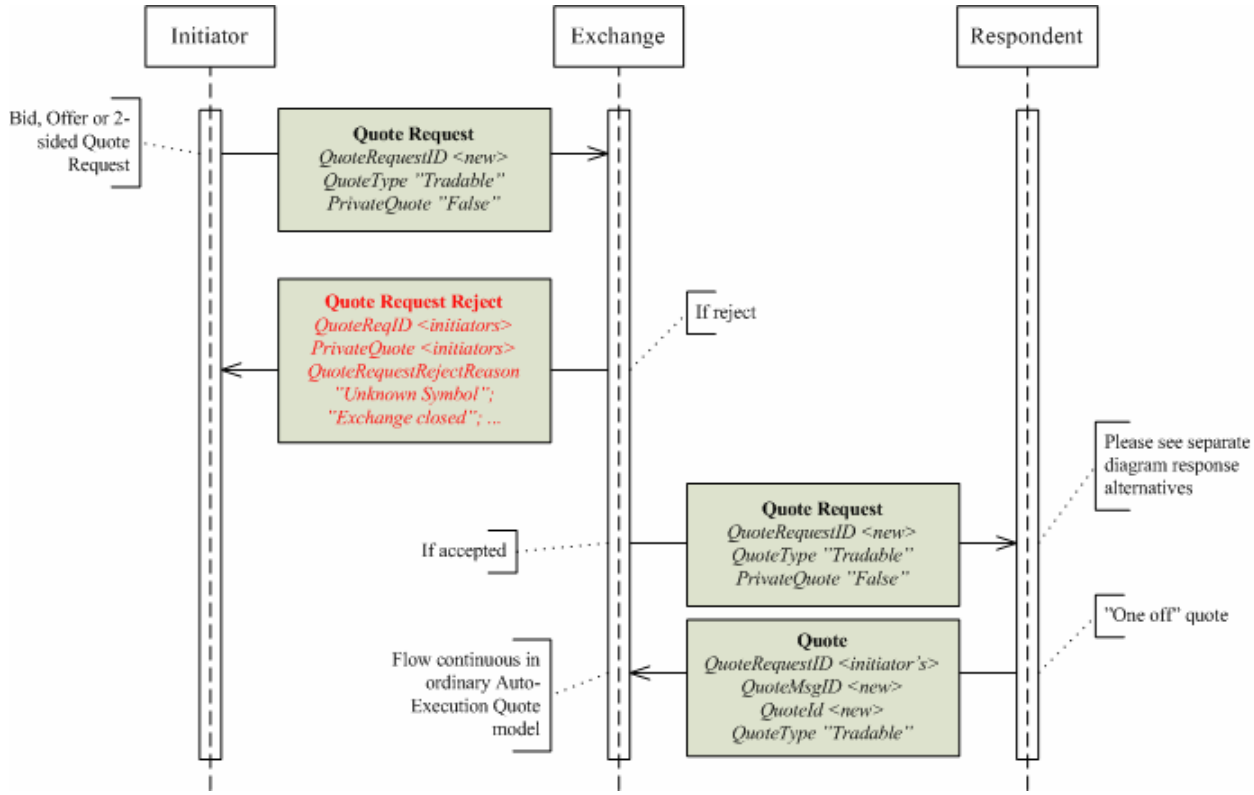
© Copyright, 2007-2008, FIX Protocol, Limited

Quote Negotiation Scenarios

Public Tradable Quote

Vanilla

Figure 21: Vanilla Public Tradable Quote



Private Tradeable Quote - Three-Party Matching Model

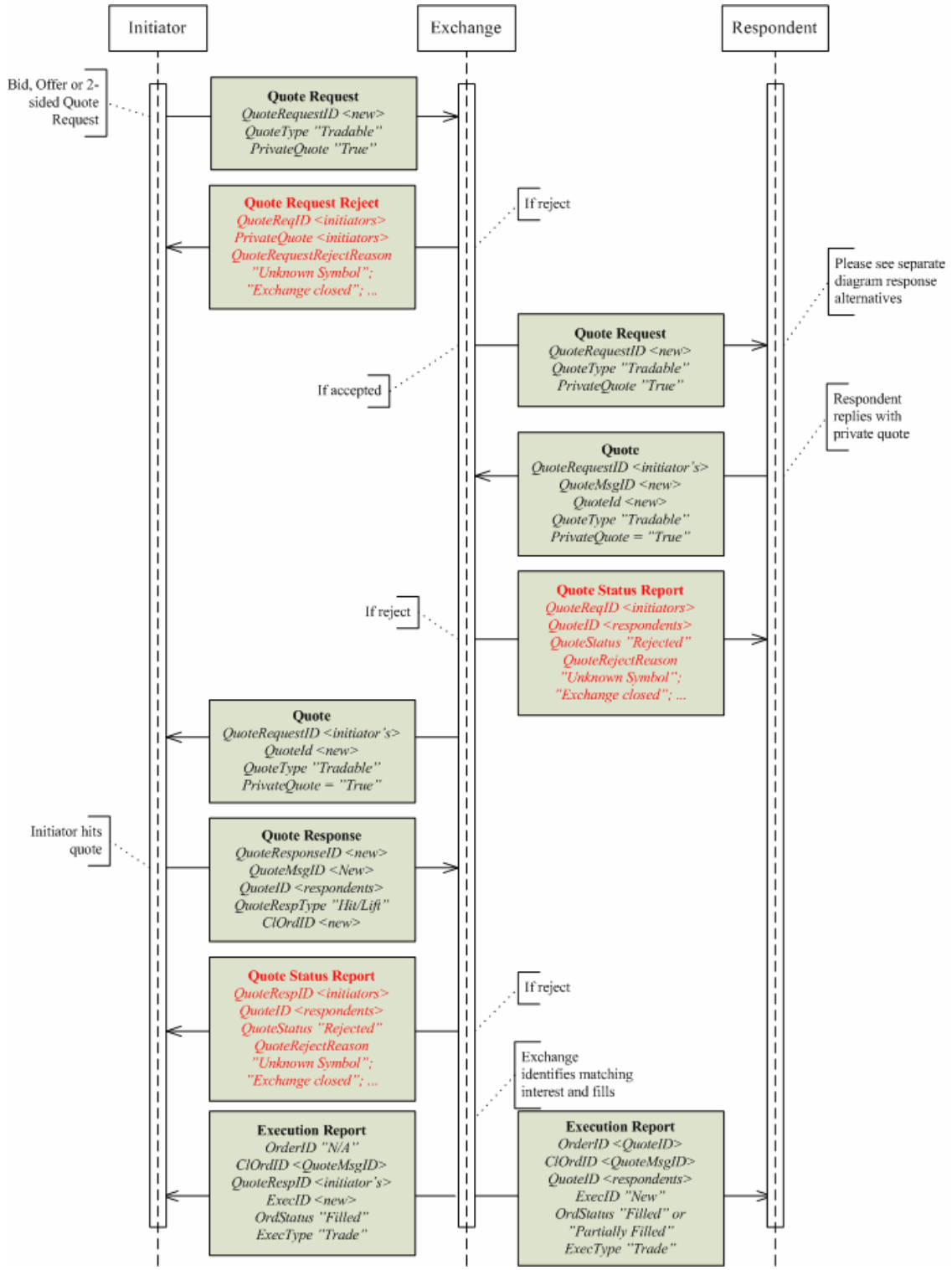
Introduction

The Three-Party matching model is a model where a third party, often a marketplace, supports Quote negotiations including execution. Execution is automated when the terms of a quote and a quote response matches. Such terms minimally includes price and quantity condition, a requirement is also that the Quote and Quote Response have not expired.

© Copyright, 2007-2008, FIX Protocol, Limited

Vanilla

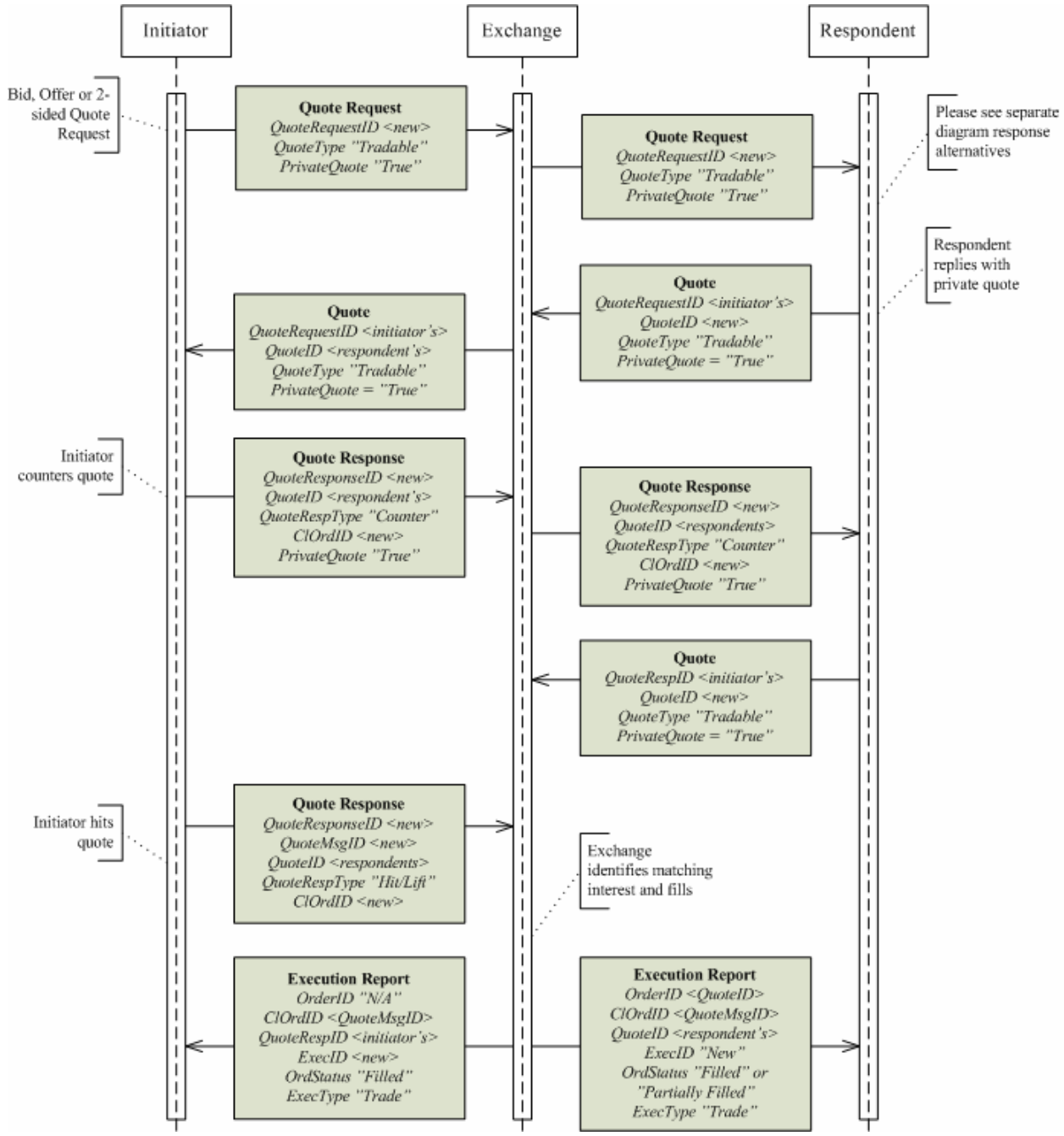
Figure 22: Private Quote



© Copyright, 2007-2008, FIX Protocol, Limited

Initiator Counters Respondent's Quote

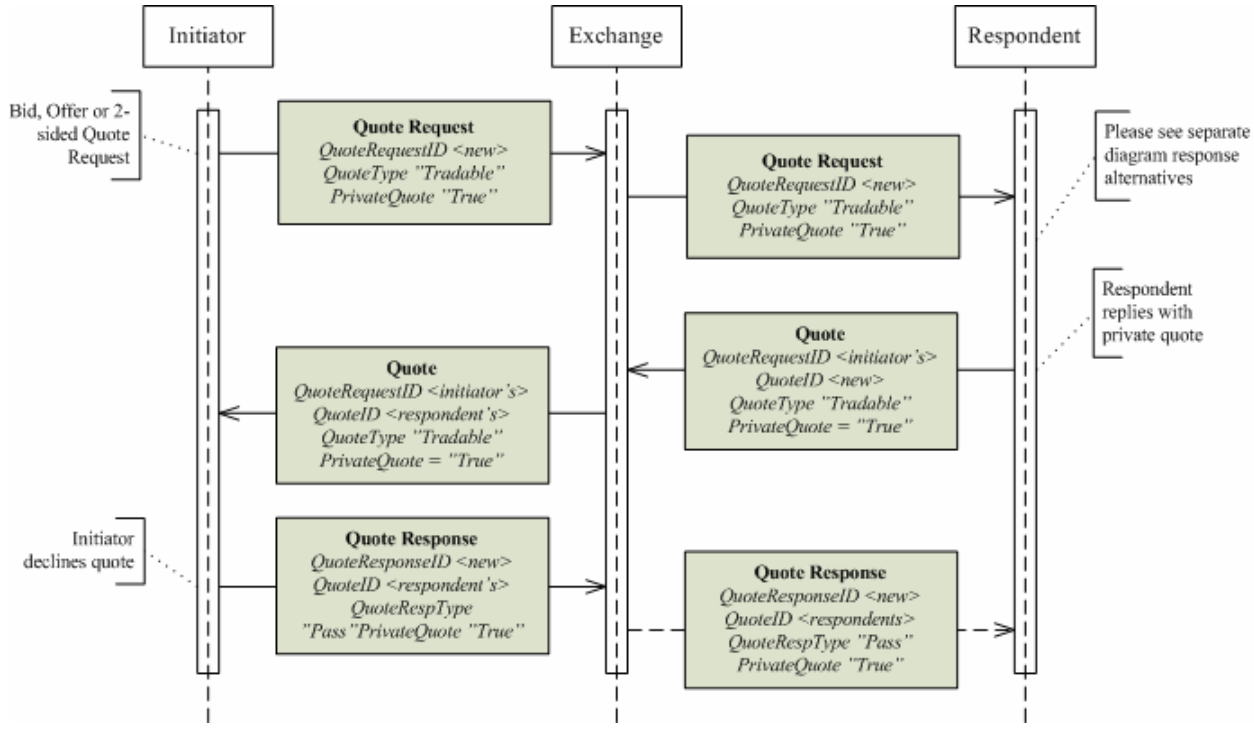
Figure 23: Intiator Counters Respondent's Quote



© Copyright, 2007-2008, FIX Protocol, Limited

Initiator Declines Respondent's Quote

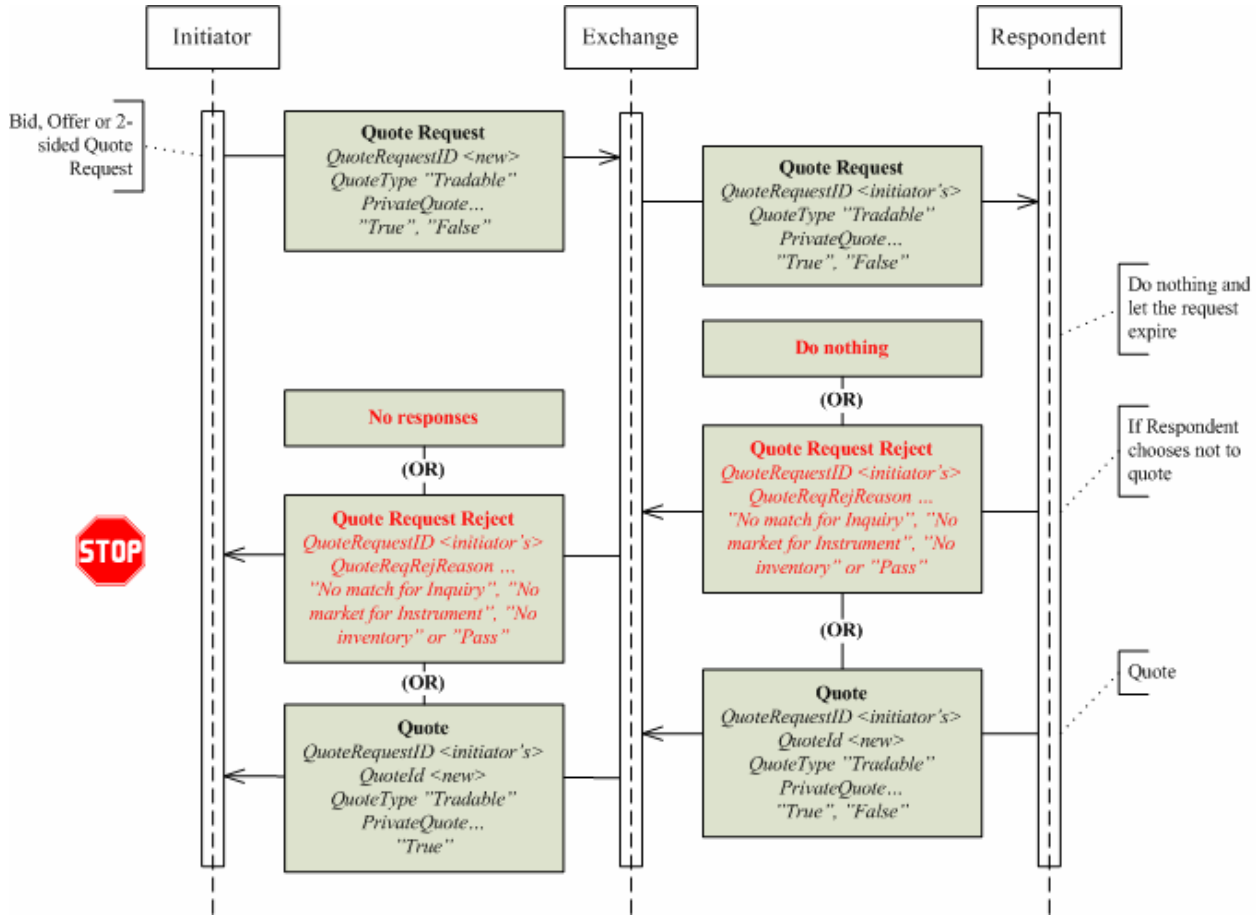
Figure 24: Initiator Declines Respondent's Quote



© Copyright, 2007-2008, FIX Protocol, Limited

Respondent's Alternative Responses to Quote Request

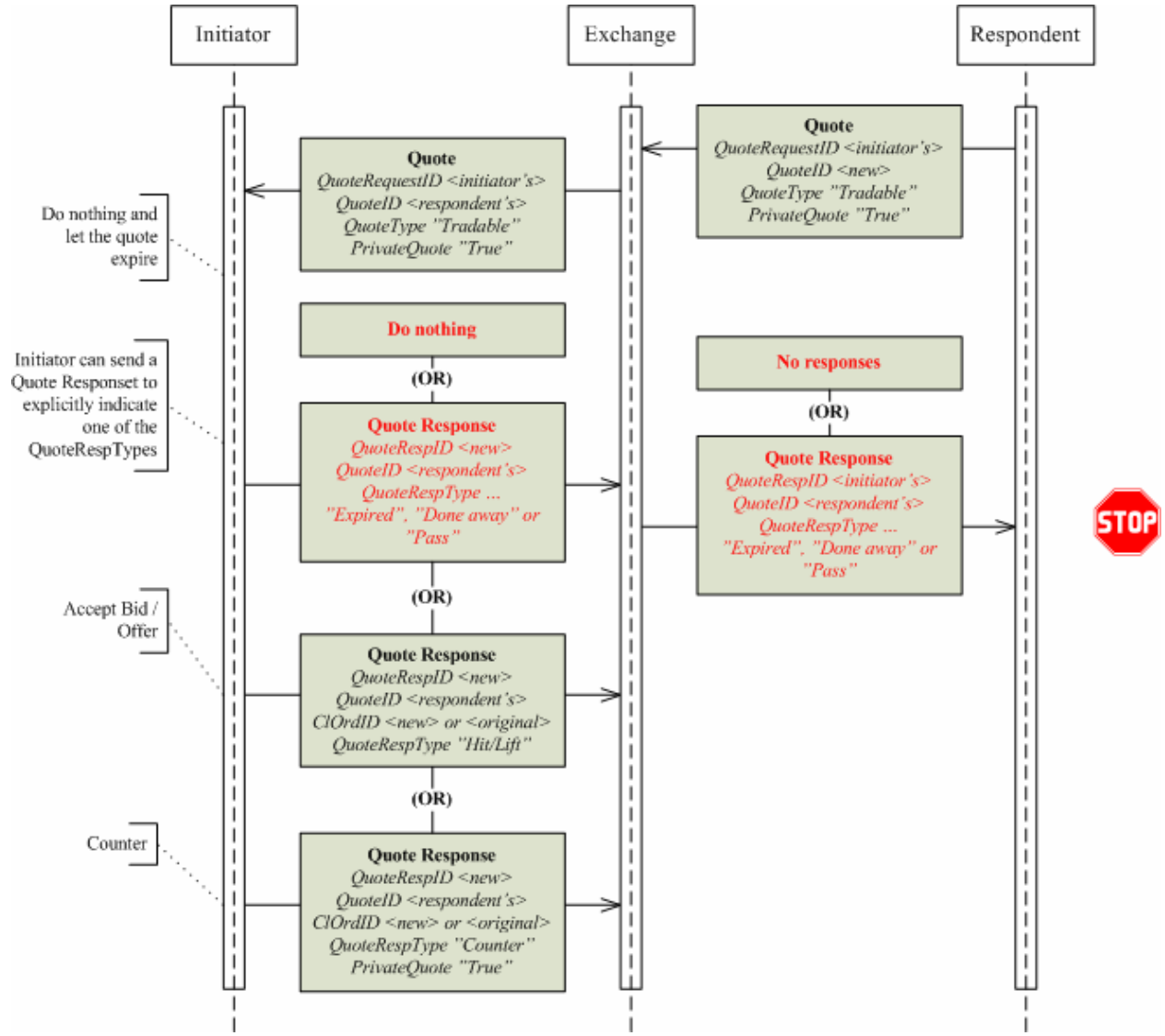
Figure 25: Respondent's Alternative Responses to Quote Request



Note that the Quote will be relayed to the initiator only in the case of a private quote; otherwise it will be published as market data.

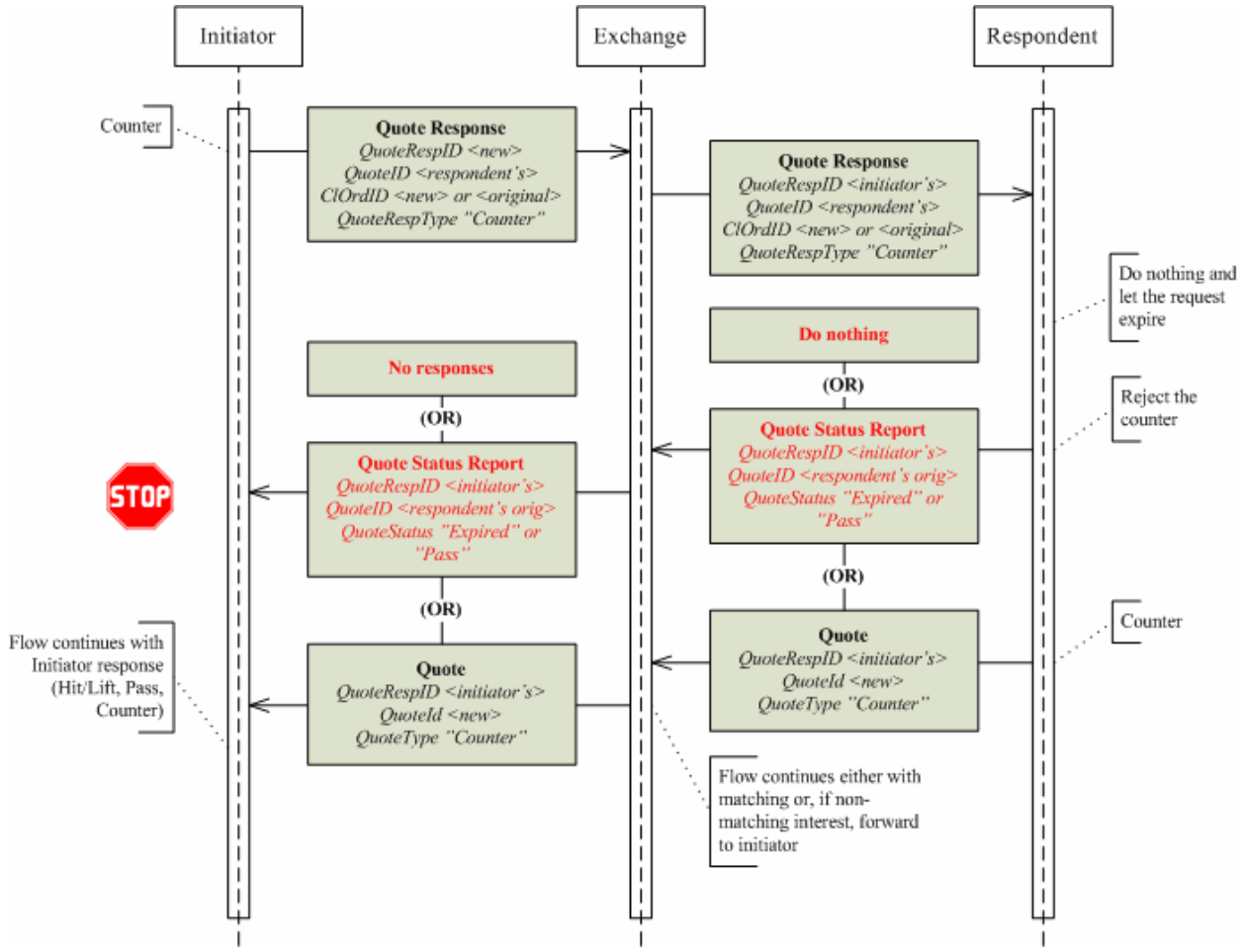
Initiator's Alternative Responses to (Private) Quote

Figure 26: Initiator's Alternative Responses to (Private) Quote



Respondent's Alternative Responses to Quote Response (Counter)

Figure 27: Respondent's Alternative Responses to Quote Response (Counter)



Multileg Orders

User-defined, Non-securitised Strategies

A user-defined non-securitized strategy (a.k.a. free combination) is a multi-leg order for a combination of instruments where there is no pre-defined security in the market place - neither will the marketplace create a security. This type of multi-leg order is more common in equities trading than in derivatives or fixed income as a stock exchange is less likely to securitize multi-legs or show them as separate books over market data. If implied orders are generated in the underlying books (Implied-Out orders), the multi-leg instrument itself does not need to be securitized. In such cases the multi-leg security does not need to be added to reference data or its book be made visible to other participants.

- Some marketplaces implement a variant of the above through creating temporary products for the multi-legs. Those products are then periodically removed, for example, at end of day or week. Those products are, however, generally made available to other brokers and displayed over market data as separate books.

The New Order Multileg requires that the <Instruments> component block root level attribute is specified. As a User-Defined, Non-Securitized Strategy will not refer to a pre-defined instrument and need not be referenced as an instrument, Symbol (55) = "`[N/A]`" (or "`MLEG`") may be used (without the quotes).

© Copyright, 2007-2008, FIX Protocol, Limited

Specifying what Multileg Model applies for an Order

A marketplace supporting more than one of the multi-leg order models (see volume 4 of the FIX specification) needs a clear indicator on the order to govern what model is to be used. This is especially relevant in order to validate that an order request is executed in accordance with the intentions of the user (creating a security / applying the order to an existing one / using a "free combination"). The MultilegModel (1377) field allows the user to define the intent with the order.

The following table depicts the models and their impact on the multi-leg order:

FIX Model	Type of order	Visibility	Comment
Predefined Multileg Security Model (FIX 4.2) (Model 1)	Normal Multileg or Single order	All actors	Uses Security Definition Request message to define the security. Assumingly also used when marketplace defines multi-leg security. The legs are not defined if the multi-leg order is used. (If a user wishes to anchor the price of a leg or otherwise provide details for a leg, he uses the Enhanced Predefined Security Model – see next row).
Enhanced Predefined Security Model (Model 2)	Normal Multileg	All actors	Uses Security Definition Request message to define the security. Same as Model 1, except that legs are defined in the multi-leg order
Product Definition Model using New Order - Multileg Message (Model 3)	Multileg including Security definition data	All actors	Uses New Order Multi-leg message to define the security. The legs are thereby defined in the order.
Single Message Model (Model 4)	Multileg including Security definition data	All actors	Uses New Order Multileg message to define the security. The legs are thereby defined in the order. Differs from Model 3 in that no security is defined, the multi-leg may however be shown as a separate book in Market Data.
Private Multileg Model	Normal Multileg	Initiator only	Does not define a security but uses the New Order Multileg message. Legs are defined. The multi-leg is not relayed to other parties; order is only visible as implied-out prices.

The above table indicates that there are three different types of multi-leg orders:

1) Orders referring to a security that is pre-defined (including the case where the Security Definition Request was used)

© Copyright, 2007-2008, FIX Protocol, Limited

2) Orders that include a request for defining a Security (securitization)

3) Orders that does not refer to an existing security and one should not be created - a “free combination”.

Multileg Price Method

A multi-leg order can be priced in a number of different ways, including:

Net Price. The price is given as the sum of the Price * Ratio for all legs.

- If buying the strategy, the price of a bought leg (which is a buy-leg in the multileg definition) is added, and the price of a sold leg is subtracted.
- If selling the strategy, the price of a bought leg (which is a sell-leg in the multileg definition) is subtracted, and the price of a sold leg is added.

Reversed Net Price. This pricing convention is often used in commodities markets. The price is given as the sum of the Price * Ratio for all legs.

- If buying the strategy, the price of a bought leg (which is a buy-leg in the multileg definition) is subtracted, and the price of a sold leg is added.
- If selling the stategy, the price of a bought leg (which is a sell-leg in the multileg definition) is added, and the price of a sold leg is subtracted.

Yield difference. The price of a strategy order is given as a yield difference between two legs.

Individual Prices. The price of the strategy is given using individual prices for the legs.

Contract Weighted Average Price (Energy Specific). The price of the strategy is given as an average price of all legs in the multileg, including adjustment for differences in contract sizes between the legs.

Multiplied Price (Cross Currency specific). The price is given as the product of the prices for all legs, independent if the leg is bought or sold.

The price given at the main level of a multileg needs to be transformed into prices applicable for the legs. When the legs are of a single asset class or otherwise quoted in a unanimous way, this is straightforward. When the asset classes are different or the legs quoted according to different conventions additional parameters are needed. Users must then indicate the price method in the Multileg order message through using the MultilegPriceMethod (1378) field.

Securitized multi-legs often have the price method defined by the market, product group, or some other means, thus the MultilegPriceMethod need not be specified in the order. However, for non-securitized multi-legs the applicable price method should be specified in the order, subject to marketplace support.

Delta Neutral Multileg Orders

A marketplace may offer delta neutral multileg orders, such as covered calls, provided one of the legs of a multi-leg order is defined to have a calculated quantity instead of a fixed quantity. A trading engine, when receiving such an order, could calculate the quantity of that leg by using the delta values for each leg and the quantities specified in the other legs.

In order to be able to perform delta-calculations, some input parameters are needed for the common option formulas. These parameters are in some cases trivial, such that they can have a single value (strike price, time to expiration, etc), but in other cases far from trivial. Investors also have different opinions on the values of those parameters. The non-trivial parameters can thereby be supplied by the investors with the order as an alternative to the “leg option ratio”. These would be:

- Volatility for the underlying – LegVolatility (1379)
- Dividend Yield - expressed as a percentage of the asset price – LegDividendYield (1381)

© Copyright, 2007-2008, FIX Protocol, Limited

- Riskfree rate – RiskFreeRate (1190)

The parameters also need to be relayed in trade and order ack messages (Execution Report and Trade Capture Report). As those messages can either report per the multi-leg security or per the security of the legs, the security related parameters need to be available for alternative representation at the root level of the respective message and per individual leg.

As a marketplace may provide some of the options parameters as part of executions and trades, and as the user may need to indicate for what leg is subject to a neutral calculation, an indicator is needed. The LegExecInst (1184) field is used for this purpose:

- r - Execute as delta neutral using volatility provided
- s - Execute as duration neutral
- t - Execute as FX neutral

The multi-leg strategies can be existing exchange products or user-defined products. In the latter case either implemented through the registration of a product or simply as an order generating implied orders in the legs.

Market Data

This section discusses the complexities of market data that occurs in some markets and provides recommendations on how to implement FIX in these environments.

Books View Complexities

Lot Types

Certain marketplaces allow a security to be traded in multiple lot types, e.g. odd lot, round lot and block lot sizes. In cases where the various lot types are traded as separate books (no matching integration), separate price feeds or books could be defined. In markets where matching is integrated (a round lot order may for example trade with odd lots), price feeds will also be integrated.

More rarely a marketplace will allow “mixed lots”, i.e. that an order is allowed a size that is not an increment of the lot size. Say e.g. that the round lot is 100 – a user may then be allowed to enter an order for 145 (a mix of one round lot and 45 odd lots). Depending on market rules for round/odd lot integrated matching; the order could also have been traded down to that size. Taking this example a further step, there are some markets that allow the user to decide whether a mixed lot order should be placed in the round lot or odd lot book, i.e. the user defines the lot type of the order. In this last case, the LotType (1093) may have to be explicitly specified in the market data messages.

Subordinate Books

Marketplaces trading securities in different lot types stipulate rules for how orders may trade between these lot types. In some markets lot types are integrated whereas other markets keep lot types separated. This is commonly referred to as integrated vs non-integrated matching.

In the non-integrated matching model, orders are usually received for one order book and the marketplace determines the appropriate lot type. This is done mainly to hide complexity of selecting the appropriate lot type for clients; the order flow will still be distributed for just one order book. However, since they are not integrated the marketplace will have to rank orders of different lot types independently. It would make no sense to provide a common ranking since the orders are not allowed to trade between lot types.

This constitutes a problem in pre 5.0 SP1 FIX versions, as the messages did not facilitate the non integrated matching. At that time only MDPriceLevel (1023) and MDEntryPositionNo (290) could be used to identify the position/ranking of an order. Round Lot, Odd Lot and Block Lot would all have separate ranking and individual

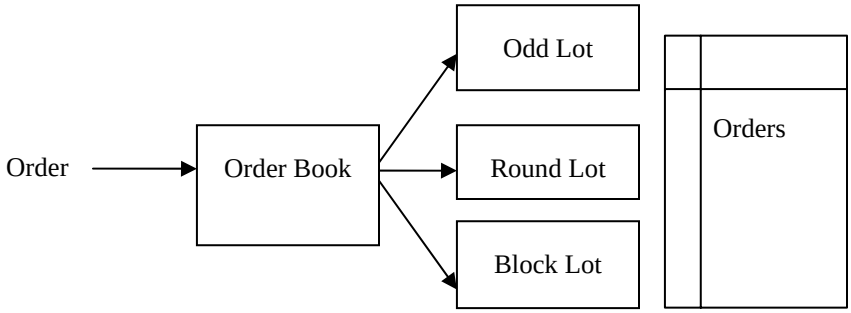
© Copyright, 2007-2008, FIX Protocol, Limited

MDEntryPosition per price level. Even if the LotType (1093) field is added to communicate the lot size, clients would not necessarily know if the market is operating a non-integrated or integrated matching model. Hence, a client cannot rely on just MDPriceLevel and MDEntryPositionNo to sort orders.

In order to divulge the sorting/ranking of orders when this cannot be derived from MDPriceLevel and MDEntryPosition, the MDSubBookType (tbd) field is added to market data messages. The field is optional and FIX clients in those complex environments must always consider MDSubBookType, MDPriceLevel, and MDEntryPositionNo to be able to sort orders and quotes accordingly.

Example Integrated Matching

Consider a market place with integrated matching of lot types. Since the orders are tradable between lot types, they have to be ranked and sorted accordingly. The complexity of selecting Lot Type would be hidden for the Client.



The following orders entered with no previous orders on the book.

Order 1: Id=1, Price=10.00, Quantity=13, Side=Bid

Order 2: Id=2, Price=10.00, Quantity=100, Side=Bid

Order 3: Id=3, Price=10.00, Quantity=1000, Side=Bid

Order 4: Id=4, Price=9.00, Quantity=200, Side=Bid

Order 5: Id=5, Price=10.00, Quantity=100, Side=Bid

With integrated matching the orders of different lot types will be ranked and sorted jointly.

Price Level	Position	Order Id	Price	Quantity	Lot Type
1	1	1	10.00	13	Odd Lot
1	2	2	10.00	100	Round Lot
1	3	3	10.00	1000	Block Lot
1	4	5	10.00	100	Round Lot
2	1	4	9.00	200	Round Lot

As a result the sequence of new orders on the MarketDataIncrementalRefresh will be:

MDEntryId=1, MDPriceLevel=1, MDEntryPositionNo=1, MDEntryPx=10.00, MDEntrySize=13

MDEntryId=2, MDPriceLevel=1, MDEntryPositionNo=2, MDEntryPx=10.00, MDEntrySize=100

MDEntryId=3, MDPriceLevel=1, MDEntryPositionNo=3, MDEntryPx=10.00, MDEntrySize=1000

© Copyright, 2007-2008, FIX Protocol, Limited

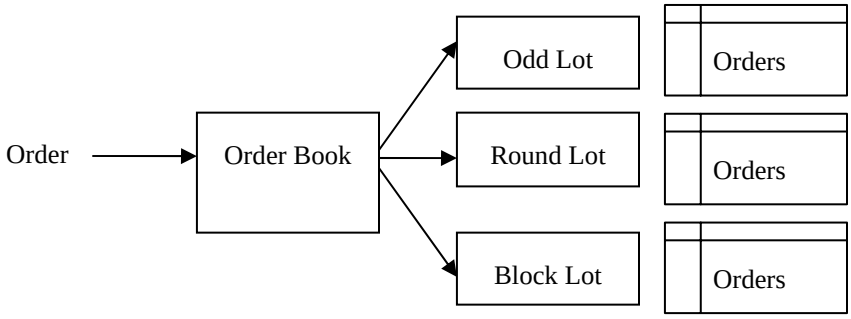
MDEntryId=4, MDPriceLevel=2, MDEntryPositionNo=1, MDEntryPx=9.00, MDEntrySize=100

MDEntryId=5, MDPriceLevel=1, MDEntryPositionNo=4, MDEntryPx=10.00, MDEntrySize=100

The Client will be able to sort orders correctly and build a corresponding order book copy.

Example Non-Integrated Matching

Now consider the same example as before, but the exchange has defined a non-integrated matching mode. This means that orders will not be able to match between lot types, and must be ranked and sorted independently. One can view this as having separate virtual order books. The complexity of selecting Lot Type and thus virtual order book would be hidden for the Client.



Consider the following orders entered with no previous orders on the book.

Order 1: Id=1, Price=10.00, Quantity=13, Side=Bid

Order 2: Id=2, Price=10.00, Quantity=100, Side=Bid

Order 3: Id=3, Price=10.00, Quantity=1000, Side=Bid

Order 4: Id=4, Price=9.00, Quantity=200, Side=Bid

Order 5: Id=5, Price=10.00, Quantity=100, Side=Bid

With integrated matching the orders of different lot types will be ranked and sorted independently.

Odd Lots

Price Level	Position	Order Id	Price	Quantity	Lot Type
1	1	1	10.00	13	Odd Lot

Round Lots

Price Level	Position	Order Id	Price	Quantity	Lot Type
1	1	2	10.00	100	Round Lot
1	2	5	10.00	100	Round Lot
2	1	4	9.00	200	Round Lot

Block Lots

Price Level	Position	Order Id	Price	Quantity	Lot Type
1	1	3	10.00	1000	Block Lot

With the pre 5.0 SP1 implementation of the protocol, it would result in the following sequence of new orders on the MarketDataIncrementalRefresh:

© Copyright, 2007-2008, FIX Protocol, Limited

MDEntryId=1, MDPriceLevel=1, MDEntryPositionNo=1, MDEntryPx=10.00, MDEntrySize=13

MDEntryId=2, MDPriceLevel=1, MDEntryPositionNo=1, MDEntryPx=10.00, MDEntrySize=100

MDEntryId=3, MDPriceLevel=1, MDEntryPositionNo=1, MDEntryPx=10.00, MDEntrySize=1000

MDEntryId=4, MDPriceLevel=2, MDEntryPositionNo=1, MDEntryPx=9.00, MDEntrySize=100

MDEntryId=5, MDPriceLevel=1, MDEntryPositionNo=2, MDEntryPx=10.00, MDEntrySize=100

The Client would not be able to decipher this and build a proper order book copy. Based on the information received and implicit push operations it would look as following:

Price Level	Position	Order Id	Price	Quantity	Lot Type
1	1	3	10.00	1000	Block Lot
1	2	5	10.00	100	Round Lot
1	3	2	10.00	100	Round Lot
1	4	1	10.00	13	Odd Lot
2	1	4	9.00	100	Round Lot

However, with the additional field of MDSubBookType (tbd), clients are able to separate the lots, and sort the orders correctly.

The sequence would then be:

MDEntryId=1, MDSubBookType=OL, MDPriceLevel=1, MDEntryPositionNo=1 …

MDEntryId=2, MDSubBookType =RL, MDPriceLevel=1, MDEntryPositionNo=1 …

MDEntryId=3, MDSubBookType =BL, MDPriceLevel=1, MDEntryPositionNo=1 …

MDEntryId=4, MDSubBookType =RL, MDPriceLevel=2, MDEntryPositionNo=1 …

MDEntryId=5, MDSubBookType =RL, MDPriceLevel=1, MDEntryPositionNo=2 …

Clients can now easily offset the position on MDSubBookType.

Adding Lot Type to message

Please note that the addition of Lot Type would not by itself assist clients in building a proper order book copy.

Consider the first example and adding lot type to the message

MDEntryId=1, MDPriceLevel=1, MDEntryPositionNo=1, MDLotType=Odd Lot …

MDEntryId=2, MDPriceLevel=1, MDEntryPositionNo=2, MDLotType=Round Lot …

MDEntryId=3, MDPriceLevel=1, MDEntryPositionNo=3, MDLotType=Block Lot …

MDEntryId=4, MDPriceLevel=2, MDEntryPositionNo=1, MDLotType=Round Lot …

MDEntryId=5, MDPriceLevel=1, MDEntryPositionNo=4, MDLotType=Round Lot …

This would provide information to the Client about lot type, but not necessarily how orders are ranked. Lot Type identifies a lot type of an order and has potentially nothing to do with how orders are sorted and ranked.

By adding the MDSubBookType to the message, Clients know how to sort the orders. In the first example with integrated matching all lots will end up in the round lot book. Hence, if the MDSubBookType is provided, the message flow would be as following:

MDEntryId=1, MDSubBookType =RL, MDPriceLevel=1, MDEntryPositionNo=1, MDLotType=Odd Lot …

© Copyright, 2007-2008, FIX Protocol, Limited

MDEntryId=2, MDSubBookType =RL, MDPriceLevel=1, MDEntryPositionNo=2, MDLotType=Round Lot …

MDEntryId=3, MDSubBookType =RL, MDPriceLevel=1, MDEntryPositionNo=3, MDLotType=Block Lot …

MDEntryId=4, MDSubBookType =RL, MDPriceLevel=2, MDEntryPositionNo=1, MDLotType=Round Lot …

MDEntryId=5, MDSubBookType =RL, MDPriceLevel=1, MDEntryPositionNo=4, MDLotType=Round Lot …

This would let the client know how to interpret the market data flow. Note the field MDSubBookType is optional and possibly only used when there are more than one ranking scheme for each order book.

Conveying Execution Restrictions for a Book Entry

Many markets support orders that, if integrated in one book view, would show a market with crossed or locked prices. Such orders include:

- Odd and round lot orders (if matching is not integrated),
- AON, MinQty and other types of orders with restrictions on executable quantity,
- Indicative quotes or
- Orders specifying acceptable counterparties.

To help market data receivers to produce uncrossed book views, those orders must be categorized in some way. While some markets choose to show those orders in separate books, others leave the display decision to the user who can choose to view the orders separated, color coded, etc.

A Market could e.g. choose to publish orders with quantity restrictions as a separate price level (MDPriceLevel). This is especially relevant in markets where order depth market data is relayed. However, a marketplace might also want to include them with other orders and let users decide how they are displayed. In this latter case, recommended practice is:

- In Top-of-Book or Price Depth modes; use separate MDPriceLevel and, when relevant, specify QuoteCondition (276).
- In Order Depth mode: specify MinQty (110) and QuoteCondition (276) whenever relevant
- Use MDQuoteType (1070) = 0 (Indicative) for non-tradable quotes.

FIX provides the QuoteConditon (276) tag to indicate various conditions related to the price. The recommendation is to use QuoteCondition (276) = F (Crossed) in cases where the book is crossed or to indicate an order that may cause the book to look crossed or locked.

Note that when using the Market Data Incremental Refresh messages, the book might not be in a crossed (or locked) situation when an entry with the discussed QuoteCondition is published. The entry may however cause this situation when additional orders are received.

FIX also support the following quantity restrictions applicable for orders that can sit on the book:

- An All or None (AON) order is indicated using ExecInst (18) = G.
- Minimum quantity is expressed using the MinQty (110) tag.
- Match increment is expressed using the MatchIncrement (1089) tag.

The potential use of the following instructions in call auctions (when such orders could need pre-trade publication) is not considered particularly relevant for disclosure in Top-of-Book or Price Depth modes:

- Immediate or Cancel is indicated using TimeInForce (59) = 3
- Fill or Kill is indicated using TimeInForce (59) = 4

© Copyright, 2007-2008, FIX Protocol, Limited

Customer vs Principal Size

Some marketplaces rank and execute customer orders ahead of principal interest at each price level – or, more rarely, the other way around. Such markets may want to relay the size of the customer (or principal) interest together with the totally available size at a price level of the book. The feature applies to Top-of-Book and Price Depth feeds (not to Order Depth where other mechanisms show ranking).

The book would be displayed e.g. as in the below table, showing the customer size within parenthesis:

100 (20) @ 9.95
150 (100) @ 9.90
200 (85) @ 9.85

Other markets have separate ranking for more than two “customer capacities”, e.g. distinguish between orders for: customer, agency, principal and firm.

In order to support the relaying of both a total quantity and customer size, the following repeatable structure can be used:

- NoOfSecSizes (1177)
  - MDSecSizeType (1178)
  - MDSecSize (1179).

VWAP for not displayed part of Book

Some markets have the convention of, when publishing Price Depth books (say five levels), also publishing the VWAP and quantity for the remaining ("unpublished") part of the book. The "unpublished prices" represent "the rest of the book" and may influence users trading decisions. Say we publish three price levels + the rest. User can display that like this (bid side only):

50 @ 10.00

80 @ 9.90

95 @ 9.80

1500 @ 9.47 (VWAP)

The last row represents the VWAP for the remaining ("unpublished") orders of book. It will have to be published per side (MDEntryType = 0/1). That extra Market Data entry is indicated through QuoteCondition (276) = tbd - Rest of Book VWAP and is not included in the MarketDepth – if MarketDepth = 3, it is an additional entry with no specified price level.

Example of a “rest of book” MD Entry:

- MDBookType = 2 (Price Depth)
- MDFeedType = xxx
- TradeDate = 2007-03-06
- MDUpdateAction = 1 (Change)
- MDEntryType = 0 (Bid)
- MDEntryID = xxx
- Instrument = ...
- MDEntryPx = 9.47
- MDEntrySize = 1500

© Copyright, 2007-2008, FIX Protocol, Limited

- QuoteCondition = 3 (Rest of book VWAP)
- NumberOfOrders = xxx

Better Prices in Condition Orders

When publishing Top-of-Book only, there might still be orders out there with better prices but with conditions that exclude them from the Top-of-Book (e.g. AON-orders). A market may want to indicate this fact and the users may choose to display it as e.g.:

50 @ 10 *

Where the star (*), or whatever symbol you want to use, could indicate that there are better prices provided the conditions are right.

The Top-of-Book entry can in those cases be associated with a QuoteCondition (276) = tbd - Better Prices in Conditional Orders.

Example of a top-of-book entry indicating that there are conditional order(s) with better prices:

- MDBookType = 1 (Top of Book)
- MDFeedType = xxx
- TradeDate = 2007-03-06
- MDUpdateAction = 1 (Change)
- MDEntryType = 0 (Bid)
- MDEntryID = xxx
- Instrument = ...
- MDEntryPx = 10.00
- MDEntrySize = 100
- QuoteCondition = tbd - Better prices in conditional orders
- NumberOfOrders = xxx

Varying Book Depths

Certain markets are associated with a book depth (MDBookType) that varies during the day, e.g.:

- Pre-Open = Price Depth
- Balancing phase = no orders (only imbalance info)
- Continuous trading = Order Depth

Users often receive the depth produced by the market, i.e. they do not have a choice of selecting a certain depth.

A market can also provide a depth view with a number of levels that change depending on how much bandwidth is consumed, e.g. 50 best in low activity situations and 20 best in high ones. As market data receivers are required to know what depth is disseminated, the MD messages need to relay the actual depth used.

In order to relay changes in the market depth within the same book type, MDBookType (1021) and MarketDepth (264) fields can be provided through the Security Status message. The MarketDepth field can also be divulged in market data messages to complement the MDBookType (1021) - the field is used here to relay the current depth of book in markets where the depth can change during a trading session or day.

© Copyright, 2007-2008, FIX Protocol, Limited

MarketDepth is irrelevant for Top-of-Book, but applies to Price Depth and Order Depth data.

Trade Statistics

A marketplace can choose to let Market Data receivers calculate statistics themselves. In markets with more complex rules for what entries are eligible for inclusion in statistics, each market data entry needs an indication showing if it is eligible for a certain type of statistic.

Eligibility indicators can be relayed in the Market Data Incremental Refresh message. They are used to define whether an entry is eligible for inclusion in a piece of statistics or not:

- NoStatsIndicators (1175)
    - StatsType (1176)

Application Sequencing and Recovery

This section discusses the use of the Application Message Request, Application Message Request Ack and Application Message Report messages for recovery of application level messages.

Business Workflow

Application Sequencing within a FIX session

A FIX session may act as the mechanism for distributing a number of "secondary" data sources that are differentiated based on application, message type, or even originating party. Each application is assigned a unique ApplID (1180). Each application assigns its own set of application-level sequence numbers which are then uniquely identified using a combination of ApplID (1180) and ApplSeqNum (1181). Once a FIX session is established the data is distributed and able to be identified by application. The receiver of the data can use the application sequence number to manage resend requests based on sending application ID should the data need to be retransmitted.

When a retransmission of data is necessary the Application Message Request is sent with the necessary application ID and the sequence number range to precisely request the required data. The data will then be resent without any disruption to the FIX session using the original application sequence numbers. Upon receipt of the data the requester will process according to the application id specified.

Additionally, the ability to check for the most recent application sequence of a given application can be done using ApplReqType (1347). The Application Message Request Ack will return the last application sequence number produced by the application.

In order to allow deliberate application-level sequence gaps in the distribution of application data, ApplLastSeqNum (1350) is used report the prior application sequence number for that ApplID (1180). This will allow the receiving application to distinguish purposeful sequence gaps from application errors by comparing the value of ApplLastSeqNum (1350) to the prior application sequence number received for that same ApplID (1180).

Application Seqencing over an alternate protocol

Application sequencing provides a generally useful mechanism for sequencing streams of data. Market Data feeds, which are inherently sequential, sent over a broadcast or multicast transport which do not guarantee ordered delivery, need application sequencing to provide the ordered delivery mechanism. An application generating a stream of data can assign ApplSeqNum (1181) values to each message allowing the Receiver to detect when messages are out of sequence. The Application Message Request will allow a Receiver to recover missed messages. Resend requests and retransmitted messages will most likely take place over a separate back channel in which two-way communication can be conducted.

© Copyright, 2007-2008, FIX Protocol, Limited

Business Cases

Concrete business cases for application sequencing include the following:

1. Data providers need to be sensitive regarding the amount of data they consolidate and send out - particularly in a resend scenario. There needs to be a way to short-cut the potentially massive volumes of data and pinpoint critical business information that is also sent on the same FIX session. It is always possible that some "subscribers" may be receiving data for the first time due to delayed logon or disconnect and are not at all interested in the potentially large volume of market data messages that may be distributed before they can receive their "time and sales" information, for example.
2. Multiple FIX sessions are expensive to establish, maintain and manage both for host and client. This is not a cost effective solution, especially for smaller firms, in distributing different types of application data. The ability to send multiple sets of distinct application data over the same connection provides a way to "leverage" a single FIX session without disrupting the session protocol.
3. FIX Session Resends require that huge amounts of data be stored by the Sender in order to support the ability to resend the messages. Supporting secondary or tertiary sessions requires that the sender store possibly the same data in duplicate or triplicate. If messages are sequenced by the generating application the Sender is able to avoid duplicate data storage by resending the information from the original application store and assigning new session sequence numbers as the data is resent. In this case the Sender is able to avoid costly dual storage and processing associated with ancillary FIX sessions.
4. In the case of drop copy, data can be consolidated across multiple sessions and the original message sequence number can serve as the application sequence number. An extended request for data, if handled as an application request, can be based on the sequence number assigned by the sending application. Given the application ID, the precise set of data can be extracted from the originating session and resent. No additional data store is needed in this case.
5. Application Message Request allows the receiver of data to control what is resent. A receiver may not want, for example, the volumes of market data messages that might be retransmitted after a lengthy disconnect and may only be interested in the trade messages that are interspersed. The Application Message Request would allow the requester to request a resend of only the application messages that they are interested in by specifying previously defined application IDs and the relevant range of application sequence numbers for those IDs.
6. Application sequencing provides a means of ensuring ordered delivery of messages as well as acting as mechanism for data recovery for one-way broadcast-like transports. Application Message Requests should be executed over a back channel in this case.

Application Indentification

An application is identified by the AppID (1180) field. It is recommended that the ID value be assigned based on the business purpose of a stream of messages. The ApplID (1180) can be used very granular or very broad depending on business requirements, but in both cases it has specific meaning both to the sender and receiver of the data. The ApplID (1180) value is either agreed to on a bilateral basis or formally published by the application data provider. In either case, the ApplID (1180) has inherent business meaning and provides a large amount of flexibility in qualifying data over a single FIX session. ApplID (1180) provides a flexible way of labeling data intermixed within a session. The Application Message Request message can be used to request a list of available application IDs from the data provider (providers may implement authorisation controls to disallow this or limit the results).

Application Sequencing integration

© Copyright, 2007-2008, FIX Protocol, Limited

Application sequencing moves the responsibility for recovery from the FIX engine back to the applications which use the FIX engine. This model is one where the Receiver of data is discriminating the type of data when requesting a resend of data. In some cases, it will make sense to automate the recovery of specific applications while other applications are not recovered at all or recovered on a manual basis.

In order to automate application level recovery it will be necessary for a Receiver to detect application level gaps and generate Application Message Requests. The Receiver's internal application will need to know that the Senders's response to its session-level resend request is being suppressed in the form of a GapFill so that it can generate the appropriate application-level requests. This will be new functionality to those that traditionally rely on their black box FIX engine for recovery. However, it will provide an alternative to the session-level's comprehensive recovery model and provide one of Selective Recovery. It should be noted that FIX Session currently makes provision for the "replacement" of application messages with a GapFill - in other words, not resending the requested range of requested messages. In point #2, from the excerpt from the FIX Transport Volume (formerly Volume 2) below, recognizes the use of SequenceReset-GapFill as a valid mechanism for synchronizing the a session.

"Upon receipt of a *Resend Request*, the resender can respond in **one of three ways:**

1. retransmit the requested messages (in order) with the original sequence numbers and *PossDupFlag* set to "Y" except for the administrative messages (listed below) which are not to be resent and which require a *SeqReset-GapFill (#2)*
2. **issue a *SeqReset-GapFill with PossDupFlag set to "Y"* message to replace the retransmission of administrative and application messages**
3. issue a *SeqReset-Reset with PossDupFlag set to "Y"* to force sequence number synchronization"

Users of application sequencing will need to

(1) keep an inventory of all application id's for which data is being received

(2) track application sequence numbers by application id's

(3) detect when a gap has occurred in a given application sequence

(4) make a decision as to whether missing data should be requested

(5) generate the Application Message Request

(6) hold new messages for that application id in abeyance while the request is being processed by the Sender

(7) apply the resent messages and verify that the gap no longer exists.

Note that in some cases Receivers may choose to process newly arriving messages prior to receiving resent messages.

This represents a sizeable new set of functionality without disrupting the existing FIX engine, while allowing the Receiver to control the data that is resent to them. Receivers will be able to subscribe to those applications they want. However, the Receiver must build some new functionality into their application to make application sequencing work.

Interaction with Session Layer

In order for the selective nature of application identifiers and application level sequencing to be effective it is necessary that messages be gap filled in a normal resend/recovery scenario. In effect, the Sender would always respond to a session-level Resend Request with a Sequence Reset – GapFill for all requested messages. This shifts some of the responsibility of recovery back on the receiver who needs to decide which messages are needed in order to adequately recover. It is no longer a simple question of the last sequence number that was received.

© Copyright, 2007-2008, FIX Protocol, Limited

A Receiver would most likely leverage the benefits of application-level recovery by automating the Application Message Request in the same manner as the traditional FIX session layer's Resend Request message. While the messages are gap filled by the Sender, the next application sequence number received will be higher than expected and the Receiver will issue an Application Message Request automatically.

The fundamental session layer model is left intact but overall control is enhanced by allowing the receiver to determine which messages are resent. It should be noted that application sequencing and resends should not be used over a trading connection where guaranteed ordered delivery of all messages is generally required. Application Sequencing is relevant where the user does not want all data retransmitted - especially when that data is of lesser importance, too stale to be of use, too voluminous to be processed in timely manner or is intermingled with other application data.

Negotiating application sequencing capability at Logon

The use of Application Sequencing and Recovery should be negotiated out of band on a bilateral basis between the affected parties. **It should be noted that either one or both parties may act as an Application Sequencing and Recovery provider if the other side indicates that it is capable of acting as a Receiver.**

Application sequencing gap detection

Gap detection

A Requestor may use ApplSeqNum to ensure that application-level messages for a given ApplID are received in order. If Sending and Receiving parties have agreed that application sequence numbers will be sequential then a gap is defined as a missing or unordered sequence number in the same way that it is for session-level gap detection.

On a reconnect, gap detection can be predicated on receiving the next message for a given ApplID. If the ApplSeqNum is higher than expected when compared to the last message received prior to the disconnection, then an Application Message Request can be issued for the intervening messages. In the case where the next message is not immediately transmitted, the Receiver has the option of (1) issuing an Application Message Request for status in which the most current ApplSeqNum is returned or (2) issuing an Application Message Request in which a message range from the last ApplSeqNum received through “infinity” (represented with a "0", zero) is requested. In this case, the Sender would return messages up through the most current ApplSeqNum.

On a dropped message when the connection remains intact, gap detection takes place as defined above and an Application Message Request is sent to recover messages in the gap. **It is important to note that on a connection using Application Sequencing and Resend, the session-level Resend Request that is generated in response to a dropped message will result in the Sender responding with a Sequence Reset-GapFill and will not be an effective mechanism for filling a gap. This will have been previously arranged during the Application Sequencing Negotiation at Logon.**

Using application and session sequencing for gap detection

In this scenario, a combination of Application and Session level sequencing will be used to recover missed messages. A limited cache of session level messages may be retained by the Sender in order to recover messages that have been dropped within a pre-stated window defined by time or number of messages. When a session resend request is issued within this window the Sender’s session will resend the messages. Once the window has been exceeded an Application Message Request must be issued in order to recover dropped messages. In this case, the application level will not be aware that a gap has occurred until the session level has recovered what it available. Beyond this, the application will detect the gap according to the logic described above and issue a resend request.

Gap detection and recovery with respect to the Application Message Request and Response messages may also need to take place at the application level since session level recovery may have been suspended.

Using ApplLastSeqNum for gap detection

© Copyright, 2007-2008, FIX Protocol, Limited

A Requestor my also use ApplLastSeqNum for gap application-level gap detection. This is done by comparing the most recently received ApplLastSeqNum to the prior ApplSeqNum that was received. If they do not match, the requestor will issue an Application Message Request in order to fill the gap from the prior ApplSeqNum up through the most recently received ApplLastSeqNum.

Resetting application sequence numbers

Resetting of ApplSeqNum can be done at any point using the Application Message Report. It is up to the implementation to determine the frequency of ApplSeqNum reset. It may be done weekly, daily or otherwise depending on the requirements of the parties involved.

Use with Business-level Request Messages

FIX provides a number of messages which allow business-level data to be requested or subscribed to based on a large, fluid set of criteria. Some of the messages that fit into this category are: Market Data Request, Trade Capture Report Request, Position Request, Security Definition Request, Security List Request, Derivatives Security List Request, Security Type Request, Security Status Request, and Trading Session List Request. These request messages allow users to subscribe to business content which is then delivered over a FIX session. The request or subscription method can be defined by via bilateral agreement between the parties.

As previously stated, the purpose of Application Sequencing is to allow data to be segregated into meaningful groups by application identifier and sequenced accordingly such that data gaps can be detected and recovered via Application Message Request. Ordinarily, this is accomplished through a Receiver subscribing to one or more Application ID's using the Application Message Request. The data stream created forms the base for application level recovery. In cases where the complete application sequence dataisfiltered before being sent to a certain receiver, the ApplLastSeqNum field must be used.

It is important to note that Application Sequencing and Recovery functions strictly at the application sequence level and requires that *all data being transmitted within a session participate in Application Sequencing.* Ordinarily, business-level request messages such as the Market Data Request and Trade Capture Report Request do not generate subscriptions that are capable of participating in Application Sequencing. A business-level request or subscription is not capable of indicating that the resulting data should be sequenced at the application level. **If a business-level request were to be issued over a session using Application Sequencing session-level gaps in the data may be unable to be recovered since recovery is being accomplished at the application-level.**

However, a sender could allow subscription requests or queries to be entered for a set of data that uses Application Sequencing and recovery,for exampl, to filter the content down or request a batch of data at a certain time. In the case of subscriptions, the filters could be set at logon time and even be persistent over time (until changed). It is not recommended that changes in the subscription / query requests filters be changed during the time the application stream of data is sent as a changed filter could impact the ability to recover lost messages.

© Copyright, 2007-2008, FIX Protocol, Limited