

FINANCIAL INFORMATION EXCHANGE PROTOCOL (FIX)

Version 4.2 with Errata 20010501

Includes Errata adjustments as of May 1, 2001
Errata Purpose:

This document includes a list of minor adjustments to the FIX 4.2 Specification document due to typographical errors or ambiguities. The nature and scope of Errata adjustments do not introduce new functionality, additional fields, new values for existing fields, or new messages. **Regretably some functionality was introduced in FIX 4.2 which contained errors that required a new value or field on a specific message in order to make the intended functionality implementable. Any such exceptions to the “do not introduce” “additional fields” or “new messages” Errata rule were kept to an absolute minimum using the “required to make the intended functionality implementable” rationale.** All of the items specified in this document will be incorporated in the next release of the FIX Protocol. The list of items has been reviewed and approved by the FIX Technical Committee and Steering Committees. Implementers of FIX version 4.2 should refer to this document to ensure the most consistent implementation and clearest understanding of the FIX protocol.

The specific adjustments made to the original FIX version 4.2 specification as a result of the Errata can be seen and printed via Microsoft Word’s revision feature of this document. A separate document with an itemized list of changes is available via the FIX website.

May 1, 2001

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).

Copyright 2001 FIX Protocol Limited, all rights reserved

PREFACE

The Financial Interface eXchange (FIX) effort was initiated in 1992 by a group of institutions and brokers interested in streamlining their trading processes. These firms felt that they, and the industry as a whole, could benefit from efficiencies derived through the electronic communication of indications, orders and executions. The result is FIX, an open message standard controlled by no single entity, that can be structured to match the business requirements of each firm. The benefits are:

- * From the business flow perspective, FIX provides institutions and brokers a means of reducing the clutter of unnecessary telephone calls and scraps of paper, and facilitates targeting high quality information to specific individuals.
- * For technologists, FIX provides an open standard that leverages the development effort so that they can efficiently create links with a wide range of counter-parties.
- * For vendors, FIX provides ready access to the industry, with the incumbent reduction in marketing effort and increase in potential client base.

Openness has been the key to FIX's success. For that reason, while encouraging vendors to participate with the standard, FIX has remained vendor neutral. Similarly, FIX avoids over-standardization. It does not demand a single type of carrier (e.g., it will work with leased lines, frame relay, Internet, etc.), nor a single security protocol. It leaves many of these decisions to the individual firms that are using it. We do expect that, over time, the rules of engagement in these non-standardized areas will converge as technologies mature.

FIX is now used by a variety of firms and vendors. It has clearly emerged as the inter-firm messaging protocol of choice. Periodic Technical Forum meetings are held to discuss modification of the specification and are open for all to attend. Those interested in providing input to the protocol are encouraged to contact the Technical Committee Chairpersons, Scott Atwell, American Century Investments, (US) 816-340-7053 (scott_atwell@americancentury.com) or Peter White, Goldman, Sachs & Co., (UK) 011-44-207-552-1315 (peter.white@gs.com). Periodic Technical Forum meetings are announced via email and on the WWW, go to <http://www.fixprotocol.org> for details.

We look forward to your participation.

FIX Protocol Ltd

May 2001

US Steering Committee	European Steering Committee	Japanese Steering Committee	Asia/Pacific Steering Committee
Alliance Capital	Abn Amro Bank NV.	Barclays Capital Japan Ltd.	American Century Investments
American Century	Alliance Capital	Daiwa Securities Group Inc.	Barring Asset Management
Capital Research			Capital Research
Credit Suisse Asset Management	Baillie Gifford	Goldman Sachs (Japan) Ltd.	Credit Suisse First Boston
Credit Suisse First Boston	Barclays Global Investors	Lehman Brothers Tokyo	Deutsche Securities Asia Ltd.

Fidelity Capital Markets	Barclays Stockbrokers	Merrill Lynch Japan Incorporated	Dresdner RCM Global Investors (Asia) Ltd.
Fidelity Mgmt & Res. Co.	Cazenove & Co.	Morgan Stanley Dean Witter Japan Limited	Fidelity Investments HK
The Goldman Sachs Group, Inc.		Nikko Asset Management Co., Ltd.	Goldman Sachs
Merrill Lynch	Commerzbank AG	Nikko Salomon Smith Barney	ING-Baring
Morgan Stanley DW&D	Credit Suisse First Boston		Janus International (Asia) Ltd.
		Nomura Asset Management Co., Ltd.	JF Asset Management
Putnam Investments		Nomura Securities Co., Ltd	Jardine Fleming Securities Ltd.
Salomon Smith Barney	Dresdner RCM Global Investors	The Chuomitsui Trust and Banking	Merrill Lynch Asia
State Street Global Advisors		The Mitsubishi Trust and Banking Corporation	Morgan Stanley Dean Witter
UBS Warburg	Foreign and Colonial Mgmt Limited		Salomon Smith Barney
	Goldman, Sachs International	The Sumitomo Trust & Banking Co., Ltd.	UBS Warburg
	HSBC	UBS Warburg	
	Instinet		
	ITG Europe		
	J.P. Morgan Investment Mgmt, Inc.		
	Merrill Lynch		
	Merrill Lynch Investment Management		
	Morgan Stanley DW&D		
	SG Securities London		
	Salomon Smith Barney		
	UBS Warburg		

Contents

PREFACE	1
CONTENTS	3
FINANCIAL INFORMATION EXCHANGE PROTOCOL	8
INTRODUCTION	8
FIX MESSAGE FORMAT AND DELIVERY	8
Message Format	8
Field Delimiter:	9
Repeating Groups:	9
Data Types:	9
Sequence Numbers:	11
Heartbeats:	11
Ordered Message Processing:	12
Possible Duplicates:	12
Possible Resends:	12
Data Integrity:	12
Required Fields:	12
Message Acknowledgment:	13
Encryption:	13
User Defined Fields:	13
SESSION PROTOCOL	14
Logon -	14
Message exchange -	15
Logout -	15
Message Recovery -	17
Standard Message header	19
Standard Message trailer	22
ADMINISTRATIVE MESSAGES	23
Heartbeat -	23
Logon -	24
Test Request -	26
Resend Request -	27
Reject (session-level) -	28
Sequence Reset (Gap Fill) -	30
Logout -	32
APPLICATION MESSAGES	33
Advertisements -	33
Indications of Interest -	35
News -	38
Email -	40
Quote Request -	42

Quote -	45
Mass Quote –	48
Quote Cancel -	55
Quote Status Request -	57
Quote Acknowledgement -	59
Market Data Request -	62
Market Data – Snapshot / Full Refresh -	65
Market Data – Incremental Refresh -	69
Market Data Request Reject -	74
Security Definition Request -	75
Security Definition -	78
Security Status Request -	81
Security Status -	83
Trading Session Status Request -	85
Trading Session Status -	86
New Order - Single -	87
Execution Reports -	92
Don't Know Trade (DK) -	101
Order Cancel/Replace Request (a.k.a. Order Modification Request) -	103
Order Cancel Request -	107
Order Cancel Reject -	110
Order Status Request -	112
Allocation -	114
Allocation ACK -	119
Settlement Instructions -	120
Bid Request -	123
Bid Response -	126
New Order - List -	128
List Strike Price -	133
List Status -	135
List Execute -	137
List Cancel Request -	138
List Status Request -	139
Fragmentation for List Order Messages	140
Business Message Reject -	141
Field Definitions	144
FIX Field Index sorted by tag number:	187
FIX Field Index sorted by field name:	191
Appendix A	195
Valid Currency Codes	195
Appendix B	196
Checksum Calculation	196
Appendix C	197
Reuters Exchange Mnemonics	197
Appendix D	199
Order State Change Matrices	199
D1 - Filled order	203
D2 – Part-filled day order, done for day	203

D3 – Cancel request issued for a zero-filled order	204
D4 – Cancel request issued for a part-filled order – executions occur whilst cancel request is active	204
D5 – Cancel request issued for an order that becomes filled before cancel request can be accepted	205
D6 – Zero-filled order, cancel/replace request issued to increase order qty	206
D7 – Part-filled order, followed by cancel/replace request to increase order qty, execution occurs whilst order is pending replace	207
D8 – Filled order, followed by cancel/replace request to increase order quantity	207
D9 – Cancel/replace request (not for quantity change) is rejected as a fill has occurred	208
D10 – Cancel/replace request sent whilst execution is being reported – the requested order qty exceeds the cum qty. Order is replaced then filled	208
D11 – Cancel/replace request sent whilst execution is being reported – the requested order qty equals the cum qty – order qty is amended to cum qty	210
D12 – Cancel/replace request sent whilst execution is being reported – the requested order qty is below cum qty – order qty is amended to cum qty	210
D13 – One cancel/replace request is issued which is accepted – another one is issued which is also accepted	211
D14 – One cancel/replace request is issued which is rejected before order becomes pending replace – then another one is issued which is accepted	212
D15 One cancel/replace request is issued which is rejected after it is in pending replace – then another one is issued which is accepted	213
D16– One cancel/replace request is issued followed immediately by another – broker processes sequentially	214
D17– One cancel/replace request is issued followed immediately by another – broker rejects the second as order is pending replace	214
D18 – Telephoned order	215
D19 – Unsolicited cancel of a part-filled order	215
D20 – Unsolicited replacement of a part-filled order	217
D23 - Order rejected because the order has already been verbally submitted	219
D24 - Order status request rejected for unknown order	219
D26 - Order sent, immediately followed by a status request. Subsequent status requests sent during life of order	220
D27 - GTC order partially filled, restated (renewed) and partially filled the following day	222
D28 - GTC order with partial fill, a 2:1 stock split then a partial fill and fill the following day	222
D29 - GTC order partially filled, restated(renewed) and canceled the following day	224
D30 - GTC order partially filled, restated(renewed) followed by replace request to increase quantity	225
D31 - Poss resend order	225
D32 – Fill or Kill order cannot be filled	226
D33 – Immediate or Cancel order that cannot be immediately hit	226
D34 – Filled order, followed by correction and cancellation of executions	226
D36 - GTC order partially filled, restated (renewed) and partially filled the following day, with corrections of quantity on both executions	228

Appendix E	231
Order Handling and Instruction Semantics	231
London SETS Order Types Matrix	231
Asia/Pacific Regional Order Handling	231
Handling Instructions (HandlInst) field	231
Pegged Orders	232
“Reserve Quantity” Orders	233
Appendix F	233
Settlement Instructions Field Usage Matrix	233

Appendix G	237
Rule80A (aka OrderCapacity) Usage by Market	237
Appendix H	239
Mass Quote Message Scenarios	239
Unsolicited quote(s) no response requested	239
Unsolicited quote(s) negative response only requested	239
Unsolicited quote(s) full response requested	240
Cancel All Quotes	240
Appendix I	241
Security Definition, Security Status, and Trading Session Message Scenarios	241
Overview	241
Background	241
Definitions	241
Approach	242
Extensions to other messages	242
Rules	243
Specifying Derivative Trading Strategies using the Security Definition message	243
Scenario 1 - Typical use of Security Definition message in placing an Order	244
Scenario 2 - Inquire Securities Types Available	244
Scenario 3 – Inquire Common Stocks Available for Trading with Counterparty.	244
Scenario 4 - Inquire all securities traded by a trading party	245
Scenario 5 – Inquire Option Classes Available for Trading with Counterparty.	246
Scenario 6 - Inquire list of option series for a class	246
Appendix J	248
Example Usage of Encoded Fields For Japanese Language Support	248
Appendix K	249
Example Usage of Allocations	249
Appendix L	258
Pre-Trade Message Targeting/Routing	258
Targeting	258
Blocking	259
Other Issues	260
Appendix M	261
FIXML Support	261
Appendix N	262
Program/Basket/List Trading	262
Overview	262
Message Flow Diagrams	263
Scenario 1 Bidding performed by Telephone and List provided via FIX	265
Scenario 2 Fully Disclosed Program Trade – with bidding stage through FIX	266
Scenario 3 Non-Disclosed Program Trade – with bidding stage through FIX	266
Illustration of liquidity indicator fields usage	267
Appendix O	271
Foreign Exchange (F/X) Trading	271

FINANCIAL INFORMATION EXCHANGE PROTOCOL

INTRODUCTION

The Financial Information Exchange (FIX) Protocol is a message standard developed to facilitate the electronic exchange of information related to securities transactions. It is intended for use between trading partners wishing to automate communications.

The message protocol, as defined, will support a variety of business functions. FIX was originally defined for use in supporting US domestic equity trading with message traffic flowing directly between principals. As the protocol evolved, a number of fields were added to support limited cross-border and fixed income trading. Similarly, the protocol was expanded to allow third parties to participate in the delivery of messages between trading partners. As subsequent versions of FIX are released, it is expected that functionality will continue to expand.

FIX was written to be independent of any specific communications protocol (X.25, asynch, TCP/IP, etc.) or physical medium (copper, fiber, satellite, etc.) chosen for electronic data delivery. It should be noted that if an “unreliable” or non-stream protocol is used, the Logon, Logout, and ResendRequest message processing is particularly susceptible to unordered delivery and/or message loss.

The protocol is defined at two levels: session and application. The session level is concerned with the delivery of data while the application level defines business related data content. This document is organized to reflect the distinction.

FIX MESSAGE FORMAT AND DELIVERY

The following section summarizes general specifications for constructing and transmitting FIX messages.

Message Format

The general format of a FIX message is a standard header followed by the message body fields and terminated with a standard trailer.

Each message is constructed of a stream of <tag>=<value> fields with a field delimiter between fields in the stream. **All tags must have a value specified. Optional fields without values should simply not be specified in the FIX message. A Reject message is the appropriate response to a tag with no value.**

Except where noted, fields within a message can be defined in any sequence (Relative position of a field within a message is inconsequential.) The exceptions to this rule are:

- 1. General message format is composed of the standard header followed by the body followed by the standard trailer.**
- 2. The first three fields in the standard header are BeginString (tag #8) followed by BodyLength (tag #9) followed by MsgType (tag #35).**
- 3. The last field in the standard trailer is the CheckSum (tag #10).**
- 4. Fields within repeating data groups must be specified in the order that the fields are specified in the message definition within the FIX specification document. The NoXXX field where XXX is the field being counted specifies the number of repeating group instances that must immediately precede the repeating group contents.**

•

In addition, certain fields of the data type *MultipleValueString* can contain multiple individual values separated by a space within the "value" portion of that field followed by a single "SOH" character (e.g. "18=2 9 C<SOH>" represents 3 individual values: '2', '9', and 'C').

It is also possible for a field to be contained in both the clear text portion and the encrypted data sections of the same message. This is normally used for validation and verification. For example, sending the *SenderCompID* in the encrypted data section can be used as a rudimentary validation technique. In the cases where the clear text data differs from the encrypted data, the encrypted data should be considered more reliable. (A security warning should be generated).

Field Delimiter:

All fields (including those of data type *data* i.e. *SecureData*, *RawData*, *SignatureData*, *XmlData*, etc.) in a FIX message are terminated by a delimiter character. The non-printing, ASCII "SOH" (#001, hex: 0x01, referred to in this document as <SOH>), is used for field termination. Messages are delimited by the "SOH" character following the CheckSum field. All messages begin with the "8=FIX.x.y<SOH>" string and terminate with "10=nnn<SOH>".

There shall be no embedded delimiter characters within fields except for data type *data*.

Repeating Groups:

It is permissible for fields to be repeated within a repeating group (e.g. "384=2<SOH>372=6<SOH>385=R<SOH>372=7<SOH>385=R<SOH>" represents a repeating group with two repeating instances "delimited" by tag 372 (first field in the repeating group).)

- If the repeating group is used, the first field of the repeating group is required. This allows implementations of the protocol to use the first field as a "delimiter" indicating a new repeating group entry.
- The NoXXX field which specifies the number of repeating group instances occurs once for a repeating group and must immediately precede the repeating group contents.
- If a repeating group field is listed as required, then it must appear in every repeated instance of that repeating group.
- Repeating groups are designated within the message definition via indentation and the → symbol.

Some repeating groups are nested within another repeating group (potentially more than one level of nesting).

- Nested repeating groups are designated within the message definition via indentation and the → symbol followed by another → symbol..
- If a nested repeating group is used, then the outer repeating group must be specified

Data Types:

Data types (with the exception of those of type "data") are mapped to ASCII strings as follows:

- **int:** Sequence of digits without commas or decimals and optional sign character (ASCII characters "-" and "0" - "9"). The sign character utilizes one byte (i.e. positive int is "99999")

while negative int is "-99999"). Note that int values may contain leading zeros (e.g. "00023" = "23").

Examples: 723 in field 21 would be mapped int as |21=723|.

-723 in field 12 would be mapped int as |12=-723|

- **float:** Sequence of digits with optional decimal point and sign character (ASCII characters "-", "0" - "9" and "."); the absence of the decimal point within the string will be interpreted as the float representation of an integer value. All float fields must accommodate up to fifteen significant digits. The number of decimal places used should be a factor of business/market needs and mutual agreement between counterparties. Note that float values may contain leading zeros (e.g. "00023.23" = "23.23") and may contain or omit trailing zeros after the decimal point (e.g. "23.0" = "23.0000" = "23").
- **Qty:** float field (see definition of "float" above) capable of storing either a whole number (no decimal places) of "shares" or a decimal value containing decimal places for non-share quantity asset classes.
- **Price:** float field (see definition of "float" above) representing a price. Note the number of decimal places may vary.
- **PriceOffset:** float field (see definition of "float" above) representing a price offset, which can be mathematically added to a "Price". Note the number of decimal places may vary and some fields such as LastForwardPoints may be negative.
- **Amt:** float field (see definition of "float" above) typically representing a Price times a Qty.
- **char:** Single character value, can include any alphanumeric character or punctuation except the delimiter. All char fields are case sensitive (i.e. **m** ≠ **M**).
- **Boolean:** a char field (see definition of "char" above) containing one of two values:
 - 'Y' = True/Yes
 - 'N' = False/No
- **String:** Alpha-numeric free format strings, can include any character or punctuation except the delimiter. All char fields are case sensitive (i.e. **morstatt** ≠ **Morstatt**).
- **MultipleValueString:** String field (see definition of "String" above) containing one or more space delimited values.
- **Currency:** String field (see definition of "String" above) representing a currency type.

Valid values:

- [See Appendix A – Valid Currency Codes](#)

- **Exchange:** String field (see definition of "String" above) representing a market or exchange.

Valid values:

- [See Appendix C – Reuters Exchange Mnemonics](#)

- **UTCTimestamp:** Time/date combination represented in UTC (Universal Time Coordinated, also known as "GMT") in **either** YYYYMMDD-HH:MM:SS (whole seconds) or YYYYMMDD-HH:MM:SS.sss (milliseconds) format, colons, dash, and period required.

Valid values:

- YYYY = 0000-9999, MM = 01-12, DD = 01-31, HH = 00-23, MM = 00-59, SS = 00-60 (60 only if UTC leap second) (without milliseconds).

- YYYY = 0000-9999, MM = 01-12, DD = 01-31, HH = 00-23, MM = 00-59, SS = 00-60 (60 only if UTC leap second), sss=000-999 (indicating milliseconds).

Leap Seconds: Note that UTC includes corrections for leap seconds, which are inserted to account for slowing of the rotation of the earth. Leap second insertion is declared by the International Earth Rotation Service (IERS) and has, since 1972, only occurred on the night of Dec. 31 or Jun 30. The IERS considers March 31 and September 30 as secondary dates for leap second insertion, but has never utilized these dates. During a leap second insertion, a *UTCTimestamp* field may read "19981231-23:59:59", "19981231-23:59:60", "19990101-00:00:00". (see <http://tycho.usno.navy.mil/leapsec.html>)

- **UTCTimeOnly:** Time-only represented in UTC (Universal Time Coordinated, also known as "GMT") in **either** HH:MM:SS (whole seconds) or HH:MM:SS.sss (milliseconds) format, colons, and period required.

Valid values:

- HH = 00-23, MM = 00-60 (60 only if UTC leap second), SS = 00-59. (without milliseconds)
- HH = 00-23, MM = 00-59, SS = 00-60 (60 only if UTC leap second), sss=000-999 (indicating milliseconds).
- **LocalMktDate:** Date of Local Market (vs. UTC) in YYYYMMDD format. Valid values: YYYY = 0000-9999, MM = 01-12, DD = 01-31.
- **UTCDate:** Date represented in UTC (Universal Time Coordinated, also known as "GMT") in YYYYMMDD format. Valid values: YYYY = 0000-9999, MM = 01-12, DD = 01-31.
- **data:** Raw data with no format or content restrictions. Data fields are always immediately preceded by a length field. The length field should specify the number of bytes of the value of the *data* field (up to but not including the terminating SOH). *Caution: the value of one of these fields may contain the delimiter (SOH) character. Note that the value specified for this field should be followed by the delimiter (SOH) character as all fields are terminated with an "SOH".*
- **month-year:** char field representing month of a year in YYYYMM format. Valid values: YYYY = 0000-9999, MM = 01-12.
- **day-of-month:** int field representing a particular day of a month. Valid values: 1-31.

Sequence Numbers:

All FIX messages are identified by a unique sequence number. Sequence numbers are initialized at the start of each FIX session (see Session Protocol section) starting at 1 (one) and increment throughout the session. Monitoring sequence numbers will enable parties to identify and react to missed messages and to gracefully synchronize applications when reconnecting during a FIX session.

Each session will establish an independent incoming and outgoing sequence series; participants will maintain a sequence series to assign to outgoing messages and a separate series to monitor for sequence gaps on incoming messages.

Heartbeats:

During periods of message inactivity, FIX applications will generate *Heartbeat* messages at regular time intervals. The heartbeat monitors the status of the communication link and identifies incoming sequence number gaps. The Heartbeat Interval is declared by the session initiator using the HeartBtInt

field in the *Logon* message. The heartbeat interval timer should be reset after every message is transmitted (not just heartbeats). The HeartBtInt value should be agreed upon by the two firms and specified by the Logon initiator and echoed back by the Logon acceptor. Note that the same HeartBtInt value is used by both sides, the Logon “initiator” and Logon “acceptor”.

Ordered Message Processing:

The FIX protocol assumes complete ordered delivery of messages between parties. Implementers should consider this when designing message gap fill processes. Two options exist for dealing with gaps, either request all messages subsequent to the last message received or ask for the specific message missed while maintaining an ordered list of all newer messages. For example, if the receiver misses the second of five messages, the application could ignore messages 3 through 5 and generate a resend request for messages 2 through 5, or, preferably 2 through 0 (where 0 represents infinity). Another option would involve saving messages 3 through 5 and resending only message 2. In both cases, messages 3 through 5 should not be processed before message 2.

Possible Duplicates:

When a FIX engine is unsure if a message was successfully received at its intended destination or when responding to a resend request, a possible duplicate message is generated. The message will be a retransmission (with the same sequence number) of the application data in question with the PossDupFlag included and set to “Y” in the header. It is the receiving application's responsibility to handle the message (i.e. treat as a new message or discard as appropriate). All messages created as the result of a resend request will contain the PossDupFlag field set to “Y”, messages lacking the PossDupFlag field or with the PossDupFlag field set to “N” should be treated as original transmissions. *Note: When retransmitting a message with the PossDupFlag set to Y, it is always necessary to recalculate the CheckSum value. The only fields that can change in a possible duplicate message are the CheckSum, OrigSendingTime, SendingTime, BodyLength and PossDupFlag. Fields related to encryption (SecureDataLen and SecureData) may also require recasting.*

Possible Resends:

Ambiguous application level messages may be resent with the PossResend flag set. This is useful when an order remains unacknowledged for an inordinate length of time and the end-user suspects it had never been sent. The receiving application must recognize this flag and interrogate internal fields (order number, etc.) to determine if this order has been previously received. *Note: The possible resend message will contain exactly the same body data but will have the PossResend flag and will have a new sequence number. In addition the CheckSum field will require recalculation and fields related to encryption (SecureDataLen and SecureData) may also require recasting.*

Data Integrity:

The integrity of message data content can be verified in two ways: verification of message length and a simple checksum of characters.

The message length is indicated in the BodyLength field and is verified by counting the number of characters in the message following the BodyLength field up to, and including, the delimiter immediately preceding the CheckSum tag (“10=”).

The CheckSum integrity check is calculated by summing the binary value of each character from the “8” of “8=” up to and including the <SOH> character immediately preceding the CheckSum tag field and comparing the least significant eight bits of the calculated value to the CheckSum value ([see Appendix B: CheckSum Calculation](#) for a complete description).

Required Fields:

Each message within the protocol is comprised of *required*, *optional* and *conditionally required* (fields which are required based on the presence or value of other fields) fields. Systems should be designed to operate when only the required and conditionally required fields are present.

Message Acknowledgment:

The FIX session protocol is based on an optimistic model; normal delivery of data is assumed (i.e. no acknowledgment of individual messages) with errors in delivery identified by message sequence number gaps. Each message is identified by a unique sequence number. It is the receiving application's responsibility to monitor incoming sequence numbers to identify message gaps for response with resend request messages.

The FIX protocol does not support individual message acknowledgment. However, a number of application messages require explicit application level acceptance or rejection. Orders, cancel requests, cancel/replace requests and allocation require specific application level response, executions can be rejected with the DK message but do not require explicit acceptance.

Encryption:

The exchange of sensitive data across public carrier networks may make it advisable to employ data encryption techniques to mask the application messages.

The choice of encryption method will be determined by mutual agreement of the two parties involved in the connection.

Any field within a message can be encrypted and included in the SecureData field, however, certain explicitly identified fields must be transmitted unencrypted. The clear (unencrypted) fields can be repeated within the SecureData field to serve as an integrity check of the clear data.

When encryption is employed, it is recommended but not required that all fields within the message body be encrypted. If repeating groups are used within a message and encryption is applied to part of the repeating group, then the entire repeating group must be encrypted.

Embedded in the protocol are fields, which enable the implementation of a public key signature and encryption methodology, straight DES encryption and clear text. The previously agreed upon encryption methodology is declared in the *Logon* message. (For more detail on implementation of various encryption techniques see the application notes section on the FIX Web Site.)

User Defined Fields:

In order to provide maximum flexibility for its users, the FIX protocol accommodates *User Defined Fields*. These fields are intended to be implemented between consenting trading partners and should be used with caution to avoid conflicts, which will arise as multiple parties begin implementation of the protocol. It is suggested that if trading partners find that particular User Defined Fields add value, they should be recommended to the FIX Technical Committee for inclusion in a future FIX version.

The tag numbers 5000 to 9999 have been reserved for use with user defined fields, which are used as part of inter-firm communication. These tags can be registered/reserved via the FIX website.

The tag numbers greater than or equal to 10000 have been reserved for internal use (within a single firm) and do not need to be registered/reserved via the FIX website.

SESSION PROTOCOL

A FIX session is defined as a bi-directional stream of ordered messages between two parties within a continuous sequence number series. A single FIX session can exist across multiple physical connections. Parties can connect and disconnect multiple times while maintaining a single FIX session. Connecting parties must bi-laterally agree as to when sessions are to be started/stopped based upon individual system and time zone requirements. It is recommended that a new FIX session be established once within each 24 hour period. It is possible to maintain 24 hour connectivity and establish a new set of sequence numbers by sending a Logon message with the ResetSeqNumFlag set.

The FIX session protocol is based on an optimistic model. Normal delivery of data is assumed (i.e. no communication level acknowledgment of individual messages) with errors in delivery identified by message sequence number gaps. This section provides details on the implementation of the FIX session layer and dealing with message sequence gaps.

The following terms are used throughout this section:

- **Valid FIX Message** is a message that is properly formed according to this specification and contains a valid body length and checksum field
- **Initiator** establishes the telecommunications link and initiates the session via transmission of the initial *Logon* message.
- **Acceptor** is the receiving party of the FIX session. This party has responsibility to perform first level authentication and formally declare the connection request “accepted” through transmission of an acknowledgment *Logon* message.

A FIX session is comprised of three parts: logon, message exchange and logout.

Logon -

Establishing a FIX connection involves three distinct operations: creation of a telecommunications level link, authentication/acceptance of the initiator by the acceptor and message synchronization (initialization). The sequence of connection follows:

- The session initiator establishes a telecommunication link with the session acceptor.
- The initiator sends a *Logon* message. The acceptor will authenticate the identity of the initiator by examining the *Logon* message. The *Logon* message will contain the data necessary to support the previously agreed upon authentication method. If the initiator is successfully authenticated, the acceptor responds with a *Logon* message. If authentication fails, the session acceptor should shut down the connection after optionally sending a Logout message to indicate the reason of failure. Sending a Logout in this case is not required because doing so would consume a sequence number for that session, which in some cases may be problematic. The session initiator may begin to send messages immediately following the *Logon* message, however, the acceptor may not be ready to receive them. The initiator must wait for the confirming *Logon* message from the acceptor before declaring the session fully established.

After the initiator has been authenticated, the acceptor will respond immediately with a confirming *Logon* message. Depending on the encryption method being used for that session, this *Logon* message may or may not contain the same session encryption key. The initiator side will use the *Logon* message being returned from the acceptor as confirmation that a FIX session has been established. If the session acceptor has chosen to change the session encryption key, the session initiator must send a third *Logon* back to the other side in order to acknowledge the key change request. This also allows the session acceptor to know when the session initiator has started to

encrypt using the new session key. Both parties are responsible for infinite loop detection and prevention during this phase of the session.

- **After authentication, the initiator and acceptor must synchronize their messages through interrogation of the *MsgSeqNum* field before sending any queued or new messages.** A comparison of the *MsgSeqNum* in the *Logon* message to the internally monitored next expected sequence number will indicate any message gaps. Likewise, the initiator can detect gaps by comparing the acknowledgment *Logon* message *MsgSeqNum* to the next expected value. The section on message recovery later in this document deals with message gap handling.
- It is recommended to wait a short period of time following the *Logon* or to send a *TestRequest* and wait for a response to it before sending queued or new messages in order to allow both sides to handle resend request processing. Failure to do this could result in a *ResendRequest* message being issued by one's counterparty for each queued or new message sent.
- It is also recommended that an engine should store out of sequence messages in a temporary queue and process them in order when the gap is closed. This prevents generating resend requests for $n > m$, $n = m+1$, $n = m+2$, ... which can result in many resent *PossDupFlag=Y* messages.
- When using the *ResetSeqNumFlag* to maintain 24 hour connectivity and establish a new set of sequence numbers, the process should be as follows. Both sides should agree on a reset time and the party that will be the initiator of the process. Note that the initiator of the *ResetSeqNum* process may be different than the initiator of the *Logon* process. One side will initiate the process by sending a *TestRequest* and wait for a *Heartbeat* in response to ensure of no sequence number gaps. Once the *Heartbeat* has been received, the initiator should send a *Logon* with *ResetSeqNumFlag* set to Y and with *MsgSeqNum* of 1. The acceptor should respond with a *Logon* with *ResetSeqNumFlag* set to Y and with *MsgSeqNum* of 1. At this point new messages from either side should continue with *MsgSeqNum* of 2. It should be noted that once the initiator sends the *Logon* with the *ResetSeqNumFlag* set, the acceptor must obey this request and the message with the last sequence number transmitted "yesterday" may no longer be available. The connection should be shutdown and manual intervention taken, if this process is initiated but not followed properly.

Message exchange -

After completion of the initialization process, normal message exchange begins. The formats for all valid messages are detailed in the sections 'Administrative Messages' and 'Application Messages'.

Logout -

Normal termination of the message exchange session will be completed via the exchange of *Logout* messages. Termination by other means should be considered an abnormal condition and dealt with as an error. Session termination without receiving a *Logout* should treat the counterparty as logged out.

It is recommended that before sending the *Logout* message, a *TestRequest* should be issued to force a *Heartbeat* from the other side. This helps to ensure that there are no sequence number gaps.

Before actually closing the session, the *Logout* initiator should wait for the opposite side to respond with a confirming *Logout* message. This gives the acceptor a chance to perform any Gap Fill operations that may be necessary. Once the messages from the *ResendRequest* have been received, the acceptor should issue the *Logout*. The session may be terminated if the acceptor does not respond in an appropriate timeframe.

Note: Logging out does not affect the state of any orders. All active orders will continue to be eligible for execution after logout.

Message Recovery -

During initialization, or in the middle of a FIX session, message gaps may occur which are detected via the tracking of incoming sequence numbers. The following section provides details on how to recover messages.

As previously stated, each FIX participant must maintain two sequence numbers for each FIX session, one each for incoming and outgoing messages which are initialized at '1' at the beginning of the FIX session. Each message is assigned a unique (by connection) sequence number, which is incremented after each message. Likewise, every message received has a unique sequence number and the incoming sequence counter is incremented after each message.

When the incoming sequence number does not match the expected number corrective processing is required. Note that the *SeqReset-Reset* message (used only to recover from a disaster scenario vs. normal resend request processing) is an exception to this rule as it should be processed without regards to its *MsgSeqNum*. **If the incoming message has a sequence number less than expected and the *PossDupFlag* is not set, it indicates a serious error. It is strongly recommended that the session be terminated and manual intervention be initiated.** If the incoming sequence number is greater than expected, it indicates that messages were missed and retransmission of the messages is requested via the *Resend Request* (see the earlier section, *Ordered Message Processing*).

Note: For the purposes of the following paragraphs *requester* refers to the party requesting the resend and *resender* refers to the party responding to the request. The process of resending and synchronizing messages is referred to as "gap filling".

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"
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

During the gap fill process, certain administrative messages should not be retransmitted. Instead, a special *SeqReset-GapFill* message is generated. The administrative messages which are not to be resent are: *Logon*, *Logout*, *ResendRequest*, *Heartbeat*, *TestRequest* and *SeqReset-Reset* and *SeqReset-GapFill*. The *SeqReset-GapFill* can also be used to skip application messages that the sender chooses not to retransmit (e.g. aged orders). This leaves *Reject* as the only administrative message- which can be resent.

All FIX implementations must monitor incoming messages to detect inadvertently retransmitted administrative messages (*PossDupFlag* flag set indicating a resend). When received, these messages should be processed for sequence number integrity only; the business/application processing of these message should be skipped (e.g. do not initiate gap fill processing based on a resent *ResendRequest*).

If there are consecutive administrative messages to be resent, it is suggested that only one *SeqReset-GapFill* message be sent in their place. The sequence number of the *SeqReset-GapFill* message is the next expected outbound sequence number. **The *NewSeqNo* field of the GapFill message contains the sequence number of the highest administrative message in this group plus 1.** For example, during a Resend operation there are 7 sequential administrative messages waiting to be resent. They start with sequence number 9 and end with sequence number 15. Instead of transmitting 7 Gap Fill messages (which is perfectly legal, but not network friendly), a *SeqReset-GapFill* message may be sent. **The sequence number of the Gap Fill message is set to 9 because the remote side is expecting that as the next next sequence number.** The *NewSeqNo* field of the GapFill message contains the number 16, because that will be the sequence number of the next message to be transmitted.

Sequence number checking is a vital part of FIX session management. However, a discrepancy in the sequence number stream is handled differently for certain classes of FIX messages. The table below lists the actions to be taken when the incoming sequence number is greater than the expected incoming sequence number.

NOTE: In **ALL cases except the Sequence Reset - Reset message, the FIX session should be terminated if the incoming sequence number is less than expected and the PossDupFlag is not set. A Logout message with some descriptive text should be sent to the other side before closing the session.**

Response by Message Type

Message Type	Action to Be Taken on Sequence # mismatch
Logon	Must always be the first message transmitted. Authenticate and accept the connection. After sending a <i>Logon</i> confirmation back, send a <i>ResendRequest</i> if a message gap was detected in the <i>Logon</i> sequence number.
Logout	<p>If a message gap was detected, issue a <i>ResendRequest</i> to retrieve all missing messages followed by a <i>Logout</i> message which serves as a confirmation of the logout request. <u>DO NOT</u> terminate the session. The initiator of the <i>Logout</i> sequence has responsibility to terminate the session. This allows the <i>Logout</i> initiator to respond to any <i>ResendRequest</i> message.</p> <p>If this side was the initiator of the <i>Logout</i> sequence, then this is a <i>Logout</i> confirmation and the session should be immediately terminated upon receipt.</p> <p>The only exception to the “do not terminate the session” rule is for an invalid Logon attempt. The session acceptor has the right to send a Logout message and terminate the session immediately. This minimizes the threat of unauthorized connection attempts.</p>
ResendRequest	Perform the Resend processing first, followed by a <i>ResendRequest</i> of your own in order to fill the incoming message gap.
SeqReset-Reset	Ignore the incoming sequence number. The <i>NewSeqNo</i> field of the <i>SeqReset</i> message will contain the sequence number of the next message to be transmitted.
SeqReset-GapFill	Send a <i>ResendRequest</i> back. Gap Fill messages behave similar to a <i>SeqReset</i> message. However, it is important to insure that no messages have been inadvertently skipped over. This means that <i>GapFill</i> messages must be received in sequence. An out of sequence <i>GapFill</i> is an abnormal condition
All Other Messages	Perform Gap Fill operations.

Standard Message header

Each administrative or application message is preceded by a standard header. The header identifies the message type, length, destination, sequence number, origination point and time.

Two fields help with resending messages. The PossDupFlag is set to Y when resending a message as the result of a session level event (i.e. the retransmission of a message reusing a sequence number). The PossResend is set to Y when reissuing a message with a new sequence number (e.g. resending an order). The receiving application should process these messages as follows:

PossDupFlag - if a message with this sequence number has been previously received, ignore message, if not, process normally.

PossResend - forward message to application and determine if previously received (i.e. verify order id and parameters).

Note that if OnBehalfOfCompID or DeliverToCompID message source identification/routing is used for a FIX session, then it must be used on all Application messages transmitted via that session accordingly (Reject message if not).

The following table provides examples regarding the use of SenderCompID, TargetCompID, DeliverToCompID, and OnBehalfOfCompID when using a single point-to-point FIX session between two firms. Assumption (A=sellside, B =buyside):

	SenderCompID	OnBehalfOfCompID	TargetCompID	DeliverToCompID
A to B directly	A		B	
B to A directly	B		A	

The following table provides examples regarding the use of SenderCompID, TargetCompID, DeliverToCompID, and OnBehalfOfCompID when using a single FIX session to represent multiple firms. Assumption (A=sellside, B and C=buyside, Q=third party):

	SenderCompID	OnBehalfOfCompID	TargetCompID	DeliverToCompID	OnBehalfOfSending Time
Send from A to B via Q					
1)	A sends to Q	A		Q	B
2)	Q sends to B	Q	A	B	A's SendingTime
B responds to A via Q					
1)	B sends to Q	B		Q	A
2)	Q sends to A	Q	B	A	B's SendingTime
Send from A to B *AND* C via Q					
1)	A sends to Q	A		Q	B
2)	Q sends to B	Q	A	B	A's SendingTime
3)	A sends to Q	A		Q	C
4)	Q sends to C	Q	A	C	A's SendingTime
B *AND* C send to A via Q					
1)	B sends to Q	B		Q	A

2)	Q sends to A	Q	B	A		B's SendingTime
3)	C sends to Q	C		Q	A	
4)	Q sends to A	Q	C	A		C's SendingTime

The standard message header format is as follows:

Standard Message Header

Tag	Field Name	Req'd	Comments
8	BeginString	Y	FIX.4.2 (Always unencrypted, must be first field in message)
9	BodyLength	Y	(Always unencrypted, must be second field in message)
35	MsgType	Y	(Always unencrypted, must be third field in message)
49	SenderCompID	Y	(Always unencrypted)
56	TargetCompID	Y	(Always unencrypted)
115	OnBehalfOfCompID	N	Trading partner company ID used when sending messages via a third party (Can be embedded within encrypted data section.)
128	DeliverToCompID	N	Trading partner company ID used when sending messages via a third party (Can be embedded within encrypted data section.)
90	SecureDataLen	N	Required to identify length of encrypted section of message. (Always unencrypted)
91	SecureData	N	Required when message body is encrypted. Always immediately follows SecureDataLen field.
34	MsgSeqNum	Y	(Can be embedded within encrypted data section.)
50	SenderSubID	N	(Can be embedded within encrypted data section.)
142	SenderLocationID	N	Sender's LocationID (i.e. geographic location and/or desk) (Can be embedded within encrypted data section.)
57	TargetSubID	N	"ADMIN" reserved for administrative messages not intended for a specific user. (Can be embedded within encrypted data section.)
143	TargetLocationID	N	Trading partner LocationID (i.e. geographic location and/or desk) (Can be embedded within encrypted data section.)
116	OnBehalfOfSubID	N	Trading partner SubID used when delivering messages via a third party. (Can be embedded within encrypted data section.)
144	OnBehalfOfLocationID	N	Trading partner LocationID (i.e. geographic location and/or desk) used when delivering messages via a third party. (Can be embedded within encrypted data section.)
129	DeliverToSubID	N	Trading partner SubID used when delivering messages via a third party. (Can be embedded within encrypted data section.)
145	DeliverToLocationID	N	Trading partner LocationID (i.e. geographic location and/or desk) used when delivering messages via a third party. (Can be embedded within encrypted data section.)

43	PossDupFlag	N	Always required for retransmitted messages, whether prompted by the sending system or as the result of a resend request. <i>(Can be embedded within encrypted data section.)</i>
97	PossResend	N	Required when message may be duplicate of another message sent under a different sequence number. <i>(Can be embedded within encrypted data section.)</i>
52	SendingTime	Y	<i>(Can be embedded within encrypted data section.)</i>
122	OrigSendingTime	N	Required for message resent as a result of a ResendRequest. If data is not available set to same value as SendingTime <i>(Can be embedded within encrypted data section.)</i>
212	XmlDataLen	N	Required when specifying XmlData to identify the length of a XmlData message block. <i>(Can be embedded within encrypted data section.)</i>
213	XmlData	N	Can contain a XML formatted message block (e.g. FIXML). Always immediately follows XmlDataLen field. <i>(Can be embedded within encrypted data section.)</i> <u>See Appendix M – FIXML Support</u>
347	MessageEncoding	N	Type of message encoding (non-ASCII characters) used in a message's "Encoded" fields. Required if any "Encoding" fields are used.
369	LastMsgSeqNumProcessed	N	The last MsgSeqNum value received and processed. Can be specified on every message sent. Useful for detecting a backlog with a counterparty.
370	OnBehalfOfSendingTime	N	Used when a message is sent via a "hub" or "service bureau". If A sends to Q (the hub) who then sends to B via a separate FIX session, then when Q sends to B the value of this field should represent the SendingTime on the message A sent to Q. (always expressed in UTC (Universal Time Coordinated, also known as "GMT"))

Standard Message trailer

Each message, administrative or application, is terminated by a standard trailer. The trailer is used to segregate messages and contains the three digit character representation of the Checksum value.

The standard message trailer format is as follows:

Standard Message Trailer

<i>Tag</i>	<i>Field Name</i>	<i>Req'd</i>	<i>Comments</i>
93	SignatureLength	N	Required when trailer contains signature. <i>Note: Not to be included within SecureData field</i>
89	Signature	N	<i>Note: Not to be included within SecureData field</i>
10	Checksum	Y	<i>(Always unencrypted, always last field in message)</i>

ADMINISTRATIVE MESSAGES

The administrative messages address the utility needs of the protocol. The following section describes each message and provides the message layout.

Administrative messages will be generated from both sides of the connection.

Heartbeat -

The Heartbeat monitors the status of the communication link and identifies when the last of a string of messages was not received.

When either end of a FIX connection has not sent any data for [HeartBtInt] seconds, it will transmit a Heartbeat message. When either end of the connection has not received any data for (HeartBtInt + “some reasonable transmission time”) seconds, it will transmit a Test Request message. If there is still no Heartbeat message received after (HeartBtInt + “some reasonable transmission time”) seconds then the connection should be considered lost and corrective action be initiated. If HeartBtInt is set to zero then no regular heartbeat messages will be generated. Note that a test request message can still be sent independent of the value of the HeartBtInt, which will force a Heartbeat message.

Heartbeats issued as the result of Test Request must contain the TestReqID transmitted in the Test Request message. This is useful to verify that the Heartbeat is the result of the Test Request and not as the result of a regular timeout.

The heartbeat format is as follows:

Heartbeat

<i>Tag</i>	<i>Field Name</i>	<i>Req'd</i>	<i>Comments</i>
	<i>Standard Header</i>	Y	MsgType = 0
112	TestReqID	N	Required when the heartbeat is the result of a Test Request message.
	<i>Standard Trailer</i>	Y	

Logon -

The logon message authenticates a user establishing a connection to a remote system. The logon message must be the first message sent by the application requesting to initiate a FIX session.

The HeartBtInt (108) field is used to declare the timeout interval for generating heartbeats (same value used by both sides). The HeartBtInt value should be agreed upon by the two firms and specified by the Logon initiator and echoed back by the Logon acceptor.

Upon receipt of a Logon message, the session acceptor will authenticate the party requesting connection and issue a Logon message as acknowledgment that the connection request has been accepted. The acknowledgment Logon can also be used by the initiator to validate that the connection was established with the correct party.

The session acceptor must be prepared to immediately begin processing messages after receipt of the Logon. The session initiator can choose to begin transmission of FIX messages before receipt of the confirmation Logon, however it is recommended that normal message delivery wait until after the return Logon is received to accommodate encryption key negotiation.

The confirmation Logon can be used for encryption key negotiation. If a session key is deemed to be weak, a stronger session key can be suggested by returning a Logon message with a new key. This is only valid for encryption protocols that allow for key negotiation. (See the FIX Web Site's Application notes for more information on a method for encryption and key passing.)

The Logon message can be used to specify the MaxMessageSize supported (e.g. can be used to control fragmentation rules for very large messages which support fragmentation). It can also be used to specify the MsgTypes supported for both sending and receiving.

The logon format is as follows:

Logon

Tag	Field Name		Req'd	Comments
	Standard Header		Y	MsgType = A
98	EncryptMethod		Y	(Always unencrypted)
108	HeartBtInt		Y	Note same value used by both sides
95	RawDataLength		N	Required for some authentication methods
96	RawData		N	Required for some authentication methods
141	ResetSeqNumFlag		N	Indicates both sides of a FIX session should reset sequence numbers
383	MaxMessageSize		N	Can be used to specify the maximum number of bytes supported for messages received
384	NoMsgTypes		N	Specifies the number of repeating MsgTypes specified
→	372	RefMsgType	N	Specifies a specific, supported MsgType. Required if NoMsgTypes is > 0. Should be specified from the point of view of the sender of the Logon message
→	385	MsgDirection	N	Indicates direction (send vs. receive) of a supported MsgType. Required if NoMsgTypes is > 0. Should be specified from the point of view of the sender of the Logon message

	<i>Standard Trailer</i>	Y	
--	-------------------------	---	--

Test Request -

The test request message forces a heartbeat from the opposing application. The test request message checks sequence numbers or verifies communication line status. The opposite application responds to the Test Request with a Heartbeat containing the TestReqID.

The TestReqID verifies that the opposite application is generating the heartbeat as the result of Test Request and not a normal timeout. The opposite application includes the TestReqID in the resulting Heartbeat. Any string can be used as the TestReqID (one suggestion is to use a timestamp string).

The test request format is as follows:

Test Request

<i>Tag</i>	<i>Field Name</i>	<i>Req'd</i>	<i>Comments</i>
	<i>Standard Header</i>	Y	MsgType = 1
112	TestReqID	Y	
	<i>Standard Trailer</i>	Y	

Resend Request -

The resend request is sent by the receiving application to initiate the retransmission of messages. This function is utilized if a sequence number gap is detected, if the receiving application lost a message, or as a function of the initialization process.

The resend request can be used to request a single message, a range of messages or all messages subsequent to a particular message.

Note: the sending application may wish to consider the message type when resending messages; e.g. if a new order is in the resend series and a significant time period has elapsed since its original inception, the sender may not wish to retransmit the order given the potential for changed market conditions. (The Sequence Reset-GapFill message is used to skip messages that a sender does not wish to resend.)

Note: it is imperative that the receiving application process messages in sequence order, e.g. if message number 7 is missed and 8-9 received, the application should ignore 8 and 9 and ask for a resend of 7-9, or, preferably, 7-0 (0 represents infinity). This latter approach is strongly recommended to recover from out of sequence conditions as it allows for faster recovery in the presence of certain race conditions when both sides are simultaneously attempting to recover a gap.

- To request a single message: BeginSeqNo = EndSeqNo
- To request a range of messages: BeginSeqNo = first message of range, EndSeqNo = last message of range
- To request all messages subsequent to a particular message: BeginSeqNo = first message of range, EndSeqNo = 0 (represents infinity) .

The resend request format is as follows:

Resend Request

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = 2
7	BeginSeqNo	Y	
16	EndSeqNo	Y	
	Standard Trailer	Y	

Reject (session-level) -

The reject message should be issued when a message is received but cannot be properly processed due to a session-level rule violation. An example of when a reject may be appropriate would be the receipt of a message with invalid basic data (e.g. MsgType=8) which successfully passes de-encryption, CheckSum and BodyLength checks. As a rule, messages should be forwarded to the trading application for business level rejections whenever possible.

Rejected messages should be logged and the incoming sequence number incremented.

Note: The receiving application should disregard any message that is garbled, cannot be parsed or fails a data integrity check. Processing of the next valid FIX message will cause detection of a sequence gap and a Resend Request will be generated. Logic should be included in the FIX engine to recognize the possible infinite resend loop, which may be encountered in this situation.

Generation and receipt of a Reject message indicates a serious error that may be the result of faulty logic in either the sending or receiving application.

If the sending application chooses to retransmit the rejected message, it should be assigned a new sequence number and sent with PossResend=Y.

Whenever possible, it is strongly recommended that the cause of the failure be described in the Text field (e.g. INVALID DATA - FIELD 35).

If an application-level message received fulfills session-level rules, it should then be processed at a business message-level. If this processing detects a rule violation, a business-level reject should be issued. Many business-level messages have specific “reject” messages, which should be used. All others can be rejected at a business-level via the Business Message Reject message. [See the Business Message Reject message](#)

Note that in the event a business message is received, fulfills session-level rules, however, the message cannot be communicated to the business-level processing system, a Business Message Reject with BusinessRejectReason = “Application not available at this time” should be issued.

Scenarios for session-level Reject:

SessionRejectReason
0 = Invalid tag number
1 = Required tag missing
2 = Tag not defined for this message type
3 = Undefined Tag
4 = Tag specified without a value
5 = Value is incorrect (out of range) for this tag
6 = Incorrect data format for value
7 = Decryption problem
8 = Signature problem
9 = CompID problem
10 = SendingTime accuracy problem
11 = Invalid MsgType

(Note other session-level rule violations may exist in which case SessionRejectReason is not specified)

The reject format is as follows:

Reject

<i>Tag</i>	<i>Field Name</i>	<i>Req'd</i>	<i>Comments</i>
	<i>Standard Header</i>	Y	MsgType = 3
45	RefSeqNum	Y	MsgSeqNum of rejected message
371	RefTagID	N	The tag number of the FIX field being referenced.
372	RefMsgType	N	The MsgType of the FIX message being referenced.
373	SessionRejectReason	N	Code to identify reason for a session-level Reject message.
58	Text	N	Where possible, message to explain reason for rejection
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
	<i>Standard Trailer</i>	Y	

Sequence Reset (Gap Fill) -

The sequence reset message is used by the sending application to reset the incoming sequence number on the opposing side. This message has two modes: "Sequence Reset-Gap Fill" when GapFillFlag is 'Y' and "Sequence Reset-Reset" when GapFillFlag is N or not present. The "Sequence Reset-Reset" mode should **ONLY** be used to recover from a disaster situation which cannot be otherwise recovered via "Gap Fill" mode. The sequence reset message can be used in the following situations:

- During normal resend processing, the sending application may choose not to send a message (e.g. an aged order). The Sequence Reset – Gap Fill is used to mark the place of that message.
- During normal resend processing, a number of administrative messages are not resent, the Sequence Reset – Gap Fill message is used to fill the sequence gap created.
- In the event of an application failure, it may be necessary to force synchronization of sequence numbers on the sending and receiving sides via the use of Sequence Reset - Reset

The sending application will initiate the sequence reset. **The message in all situations specifies NewSeqNo to reset as the value of the next sequence number immediately following the messages and/or sequence numbers being skipped.**

If the GapFillFlag field is not present (or set to N), it can be assumed that the purpose of the sequence reset message is to recover from an out-of-sequence condition. The MsgSeqNum in the header should be ignored (i.e. the receipt of a Sequence Reset - Reset message with an out of sequence MsgSeqNum should not generate resend requests). ***Sequence Reset – Reset should NOT be used as a normal response to a Resend Request (use Sequence Reset – Gap Fill). The Sequence Reset – Reset should ONLY be used to recover from a disaster situation which cannot be recovered via the use of Sequence Reset – Gap Fill. Note that the use of Sequence Reset – Reset may result in the possibility of lost messages***

If the GapFillFlag field is present (and equal to Y), the MsgSeqNum should conform to standard message sequencing rules (i.e. the MsgSeqNum of the Sequence Reset-GapFill message should represent the beginning MsgSeqNum in the GapFill range because the remote side is expecting that next message).

The sequence reset can only increase the sequence number. If a sequence reset is received attempting to decrease the next expected sequence number the message should be rejected and treated as a serious error. It is possible to have multiple ResendRequests issued in a row (i.e. 5 to 10 followed by 5 to 11). If sequence number 8, 10, and 11 represent application messages while the 5-7 and 9 represent administrative messages, the series of messages as result of the Resend Request may appear as SeqReset-GapFill with NewSeqNo of 8, message 8, SeqReset-GapFill with NewSeqNo of 10, and message 10. This could then followed by SeqReset-GapFill with NewSeqNo of 8, message 8, SeqReset-GapFill with NewSeqNo of 10, message 10, and message 11. One must be careful to ignore the duplicate SeqReset-GapFill which is attempting to lower the next expected sequence number. This can be detected by checking to see if its MsgSeqNum is less than expected. If so, the SeqReset-GapFill is a duplicate and should be discarded.

The Sequence Reset format is as follows:

Sequence Reset

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = 4

123	GapFillFlag	N	
36	NewSeqNo	Y	
	<i>Standard Trailer</i>	Y	

Logout -

The logout message initiates or confirms the termination of a FIX session. Disconnection without the exchange of logout messages should be interpreted as an abnormal condition.

Before actually closing the session, the logout initiator should wait for the opposite side to respond with a confirming logout message. This gives the remote end a chance to perform any Gap Fill operations that may be necessary. The session may be terminated if the remote side does not respond in an appropriate timeframe.

After sending the Logout message, the logout initiator should not send any messages unless requested to do so by the logout acceptor via a ResendRequest.

The logout format is as follows:

Logout

<i>Tag</i>	<i>Field Name</i>	<i>Req'd</i>	<i>Comments</i>
	<i>Standard Header</i>	Y	MsgType = 5
58	Text	N	
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
	<i>Standard Trailer</i>	Y	

APPLICATION MESSAGES

The exchange of business related information is accomplished through the passing of application messages. The application message is composed of the standard header followed by the message body and trailer.

Descriptions and formats of the specific messages follow.

Advertisements -

Advertisement messages are used to announce completed transactions. The advertisement message can be transmitted in various transaction types; NEW, CANCEL and REPLACE. All message types other than NEW modify the state of a previously transmitted advertisement identified in AdvRefID.

The advertisement message format is as follows:

Advertisement

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = 7
2	AdvId	Y	
5	AdvTransType	Y	
3	AdvRefID	N	Required for Cancel and Replace AdvTransType messages
55	Symbol	Y	
65	SymbolSfx	N	
48	SecurityID	N	
22	IDSource	N	
167	SecurityType	N	Must be specified if a Future or Option. If a Future: Symbol, SecurityType, and MaturityMonthYear are required. If an Option: Symbol, SecurityType, MaturityMonthYear, PutOrCall, and StrikePrice are required.
200	MaturityMonthYear	N	Specifies the month and year of maturity. Required if MaturityDay is specified.
205	MaturityDay	N	Can be used in conjunction with MaturityMonthYear to specify a particular maturity date.
201	PutOrCall	N	For Options.
202	StrikePrice	N	For Options.
206	OptAttribute	N	For Options.
231	ContractMultiplier	N	For Fixed Income, Convertible Bonds, Derivatives, etc. Note: If used, quantities should be expressed in the "nominal" (e.g. contracts vs. shares) amount.
223	CouponRate	N	For Fixed Income.
207	SecurityExchange	N	Can be used to identify the security.

106	Issuer	N	
348	EncodedIssuerLen	N	Must be set if EncodedIssuer field is specified and must immediately precede it.
349	EncodedIssuer	N	Encoded (non-ASCII characters) representation of the Issuer field in the encoded format specified via the MessageEncoding field.
107	SecurityDesc	N	
350	EncodedSecurityDescLen	N	Must be set if EncodedSecurityDesc field is specified and must immediately precede it.
351	EncodedSecurityDesc	N	Encoded (non-ASCII characters) representation of the SecurityDesc field in the encoded format specified via the MessageEncoding field.
4	AdvSide	Y	
53	Shares	Y	
44	Price	N	
15	Currency	N	
75	TradeDate	N	
60	TransactTime	N	
58	Text	N	
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
149	URLLink	N	A URL (Uniform Resource Locator) link to additional information (i.e. http://www.XYZ.com/research.html)
30	LastMkt	N	
336	TradingSessionID	N	
	<i>Standard Trailer</i>	Y	

Indications of Interest -

Indication of interest messages market merchandise which the broker is buying or selling in either a proprietary or agency capacity. The indications can be time bound with a specific expiration value. Indications are distributed with the understanding that other firms may react to the message first and that the merchandise may no longer be available due to prior trade.

Indication messages can be transmitted in various transaction types; NEW, CANCEL, and REPLACE. All message types other than NEW modify the state of the message identified in IOIRefID.

The indication of interest message format is as follows:

Indication of Interest

Tag	Field Name	Req'd	Comments
	<i>Standard Header</i>	Y	MsgType = 6
23	IOIId	Y	
28	IOITransType	Y	
26	IOIRefID	N	Required for Cancel and Replace IOITransType messages
55	Symbol	Y	
65	SymbolSfx	N	
48	SecurityID	N	
22	IDSSource	N	
167	SecurityType	N	Must be specified if a Future or Option. If a Future: Symbol, SecurityType, and MaturityMonthYear are required. If an Option: Symbol, SecurityType, MaturityMonthYear, PutOrCall, and StrikePrice are required.
200	MaturityMonthYear	N	Specifies the month and year of maturity. Required if MaturityDay is specified.
205	MaturityDay	N	Can be used in conjunction with MaturityMonthYear to specify a particular maturity date.
201	PutOrCall	N	For Options.
202	StrikePrice	N	For Options.
206	OptAttribute	N	For Options.
231	ContractMultiplier	N	For Fixed Income, Convertible Bonds, Derivatives, etc. Note: If used, quantities should be expressed in the "nominal" (e.g. contracts vs. shares) amount.
223	CouponRate	N	For Fixed Income.
207	SecurityExchange	N	Can be used to identify the security.
106	Issuer	N	
348	EncodedIssuerLen	N	Must be set if EncodedIssuer field is specified and must immediately precede it.

349	EncodedIssuer	N	Encoded (non-ASCII characters) representation of the Issuer field in the encoded format specified via the MessageEncoding field.
107	SecurityDesc	N	
350	EncodedSecurityDescLen	N	Must be set if EncodedSecurityDesc field is specified and must immediately precede it.
351	EncodedSecurityDesc	N	Encoded (non-ASCII characters) representation of the SecurityDesc field in the encoded format specified via the MessageEncoding field.
54	Side	Y	Side of Indication Valid values: 1 = Buy 2 = Sell 7 = Undisclosed (for IOIs)
27	IOIShares	Y	
44	Price	N	
15	Currency	N	
62	ValidUntilTime	N	
25	IOIQltyInd	N	
130	IOINaturalFlag	N	
199	NoIOIQualifiers	N	Required if any IOIQualifiers are specified. Indicates the number of repeating IOIQualifiers.
→	104	IOIQualifier	N Required if NoIOIQualifiers > 0
58	Text	N	
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
60	TransactTime	N	
149	URLLink	N	A URL (Uniform Resource Locator) link to additional information (i.e. http://www.XYZ.com/research.html)
215	NoRoutingIDs	N	Required if any RoutingType and RoutingIDs are specified. Indicates the number within repeating group.
→	216	RoutingType	N Indicates type of RoutingID. Required if NoRoutingIDs is > 0.
→	217	RoutingID	N Identifies routing destination. Required if NoRoutingIDs is > 0.
218	SpreadToBenchmark	N	For Fixed Income
219	Benchmark	N	For Fixed Income

	<i>Standard Trailer</i>	Y	
--	-------------------------	---	--

News -

The news message is a general free format message between the broker and institution. The message contains flags to identify the news item's urgency and to allow sorting by subject company (symbol). The News message can be originated at either the broker or institution side.

The news message format is as follows:

News

Tag	Field Name		Req'd	Comments
	<i>Standard Header</i>		Y	MsgType = B
42	OrigTime		N	
61	Urgency		N	
148	Headline		Y	Specifies the headline text
358	EncodedHeadlineLen		N	Must be set if EncodedHeadline field is specified and must immediately precede it.
359	EncodedHeadline		N	Encoded (non-ASCII characters) representation of the Headline field in the encoded format specified via the MessageEncoding field.
215	NoRoutingIDs		N	Required if any RoutingType and RoutingIDs are specified. Indicates the number within repeating group.
→	216	<i>RoutingType</i>	N	Indicates type of RoutingID. Required if NoRoutingIDs is > 0.
→	217	<i>RoutingID</i>	N	Identifies routing destination. Required if NoRoutingIDs is > 0.
146	NoRelatedSym		N	Specifies the number of repeating symbols specified
→	46	<i>RelatdSym</i>	N	Can be repeated multiple times if message is related to multiple symbols.
→	65	<i>SymbolSfx</i>	N	Can be repeated multiple times if message is related to multiple symbols.
→	48	<i>SecurityID</i>	N	Can be repeated multiple times if message is related to multiple symbols.
→	22	<i>IDSource</i>	N	Can be repeated multiple times if message is related to multiple symbols.
→	167	<i>SecurityType</i>	N	Must be specified if a Future or Option. If a Future: RelatdSym, SecurityType, and MaturityMonthYear are required. If an Option: RelatdSym, SecurityType, MaturityMonthYear, PutOrCall, and StrikePrice are required.
→	200	<i>MaturityMonthYear</i>	N	Specifies the month and year of maturity. Required if MaturityDay is specified.
→	205	<i>MaturityDay</i>	N	Can be used in conjunction with MaturityMonthYear to specify a particular maturity date.

→	201	<i>PutOrCall</i>	N	For Options.
→	202	<i>StrikePrice</i>	N	For Options.
→	206	<i>OptAttribute</i>	N	For Options.
→	231	<i>ContractMultiplier</i>	N	For Fixed Income, Convertible Bonds, Derivatives, etc. Note: If used, quantities should be expressed in the "nominal" (e.g. contracts vs. shares) amount.
→	223	<i>CouponRate</i>	N	For Fixed Income.
→	207	<i>SecurityExchange</i>	N	Can be used to identify the security.
→	106	<i>Issuer</i>	N	Can be repeated multiple times if message is related to multiple symbols.
→	348	<i>EncodedIssuerLen</i>	N	Must be set if EncodedIssuer field is specified and must immediately precede it.
→	349	<i>EncodedIssuer</i>	N	Encoded (non-ASCII characters) representation of the Issuer field in the encoded format specified via the MessageEncoding field.
→	107	<i>SecurityDesc</i>	N	Can be repeated multiple times if message is related to multiple symbols.
→	350	<i>EncodedSecurityDesc Len</i>	N	Must be set if EncodedSecurityDesc field is specified and must immediately precede it.
→	351	<i>EncodedSecurityDesc</i>	N	Encoded (non-ASCII characters) representation of the SecurityDesc field in the encoded format specified via the MessageEncoding field.
33	LinesOfText		Y	Specifies the number of repeating lines of text specified
→	58	<i>Text</i>	Y	Repeating field, number of instances defined in LinesOfText
→	354	<i>EncodedTextLen</i>	N	Must be set if EncodedText field is specified and must immediately precede it.
→	355	<i>EncodedText</i>	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
149	URLLink		N	A URL (Uniform Resource Locator) link to additional information (i.e. http://www.XYZ.com/research.html)
95	RawDataLength		N	
96	RawData		N	
	<i>Standard Trailer</i>		Y	

Email -

The email message is similar to the format and purpose of to the News message, however, it is intended for private use between two parties.

The email message format is as follows:

Email

Tag	Field Name		Req'd	Comments
	<i>Standard Header</i>		Y	MsgType = C
164	EmailThreadID		Y	Unique identifier for the email message thread
94	EmailType		Y	
42	OrigTime		N	
147	Subject		Y	Specifies the Subject text
356	EncodedSubjectLen		N	Must be set if EncodedSubject field is specified and must immediately precede it.
357	EncodedSubject		N	Encoded (non-ASCII characters) representation of the Subject field in the encoded format specified via the MessageEncoding field.
215	NoRoutingIDs		N	Required if any RoutingType and RoutingIDs are specified. Indicates the number within repeating group.
→	216	<i>RoutingType</i>	N	Indicates type of RoutingID. Required if NoRoutingIDs is > 0.
→	217	<i>RoutingID</i>	N	Identifies routing destination. Required if NoRoutingIDs is > 0.
146	NoRelatedSym		N	Specifies the number of repeating symbols specified
→	46	<i>RelatdSym</i>	N	Can be repeated multiple times if message is related to multiple symbols.
→	65	<i>SymbolSfx</i>	N	Can be repeated multiple times if message is related to multiple symbols.
→	48	<i>SecurityID</i>	N	Can be repeated multiple times if message is related to multiple symbols.
→	22	<i>IDSource</i>	N	Can be repeated multiple times if message is related to multiple symbols.
→	167	<i>SecurityType</i>	N	Must be specified if a Future or Option. If a Future: RelatdSym, SecurityType, and MaturityMonthYear are required. If an Option: RelatdSym, SecurityType, MaturityMonthYear, PutOrCall, and StrikePrice are required.
→	200	<i>MaturityMonthYear</i>	N	Specifies the month and year of maturity. Required if MaturityDay is specified.
→	205	<i>MaturityDay</i>	N	Can be used in conjunction with MaturityMonthYear to specify a particular maturity date.

→	201	<i>PutOrCall</i>	N	For Options.
→	202	<i>StrikePrice</i>	N	For Options.
→	206	<i>OptAttribute</i>	N	For Options.
→	231	<i>ContractMultiplier</i>	N	For Fixed Income, Convertible Bonds, Derivatives, etc. Note: If used, quantities should be expressed in the "nominal" (e.g. contracts vs. shares) amount.
→	223	<i>CouponRate</i>	N	For Fixed Income.
→	207	<i>SecurityExchange</i>	N	Can be used to identify the security.
→	106	<i>Issuer</i>	N	Can be repeated multiple times if message is related to multiple symbols.
→	348	<i>EncodedIssuerLen</i>	N	Must be set if EncodedIssuer field is specified and must immediately precede it.
→	349	<i>EncodedIssuer</i>	N	Encoded (non-ASCII characters) representation of the Issuer field in the encoded format specified via the MessageEncoding field.
→	107	<i>SecurityDesc</i>	N	Can be repeated multiple times if message is related to multiple symbols.
→	350	<i>EncodedSecurityDesc Len</i>	N	Must be set if EncodedSecurityDesc field is specified and must immediately precede it.
→	351	<i>EncodedSecurityDesc</i>	N	Encoded (non-ASCII characters) representation of the SecurityDesc field in the encoded format specified via the MessageEncoding field.
37	OrderID		N	
11	ClOrdID		N	
33	LinesOfText		Y	Specifies the number of repeating lines of text specified
→	58	<i>Text</i>	Y	Repeating field, number of instances defined in LinesOfText
→	354	<i>EncodedTextLen</i>	N	Must be set if EncodedText field is specified and must immediately precede it.
→	355	<i>EncodedText</i>	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
95	RawDataLength		N	
96	RawData		N	
	<i>Standard Trailer</i>		Y	

Quote Request -

In some markets it is the practice to request quotes from brokers prior to placement of an order. The quote request message is used for this purpose.

Quotes can be requested on specific securities or forex rates.

Securities quotes can be requested as either market quotes or for a specific quantity and side. If OrderQty and Side are absent, a market-style quote (bid x offer, size x size) will be returned.

If the message is used for foreign exchange, conventions for identifying the forex transaction are as follows:

- The forex Symbol is defined in "EBS" (Electronic Banking System) format: "CCY1/CCY2".
 - Rates are expressed as "currency1 **in** currency2" (or "currency2 per currency1") and are calculated as CCY2 divided by CCY1 (**NOT** CCY1 divided by CCY2)
 - e.g. "GBP/USD" represents a rate expressed as USD per GBP, "USD/JPY" represents a rate expressed as JPY per USD, etc.).
 - CCY1 and CCY2 are ISO currency codes
- The value of the Currency field represents the denomination of the quantity fields (e.g. JPY represents quantity of JPY).
- [See Appendix O – Foreign Exchange Trading](#)
- Forex quotes can be requested as indicative or at a specific quantity level. If an indicative quote is requested (OrderQty and Side are absent), the broker has discretion to quote at either a specific trade level and side or to provide an indicative quote at the mid-point of the spread. The broker can also choose to respond to an indicative quote by sending multiple quote messages specifying various levels and sides.

The quote request message format is as follows:

Quote Request

Tag	Field Name		Req'd	Comments
	<i>Standard Header</i>		Y	MsgType = R
131	QuoteReqID		Y	
146	NoRelatedSym		Y	Number of related symbols in Request
→	55	<i>Symbol</i>	Y	Must be the first field in the repeating group.
→	65	<i>SymbolSfx</i>	N	Can be repeated multiple times if message is related to multiple symbols.
→	48	<i>SecurityID</i>	N	Can be repeated multiple times if message is related to multiple symbols.
→	22	<i>IDSource</i>	N	Can be repeated multiple times if message is related to multiple symbols.

→	167	<i>SecurityType</i>	N	Must be specified if a Future or Option. If a Future:Symbol, SecurityType, and MaturityMonthYear are required. If an Option:Symbol, SecurityType, MaturityMonthYear, PutOrCall, and StrikePrice are required.
→	200	<i>MaturityMonthYear</i>	N	Specifies the month and year of maturity. Required if MaturityDay is specified.
→	205	<i>MaturityDay</i>	N	Can be used in conjunction with MaturityMonthYear to specify a particular maturity date.
→	201	<i>PutOrCall</i>	N	For Options.
→	202	<i>StrikePrice</i>	N	For Options.
→	206	<i>OptAttribute</i>	N	For Options.
→	231	<i>ContractMultiplier</i>	N	For Fixed Income, Convertible Bonds, Derivatives, etc. Note: If used, quantities should be expressed in the "nominal" (e.g. contracts vs. shares) amount.
→	223	<i>CouponRate</i>	N	For Fixed Income.
→	207	<i>SecurityExchange</i>	N	Can be used to identify the security.
→	106	<i>Issuer</i>	N	Can be repeated multiple times if message is related to multiple symbols.
→	348	<i>EncodedIssuerLen</i>	N	Must be set if EncodedIssuer field is specified and must immediately precede it.
→	349	<i>EncodedIssuer</i>	N	Encoded (non-ASCII characters) representation of the Issuer field in the encoded format specified via the MessageEncoding field.
→	107	<i>SecurityDesc</i>	N	Can be repeated multiple times if message is related to multiple symbols.
→	350	<i>EncodedSecurityDescLen</i>	N	Must be set if EncodedSecurityDesc field is specified and must immediately precede it.
→	351	<i>EncodedSecurityDesc</i>	N	Encoded (non-ASCII characters) representation of the SecurityDesc field in the encoded format specified via the MessageEncoding field.
→	140	<i>PrevClosePx</i>	N	Useful for verifying security identification
→	303	<i>QuoteRequestType</i>	N	Indicates the type of Quote Request (e.g. Manual vs. Automatic) being generated.
→	336	<i>TradingSessionID</i>	N	
→	54	<i>Side</i>	N	If OrdType = "Forex - Swap", should be the side of the future portion of a F/X swap
→	38	<i>OrderQty</i>	N	
→	64	<i>FutSettDate</i>	N	Can be used with forex quotes to specify the desired "value date"

→	40	<i>OrdType</i>	N	Can be used to specify the type of order the quote request is for
→	193	<i>FutSettDate2</i>	N	Can be used with OrdType = “Forex - Swap” to specify the “value date” for the future portion of a F/X swap.
→	192	<i>OrderQty2</i>	N	Can be used with OrdType = “Forex - Swap” to specify the order quantity for the future portion of a F/X swap.
→	126	<i>ExpireTime</i>	N	The time when Quote Request will expire.
→	60	<i>TransactTime</i>	N	Time transaction was entered
→	15	<i>Currency</i>	N	Can be used to specify the currency of the quoted price.
	<i>Standard Trailer</i>		Y	

Quote -

The quote message is used as the response to a Quote Request message and can be used to publish unsolicited quotes.

Quotes supplied as the result of a Quote Request message are tagged with the appropriate QuoteReqID, unsolicited quotes can be identified by the absence of a QuoteReqID.

If the message is used for foreign exchange, conventions for identifying the forex transaction are as follows:

- The forex Symbol is defined in "EBS" (Electronic Banking System) format: "CCY1/CCY2".
 - Rates are expressed as "currency1 **in** currency2" (or "currency2 per currency1") and are calculated as CCY2 divided by CCY1 (**NOT** CCY1 divided by CCY2)
 - e.g. "GBP/USD" represents a rate expressed as USD per GBP, "USD/JPY" represents a rate expressed as JPY per USD, etc.).
 - CCY1 and CCY2 are ISO currency codes
- The value of the Currency field represents the denomination of the quantity fields (e.g. JPY represents quantity of JPY).
- [See Appendix O – Foreign Exchange Trading](#)

Orders can be generated based on Quotes. Quoted orders include the QuoteID and are OrdType=Previously Quoted or Forex - Previously Quoted.

The quote message format is as follows:

Quote

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = S
131	QuoteReqID	N	Required when quote is in response to a Quote Request message
117	QuoteID	Y	
301	QuoteResponseLevel	N	Level of Response requested from receiver of quote messages.
336	TradingSessionID	N	
55	Symbol	Y	
65	SymbolSfx	N	
48	SecurityID	N	
22	IDSSource	N	

167	SecurityType	N	Must be specified if a Future or Option. If a Future: Symbol, SecurityType, and MaturityMonthYear are required. If an Option: Symbol, SecurityType, MaturityMonthYear, PutOrCall, and StrikePrice are required.
200	MaturityMonthYear	N	Specifies the month and year of maturity. Required if MaturityDay is specified.
205	MaturityDay	N	Can be used in conjunction with MaturityMonthYear to specify a particular maturity date.
201	PutOrCall	N	For Options.
202	StrikePrice	N	For Options.
206	OptAttribute	N	For Options.
231	ContractMultiplier	N	For Fixed Income, Convertible Bonds, Derivatives, etc. Note: If used, quantities should be expressed in the "nominal" (e.g. contracts vs. shares) amount.
223	CouponRate	N	For Fixed Income.
207	SecurityExchange	N	Can be used to identify the security.
106	Issuer	N	
348	EncodedIssuerLen	N	Must be set if EncodedIssuer field is specified and must immediately precede it.
349	EncodedIssuer	N	Encoded (non-ASCII characters) representation of the Issuer field in the encoded format specified via the MessageEncoding field.
107	SecurityDesc	N	
350	EncodedSecurityDescLen	N	Must be set if EncodedSecurityDesc field is specified and must immediately precede it.
351	EncodedSecurityDesc	N	Encoded (non-ASCII characters) representation of the SecurityDesc field in the encoded format specified via the MessageEncoding field.
132	BidPx	N	If F/X quote, should be the "all-in" rate (spot rate adjusted for forward points). Note that either BidPx, OfferPx or both must be specified.
133	OfferPx	N	If F/X quote, should be the "all-in" rate (spot rate adjusted for forward points). Note that either BidPx, OfferPx or both must be specified.
134	BidSize	N	
135	OfferSize	N	
62	ValidUntilTime	N	
188	BidSpotRate	N	May be applicable for F/X quotes
190	OfferSpotRate	N	May be applicable for F/X quotes

189	BidForwardPoints	N	May be applicable for F/X quotes
191	OfferForwardPoints	N	May be applicable for F/X quotes
60	TransactTime	N	
64	FutSettDate	N	Can be used with forex quotes to specify a specific “value date”
40	OrdType	N	Can be used to specify the type of order the quote is for
193	FutSettDate2	N	Can be used with OrdType = “Forex - Swap” to specify the “value date” for the future portion of a F/X swap.
192	OrderQty2	N	Can be used with OrdType = “Forex - Swap” to specify the order quantity for the future portion of a F/X swap.
15	Currency	N	Can be used to specify the currency of the quoted price.
	<i>Standard Trailer</i>	Y	

Example: Quote for Single Security

QuoteID=XXX
 QuoteReqID=YYY
 Symbol=AA
 MaturityMonthYear=199901
 StrikePrice=25.00
 PutOrCall=1
 BixPx=5.00
 OfferPx=5.25
 BidSize=10
 OfferSize=10

Mass Quote –

The Mass Quote message can contain quotes for multiple securities to support applications that allow for the mass quoting of an option series. Two levels of repeating groups have been provided to minimize the amount of data required to submit a set of quotes for a class of options (e.g. all option series for IBM).

A QuoteSet specifies the first level of repeating tags for the Mass Quote message. It represents a group of related quotes and can, for example, represent an option class.

Each QuoteSet contains an optional repeating group of QuoteEntries which can represent an option series.

It is possible the number of Quote Entries for a Quote Set (option class) could exceed one's physical or practical message size. It may be necessary to fragment a message across multiple quote messages. Message size limits must be mutually agreed to with one's counterparties.

The grouping of quotes is as follows:

NoQuoteSets – specifies the number of sets of quotes contained in the message

QuoteSetID – Is a unique ID given to the quote set

Information regarding the security to which all of the quotes belong

TotQuoteEntries – defines the number of quotes for the quote set across all messages

NoQuoteEntries – defines the number of quotes contained within this message for this quote set

QuoteEntryID – Is a unique ID given to a specific quote entry

Information regarding the specific quote (bid/ask size and price)

If there are too many Quote Entries for a Quote Set to fit into one physical message, then the quotes can be continued in another Mass Quote message by repeating all of the QuoteSet information and then specifying the number of Quote Entries (related symbols) in the continued message. The TotQuoteEntries is provided to optionally indicate to the counterparty the total number of Quote Entries for a Quote Set in multiple quote messages. This permits, but does not require, a receiving application to react in a stateful manner where it can determine if it has received all quotes for a Quote Set before carrying out some action. However, the overall approach to fragmentation is to permit each mass quote message to be processed in a stateless manner as it is received. Each mass quote message should contain enough information to have the Quote Entries applied to a market without requiring the next message if fragmentation has occurred. Also, a continued message should not require any information from the previous message.

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

Requesting Acknowledgement for Mass Quotes

Applications can optionally support acknowledgement of quotes using the *QuoteResponseLevel* tag. The *QuoteResponseLevel* is used to specify the level of acknowledgement requested from the counterparty. A *QuoteResponseLevel* of 0 indicates that no acknowledgement is requested. A *ResponseLevel* of 1 requests acknowledgement of invalid or erroneous quotes. A *QuoteResponseLevel* of 2 requests acknowledgement of each Mass Quote message.

[See Appendix H: Mass Quote Message Scenarios](#)

The Mass Quote message format is as follows:

Mass Quote

Tag	Field Name		Req'd	Comments
	<i>Standard Header</i>		Y	MsgType = i (lowercase)
131	QuoteReqID		N	Required when quote is in response to a Quote Request message
117	QuoteID		Y	
301	QuoteResponseLevel		N	Level of Response requested from receiver of quote messages.
293	DefBidSize		N	Default Bid Size for quote contained within this quote message – if not explicitly provided.
294	DefOfferSize		N	Default Offer Size for quotes contained within this quote message – if not explicitly provided.
296	NoQuoteSets		Y	The number of sets of quotes in the message
→	302	<i>QuoteSetID</i>	Y	Sequential number for the Quote Set. For a given QuoteID – assumed to start at 1. Must be the first field in the repeating group.
→	311	<i>UnderlyingSymbol</i>	Y	
→	312	<i>UnderlyingSymbolSfx</i>	N	
→	309	<i>UnderlyingSecurityID</i>	N	
→	305	<i>UnderlyingIDSource</i>	N	
→	310	<i>UnderlyingSecurityType</i>	N	
→	313	<i>UnderlyingMaturityMonthYear</i>	N	Required if UnderlyingMaturityDay is specified.
→	314	<i>UnderlyingMaturityDay</i>	N	
→	315	<i>UnderlyingPutOrCall</i>	N	
→	316	<i>UnderlyingStrikePrice</i>	N	

→	317	<i>UnderlyingOptAttribute</i>		N	
→	436	<i>UnderlyingContractMultiplier</i>		N	For Fixed Income, Convertible Bonds, Derivatives, etc.
→	435	<i>UnderlyingCouponRate</i>		N	For Fixed Income.
→	308	<i>UnderlyingSecurityExchange</i>		N	
→	306	<i>UnderlyingIssuer</i>		N	
→	362	<i>EncodedUnderlyingIssuerLen</i>		N	Must be set if EncodedUnderlyingIssuer field is specified and must immediately precede it.
→	363	<i>EncodedUnderlyingIssuer</i>		N	Encoded (non-ASCII characters) representation of the UnderlyingIssuer field in the encoded format specified via the MessageEncoding field.
→	307	<i>UnderlyingSecurityDesc</i>		N	
→	364	<i>EncodedUnderlyingSecurityDescLen</i>		N	Must be set if EncodedUnderlyingSecurityDesc field is specified and must immediately precede it.
→	365	<i>EncodedUnderlyingSecurityDesc</i>		N	Encoded (non-ASCII characters) representation of the UnderlyingSecurityDesc field in the encoded format specified via the MessageEncoding field.
→	367	<i>QuoteSetValidUntilTime</i>		N	
→	304	<i>TotQuoteEntries</i>		Y	Total number of quotes for the quote set across all messages. Should be the sum of all NoQuoteEntries in each message that has repeating quotes that are part of the same quote set.
→	295	<i>NoQuoteEntries</i>		Y	The number of quotes for this Symbol (QuoteSet) that follow in this message. ** Nested Repeating Group follows **
→	→	299	<i>QuoteEntryID</i>	Y	Uniquely identifies the quote as part of a QuoteSet. Must be used if NoQuoteEntries is used
→	→	55	<i>Symbol</i>	N	
→	→	65	<i>SymbolSfx</i>	N	
→	→	48	<i>SecurityID</i>	N	
→	→	22	<i>IDSource</i>	N	

→	→	167	<i>SecurityType</i>	N	Must be specified if a Future or Option. If a Future: Symbol, SecurityType, and MaturityMonthYear are required. If an Option: Symbol, SecurityType, MaturityMonthYear, PutOrCall, and StrikePrice are required.
→	→	200	<i>MaturityMonthYear</i>	N	Specifies the month and year of maturity. Required if MaturityDay is specified.
→	→	205	<i>MaturityDay</i>	N	Can be used in conjunction with MaturityMonthYear to specify a particular maturity date.
→	→	201	<i>PutOrCall</i>	N	For Options.
→	→	202	<i>StrikePrice</i>	N	For Options.
→	→	206	<i>OptAttribute</i>	N	For Options.
→	→	231	<i>ContractMultiplier</i>	N	For Fixed Income, Convertible Bonds, Derivatives, etc. Note: If used, quantities should be expressed in the "nominal" (e.g. contracts vs. shares) amount.
→	→	223	<i>CouponRate</i>	N	For Fixed Income.
→	→	207	<i>SecurityExchange</i>	N	Can be used to identify the security.
→	→	106	<i>Issuer</i>	N	
→	→	348	<i>EncodedIssuerLen</i>	N	Must be set if EncodedIssuer field is specified and must immediately precede it.
→	→	349	<i>EncodedIssuer</i>	N	Encoded (non-ASCII characters) representation of the Issuer field in the encoded format specified via the MessageEncoding field.
→	→	107	<i>SecurityDesc</i>	N	
→	→	350	<i>EncodedSecurityDescLen</i>	N	Must be set if EncodedSecurityDesc field is specified and must immediately precede it.
→	→	351	<i>EncodedSecurityDesc</i>	N	Encoded (non-ASCII characters) representation of the SecurityDesc field in the encoded format specified via the MessageEncoding field.
→	→	132	<i>BidPx</i>	N	If F/X quote, should be the "all-in" rate (spot rate adjusted for forward points). Note that either BidPx, OfferPx or both must be specified.
→	→	133	<i>OfferPx</i>	N	If F/X quote, should be the "all-in" rate (spot rate adjusted for forward points). Note that either BidPx, OfferPx or both must be specified.
→	→	134	<i>BidSize</i>	N	

→	→	135	<i>OfferSize</i>	N	
→	→	62	<i>ValidUntilTime</i>	N	
→	→	188	<i>BidSpotRate</i>	N	May be applicable for F/X quotes
→	→	190	<i>OfferSpotRate</i>	N	May be applicable for F/X quotes
→	→	189	<i>BidForwardPoints</i>	N	May be applicable for F/X quotes
→	→	191	<i>OfferForwardPoints</i>	N	May be applicable for F/X quotes
→	→	60	<i>TransactTime</i>	N	
→	→	336	<i>TradingSessionID</i>	N	
→	→	64	<i>FutSettDate</i>	N	Can be used with forex quotes to specify a specific "value date"
→	→	40	<i>OrdType</i>	N	Can be used to specify the type of order the quote is for
→	→	193	<i>FutSettDate2</i>	N	Can be used with OrdType = "Forex - Swap" to specify the "value date" for the future portion of a F/X swap.
→	→	192	<i>OrderQty2</i>	N	Can be used with OrdType = "Forex - Swap" to specify the order quantity for the future portion of a F/X swap.
→	→	15	<i>Currency</i>	N	Can be used to specify the currency of the quoted price.
			<i>Standard Trailer</i>	Y	

Notes on usage for Options Markets:

It is assumed that for many options markets, the Mass Quote message will be used to generate quotes in high volumes in an unsolicited manner. This means that multiple quotes will be sent to the counterparty (an exchange) without acknowledgement. The Mass Quote message can be used to send quotes for multiple classes, each with multiple series.

Example: Multiple Option Series for a single Option Class (No Fragmentation)

QuoteID=XXX
QuoteReqID=YYY
NoQuoteSets=1
QuoteSetID=1
Symbol=AA
TotQuoteEntries=2
NoQuoteEntries=2
Other quote set fields
QuoteEntryID=1
MaturyMonthYear=199901
StrikePrice=25.00
PutOrCall=1

BixPx=5.00
OfferPx=5.25
BidSize=10
OfferSize=10
QuoteEntryID=2
MaturityMonthYear=199901
StrikePrice=30.00
PutOrCall=1
BixPx=3.00
OfferPx=3.25
BidSize=10
OfferSize=10

Example: Multiple Option Series for a single Option Class (Fragmentation)

First Message:

QuoteID=XXX
QuoteReqID=YYY
NoQuoteSets=1
QuoteSetID=1
Symbol=AA
TotQuoteEntries=3
NoQuoteEntries=2
Other quote set fields
QuoteEntryID=1
MaturityMonthYear=199901
StrikePrice=25.00
PutOrCall=1
BixPx=5.00
OfferPx=5.25
BidSize=10
OfferSize=10
QuoteEntryID=2
MaturityMonthYear=199901
StrikePrice=30.00
PutOrCall=1
BixPx=3.00
OfferPx=3.25
BidSize=10
OfferSize=10

Second Message:

QuoteID=XXX
QuoteReqID=YYY
NoQuoteSets=1
QuoteSetID=1
Symbol=AA
Other quote set fields
TotQuoteEntries=3
NoQuoteEntries=1
QuoteEntryID=3
MaturityMonthYear=199901
StrikePrice=35.00

PutOrCall=1
BixPx=2.00
OfferPx=2.25
BidSize=10
OfferSize=10

Quote Cancel -

The Quote Cancel message is used by an originator of quotes to cancel quotes.

The Quote Cancel message supports cancellation of:

- All quotes
- Quotes for a specific symbol or security ID
- All quotes for a security type
- All quotes for an underlying

Canceling a Quote is accomplished by indicating the type of cancellation in the QuoteCancelType field.

The Quote Cancellation only applies to quotes made by the current FIX user.

The Quote Cancel message format is as follows:

Quote Cancel

Tag	Field Name		Req'd	Comments
	<i>Standard Header</i>		Y	MsgType = Z
131	QuoteReqID		N	Required when quote is in response to a Quote Request message
117	QuoteID		Y	
298	QuoteCancelType		Y	Identifies the type of Quote Cancel request.
301	QuoteResponseLevel		N	Level of Response requested from receiver of quote messages.
336	TradingSessionID		N	
295	NoQuoteEntries		Y	The number of securities whose quotes are to be canceled
→	55	<i>Symbol</i>	Y	Must be the first field in the repeating group.
→	65	<i>SymbolSfx</i>	N	Can be repeated multiple times if message is related to multiple symbols.
→	48	<i>SecurityID</i>	N	Can be repeated multiple times if message is related to multiple symbols.
→	22	<i>IDSource</i>	N	Can be repeated multiple times if message is related to multiple symbols.
→	167	<i>SecurityType</i>	N	Must be specified if a Future or Option. If a Future:Symbol, SecurityType, and MaturityMonthYear are required. If an Option:Symbol, SecurityType, MaturityMonthYear, PutOrCall, and StrikePrice are required.
→	200	<i>MaturityMonthYear</i>	N	Specifies the month and year of maturity. Required if MaturityDay is specified.

→	205	<i>MaturityDay</i>	N	Can be used in conjunction with MaturityMonthYear to specify a particular maturity date.
→	201	<i>PutOrCall</i>	N	For Options.
→	202	<i>StrikePrice</i>	N	For Options.
→	206	<i>OptAttribute</i>	N	For Options.
→	231	<i>ContractMultiplier</i>	N	For Fixed Income, Convertible Bonds, Derivatives, etc. Note: If used, quantities should be expressed in the "nominal" (e.g. contracts vs. shares) amount.
→	223	<i>CouponRate</i>	N	For Fixed Income.
→	207	<i>SecurityExchange</i>	N	Can be used to identify the security.
→	106	<i>Issuer</i>	N	Can be repeated multiple times if message is related to multiple symbols.
→	348	<i>EncodedIssuerLen</i>	N	Must be set if EncodedIssuer field is specified and must immediately precede it.
→	349	<i>EncodedIssuer</i>	N	Encoded (non-ASCII characters) representation of the Issuer field in the encoded format specified via the MessageEncoding field.
→	107	<i>SecurityDesc</i>	N	Can be repeated multiple times if message is related to multiple symbols.
→	350	<i>EncodedSecurityDescLen</i>	N	Must be set if EncodedSecurityDesc field is specified and must immediately precede it.
→	351	<i>EncodedSecurityDesc</i>	N	Encoded (non-ASCII characters) representation of the SecurityDesc field in the encoded format specified via the MessageEncoding field.
→	311	<i>UnderlyingSymbol</i>	N	The symbol of the underlying security of options that should be canceled.
	<i>Standard Trailer</i>		Y	

Options usage notes:

Normal usage would be to cancel the quotes for a symbol. This is the reason that the use of further nesting similar to the quote is not used in this message. You are able to cancel quotes for specific series by specifying each option series in the repeating group.

It is recommended that all Cancel messages be acknowledged using the Quote Acknowledgement message

Quote Status Request -

The quote status request message is used by the institution to generate an execution report that contains the quote status message back from the counterparty. It is used to retrieve the status of the originating party's quotes and not to obtain market data information.

The format of the quote status request message is:

Quote Status Request

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = a (lowercase)
117	QuoteID	N	
55	Symbol	Y	
65	SymbolSfx	N	
48	SecurityID	N	
22	IDSSource	N	
167	SecurityType	N	Must be specified if a Future or Option. If a Future: Symbol, SecurityType, and MaturityMonthYear are required. If an Option: Symbol, SecurityType, MaturityMonthYear, PutOrCall, and StrikePrice are required.
200	MaturityMonthYear	N	Specifies the month and year of maturity. Required if MaturityDay is specified.
205	MaturityDay	N	Can be used in conjunction with MaturityMonthYear to specify a particular maturity date.
201	PutOrCall	N	For Options.
202	StrikePrice	N	For Options.
206	OptAttribute	N	For Options.
231	ContractMultiplier	N	For Fixed Income, Convertible Bonds, Derivatives, etc. Note: If used, quantities should be expressed in the "nominal" (e.g. contracts vs. shares) amount.
223	CouponRate	N	For Fixed Income.
207	SecurityExchange	N	Can be used to identify the security.
106	Issuer	N	
348	EncodedIssuerLen	N	Must be set if EncodedIssuer field is specified and must immediately precede it.
349	EncodedIssuer	N	Encoded (non-ASCII characters) representation of the Issuer field in the encoded format specified via the MessageEncoding field.
107	SecurityDesc	N	
350	EncodedSecurityDesc Len	N	Must be set if EncodedSecurityDesc field is specified and must immediately precede it.

351	EncodedSecurityDesc	N	Encoded (non-ASCII characters) representation of the SecurityDesc field in the encoded format specified via the MessageEncoding field.
54	Side	N	
336	TradingSessionID	N	
	<i>Standard Trailer</i>	Y	

Option Usage:

To retrieve all quotes for a given underlying symbol, use part of the symbology fields.

Quote Acknowledgement -

An optional response to Quote, Mass Quote, Quote Cancel, and Quote Request message is the Quote Acknowledgement message. The use of the Quote Acknowledgement message is optional per Quote, Mass Quote, or Quote Request message. It is intended to provide application level acknowledgement of quotes. The level of response requested from a receiver of the Quote Acknowledgement message is specified in the QuoteResponseLevel tag.

The Quote Acknowledgement message format is as follows:

Quote Acknowledgement

Tag	Field Name		Req'd	Comments
	<i>Standard Header</i>		Y	MsgType = b (lowercase)
131	QuoteReqID		N	Required when acknowledgment is in response to a Quote Request message
117	QuoteID		N	Required when acknowledgment is in response to a Quote message
297	QuoteAckStatus		Y	Status of the quote acknowledgement.
300	QuoteRejectReason		N	Reason Quote was rejected.
301	QuoteResponseLevel		N	Level of Response requested from receiver of quote messages. Is echoed back to the counterparty.
336	TradingSessionID		N	
58	Text		N	
296	NoQuoteSets		N	The number of sets of quotes in the message
→	302	<i>QuoteSetID</i>	N	First field in repeating group. Required if NoQuoteSets > 0
→	311	<i>UnderlyingSymbol</i>	N	Required if NoQuoteSets > 0
→	312	<i>UnderlyingSymbolSfx</i>	N	
→	309	<i>UnderlyingSecurityID</i>	N	
→	305	<i>UnderlyingISource</i>	N	
→	310	<i>UnderlyingSecurityType</i>	N	
→	313	<i>UnderlyingMaturityMonthYear</i>	N	Required if UnderlyingMaturityDay is specified.
→	314	<i>UnderlyingMaturityDay</i>	N	
→	315	<i>UnderlyingPutOrCall</i>	N	

→	316	<i>UnderlyingStrikePrice</i>		N	
→	317	<i>UnderlyingOptAttribute</i>		N	
→	436	<i>UnderlyingContractMultiplier</i>		N	For Fixed Income, Convertible Bonds, Derivatives, etc.
→	435	<i>UnderlyingCouponRate</i>		N	For Fixed Income.
→	308	<i>UnderlyingSecurityExchange</i>		N	
→	306	<i>UnderlyingIssuer</i>		N	
→	362	<i>EncodedUnderlyingIssuerLen</i>		N	Must be set if EncodedUnderlyingIssuer field is specified and must immediately precede it.
→	363	<i>EncodedUnderlyingIssuer</i>		N	Encoded (non-ASCII characters) representation of the UnderlyingIssuer field in the encoded format specified via the MessageEncoding field.
→	307	<i>UnderlyingSecurityDesc</i>		N	
→	364	<i>EncodedUnderlyingSecurityDescLen</i>		N	Must be set if EncodedUnderlyingSecurityDesc field is specified and must immediately precede it.
→	365	<i>EncodedUnderlyingSecurityDesc</i>		N	Encoded (non-ASCII characters) representation of the UnderlyingSecurityDesc field in the encoded format specified via the MessageEncoding field.
→	304	<i>TotQuoteEntries</i>		N	Total number of quotes for the quote set across all messages. Should be the sum of all NoQuoteEntries in each message that has repeating quotes that are part of the same quote set. Required if NoQuoteEntries > 0
→	295	<i>NoQuoteEntries</i>		N	The number of quotes for this Symbol (QuoteSet) that follow in this message.
→	→	299	<i>QuoteEntryID</i>	N	Uniquely identifies the quote as part of a QuoteSet. First field in repeating group. Required if NoQuoteEntries > 0.
→	→	55	<i>Symbol</i>	N	
→	→	65	<i>SymbolSfx</i>	N	
→	→	48	<i>SecurityID</i>	N	
→	→	22	<i>IDSource</i>	N	

→	→	167	<i>SecurityType</i>	N	Must be specified if a Future or Option. If a Future: Symbol, SecurityType, and MaturityMonthYear are required. If an Option: Symbol, SecurityType, MaturityMonthYear, PutOrCall, and StrikePrice are required.
→	→	200	<i>MaturityMonthYear</i>	N	Specifies the month and year of maturity. Required if MaturityDay is specified.
→	→	205	<i>MaturityDay</i>	N	Can be used in conjunction with MaturityMonthYear to specify a particular maturity date.
→	→	201	<i>PutOrCall</i>	N	For Options.
→	→	202	<i>StrikePrice</i>	N	For Options.
→	→	206	<i>OptAttribute</i>	N	For Options.
→	→	231	<i>ContractMultiplier</i>	N	For Fixed Income, Convertible Bonds, Derivatives, etc. Note: If used, quantities should be expressed in the "nominal" (e.g. contracts vs. shares) amount.
→	→	223	<i>CouponRate</i>	N	For Fixed Income.
→	→	207	<i>SecurityExchange</i>	N	Can be used to identify the security.
→	→	106	<i>Issuer</i>	N	
→	→	348	<i>EncodedIssuerLen</i>	N	Must be set if EncodedIssuer field is specified and must immediately precede it.
→	→	349	<i>EncodedIssuer</i>	N	Encoded (non-ASCII characters) representation of the Issuer field in the encoded format specified via the MessageEncoding field.
→	→	107	<i>SecurityDesc</i>	N	
→	→	350	<i>EncodedSecurityDescLen</i>	N	Must be set if EncodedSecurityDesc field is specified and must immediately precede it.
→	→	351	<i>EncodedSecurityDesc</i>	N	Encoded (non-ASCII characters) representation of the SecurityDesc field in the encoded format specified via the MessageEncoding field.
→	→	368	<i>QuoteEntryRejectReason</i>	N	Reason Quote Entry was rejected.
		<i>Standard Trailer</i>		Y	

Market Data Request -

Some systems allow the transmission of real-time quote, order, trade and/or other price information on a subscription basis. A Market Data Request is a general request for market data on specific securities or forex quotes.

A successful Market Data Request returns one or more Market Data messages containing one or more Market Data Entries. Each Market Data Entry is a Bid, an Offer, a Trade associated with a security, the opening, closing, or settlement price of a security, the value of an index, or the trading session high price, low price, or VWAP. Market Data Entries have a price and usually a quantity associated with them. For example, in an order book environment, requesting just the top of book will result in only two active Market Data Entries at a time – one for the best Bid and one for the best Offer. For a full book, the Bid and Offer side may each have several Market Data Entries. Each Market Data Entry might represent an aggregate for each price tier, and only one Market Data Entry per side per price would be active at a time. This is referred to as an Aggregated book. Or several Market Data Entries at one price tier could each represent a broker, Market Maker, ECN or Exchange's quote in a security, or individual orders in a book. This is a Non-Aggregated book. Alternately, a Market Data Entry could represent a completed trade in a security, the value of an index, the opening, closing, or settlement price of an instrument, or the trading session high price, low price, or VWAP.

If the message is used for foreign exchange, conventions for identifying the forex transaction are as follows:

- The forex Symbol is defined in "EBS" (Electronic Banking System) format: "CCY1/CCY2".
 - Rates are expressed as "currency1 **in** currency2" (or "currency2 per currency1") and are calculated as CCY2 divided by CCY1 (**NOT** CCY1 divided by CCY2)
 - e.g. "GBP/USD" represents a rate expressed as USD per GBP, "USD/JPY" represents a rate expressed as JPY per USD, etc.).
 - CCY1 and CCY2 are ISO currency codes
- The value of the Currency field represents the denomination of the quantity fields (e.g. JPY represents quantity of JPY).
- [See Appendix O – Foreign Exchange Trading](#)

A Snapshot causes the current state of the market to be sent. A Snapshot + Updates causes the current state of the market to be sent, and any updates as they occur, until the client requests that the Snapshot + Updates be disabled.

When just a Snapshot is requested, the complete data for only one security or forex quote will be returned per FIX Market Data message.

When Snapshot + Updates is requested, updates may be full or incremental:

- Full Refresh. This mode is optimized to trade off increased bandwidth for simplicity in processing and is intended for requests on only a few instruments. Each FIX Market Data message in response to the request will contain the complete data requested for one instrument. If more than just the top of book is requested, this means that both sides, and all price tiers, must be reported in that Market Data message.
- Incremental Refresh. This mode is optimized for handling requests for many instruments while conserving bandwidth. Each Market Data Entry is assigned an MDEntryID unique among all other active entries, and several incremental updates of entries for different instruments can be included in one FIX Market Data message.

One specifies whether a list of trades, a 1-sided or 2-sided book, index, opening, closing, settlement, high, low and VWAP prices should be returned by using the NoMDEntryTypes field and MDEntryType repeating group to list all MDEntryType values that should be returned.

While this document specifies many parameters and modes in a request, the recipient of the request is not required to support all of them. A Market Data Request Reject may be sent in response to a request indicating that it cannot be honored.

The Market Data Request message format is as follows:

Market Data Request

Tag	Field Name		Req'd	Comments
	<i>Standard Header</i>		Y	MsgType = V
262	MDReqID		Y	Must be unique, or the ID of previous Market Data Request to disable if SubscriptionRequestType = Disable previous Snapshot + Updates Request (2).
263	SubscriptionRequestType		Y	SubscriptionRequestType indicates to the other party what type of response is expected. A snapshot request only asks for current information. A subscribe request asks for updates as the status changes. Unsubscribe will cancel any future update messages from the counter party.
264	MarketDepth		Y	
265	MDUpdateType		N	Required if SubscriptionRequestType = Snapshot + Updates (1).
266	AggregatedBook		N	
267	NoMDEntryTypes		Y	Number of MDEntryType fields requested.
→	269	<i>MDEntryType</i>	Y	Must be the first field in this repeating group. This is a list of all the types of Market Data Entries that the firm requesting the Market Data is interested in receiving.
146	NoRelatedSym		Y	Number of symbols requested.
→	55	<i>Symbol</i>	Y	Must be the first field in the repeating group.
→	65	<i>SymbolSfx</i>	N	Can be repeated multiple times if message is related to multiple symbols.
→	48	<i>SecurityID</i>	N	Can be repeated multiple times if message is related to multiple symbols.
→	22	<i>IDSource</i>	N	Can be repeated multiple times if message is related to multiple symbols.
→	167	<i>SecurityType</i>	N	Must be specified if a Future or Option. If a Future:Symbol, SecurityType, and MaturityMonthYear are required. If an Option:Symbol, SecurityType, MaturityMonthYear, PutOrCall, and StrikePrice are required.
→	200	<i>MaturityMonthYear</i>	N	Specifies the month and year of maturity. Required if MaturityDay is specified.

→	205	<i>MaturityDay</i>	N	Can be used in conjunction with MaturityMonthYear to specify a particular maturity date.
→	201	<i>PutOrCall</i>	N	For Options.
→	202	<i>StrikePrice</i>	N	For Options.
→	206	<i>OptAttribute</i>	N	For Options.
→	231	<i>ContractMultiplier</i>	N	For Fixed Income, Convertible Bonds, Derivatives, etc. Note: If used, quantities should be expressed in the "nominal" (e.g. contracts vs. shares) amount.
→	223	<i>CouponRate</i>	N	For Fixed Income.
→	207	<i>SecurityExchange</i>	N	Can be used to identify the security.
→	106	<i>Issuer</i>	N	Can be repeated multiple times if message is related to multiple symbols.
→	348	<i>EncodedIssuerLen</i>	N	Must be set if EncodedIssuer field is specified and must immediately precede it.
→	349	<i>EncodedIssuer</i>	N	Encoded (non-ASCII characters) representation of the Issuer field in the encoded format specified via the MessageEncoding field.
→	107	<i>SecurityDesc</i>	N	Can be repeated multiple times if message is related to multiple symbols.
→	350	<i>EncodedSecurityDescLen</i>	N	Must be set if EncodedSecurityDesc field is specified and must immediately precede it.
→	351	<i>EncodedSecurityDesc</i>	N	Encoded (non-ASCII characters) representation of the SecurityDesc field in the encoded format specified via the MessageEncoding field.
→	336	<i>TradingSessionID</i>	N	
	<i>Standard Trailer</i>		Y	

Market Data – Snapshot / Full Refresh -

The Market Data messages are used as the response to a Market Data Request message. In all cases, one Market Data message refers only to one Market Data Request. It can be used to transmit a 2-sided book of orders or list of quotes, a list of trades, index values, opening, closing, settlement, high, low, or VWAP prices, or any combination of these.

Market Data messages sent as the result of a Market Data Request message are tagged with the appropriate MDReqID. Unsolicited Market Data messages can be sent; in such cases, MDReqID will not be present.

If the message is used for foreign exchange, conventions for identifying the forex transaction are as follows:

- The forex Symbol is defined in "EBS" (Electronic Banking System) format: "CCY1/CCY2".
 - Rates are expressed as "currency1 **in** currency2" (or "currency2 per currency1") and are calculated as CCY2 divided by CCY1 (**NOT** CCY1 divided by CCY2)
 - e.g. "GBP/USD" represents a rate expressed as USD per GBP, "USD/JPY" represents a rate expressed as JPY per USD, etc.).
 - CCY1 and CCY2 are ISO currency codes
- The value of the Currency field represents the denomination of the quantity fields (e.g. JPY represents quantity of JPY).
- [See Appendix O – Foreign Exchange Trading](#)

Market Data messages include many fields, and not all are required to be used. A firm may, at its option, choose to send the minimum fields required, or may choose to send more information, such as tick direction, tagging of best quotes, etc.

Market Data messages can take two forms. The first Market Data message format used for a Snapshot, or a Snapshot + Updates where MDUpdateType = Full Refresh (0) is as follows:

- For Market Data Requests where a Bid or Offer is added, changed, or deleted, every update to a Market Data Entry results in a new Market Data message that contains the entirety of the data requested for that instrument, not just the changed Market Data Entry. In other words, both sides of the market, or just one side in the case of a request of only bids or offers, for the depth requested, must be sent in one FIX Market Data message.
- A Market Data message may contain several trades, an index value, opening, closing, settlement, high, low, and/or VWAP price for one instrument, but only one instrument per message.
- Messages containing bids and/or offers cannot contain trades, index value, opening, closing, settlement, high, low, and/or VWAP prices.

Market Data - Snapshot / Full Refresh

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = W
262	MDReqID	N	Conditionally required if this message is in response to a Market Data Request.

55	Symbol	Y	
65	SymbolSfx	N	
48	SecurityID	N	
22	IDSSource	N	
167	SecurityType	N	Must be specified if a Future or Option. If a Future: Symbol, SecurityType, and MaturityMonthYear are required. If an Option: Symbol, SecurityType, MaturityMonthYear, PutOrCall, and StrikePrice are required.
200	MaturityMonthYear	N	Specifies the month and year of maturity. Required if MaturityDay is specified.
205	MaturityDay	N	Can be used in conjunction with MaturityMonthYear to specify a particular maturity date.
201	PutOrCall	N	For Options.
202	StrikePrice	N	For Options.
206	OptAttribute	N	For Options.
231	ContractMultiplier	N	For Fixed Income, Convertible Bonds, Derivatives, etc. Note: If used, quantities should be expressed in the "nominal" (e.g. contracts vs. shares) amount.
223	CouponRate	N	For Fixed Income.
207	SecurityExchange	N	Can be used to identify the security.
106	Issuer	N	
348	EncodedIssuerLen	N	Must be set if EncodedIssuer field is specified and must immediately precede it.
349	EncodedIssuer	N	Encoded (non-ASCII characters) representation of the Issuer field in the encoded format specified via the MessageEncoding field.
107	SecurityDesc	N	
350	EncodedSecurityDescLen	N	Must be set if EncodedSecurityDesc field is specified and must immediately precede it.
351	EncodedSecurityDesc	N	Encoded (non-ASCII characters) representation of the SecurityDesc field in the encoded format specified via the MessageEncoding field.
291	FinancialStatus	N	
292	CorporateAction	N	
387	TotalVolumeTraded	N	Total volume traded in this trading session for this security.
268	NoMDEntries	Y	Number of entries following.

→	269	<i>MDEntryType</i>	Y	Must be the first field in this repeating group.
→	270	<i>MDEntryPx</i>	Y	
→	15	<i>Currency</i>	N	Can be used to specify the currency of the quoted price.
→	271	<i>MDEntrySize</i>	N	Conditionally required if MDEntryType = Bid(0), Offer(1), or Trade(2)
→	272	<i>MDEntryDate</i>	N	
→	273	<i>MDEntryTime</i>	N	
→	274	<i>TickDirection</i>	N	
→	275	<i>MDMkt</i>	N	Market posting quote / trade. Valid values: See Appendix C
→	336	<i>TradingSessionID</i>	N	
→	276	<i>QuoteCondition</i>	N	Space-delimited list of conditions describing a quote.
→	277	<i>TradeCondition</i>	N	Space-delimited list of conditions describing a trade
→	282	<i>MDEntryOriginator</i>	N	
→	283	<i>LocationID</i>	N	
→	284	<i>DeskID</i>	N	
→	286	<i>OpenCloseSettleFlag</i>	N	Used if MDEntryType = Opening Price(4), Closing Price(5), or Settlement Price(6).
→	59	<i>TimeInForce</i>	N	For optional use when this Bid or Offer represents an order
→	432	<i>ExpireDate</i>	N	For optional use when this Bid or Offer represents an order. ExpireDate and ExpireTime cannot both be specified in one Market Data Entry.
→	126	<i>ExpireTime</i>	N	For optional use when this Bid or Offer represents an order. ExpireDate and ExpireTime cannot both be specified in one Market Data Entry.
→	110	<i>MinQty</i>	N	For optional use when this Bid or Offer represents an order
→	18	<i>ExecInst</i>	N	Can contain multiple instructions, space delimited.
→	287	<i>SellerDays</i>	N	
→	37	<i>OrderID</i>	N	For optional use when this Bid, Offer, or Trade represents an order

→	299	<i>QuoteEntryID</i>	N	For optional use when this Bid, Offer, or Trade represents a quote
→	288	<i>MDEntryBuyer</i>	N	For optional use in reporting Trades
→	289	<i>MDEntrySeller</i>	N	For optional use in reporting Trades
→	346	<i>NumberOfOrders</i>	N	In an Aggregated Book, used to show how many individual orders make up an MDEntry
→	290	<i>MDEntryPositionNo</i>	N	Display position of a bid or offer, numbered from most competitive to least competitive, per market side, beginning with 1
→	58	<i>Text</i>	N	Text to describe the Market Data Entry. Part of repeating group.
→	354	<i>EncodedTextLen</i>	N	Must be set if EncodedText field is specified and must immediately precede it.
→	355	<i>EncodedText</i>	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
	<i>Standard Trailer</i>		Y	

Market Data – Incremental Refresh -

The second Market Data message format is used for incremental updates. Market Data Entries may have an MDEntryID unique among all currently active Market Data Entries so they can be referenced for the purposes of deleting and changing them later. When changing a Market Data Entry, it may keep the same MDEntryID, in which case only MDEntryID would be populated, or the MDEntryID may change, in which case MDEntryID will contain the new ID, and MDEntryRefID will contain the ID of the Market Data Entry being changed. An MDEntryID can be reused within a day only if it has first been deleted.

Alternately, in the case of displaying the best quotes of Market Makers or Exchanges, and not orders in an order book, MDEntryID can be omitted for simplification. In this case, a New Market Data Entry will replace the previous best quote for that side and symbol for the specified Market Maker or Exchange. Deletion of a Market Data Entry would not specify an MDEntryID or MDRefID, and would remove the most recent Market Data Entry for the specified symbol, side, and Market Maker or Exchange. A Change of a Market Data Entry would not specify an MDEntryID or MDRefID, and would replace the most recent Market Data Entry for the specified symbol, side, and Market Maker or Exchange.

The Market Data message for incremental updates may contain any combination of new, changed, or deleted Market Data Entries, for any combination of instruments, with any combination of trades, quotes, index values, open, close, settlement, high, low, and VWAP prices, so long as the maximum FIX message size is not exceeded. All of these types of Market Data Entries can be changed and deleted.

Adding, Changing, or Deleting Market Data Entries requires special consideration of the MDEntryPositionNo field, if the sender wishes to specify it and the receiver wishes to process it. For example, assume ten bids for a security. Adding a bid with MDEntryPositionNo = 4 requires the receiver to shift down other Market Data Entries, i.e. the Market Data Entry in the 4th display position will shift to the 5th, the 5th shifts to the 6th, etc. until the 10th shifts to the 11th. The sender must **NOT** send a modification of all MDEntries in the 4th through 10th positions just to update the MDEntryPositionNo field; the recipient must infer the change. Similarly, deleting a Market Data Entry in the 7th position causes the 8th Market Data Entry to move into the 7th position, the 9th to shift into the 8th position, etc. A Change of the MDEntryPositionNo field of a Market Data Entry causes the Market Data Entries lying between the old and new positions to shift. For instance, a Market Data Entry that occupied the 5th position is changed to the 8th position. This means that the Market Data Entry in the 6th position shifts up to the 5th position, the 7th position shifts to the 6th, and what was in the 8th position shifts into the 7th to make room for the changed Market Data Entry that is being moved into the 8th position.

Several techniques are employed to conserve bandwidth:

- An instrument only needs to be identified when a Market Data Entry is first created.
- In cases where the identification of an instrument is long, the sender has the option of referring to a previous active Market Data Entry of the same instrument instead of duplicating the information.
- A new Market Data Entry will default to the same instrument of the previous Market Data Entry in the same Market Data message if neither Symbol nor MDEntryRefID are specified.
- In the case of a change in a Market Data Entry, only the fields changing need to be sent as part of the change to the Market Data Entry; for example, a change of the MDEntrySize but not the MDEntryPx or other attributes of the Market Data Entry only requires listing the MDEntrySize field, in addition to MDUpdateAction and MDEntryID if used in the original Market Data Entry
- When creating a new Market Data Entry with a future or option instrument similar to the instrument in the previous Market Data Entry in the same FIX message, one may send just symbol

identification fields that have changed, such as MaturityMonthYear, MaturityDay, PutOrCall, StrikePrice, OptAttribute, and SecurityExchange.

- MDEntryID can be reused within the same day after it is deleted. This is helpful for distributing order books because an order that is suspended and then reinstated can have its MDEntryID deleted upon suspension and later reused, with MDUpdateAction = New(0) upon reinstatement, thus avoiding having to re-map the MDEntryID.

Market Data - Incremental Refresh

Tag	Field Name		Req'd	Comments
	<i>Standard Header</i>		Y	MsgType = X
262	MDReqID		N	Conditionally required if this message is in response to a Market Data Request.
268	NoMDEntries		Y	Number of entries following.
→	279	MDUpdateAction	Y	Must be first field in this repeating group.
→	285	DeleteReason	N	If MDUpdateAction = Delete(2), can be used to specify a reason for the deletion.
→	269	MDEntryType	N	Conditionally required if MDUpdateAction = New(0). Cannot be changed.
→	278	MDEntryID	N	If specified, must be unique among currently active entries if MDUpdateAction = New (0), must be the same as a previous MDEntryID if MDUpdateAction = Delete (2), and must be the same as a previous MDEntryID if MDUpdateAction = Change (1) and MDEntryRefID is not specified, or must be unique among currently active entries if MDUpdateAction = Change(1) and MDEntryRefID is specified..
→	280	MDEntryRefID	N	If MDUpdateAction = New(0), for the first Market Data Entry in a message, either this field or a Symbol must be specified. If MDUpdateAction = Change(1), this must refer to a previous MDEntryID.
→	55	Symbol	N	Either Symbol or MDEntryRefID must be specified if MDUpdateAction = New(0) for the first Market Data Entry in a message. For subsequent Market Data Entries where MDUpdateAction = New(0), the default is the instrument used in the previous Market Data Entry if neither Symbol nor MDEntryRefID are specified, or in the case of options and futures, the previous instrument with changes specified in MaturityMonthYear, MaturityDay, PutOrCall, StrikePrice, OptAttribute, and SecurityExchange. May not be changed.
→	65	SymbolSfx	N	May not be changed.
→	48	SecurityID	N	May not be changed.

→	22	<i>IDSource</i>	N	May not be changed.
→	167	<i>SecurityType</i>	N	Must be specified if a Future or Option. If a Future: Symbol, SecurityType, and MaturityMonthYear are required. If an Option: Symbol, SecurityType, MaturityMonthYear, PutOrCall, and StrikePrice are required. May not be changed.
→	200	<i>MaturityMonthYear</i>	N	Specifies the month and year of maturity. Required if MaturityDay is specified. May not be changed.
→	205	<i>MaturityDay</i>	N	Can be used in conjunction with MaturityMonthYear to specify a particular maturity date. May not be changed.
→	201	<i>PutOrCall</i>	N	For Options. May not be changed.
→	202	<i>StrikePrice</i>	N	For Options. May not be changed.
→	206	<i>OptAttribute</i>	N	For Options. May not be changed.
→	231	<i>ContractMultiplier</i>	N	For Fixed Income, Convertible Bonds, Derivatives, etc. Note: If used, quantities should be expressed in the "nominal" (e.g. contracts vs. shares) amount.
→	223	<i>CouponRate</i>	N	For Fixed Income.
→	207	<i>SecurityExchange</i>	N	Can be used to identify the security. May not be changed.
→	106	<i>Issuer</i>	N	May not be changed.
→	348	<i>EncodedIssuerLen</i>	N	Must be set if EncodedIssuer field is specified and must immediately precede it.
→	349	<i>EncodedIssuer</i>	N	Encoded (non-ASCII characters) representation of the Issuer field in the encoded format specified via the MessageEncoding field.
→	107	<i>SecurityDesc</i>	N	May not be changed.
→	350	<i>EncodedSecurityDescLen</i>	N	Must be set if EncodedSecurityDesc field is specified and must immediately precede it.
→	351	<i>EncodedSecurityDesc</i>	N	Encoded (non-ASCII characters) representation of the SecurityDesc field in the encoded format specified via the MessageEncoding field.
→	291	<i>FinancialStatus</i>	N	
→	292	<i>CorporateAction</i>	N	
→	270	<i>MDEntryPx</i>	N	Conditionally required when MDUpdateAction = New(0).
→	15	<i>Currency</i>	N	Can be used to specify the currency of the quoted price.
→	271	<i>MDEntrySize</i>	N	Conditionally required when MDUpdateAction = New(0) and MDEntryType = Bid(0), Offer(1), or Trade(2).

→	272	<i>MDEntryDate</i>	N	
→	273	<i>MDEntryTime</i>	N	
→	274	<i>TickDirection</i>	N	
→	275	<i>MDMkt</i>	N	Market posting quote / trade. Valid values: See Appendix C
→	336	<i>TradingSessionID</i>	N	
→	276	<i>QuoteCondition</i>	N	Space-delimited list of conditions describing a quote.
→	277	<i>TradeCondition</i>	N	Space-delimited list of conditions describing a trade
→	282	<i>MDEntryOriginator</i>	N	
→	283	<i>LocationID</i>	N	
→	284	<i>DeskID</i>	N	
→	286	<i>OpenCloseSettleFlag</i>	N	Used if MDEntryType = Opening Price(4), Closing Price(5), or Settlement Price(6).
→	59	<i>TimeInForce</i>	N	For optional use when this Bid or Offer represents an order
→	432	<i>ExpireDate</i>	N	For optional use when this Bid or Offer represents an order. ExpireDate and ExpireTime cannot both be specified in one Market Data Entry.
→	126	<i>ExpireTime</i>	N	For optional use when this Bid or Offer represents an order. ExpireDate and ExpireTime cannot both be specified in one Market Data Entry.
→	110	<i>MinQty</i>	N	For optional use when this Bid or Offer represents an order
→	18	<i>ExecInst</i>	N	Can contain multiple instructions, space delimited.
→	287	<i>SellerDays</i>	N	
→	37	<i>OrderID</i>	N	For optional use when this Bid, Offer, or Trade represents an order
→	299	<i>QuoteEntryID</i>	N	For optional use when this Bid, Offer, or Trade represents a quote
→	288	<i>MDEntryBuyer</i>	N	For optional use in reporting Trades
→	289	<i>MDEntrySeller</i>	N	For optional use in reporting Trades
→	346	<i>NumberOfOrders</i>	N	In an Aggregated Book, used to show how many individual orders make up an MDEntry
→	290	<i>MDEntryPositionNo</i>	N	Display position of a bid or offer, numbered from most competitive to least competitive, per market side, beginning with 1
→	387	<i>TotalVolumeTraded</i>	N	Total volume traded in this trading session for this security.

→	58	<i>Text</i>	N	Text to describe the Market Data Entry. Part of repeating group.
→	354	<i>EncodedTextLen</i>	N	Must be set if EncodedText field is specified and must immediately precede it.
→	355	<i>EncodedText</i>	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
	<i>Standard Trailer</i>		Y	

Market Data Request Reject -

The Market Data Request Reject is used when the broker cannot honor the Market Data Request, due to business or technical reasons. Brokers may choose to limit various parameters, such as the size of requests, whether just the top of book or the entire book may be displayed, and whether Full or Incremental updates must be used.

The market data request reject message format is as follows:

Market Data Request Reject

<i>Tag</i>	<i>Field Name</i>	<i>Req'd</i>	<i>Comments</i>
	<i>Standard Header</i>	Y	MsgType = Y
262	MDReqID	Y	Must refer to the MDReqID of the request.
281	MDReqRejReason	N	
58	Text	N	
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
	<i>Standard Trailer</i>	Y	

Security Definition Request -

The Security Definition Request message is used for the following:

1. Request a specific Security to be traded with the second party. The request security can be defined as a complex security made up of one or more underlying securities.
2. Request a list of the Security Types that can be traded with the second party.
3. Request a list of Securities that can be traded with the second party. This request can optionally be qualified with Symbol, TradingSessionID, SecurityExchange, and Security Type.

[See Appendix I: Security Definition, Security Status, and Trading Session Message Scenarios](#)

Security Definition Request

Tag	Field Name	Req'd	Comments
	<i>Standard Header</i>	Y	MsgType = c (lowercase)
320	SecurityReqID	Y	
321	SecurityRequestType	Y	
55	Symbol	N	Symbol of the requested Security
65	SymbolSfx	N	Suffix of the Requested Security
48	SecurityID	N	Security ID of the requested Security
22	IDSSource	N	Source of the Security ID
167	SecurityType	N	Must be specified if a Future or Option. If a Future: Symbol, SecurityType, and MaturityMonthYear are required. If an Option: Symbol, SecurityType, MaturityMonthYear, PutOrCall, and StrikePrice are required.
200	MaturityMonthYear	N	Specifies the month and year of maturity. Required if MaturityDay is specified.
205	MaturityDay	N	Can be used in conjunction with MaturityMonthYear to specify a particular maturity date.
201	PutOrCall	N	For Options.
202	StrikePrice	N	For Options.
206	OptAttribute	N	For Options.
231	ContractMultiplier	N	For Fixed Income, Convertible Bonds, Derivatives, etc. Note: If used, quantities should be expressed in the "nominal" (e.g. contracts vs. shares) amount.

223	CouponRate		N	For Fixed Income.
207	SecurityExchange		N	
106	Issuer		N	
348	EncodedIssuerLen		N	Must be set if EncodedIssuer field is specified and must immediately precede it.
349	EncodedIssuer		N	Encoded (non-ASCII characters) representation of the Issuer field in the encoded format specified via the MessageEncoding field.
107	SecurityDesc		N	
350	EncodedSecurityDescLen		N	Must be set if EncodedSecurityDesc field is specified and must immediately precede it.
351	EncodedSecurityDesc		N	Encoded (non-ASCII characters) representation of the SecurityDesc field in the encoded format specified via the MessageEncoding field.
15	Currency		N	
58	Text		N	Comment, instructions, or other identifying information.
354	EncodedTextLen		N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText		N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
336	TradingSessionID		N	Optional Trading Session Identifier to specify a particular trading session for which you want to obtain a list of securities that are tradeable.
146	NoRelatedSym		N	Number of legs that make up the Security
→	311	UnderlyingSymbol	N	The UnderlyingSymbol must be specified as the first field in the repeating group. Required if NoRelatedSym > 0.
→	312	UnderlyingSymbolSfx	N	
→	309	UnderlyingSecurityID	N	
→	305	UnderlyingIDSource	N	
→	310	UnderlyingSecurityType	N	Must be specified if a Future or Option. If a Future: UnderlyingSymbol, UnderlyingSecurityType, and UnderlyingMaturityMonthYear are required. If an Option: UnderlyingSymbol, UnderlyingSecurityType, UnderlyingMaturityMonthYear, UnderlyingPutOrCall, and UnderlyingStrikePrice are required.

→	313	<i>UnderlyingMaturityMonthYear</i>	N	Specifies the month and year of maturity. Required if UnderlyingMaturityDay is specified.
→	314	<i>UnderlyingMaturityDay</i>	N	Can be used in conjunction with UnderlyingMaturityMonthYear to specify a particular maturity date.
→	315	<i>UnderlyingPutOrCall</i>	N	For Options.
→	316	<i>UnderlyingStrikePrice</i>	N	For Options.
→	317	<i>UnderlyingOptAttribute</i>	N	For Options.
→	436	<i>UnderlyingContractMultiplier</i>	N	For Fixed Income, Convertible Bonds, Derivatives, etc.
→	435	<i>UnderlyingCouponRate</i>	N	For Fixed Income.
→	308	<i>UnderlyingSecurityExchange</i>	N	Can be used to identify the security.
→	306	<i>UnderlyingIssuer</i>	N	
→	362	<i>EncodedUnderlyingIssuerLen</i>	N	Must be set if EncodedUnderlyingIssuer field is specified and must immediately precede it.
→	363	<i>EncodedUnderlyingIssuer</i>	N	Encoded (non-ASCII characters) representation of the UnderlyingIssuer field in the encoded format specified via the MessageEncoding field.
→	307	<i>UnderlyingSecurityDesc</i>	N	
→	364	<i>EncodedUnderlyingSecurityDescLen</i>	N	Must be set if EncodedUnderlyingSecurityDesc field is specified and must immediately precede it.
→	365	<i>EncodedUnderlyingSecurityDesc</i>	N	Encoded (non-ASCII characters) representation of the UnderlyingSecurityDesc field in the encoded format specified via the MessageEncoding field.
→	319	<i>RatioQty</i>	N	Quantity of particular leg in the Security
→	54	<i>Side</i>	N	Indicates if this leg of the security is to be Bought or Sold as part of this complex security.
→	318	<i>UnderlyingCurrency</i>	N	
	<i>Standard Trailer</i>		Y	

Security Definition -

The Security Definition message is used for the following:

1. Accept the security defined in a *Security Definition* message.
2. Accept the security defined in a *Security Definition* message with changes to the definition and/or identity of the security.
3. Reject the security requested in a *Security Definition* message
4. Return a list of Security Types
5. Return a list of Securities

Security Definition

Tag	Field Name	Req'd	Comments
	<i>Standard Header</i>	Y	MsgType = d (lowercase)
320	SecurityReqID	Y	
322	SecurityResponseID	Y	Identifier for the Security Definition message
323	SecurityResponseType	N	
393	TotalNumSecurities	Y	
55	Symbol	N	Symbol of the requested Security
65	SymbolSfx	N	Suffix of the Requested Security
48	SecurityID	N	Security ID of the requested Security
22	IDSSource	N	Source of the Security ID
167	SecurityType	N	Must be specified if a Future or Option. If a Future: Symbol, SecurityType, and MaturityMonthYear are required. If an Option: Symbol, SecurityType, MaturityMonthYear, PutOrCall, and StrikePrice are required. Set to "?" if Security Definition Request is looking for the Security Types
200	MaturityMonthYear	N	Specifies the month and year of maturity. Required if MaturityDay is specified.
205	MaturityDay	N	Can be used in conjunction with MaturityMonthYear to specify a particular maturity date.
201	PutOrCall	N	For Options.
202	StrikePrice	N	For Options.
206	OptAttribute	N	For Options.

231	ContractMultiplier		N	For Fixed Income, Convertible Bonds, Derivatives, etc. Note: If used, quantities should be expressed in the "nominal" (e.g. contracts vs. shares) amount.
223	CouponRate		N	For Fixed Income.
207	SecurityExchange		N	
106	Issuer		N	
348	EncodedIssuerLen		N	Must be set if EncodedIssuer field is specified and must immediately precede it.
349	EncodedIssuer		N	Encoded (non-ASCII characters) representation of the Issuer field in the encoded format specified via the MessageEncoding field.
107	SecurityDesc		N	
350	EncodedSecurityDescLen		N	Must be set if EncodedSecurityDesc field is specified and must immediately precede it.
351	EncodedSecurityDesc		N	Encoded (non-ASCII characters) representation of the SecurityDesc field in the encoded format specified via the MessageEncoding field.
15	Currency		N	
336	TradingSessionID		N	
58	Text		N	Comment, instructions, or other identifying information.
354	EncodedTextLen		N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText		N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
146	NoRelatedSym		N	Number of legs that make up the Security
→	311	UnderlyingSymbol	N	Must be specified as the first field in the repeating group. Required if NoRelatedSym > 0.
→	312	UnderlyingSymbolSfx	N	
→	309	UnderlyingSecurityID	N	
→	305	UnderlyingIDSource	N	

→	310	<i>UnderlyingSecurityType</i>	N	Must be specified if a Future or Option. If a Future: UnderlyingSymbol, UnderlyingSecurityType, and UnderlyingMaturityMonthYear are required. If an Option: UnderlyingSymbol, UnderlyingSecurityType, UnderlyingMaturityMonthYear, PutOrCall, and UnderlyingStrikePrice are required.
→	313	<i>UnderlyingMaturityMonthYear</i>	N	Specifies the month and year of maturity. Required if UnderlyingMaturityDay is specified.
→	314	<i>UnderlyingMaturityDay</i>	N	Can be used in conjunction with UnderlyingMaturityMonthYear to specify a particular maturity date.
→	315	<i>UnderlyingPutOrCall</i>	N	For Options.
→	316	<i>UnderlyingStrikePrice</i>	N	For Options.
→	317	<i>UnderlyingOptAttribute</i>	N	For Options.
→	436	<i>UnderlyingContractMultiplier</i>	N	For Fixed Income, Convertible Bonds, Derivatives, etc.
→	435	<i>UnderlyingCouponRate</i>	N	For Fixed Income.
→	308	<i>UnderlyingSecurityExchange</i>	N	Can be used to identify the security.
→	306	<i>UnderlyingIssuer</i>	N	
→	362	<i>EncodedUnderlyingIssuerLen</i>	N	Must be set if EncodedUnderlyingIssuer field is specified and must immediately precede it.
→	363	<i>EncodedUnderlyingIssuer</i>	N	Encoded (non-ASCII characters) representation of the UnderlyingIssuer field in the encoded format specified via the MessageEncoding field.
→	307	<i>UnderlyingSecurityDesc</i>	N	
→	364	<i>EncodedUnderlyingSecurityDescLen</i>	N	Must be set if EncodedUnderlyingSecurityDesc field is specified and must immediately precede it.
→	365	<i>EncodedUnderlyingSecurityDesc</i>	N	Encoded (non-ASCII characters) representation of the UnderlyingSecurityDesc field in the encoded format specified via the MessageEncoding field.
→	319	<i>RatioQty</i>	N	Quantity of particular leg in the Security
→	54	<i>Side</i>	N	Indicates if this leg of the security is to be Bought or Sold as part of this complex security.
→	318	<i>UnderlyingCurrency</i>	N	
	<i>Standard Trailer</i>		Y	

Security Status Request -

The Security Status Request message provides for the ability to request the status of a security. One or more Security Status messages are returned as a result of a Security Status Request message.

The Security Status Request message contains a *SubscriptionRequestType* field. This tells the counter party what type of request is being made:

0 – indicates that the requestor only wants a snapshot or the current status.

1 – indicates that the requestor wants a snapshot (the current status) plus updates as the status changes. This is similar to subscribing for information and can be implemented in applications as a subscription mechanism.

2 – indicates that the requestor wishes to cancel any pending snapshots or updates – in essence making this an unsubscribe operation.

Security Status Request

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = e (lowercase)
324	SecurityStatusReqID	Y	Must be unique, or the ID of previous Security Status Request to disable if SubscriptionRequestType = Disable previous Snapshot + Updates Request (2).
55	Symbol	Y	
65	SymbolSfx	N	
48	SecurityID	N	
22	IDSOURCE	N	
167	SecurityType	N	Must be specified if a Future or Option. If a Future: Symbol, SecurityType, and MaturityMonthYear are required. If an Option: Symbol, SecurityType, MaturityMonthYear, PutOrCall, and StrikePrice are required.
200	MaturityMonthYear	N	Specifies the month and year of maturity. Required if MaturityDay is specified.
205	MaturityDay	N	Can be used in conjunction with MaturityMonthYear to specify a particular maturity date.
201	PutOrCall	N	For Options.
202	StrikePrice	N	For Options.
206	OptAttribute	N	For Options.
231	ContractMultiplier	N	For Fixed Income, Convertible Bonds, Derivatives, etc. Note: If used, quantities should be expressed in the "nominal" (e.g. contracts vs. shares) amount.
223	CouponRate	N	For Fixed Income.
207	SecurityExchange	N	Can be used to identify the security.

106	Issuer	N	
348	EncodedIssuerLen	N	Must be set if EncodedIssuer field is specified and must immediately precede it.
349	EncodedIssuer	N	Encoded (non-ASCII characters) representation of the Issuer field in the encoded format specified via the MessageEncoding field.
107	SecurityDesc	N	
350	EncodedSecurityDescLen	N	Must be set if EncodedSecurityDesc field is specified and must immediately precede it.
351	EncodedSecurityDesc	N	Encoded (non-ASCII characters) representation of the SecurityDesc field in the encoded format specified via the MessageEncoding field.
15	Currency	N	
263	SubscriptionRequestType	Y	SubscriptionRequestType indicates to the other party what type of response is expected. A snapshot request only asks for current information. A subscribe request asks for updates as the status changes. Unsubscribe will cancel any future update messages from the counter party.)
336	TradingSessionID	N	
	<i>Standard Trailer</i>	Y	

Security Status -

The Security Status message provides for the ability to report changes in status to a security. The Security Status message contains fields to indicate trading status, corporate actions, financial status of the company. The Security Status message is used by one trading entity (for instance an exchange) to report changes in the state of a security.

It is expected that the Security Status message that is sent as a response should indicate what type of request is being provided. If the message is being generated as a result of a RequestType =1, then the response should have a RequestType=1 to permit the requestor to determine why the message was sent.

Security Status

Tag	Field Name	Req'd	Comments
	<i>Standard Header</i>	Y	MsgType = f (lowercase)
324	SecurityStatusReqID	N	
55	Symbol	Y	
65	SymbolSfx	N	
48	SecurityID	N	
22	IDSSource	N	
167	SecurityType	N	Must be specified if a Future or Option. If a Future: Symbol, SecurityType, and MaturityMonthYear are required. If an Option: Symbol, SecurityType, MaturityMonthYear, PutOrCall, and StrikePrice are required.
200	MaturityMonthYear	N	Specifies the month and year of maturity. Required if MaturityDay is specified.
205	MaturityDay	N	Can be used in conjunction with MaturityMonthYear to specify a particular maturity date.
201	PutOrCall	N	For Options.
202	StrikePrice	N	For Options.
206	OptAttribute	N	For Options.
231	ContractMultiplier	N	For Fixed Income, Convertible Bonds, Derivatives, etc. Note: If used, quantities should be expressed in the "nominal" (e.g. contracts vs. shares) amount.
223	CouponRate	N	For Fixed Income.
207	SecurityExchange	N	Can be used to identify the security.
106	Issuer	N	
348	EncodedIssuerLen	N	Must be set if EncodedIssuer field is specified and must immediately precede it.

349	EncodedIssuer	N	Encoded (non-ASCII characters) representation of the Issuer field in the encoded format specified via the MessageEncoding field.
107	SecurityDesc	N	
350	EncodedSecurityDescLen	N	Must be set if EncodedSecurityDesc field is specified and must immediately precede it.
351	EncodedSecurityDesc	N	Encoded (non-ASCII characters) representation of the SecurityDesc field in the encoded format specified via the MessageEncoding field.
15	Currency	N	
336	TradingSessionID	N	
325	UnsolicitedIndicator	N	Set to 'Y' if message is sent as a result of a subscription request not a snapshot request
326	SecurityTradingStatus	N	Identifies the trading status applicable to the transaction.
291	FinancialStatus	N	
292	CorporateAction	N	
327	HaltReason	N	Denotes the reason for the Opening Delay or Trading Halt.
328	InViewOfCommon	N	
329	DueToRelated	N	
330	BuyVolume	N	
331	SellVolume	N	
332	HighPx	N	
333	LowPx	N	
31	LastPx	N	Represents the last price for that security either on a Consolidated or an individual participant basis at the time it is disseminated.
60	TransactTime	N	Trade Dissemination Time
334	Adjustment	N	
	<i>Standard Trailer</i>	Y	

Trading Session Status Request -

The Trading Session Status Request is used to request information on the status of a market. With the move to multiple sessions occurring for a given trading party (morning and evening sessions for instance) there is a need to be able to provide information on what product is trading on what market.

The Trading Session Status Request message can be used to inquire the trading status of a trading party. The Trading Session Status message can be used to subscribe to updates to the status of a trading session by setting the RequestType field to 1.

To list the securities available during a particular trading session, [see the SecurityDefinitionRequest message](#).

Trading Session Status Request

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = g (lowercase)
335	TradSesReqID	Y	Must be unique, or the ID of previous Market Data Request to disable if SubscriptionRequestType = Disable previous Snapshot + Updates Request (2).
336	TradingSessionID	N	Trading Session for which status is being requested
338	TradSesMethod	N	Method of trading
339	TradSesMode	N	Trading Session Mode
263	SubscriptionRequestType	Y	
	Standard Trailer	Y	

Trading Session Status -

The Trading Session Status provides information on the status of a market. With the move to multiple sessions occurring for a given trading party (morning and evening sessions for instance) there is a need to be able to provide information on what product is trading on what market.

The Trading Session Status can provide an optional repeating group of securities that are available for trading during that session.

Trading Session Status

Tag	Field Name	Req'd	Comments
	<i>Standard Header</i>	Y	MsgType = h (lowercase)
335	TradSesReqID	N	Provided for a response to a specific Trading Session Status Request message (snapshot).
336	TradingSessionID	Y	Identifier for Trading Session
338	TradSesMethod	N	Method of trading:
339	TradSesMode	N	Trading Session Mode
325	UnsolicitedIndicator	N	'Y' if message is sent unsolicited as a result of a previous subscription request.
340	TradSesStatus	Y	State of the trading session
341	TradSesStartTime	N	Starting time of the trading session
342	TradSesOpenTime	N	Time of the opening of the trading session
343	TradSesPreCloseTime	N	Time of the pre-close of the trading session
344	TradSesCloseTime	N	Closing time of the trading session
345	TradSesEndTime	N	End time of the trading session
387	TotalVolumeTraded	N	
58	Text	N	
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
	<i>Standard Trailer</i>	Y	

New Order - Single -

The new order message type is used by institutions wishing to electronically submit securities and forex orders to a broker for execution.

Orders can be submitted with special handling instructions and execution instructions. Handling instructions refer to how the broker should handle the order on its trading floor (see `HandlInst` field). Execution instructions contain explicit directions as to how the order should be executed (see `ExecInst` field).

New Order messages received with the `PossResend` flag set in the header should be validated by `ClOrdID`. Implementations should also consider checking order parameters (side, symbol, quantity, etc.) to determine if the order had been previously submitted. `PossResends` previously received should be acknowledged back to the client via an Execution - Status message. `PossResends` not previously received should be processed as a new order and acknowledged via an Execution - New message.

The value specified in the `TransactTime` field should allow the receiver of the order to apply business rules to determine if the order is potentially "stale" (e.g. in the event that there have been communication problems). To support forex accommodation trades, two fields, `ForexReq` and `SettlCurrency`, are included in the message. To request a broker to execute a forex trade in conjunction with the securities trade, the institution would set the `ForexReq` = Y and `SettlCurrency` = "intended settlement currency". The broker would then execute a forex trade from the execution currency to the settlement currency and report the results via the execution message in the `SettlCurrAmt` and `SettlCurrency` fields.

The order message can also be used to request a straight forex trade. Conventions for identifying a forex transaction are as follows:

- The forex Symbol is defined in "EBS" (Electronic Banking System) format: "CCY1/CCY2".
 - Rates are expressed as "currency1 in currency2" (or "currency2 per currency1") and are calculated as CCY2 divided by CCY1 (**NOT** CCY1 divided by CCY2)
 - (e.g. "GBP/USD" represents a rate expressed as USD per GBP, "USD/JPY" represents a rate expressed as JPY per USD, etc.).
 - CCY1 and CCY2 are ISO currency codes
- The value of the Currency field represents the denomination of the quantity fields (e.g. JPY represents quantity of JPY).
- In the case of a Forex - Swap (buying (or selling) a currency at one value date and selling (or buying) the same currency at a different value date), Side should represent the side of the `FutSettDate2` transaction.
- `OrdType` = Forex - Market, Forex - Limit, Forex- Swap, or Forex - Previously Quoted
- Netting can be specified via the `ExecInst` field.
- [See Appendix O – Foreign Exchange Trading](#)

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

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

To “take” an IOI (or Quote) from an ECN or exchange and not display the order on the book, the New Order message should contain the TimeInForce field with ImmediateOrCancel and an OrdType field with Previously Indicated (or Previously Quoted).

The presence of DiscretionInst on an order indicates that the trader wishes to display one price but will accept trades at another price. For example a sell order with OrdType = Limit, Price=50.00, DiscretionInst = Related to displayed price and DiscretionOffset = -0.25 means that the order should be displayed as an offer for 50.00, but will match any bid ≥ 49.75 . Discretionary pricing can also be used when pegging an order - for example to indicate that a buy order is to be displayed as pegged to the bid minus 0.25, but can be matched by anything \leq the offer, set OrdType=Pegged, ExecInst = Primary Peg, PegDifference = -0.25, DiscretionInst = Related to market price and DiscretionOffset = 0.

[See Appendix D: Order State Change Matrices](#)

The format for the new order message is as follows:

New Order - Single

Tag	Field Name		Req'd	Comments
	Standard Header		Y	MsgType = D
11	ClOrdID		Y	Unique identifier of the order as assigned by institution.
109	ClientID		N	Used for firm identification in third-party transactions (should not be a substitute for OnBehalfOfCompID/DeliverToCompID).
76	ExecBroker		N	Used for firm identification in third-party transactions (should not be a substitute for OnBehalfOfCompID/DeliverToCompID).
1	Account		N	
78	NoAllocs		N	Number of repeating groups for pre-trade allocation
→	79	AllocAccount	N	Required if NoAllocs > 0. Must be first field in repeating group.
→	80	AllocShares	N	
63	SettlmntTyp		N	Absence of this field is interpreted as Regular.
64	FutSettDate		N	Required when SettlmntTyp = 6 (Future) or SettlmntTyp = 8 (Sellers Option)
21	HandlInst		Y	
18	ExecInst		N	Can contain multiple instructions, space delimited. If OrdType=P, exactly one of the following values (ExecInst = L, R, M, P, O, T, or W) must be specified.
110	MinQty		N	
111	MaxFloor		N	
100	ExDestination		N	

386	NoTradingSessions	N	Specifies the number of repeating TradingSessionIDs
→	336 <i>TradingSessionID</i>	N	Required if NoTradingSessions is > 0.
81	ProcessCode	N	Used to identify soft trades at order entry.
55	Symbol	Y	
65	SymbolSfx	N	
48	SecurityID	N	
22	IDSource	N	
167	SecurityType	N	Must be specified if a Future or Option. If a Future: Symbol, SecurityType, and MaturityMonthYear are required. If an Option: Symbol, SecurityType, MaturityMonthYear, PutOrCall, and StrikePrice are required.
200	MaturityMonthYear	N	Specifies the month and year of maturity. Required if MaturityDay is specified.
205	MaturityDay	N	Can be used in conjunction with MaturityMonthYear to specify a particular maturity date.
201	PutOrCall	N	For Options.
202	StrikePrice	N	For Options.
206	OptAttribute	N	For Options.
231	ContractMultiplier	N	For Fixed Income, Convertible Bonds, Derivatives, etc. Note: If used, quantities should be expressed in the "nominal" (e.g. contracts vs. shares) amount.
223	CouponRate	N	For Fixed Income.
207	SecurityExchange	N	Can be used to identify the security.
106	Issuer	N	
348	EncodedIssuerLen	N	Must be set if EncodedIssuer field is specified and must immediately precede it.
349	EncodedIssuer	N	Encoded (non-ASCII characters) representation of the Issuer field in the encoded format specified via the MessageEncoding field.
107	SecurityDesc	N	
350	EncodedSecurityDescLen	N	Must be set if EncodedSecurityDesc field is specified and must immediately precede it.
351	EncodedSecurityDesc	N	Encoded (non-ASCII characters) representation of the SecurityDesc field in the encoded format specified via the MessageEncoding field.
140	PrevClosePx	N	Useful for verifying security identification
54	Side	Y	
114	LocateReqd	N	Required for short sell orders

60	TransactTime	Y	Time this order request was initiated/released by the trader or trading system.
38	OrderQty	N	Either CashOrderQty or OrderQty is required. Note that either, but not both, CashOrderQty or OrderQty should be specified.
152	CashOrderQty	N	Either CashOrderQty or OrderQty is required. Note that either, but not both, CashOrderQty or OrderQty should be specified. Specifies the approximate “monetary quantity” for the order. Broker is responsible for converting and calculating OrderQty in shares for subsequent messages.
40	OrdType	Y	
44	Price	N	Required for limit OrdTypes. For F/X orders, should be the “all-in” rate (spot rate adjusted for forward points). Can be used to specify a limit price for a pegged order, previously indicated, etc.
99	StopPx	N	Required for OrdType = “Stop” or OrdType = “Stop limit”.
15	Currency	N	
376	ComplianceID	N	
377	SolicitedFlag	N	
23	IOIid	N	Required for Previously Indicated Orders (OrdType=E)
117	QuoteID	N	Required for Previously Quoted Orders (OrdType=D)
59	TimeInForce	N	Absence of this field indicates Day order
168	EffectiveTime	N	Can specify the time at which the order should be considered valid
432	ExpireDate	N	Conditionally required if TimeInForce = GTD and ExpireTime is not specified.
126	ExpireTime	N	Conditionally required if TimeInForce = GTD and ExpireDate is not specified.
427	GTBookingInst	N	States whether executions are booked out or accumulated on a partially filled GT order
12	Commission	N	
13	CommType	N	
47	Rule80A (aka OrderCapacity)	N	
121	ForexReq	N	Indicates that broker is requested to execute a Forex accommodation trade in conjunction with the security trade.
120	SettlCurrency	N	Required if ForexReq = Y.

58	Text	N	
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
193	FutSettDate2	N	Can be used with OrdType = "Forex - Swap" to specify the "value date" for the future portion of a F/X swap.
192	OrderQty2	N	Can be used with OrdType = "Forex - Swap" to specify the order quantity for the future portion of a F/X swap.
77	OpenClose	N	For options
203	CoveredOrUncovered	N	For options
204	CustomerOrFirm	N	For options when delivering the order to execution system/exchange.
210	MaxShow	N	
211	PegDifference	N	Amount (signed) added to the price of the peg
388	DiscretionInst	N	Code to identify the price a DiscretionOffset is related to and should be mathematically added to. Required if DiscretionOffset is specified.
389	DiscretionOffset	N	Amount (signed) added to the "related to" price specified via DiscretionInst.
439	ClearingFirm	N	
440	ClearingAccount	N	
	<i>Standard Trailer</i>	Y	

Execution Reports -

The execution report message is used to:

1. confirm the receipt of an order
2. confirm changes to an existing order (i.e. accept cancel and replace requests)
3. relay order status information
4. relay fill information on working orders
5. reject orders
6. report post-trade fees calculations associated with a trade

NOTE: Execution reports do not replace the end-of-day confirm. Execution reports are to be regarded only as replacements for the existing fill messages currently communicated via telephone.

Each execution report contains three fields which are used to communicate both the current state of the order as understood by the broker and the purpose of the message: OrdStatus, ExecType and ExecTransType.

In an execution report the OrdStatus is used to convey the current state of the order. If an order simultaneously exists in more than one order state, the value with highest precedence is the value that is reported in the OrdStatus field. The order statuses are as follows (in highest to lowest precedence):

Precedence	OrdStatus	Description
12	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.
11	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.
10	Done for Day	Order not, or partially, filled; no further executions forthcoming for the trading day
9	Calculated	Order has been completed for the day (either filled or done for day). Commission or currency settlement details have been calculated and reported in this execution message
8	Filled	Order completely filled, no remaining quantity
7	Stopped	Order has been stopped at the exchange. Used when guaranteeing or protecting a price and quantity
6	Suspended	Order has been placed in suspended state at the request of the client.
5	Canceled	Canceled order with or without executions
5	Expired	Order has been canceled in broker's system due to time in force instructions.
4	Partially Filled	Outstanding order with executions and remaining quantity
3	Replaced	Replaced order with or without executions

2	New	Outstanding order with no executions
2	Rejected	Order has been rejected by broker. 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 brokers system but not yet accepted for execution. An execution message with this status will only be sent in response to a Status Request message.
1	Accepted for bidding	Order has been received and is being evaluated for pricing. It is anticipated that this status will only be used with the "Disclosed" BidType List Order Trading model.

The ExecType is used to identify the purpose of the execution report message. To transmit a change in OrdStatus for an order, the broker(sell side) should send an Execution Report with the new OrdStatus value in both the ExecType AND the OrdStatus fields to signify this message is changing the state of the order. The only exception to this rule is that when rejecting a cancel or cancel/replace request the CancelReject message is used both to reject the request and to communicate the current OrdStatus. An ExecType of Pending Cancel or Pending Replace is used to indicate that a cancel or cancel/replace request is being processed. An ExecType of Canceled or Replace is used to indicate that the cancel or cancel/replace request has been successfully processed.

Execution information (e.g. new partial fill or complete fill) should not be communicated in the same report as one which communicates other state changes (such as pending cancel, pending replace, canceled, replaced, accepted, done for day etc).

Any fills which occur and need to be communicated to the customer while an order is "pending" and waiting to achieve a new state (i.e. via a Order Cancel Replace (aka Order Modification) Request) must contain the "original" (current order prior to state change request) order parameters (i.e. ClOrdID, OrderQty, Price, etc). These fills will cause the CumQty and AvgPx to be updated. An order cannot be considered replaced until it has been explicitly accepted and confirmed to have reached the replaced status via an execution report with ExecType = 'Replace', at which time the effect of the replacement (ClOrdID, new quantity or limit price etc) will be seen. Note that due to the precedence rules above, in reports where ExecType=Replace, OrdStatus may not = Replaced.

Requests to cancel or cancel/replace an order are only acted upon when there is an outstanding order quantity. Requests to replace the OrderQty to a level less than the CumQty will be interpreted by the broker as requests to stop executing the order. Requests to change price on a filled order will be rejected (see Order Cancel Reject message type). The OrderQty, CumQty, LeavesQty, and AvgPx fields should be calculated to reflect the cumulative result of all versions of an order. For example, if partially filled order A were replaced by order B, the OrderQty, CumQty, LeavesQty, and AvgPx on order B's fills should represent the cumulative result of order A plus those on order B.

The general rule is: $\text{OrderQty} = \text{CumQty} + \text{LeavesQty}$.

There can be exceptions to this rule when ExecType and/or OrdStatus are Canceled, DoneForTheDay (e.g. on a day order), Expired, Calculated, or Rejected in which case the order is no longer active and LeavesQty could be 0.

Execution report messages are transmitted with a transaction type (ExecTransType) NEW, CANCEL, CORRECT or STATUS. Transaction types CANCEL and CORRECT modify the state of the message identified in field ExecRefID, and are used to cancel or correct a previously reported execution.

Transaction type STATUS indicates that the execution message contains no new information, only summary information regarding order status.

- The NEW transaction type indicates that this message represents a new order, a change in status of the order, or a new fill against an existing order. The combination of the ExecTransType, ExecType, and OrdStatus fields will indicate how the message is to be applied to an order.
- The CANCEL transaction type applies at the execution level. The Cancel transaction will be used to cancel an execution which has been reported in error. The canceled execution will be identified in the ExecRefID field. Note: ExecTransType of Cancel should not be used to cancel a previous ExecutionRpt with ExecTransType of Cancel (e.g. cannot cancel a cancel).
- The CORRECT transaction type applies at the execution level and is used to modify an incorrectly reported fill. The incorrect execution will be identified in the ExecRefID field. If a single execution is corrected more than once, ExecRefID should refer to the ExecID of the last corrected ExecutionRpt (same convention as ClOrdID and OrigClOrdID). To correct an ExecutionRpt which was previously canceled, an ExecutionRpt with ExecTransType=New should be sent (e.g. cannot send ExecTransType=Correct for an ExecutionRpt with ExecTransType=Cancel). *Note: Data reported in the CumQty, LeavesQty, and AvgPx fields represent the status of the order as of the time of the correction, not as of the time of the originally reported execution.*

[See Appendix D: Order State Change Matrices](#) for examples of key state changes, processing of cancel and cancel/replace requests, and for execution cancel/corrections.

An ExecutionRpt with ExecType = Restated represents an ExecutionRpt sent by the sellside communicating a change in the order or a restatement of the order's parameters without an electronic request from the customer. ExecRestatementReason must be set. This is used for GT orders and corporate actions (see below), changes communicated verbally to the sellside either due to normal business practices or as an emergency measure when electronic systems are not available, repricing of orders by the sellside (such as making Sell Short orders compliant with uptick / downtick rules), or other reasons (Broker option). ExecRestatementReason can also be used to communicate unsolicited cancels.

The field ClOrdID is provided for institutions to affix an identification number to an order to coincide with internal systems. The OrderID field is populated with the broker-generated order number. Unlike ClOrdID/OrigClOrdID which requires a chaining through Cancel/Replaces and Cancels, OrderID and SecondaryOrderID are not required to change through changes to an order.

The underlying business assumption of orders that can trade over multiple days, such as GTC and Good Till Date orders expiring on a future trading date (henceforth referred to as GT orders) is that a GT order that is not fully executed and has not been canceled and has not expired on a given day remains good for the broker to execute the following day. Note that the concept of day is determined by the market convention, which will be security specific. At the end of each trading day, once the order is no longer subject to execution, the broker may optionally send an Execution Report with ExecType=Done for Day(3). When the ExpireDate or ExpireTime of a Good Till Date order is reached, or a GTC order reaches a maximum age, the order is considered expired and the broker may optionally send an Execution Report with ExecType and OrdStatus=Expired(C).

In handling GT orders, the OrderQty, CumQty and AvgPx fields will represent the entirety of the order over all days. The fields DayOrderQty, DayCumQty, and DayAvgPx can be used on days following the day of the first trade on a GT order. Prior to the start of business each day, for all GT orders that have

partial fills on previous days, DayCumQty and DayAvgPx are set to zero, and DayOrderQty becomes the LeavesQty. The following relationship holds: $\text{DayOrderQty} = \text{OrderQty} - (\text{CumQty} - \text{DayCumQty})$. Since $(\text{CumQty} - \text{DayCumQty})$ represents the volume traded on all previous days, $\text{DayOrderQty} = \text{OrderQty} - \text{Volume traded on all previous days}$. Note that when changing the quantity of an order, both OrderQty and DayOrderQty will change. Requests to change or cancel an order will be made in terms of the total quantity for the order, not the quantity open today. For example, on an order where OrderQty=10000 and 2000 shares trade during the previous days, a request to change OrderQty to 15000 will mean that 13000 shares will be open. [See Appendix D – Order State Change Matrices](#) for examples of canceling and changing GT orders partially filled on previous days.

A Cancel on an execution (trade bust) happening the same day of the trade will result in CumQty and DayCumQty each decreasing by the quantity busted, and LeavesQty increasing by the quantity busted. OrderQty and DayOrderQty will remain unchanged. If the business rules allow for a trade bust to be reported on a later date than the trade being busted, the OrderQty and DayCumQty will remain unchanged, the LeavesQty and DayOrderQty will increase by the quantity busted, and the CumQty will decrease by the quantity busted.

If bilaterally agreed between counterparties, a broker may wish to transmit a list of all open GT orders, permitting reconciliation of the open orders. Typically this transmission may occur at the end of the trading day or at the start of the following trading day. There is no expected response to such retransmission; in the event of a reconciliation problem this should be resolved manually or via the DK message. Assuming no corporate actions have occurred, the broker will send an Execution Report with ExecType = Restated (D) and ExecRestatementReason = GT renewal / restatement (no corporate action) (1) for each open GT order. These Execution Reports may have DayCumQty and DayAvgPx restated to zero, and DayOrderQty restated to LeavesQty if the transmission occurs at the start of the following business day. The broker has the option of changing the OrderID and SecondaryOrderID fields, or leaving them unchanged. If they are changed, then the buy-side should use these new ID fields when sending Order Cancel Request, Order Cancel/Replace Request, and Order Status Request messages.

In the case of a corporate action resulting in the adjustment of an open GT order, the broker will send an Execution Report with ExecType = Restated (D) and ExecRestatementReason = GT Corporate action (0) with the order's state after the corporate action adjustment. In the case of stock splits, OrderQty, CumQty, AvgPx, and LeavesQty will be adjusted to reflect the order's state in terms of current shares, not pre-split shares. [See Appendix D – Order State Change Matrices](#) for examples of GT order restatement with and without a corporate action.

The execution report message format is as follows:

Execution Report

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = 8
37	OrderID	Y	OrderID is required to be unique for each chain of orders.

198	SecondaryOrderID		N	Can be used to provide order id used by exchange or executing system.
11	ClOrdID		N	Required for executions against electronically submitted orders which were assigned an ID by the institution. Not required for orders manually entered by the broker.
41	OrigClOrdID		N	Conditionally required for response to an electronic Cancel or Cancel/Replace request (ExecType=PendingCancel, Replaced, or Canceled). ClOrdID of the previous order (NOT the initial order of the day) when canceling or replacing an order.
109	ClientID		N	Used for firm identification in third-party transactions (should not be a substitute for OnBehalfOfCompID/DeliverToCompID).
76	ExecBroker		N	Used for firm identification in third-party transactions (should not be a substitute for OnBehalfOfCompID/DeliverToCompID).
382	NoContraBrokers		N	Number of ContraBrokers repeating group instances.
➔	375	ContraBroker	N	First field in repeating group. Required if NoContraBrokers > 0.
➔	337	ContraTrader	N	
➔	437	ContraTradeQty	N	
➔	438	ContraTradeTime	N	
66	ListID		N	Required for executions against orders which were submitted as part of a list.
17	ExecID		Y	Must be unique for each Execution Report message
20	ExecTransType		Y	
19	ExecRefID		N	Required for Cancel and Correct ExecTransType messages
150	ExecType		Y	Describes the type of execution report. Same possible values as OrdStatus.
39	OrdStatus		Y	Describes the current state of a CHAIN of orders, same scope as OrderQty, CumQty, LeavesQty, and AvgPx
103	OrdRejReason		N	For optional use with ExecType = 8 (Rejected)
378	ExecRestatementReason		N	Required for ExecType = D (Restated).
1	Account		N	Required for executions against electronically submitted orders <i>which were assigned an account by the institution</i>
63	SettlmntTyp		N	Absence of this field is interpreted as Regular.
64	FutSettDate		N	Required when SettlmntTyp = 6 (Future) or SettlmntTyp = 8 (Sellers Option)
55	Symbol		Y	
65	SymbolSfx		N	

48	SecurityID	N	
22	IDSSource	N	
167	SecurityType	N	Must be specified if a Future or Option. If a Future: Symbol, SecurityType, and MaturityMonthYear are required. If an Option: Symbol, SecurityType, MaturityMonthYear, PutOrCall, and StrikePrice are required.
200	MaturityMonthYear	N	Specifies the month and year of maturity. Required if MaturityDay is specified.
205	MaturityDay	N	Can be used in conjunction with MaturityMonthYear to specify a particular maturity date.
201	PutOrCall	N	For Options.
202	StrikePrice	N	For Options.
206	OptAttribute	N	For Options.
231	ContractMultiplier	N	For Fixed Income, Convertible Bonds, Derivatives, etc. Note: If used, quantities should be expressed in the "nominal" (e.g. contracts vs. shares) amount.
223	CouponRate	N	For Fixed Income.
207	SecurityExchange	N	Can be used to identify the security.
106	Issuer	N	
348	EncodedIssuerLen	N	Must be set if EncodedIssuer field is specified and must immediately precede it.
349	EncodedIssuer	N	Encoded (non-ASCII characters) representation of the Issuer field in the encoded format specified via the MessageEncoding field.
107	SecurityDesc	N	
350	EncodedSecurityDescLen	N	Must be set if EncodedSecurityDesc field is specified and must immediately precede it.
351	EncodedSecurityDesc	N	Encoded (non-ASCII characters) representation of the SecurityDesc field in the encoded format specified via the MessageEncoding field.
54	Side	Y	
38	OrderQty	N	Either CashOrderQty or OrderQty is required. Not required for a rejected cash order or an order ack for a cash order.
152	CashOrderQty	N	Either CashOrderQty or OrderQty is required. Specifies the approximate "monetary quantity" conveyed on the order. Broker is responsible for converting and calculating OrderQty in shares for subsequent messages involving fills.
40	OrdType	N	
44	Price	N	Required if specified on the order

99	StopPx	N	Required if specified on the order
211	PegDifference	N	Required if specified on the order
388	DiscretionInst	N	Code to identify the price a DiscretionOffset is related to and should be mathematically added to. Required if DiscretionOffset is specified.
389	DiscretionOffset	N	Amount (signed) added to the “related to” price specified via DiscretionInst.
15	Currency	N	
376	ComplianceID	N	
377	SolicitedFlag	N	
59	TimeInForce	N	Absence of this field indicates Day order
168	EffectiveTime	N	Time specified on the order at which the order should be considered valid
432	ExpireDate	N	Conditionally required if TimeInForce = GTD and ExpireTime is not specified.
126	ExpireTime	N	Conditionally required if TimeInForce = GTD and ExpireDate is not specified.
18	ExecInst	N	Can contain multiple instructions, space delimited.
47	Rule80A (aka OrderCapacity)	N	
32	LastShares	N	Quantity of shares bought/sold on this (last) fill. Not required ExecTransType = 3 (Status). When required, should be "0" for non-fills ("fill" defined as ExecTransType=New and ExecType=Partial Fill or Fill) unless noted below. If ExecType=Stopped, represents the quantity stopped/guaranteed/protected for.
31	LastPx	N	Price of this (last) fill. Not required for ExecTransType = 3 (Status), Should represent the “all-in” (LastSpotRate + LastForwardPoints) rate for F/X orders.). When required, should be "0" for non-fills ("fill" defined as ExecTransType=New and ExecType=Partial Fill or Fill) unless noted below. If ExecType=Stopped, represents the price stopped/guaranteed/protected at.
194	LastSpotRate	N	Applicable for F/X orders
195	LastForwardPoints	N	Applicable for F/X orders
30	LastMkt	N	
336	TradingSessionID	N	
29	LastCapacity	N	

151	LeavesQty	Y	Amount of shares open for further execution. If the OrdStatus is Canceled, DoneForTheDay, Expired, Calculated, or Rejected (in which case the order is no longer active) then LeavesQty could be 0, otherwise LeavesQty = OrderQty - CumQty.
14	CumQty	Y	Currently executed shares for chain of orders.
6	AvgPx	Y	
424	DayOrderQty	N	For GT orders on days following the day of the first trade.
425	DayCumQty	N	For GT orders on days following the day of the first trade.
426	DayAvgPx	N	For GT orders on days following the day of the first trade.
427	GTBookingInst	N	States whether executions are booked out or accumulated on a partially filled GT order
75	TradeDate	N	Used when reporting other than current day trades.
60	TransactTime	N	Time the transaction represented by this ExecutionReport occurred
113	ReportToExch	N	
12	Commission	N	On a fill/partial fill messages, it represents value for that fill/partial fill, on ExecType=Calculated, it represents cumulative value for the order. Monetary commission values are expressed in the currency reflected by the Currency field.
13	CommType	N	
381	GrossTradeAmt	N	
119	SettlCurrAmt	N	Used to report results of forex accommodation trade
120	SettlCurrency	N	Used to report results of forex accommodation trade
<u>155</u>	SettlCurrFxRate	N	Foreign exchange rate used to compute SettlCurrAmt from Currency to SettlCurrency
<u>156</u>	SettlCurrFxRateCalc	N	Specifies whether the SettlCurrFxRate should be multiplied or divided
21	HandlInst	N	
110	MinQty	N	
111	MaxFloor	N	
77	OpenClose	N	For options
210	MaxShow	N	
58	Text	N	
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.

355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
193	FutSettDate2	N	Can be used with OrdType = "Forex - Swap" to specify the "value date" for the future portion of a F/X swap.
192	OrderQty2	N	Can be used with OrdType = "Forex - Swap" to specify the order quantity for the future portion of a F/X swap.
439	ClearingFirm	N	
440	ClearingAccount	N	
442	MultiLegReportingType	N	Default is a single security if not specified.
	<i>Standard Trailer</i>	Y	

Don't Know Trade (DK) -

The Don't Know Trade (DK) message notifies a trading partner that an electronically received execution has been rejected. This message can be thought of as an execution reject message.

This message has special utility when dealing with one-way execution reporting. If the initial Order Acknowledgment message (LastShares=0 and OrdStatus=New) does not match an existing order this message can be used to notify the broker of a potential problem order.

Note that the decision to DK an execution lies with the institution. Some of the mismatches listed in the DKReason field may be acceptable and will not require a DK messages to be generated.

The Don't Know Trade (DK) format is as follows:

Don't Know Trade (DK)

Tag	Field Name	Req'd	Comments
	<i>Standard Header</i>	Y	MsgType = Q
37	OrderID	Y	Broker Order ID as identified on problem execution
17	ExecID	Y	Execution ID of problem execution
127	DKReason	Y	
55	Symbol	Y	
65	SymbolSfx	N	
48	SecurityID	N	
22	IDSSource	N	
167	SecurityType	N	Must be specified if a Future or Option. If a Future: Symbol, SecurityType, and MaturityMonthYear are required. If an Option: Symbol, SecurityType, MaturityMonthYear, PutOrCall, and StrikePrice are required.
200	MaturityMonthYear	N	Specifies the month and year of maturity. Required if MaturityDay is specified.
205	MaturityDay	N	Can be used in conjunction with MaturityMonthYear to specify a particular maturity date.
201	PutOrCall	N	For Options.
202	StrikePrice	N	For Options.
206	OptAttribute	N	For Options.
231	ContractMultiplier	N	For Fixed Income, Convertible Bonds, Derivatives, etc. Note: If used, quantities should be expressed in the "nominal" (e.g. contracts vs. shares) amount.
223	CouponRate	N	For Fixed Income.
207	SecurityExchange	N	Can be used to identify the security.
106	Issuer	N	

348	EncodedIssuerLen	N	Must be set if EncodedIssuer field is specified and must immediately precede it.
349	EncodedIssuer	N	Encoded (non-ASCII characters) representation of the Issuer field in the encoded format specified via the MessageEncoding field.
107	SecurityDesc	N	
350	EncodedSecurityDescLen	N	Must be set if EncodedSecurityDesc field is specified and must immediately precede it.
351	EncodedSecurityDesc	N	Encoded (non-ASCII characters) representation of the SecurityDesc field in the encoded format specified via the MessageEncoding field.
54	Side	Y	
38	OrderQty	N	Either CashOrderQty or OrderQty is required.
152	CashOrderQty	N	Either CashOrderQty or OrderQty is required. Specifies the approximate “monetary quantity” for the order. Broker is responsible for converting and calculating OrderQty in shares for subsequent messages.
32	LastShares	N	Required if specified on the ExecutionRpt
31	LastPx	N	Required if specified on the ExecutionRpt
58	Text	N	
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
	<i>Standard Trailer</i>	Y	

Order Cancel/Replace Request (a.k.a. Order Modification Request) -

The order cancel/replace request is used to change the parameters of an existing order.

Do not use this message to cancel the remaining quantity of an outstanding order, use the Cancel Request message for this purpose.

Cancel/Replace will be used to change any valid attribute of an open order (i.e. reduce/increase quantity, change limit price, change instructions, etc.) It can be used to re-open a filled order by increasing OrderQty.

An immediate response to this message is required. It is recommended that an ExecutionRpt with ExecType=Pending Replace be sent unless the Order Cancel/Replace Request can be immediately accepted (ExecutionRpt with ExecType=Replaced) or rejected (Order Cancel Reject message).

The Cancel/Replace request will only be accepted if the order can successfully be pulled back from the exchange floor without executing. Requests which cannot be processed will be rejected using the Cancel Reject message. The Cancel Reject message should provide the ClOrdID and OrigClOrdID values which were specified on the Cancel/Replace Request message for identification.

Note that while it is necessary for the ClOrdID to change and be unique, the broker's OrderID field does not necessarily have to change as a result of the Cancel/Replace request.

Only a limited number of fields can be changed via the cancel/replace request message. All other fields should be retransmitted as sent in the original order. The fields which can be changed via this message are:

ExecInst	ExpireDate	OrderQty2
OrderQty	ExpireTime	OpenClose
OrdType	MinQty	CoveredOrUncovered
Price	MaxFloor	Side (i.e. sell to sell plus)
HandlInst	StopPx	MaxShow
TimeInForce	PegDifference	LocateReqd
TradingSessionID	DiscretionInst	
EffectiveTime	DiscretionOffset	
	CashOrderQty	

When modifying ExecInst fields in a replacement order, it is necessary to re-declare all ExecInst in the replacement order. ExecInst's will not be carried forward from the original order to the replacement unless re-declared.

The format of the Order Cancel/Replace Request message is:

Order Cancel/Replace Request (a.k.a. Order Modification Request)

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = G
37	OrderID	N	Unique identifier of most recent order as assigned by broker.

109	ClientID	N	Used for firm identification in third-party transactions (should not be a substitute for OnBehalfOfCompID/DeliverToCompID).
76	ExecBroker	N	Used for firm identification in third-party transactions (should not be a substitute for OnBehalfOfCompID/DeliverToCompID).
41	OrigClOrdID	Y	ClOrdID of the previous order (NOT the initial order of the day) when canceling or replacing an order.
11	ClOrdID	Y	Unique identifier of <i>replacement</i> order as assigned by institution. Note that this identifier will be used in ClOrdID field of the Cancel Reject message if the replacement request is rejected.
66	ListID	N	Required for List Orders
1	Account	N	
78	NoAllocs	N	Number of repeating groups for pre-trade allocation
→	79	<i>AllocAccount</i>	Required if NoAllocs > 0. Must be first field in repeating group.
→	80	<i>AllocShares</i>	
63	SettlmntTyp	N	Absence of this field is interpreted as Regular.
64	FutSettDate	N	Required when SettlmntTyp = 6 (Future) or SettlmntTyp = 8 (Sellers Option)
21	HandlInst	Y	
18	ExecInst	N	Can contain multiple instructions, space delimited. Replacement order must be created with new parameters (i.e. original order values will not be brought forward to replacement order unless redefined within this message).
110	MinQty	N	
111	MaxFloor	N	
100	ExDestination	N	
386	NoTradingSessions	N	Specifies the number of repeating TradingSessionIDs
→	336	<i>TradingSessionID</i>	Required if NoTradingSessions is > 0.
55	Symbol	Y	Must match original order
65	SymbolSfx	N	
48	SecurityID	N	Must match original order
22	IDSSource	N	Must match original order

167	SecurityType	N	Must be specified if a Future or Option. If a Future: Symbol, SecurityType, and MaturityMonthYear are required. If an Option: Symbol, SecurityType, MaturityMonthYear, PutOrCall, and StrikePrice are required.
200	MaturityMonthYear	N	Specifies the month and year of maturity. Required if MaturityDay is specified.
205	MaturityDay	N	Can be used in conjunction with MaturityMonthYear to specify a particular maturity date.
201	PutOrCall	N	For Options.
202	StrikePrice	N	For Options.
206	OptAttribute	N	For Options.
231	ContractMultiplier	N	For Fixed Income, Convertible Bonds, Derivatives, etc. Note: If used, quantities should be expressed in the "nominal" (e.g. contracts vs. shares) amount.
223	CouponRate	N	For Fixed Income.
207	SecurityExchange	N	Can be used to identify the security.
106	Issuer	N	
348	EncodedIssuerLen	N	Must be set if EncodedIssuer field is specified and must immediately precede it.
349	EncodedIssuer	N	Encoded (non-ASCII characters) representation of the Issuer field in the encoded format specified via the MessageEncoding field.
107	SecurityDesc	N	
350	EncodedSecurityDescLen	N	Must be set if EncodedSecurityDesc field is specified and must immediately precede it.
351	EncodedSecurityDesc	N	Encoded (non-ASCII characters) representation of the SecurityDesc field in the encoded format specified via the MessageEncoding field.
54	Side	Y	Must match original side, however, Buy and Buy Minus can be interchanged as well as Sell and Sell Plus
60	TransactTime	Y	Time this order request was initiated/released by the trader or trading system.
38	OrderQty	N	Either CashOrderQty or OrderQty is required. Note that either, but not both, CashOrderQty or OrderQty should be specified. Should be the "Total Intended Order Quantity" (including the amount already executed for this chain of orders)

152	CashOrderQty	N	Either CashOrderQty or OrderQty is required. Note that either, but not both, CashOrderQty or OrderQty should be specified. Specifies the approximate “monetary quantity” for the order. Broker is responsible for converting and calculating OrderQty in shares for subsequent messages.
40	OrdType	Y	
44	Price	N	Required for limit OrdTypes. For F/X orders, should be the “all-in” rate (spot rate adjusted for forward points). Can be used to specify a limit price for a pegged order, previously indicated, etc.
99	StopPx	N	Required for OrdType = “Stop” or OrdType = “Stop limit”.
211	PegDifference	N	Amount (signed) added to the price of the peg
388	DiscretionInst	N	Code to identify the price a DiscretionOffset is related to and should be mathematically added to. Required if DiscretionOffset is specified.
389	DiscretionOffset	N	Amount (signed) added to the “related to” price specified via DiscretionInst.
376	ComplianceID	N	
377	SolicitedFlag	N	
15	Currency	N	Must match original order.
59	TimeInForce	N	Absence of this field indicates Day order
168	EffectiveTime	N	Can specify the time at which the order should be considered valid
432	ExpireDate	N	Conditionally required if TimeInForce = GTD and ExpireTime is not specified.
126	ExpireTime	N	Conditionally required if TimeInForce = GTD and ExpireDate is not specified.
427	GTBookingInst	N	States whether executions are booked out or accumulated on a partially filled GT order
12	Commission	N	
13	CommType	N	
47	Rule80A (aka OrderCapacity)	N	Must match original order
121	ForexReq	N	Indicates that broker is requested to execute a Forex accommodation trade in conjunction with the security trade.
120	SettlCurrency	N	Required if ForexReq = Y.
58	Text	N	

354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
193	FutSettDate2	N	Can be used with OrdType = "Forex - Swap" to specify the "value date" for the future portion of a F/X swap.
192	OrderQty2	N	Can be used with OrdType = "Forex - Swap" to specify the order quantity for the future portion of a F/X swap.
77	OpenClose	N	For options
203	CoveredOrUncovered	N	For options
204	CustomerOrFirm	N	For options when delivering the order to execution system/exchange.
210	MaxShow	N	
114	LocateReqd	N	
439	ClearingFirm	N	
440	ClearingAccount	N	
	<i>Standard Trailer</i>	Y	

Order Cancel Request -

The order cancel request message requests the cancellation of **all** of the remaining quantity of an existing order. Note that the Order Cancel/Replace Request should be used to partially cancel (reduce) an order).

The request will only be accepted if the order can successfully be pulled back from the exchange floor without executing.

A cancel request is assigned a ClOrdID and is treated as a separate entity. If rejected, the ClOrdID of the cancel request will be sent in the Cancel Reject message, as well as the ClOrdID of the actual order in the OrigClOrdID field. The ClOrdID assigned to the cancel request must be unique amongst the ClOrdID assigned to regular orders and replacement orders.

An immediate response to this message is required. It is recommended that an ExecutionRpt with ExecType=Pending Cancel be sent unless the Order Cancel Request can be immediately accepted (ExecutionRpt with ExecType=Canceled) or rejected (Order Cancel Reject message).

The format of the cancel request message is:

Order Cancel Request

<i>Tag</i>	<i>Field Name</i>	<i>Req'd</i>	<i>Comments</i>
	<i>Standard Header</i>	Y	MsgType = F
41	OrigClOrdID	Y	ClOrdID of the previous order (NOT the initial order of the day) when canceling or replacing an order.

37	OrderID	N	Unique identifier of most recent order as assigned by broker.
11	ClOrdID	Y	Unique ID of cancel request as assigned by the institution.
66	ListID	N	Required for List Orders
1	Account	N	
109	ClientID	N	Used for firm identification in third-party transactions (should not be a substitute for OnBehalfOfCompID/DeliverToCompID).
76	ExecBroker	N	Used for firm identification in third-party transactions (should not be a substitute for OnBehalfOfCompID/DeliverToCompID).
55	Symbol	Y	
65	SymbolSfx	N	
48	SecurityID	N	
22	IDSSource	N	
167	SecurityType	N	Must be specified if a Future or Option. If a Future: Symbol, SecurityType, and MaturityMonthYear are required. If an Option: Symbol, SecurityType, MaturityMonthYear, PutOrCall, and StrikePrice are required.
200	MaturityMonthYear	N	Specifies the month and year of maturity. Required if MaturityDay is specified.
205	MaturityDay	N	Can be used in conjunction with MaturityMonthYear to specify a particular maturity date.
201	PutOrCall	N	For Options.
202	StrikePrice	N	For Options.
206	OptAttribute	N	For Options.
231	ContractMultiplier	N	For Fixed Income, Convertible Bonds, Derivatives, etc. Note: If used, quantities should be expressed in the "nominal" (e.g. contracts vs. shares) amount.
223	CouponRate	N	For Fixed Income.
207	SecurityExchange	N	Can be used to identify the security.
106	Issuer	N	
348	EncodedIssuerLen	N	Must be set if EncodedIssuer field is specified and must immediately precede it.
349	EncodedIssuer	N	Encoded (non-ASCII characters) representation of the Issuer field in the encoded format specified via the MessageEncoding field.
107	SecurityDesc	N	
350	EncodedSecurityDescLen	N	Must be set if EncodedSecurityDesc field is specified and must immediately precede it.
351	EncodedSecurityDesc	N	Encoded (non-ASCII characters) representation of the SecurityDesc field in the encoded format specified via the MessageEncoding field.

54	Side	Y	
60	TransactTime	Y	Time this order request was initiated/released by the trader or trading system.
38	OrderQty	N	Either CashOrderQty or OrderQty is required. OrderQty = CumQty + LeavesQty (see exceptions above)
152	CashOrderQty	N	Either CashOrderQty or OrderQty is required. Specifies the approximate “monetary quantity” for the order. Broker is responsible for converting and calculating OrderQty in shares for subsequent messages.
376	ComplianceID	N	
377	SolicitedFlag	N	
58	Text	N	
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
	<i>Standard Trailer</i>	Y	

Order Cancel Reject -

The order cancel reject message is issued by the broker upon receipt of a cancel request or cancel/replace request message which cannot be honored. Requests to change price or decrease quantity are executed only when an outstanding quantity exists. Filled orders cannot be changed (i.e. quantity reduced or price change). However, the broker/sellside may support increasing the order quantity on a currently filled order).

When rejecting a Cancel/Replace Request, the Cancel Reject message should provide the ClOrdID and OrigClOrdID values which were specified on the Cancel/Replace Request message for identification

The execution message responds to accepted cancel request and cancel/replace request messages.

The order cancel reject message format is as follows:

Order Cancel Reject

Tag	Field Name	Req'd	Comments
	<i>Standard Header</i>	Y	MsgType = 9
37	OrderID	Y	If CxlRejReason="Unknown order", specify "NONE".
198	SecondaryOrderID	N	Can be used to provide order id used by exchange or executing system.
11	ClOrdID	Y	Unique order id assigned by institution to the cancel request or to the <i>replacement</i> order.
41	OrigClOrdID	Y	ClOrdID which could not be canceled/replaced. ClOrdID of the previous order (NOT the initial order of the day) when canceling or replacing an order.
39	OrdStatus	Y	OrdStatus value after this cancel reject is applied.
109	ClientID	N	Used for firm identification in third-party transactions (should not be a substitute for OnBehalfOfCompID/DeliverToCompID).
76	ExecBroker	N	Used for firm identification in third-party transactions (should not be a substitute for OnBehalfOfCompID/DeliverToCompID).
66	ListID	N	Required for rejects against orders which were submitted as part of a list.
1	Account	N	
60	TransactTime	N	
434	CxlRejResponseTo	Y	
102	CxlRejReason	N	
58	Text	N	
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
	<i>Standard Trailer</i>	Y	

Order Status Request -

The order status request message is used by the institution to generate an order status message back from the broker.

The format of the order status request message is:

Order Status Request

Tag	Field Name	Req'd	Comments
	<i>Standard Header</i>	Y	MsgType = H
37	OrderID	N	
11	ClOrdID	Y	
109	ClientID	N	Used for firm identification in third-party transactions (should not be a substitute for OnBehalfOfCompID/DeliverToCompID).
1	Account	N	
76	ExecBroker	N	Used for firm identification in third-party transactions (should not be a substitute for OnBehalfOfCompID/DeliverToCompID).
55	Symbol	Y	
65	SymbolSfx	N	
48	SecurityID	N	
22	IDSSource	N	
167	SecurityType	N	Must be specified if a Future or Option. If a Future: Symbol, SecurityType, and MaturityMonthYear are required. If an Option: Symbol, SecurityType, MaturityMonthYear, PutOrCall, and StrikePrice are required.
200	MaturityMonthYear	N	Specifies the month and year of maturity. Required if MaturityDay is specified.
205	MaturityDay	N	Can be used in conjunction with MaturityMonthYear to specify a particular maturity date.
201	PutOrCall	N	For Options.
202	StrikePrice	N	For Options.
206	OptAttribute	N	For Options.
231	ContractMultiplier	N	For Fixed Income, Convertible Bonds, Derivatives, etc. Note: If used, quantities should be expressed in the "nominal" (e.g. contracts vs. shares) amount.
223	CouponRate	N	For Fixed Income.
207	SecurityExchange	N	Can be used to identify the security.
106	Issuer	N	
348	EncodedIssuerLen	N	Must be set if EncodedIssuer field is specified and must immediately precede it.

349	EncodedIssuer	N	Encoded (non-ASCII characters) representation of the Issuer field in the encoded format specified via the MessageEncoding field.
107	SecurityDesc	N	
350	EncodedSecurityDescLen	N	Must be set if EncodedSecurityDesc field is specified and must immediately precede it.
351	EncodedSecurityDesc	N	Encoded (non-ASCII characters) representation of the SecurityDesc field in the encoded format specified via the MessageEncoding field.
54	Side	Y	
	<i>Standard Trailer</i>	Y	

Allocation -

The Allocation message provides the ability to specify how an order or set of orders should be subdivided amongst **one or more** accounts. It can also be used as a confirmation message through which third parties can communicate execution and settlement details between trading partners. In addition, the allocation message can be sent by the broker to communicate fees and other details that can only be computed once the sub-account breakdowns are known.

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

An allocation message can be submitted as preliminary, calculated, calculated without preliminary, new, cancel or replace. The AllocTransType field indicates the purpose of the message. When submitting calculated, replace, or cancel AllocTransType messages the RefAllocID field is required. Note that AllocTransType of Cancel, Reject, or Replace affects the entire Allocation message thus each AllocAccount (cannot cancel, reject, or replace a single AllocAccount instance without affecting the entire Allocation message). Replacement allocation messages must contain all data for the replacement allocation. Calculated allocations should have a unique AllocID and use RefAllocID to specify the AllocID from the preliminary. Calculated without preliminary are sent unsolicited from the sellside and do not require RefAllocID.

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

- The total shares allocated must equal the Shares value which must equal the total executed quantity of the original order. If present, the total shares in the execution section must also be equal to this value.
- The number of sub-account instances is indicated in NoAllocs.
- Multiple orders can be combined for allocation by identifying the number of orders in the NoOrders field and each individual order in the OrderID fields. Combined orders must have the same ticker, trade date, settlement date and side.

Pre-Trade Allocation consists of the following steps:

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

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

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

Post-Trade Allocation supports three different message flows:

1. Buyside initiated without Misc Fees
2. Buyside-initiated with Misc Fee computation
3. Sellside-initiated

See [Appendix K: Example Usage of Allocations](#) for more examples and details.

Allocation

Tag	Field Name		Req'd	Comments
	<i>Standard Header</i>		Y	MsgType = J
70	AllocID		Y	
71	AllocTransType		Y	
72	RefAllocID		N	Required for AllocTransType = Calculated, Replace, or Cancel
196	AllocLinkID		N	Can be used to link two different Allocation messages (each with unique AllocID) together, i.e. for F/X “Netting” or “Swaps”
197	AllocLinkType		N	Can be used to link two different Allocation messages and identifies the type of link. Required if AllocLinkID is specified.
73	NoOrders		Y*	Indicates number of orders to be combined for allocation. If order(s) were manually delivered set to 1 (one).
→	11	<i>ClOrdID</i>	Y*	Order ID assigned by client if order(s) were electronically delivered and executed. If order(s) were manually delivered this field should contain string “MANUAL”.
→	37	<i>OrderID</i>	N	
→	198	<i>SecondaryOrderID</i>	N	Can be used to provide order id used by exchange or executing system.
→	66	<i>ListID</i>	N	Required for List Orders.
→	105	<i>WaveNo</i>	N	
124	NoExecs		N	Indicates number of individual execution repeating group entries to follow. Absence of this field indicates that no individual execution entries are included. Primarily used to support step-outs.
→	32	<i>LastShares</i>	N	Number of shares in individual execution. Required if NoExecs > 0
→	17	<i>ExecID</i>	N	
→	31	<i>LastPx</i>	N	Price of individual execution. Required if NoExecs > 0
→	29	<i>LastCapacity</i>	N	Can be specified by broker for AllocTransTyp=Calculated
54	Side		Y	
55	Symbol		Y	
65	SymbolSfx		N	
48	SecurityID		N	

22	IDSource	N	
167	SecurityType	N	Must be specified if a Future or Option. If a Future: Symbol, SecurityType, and MaturityMonthYear are required. If an Option: Symbol, SecurityType, MaturityMonthYear, PutOrCall, and StrikePrice are required.
200	MaturityMonthYear	N	Specifies the month and year of maturity. Required if MaturityDay is specified.
205	MaturityDay	N	Can be used in conjunction with MaturityMonthYear to specify a particular maturity date.
201	PutOrCall	N	For Options.
202	StrikePrice	N	For Options.
206	OptAttribute	N	For Options.
231	ContractMultiplier	N	For Fixed Income, Convertible Bonds, Derivatives, etc. Note: If used, quantities should be expressed in the "nominal" (e.g. contracts vs. shares) amount.
223	CouponRate	N	For Fixed Income.
207	SecurityExchange	N	Can be used to identify the security.
106	Issuer	N	
348	EncodedIssuerLen	N	Must be set if EncodedIssuer field is specified and must immediately precede it.
349	EncodedIssuer	N	Encoded (non-ASCII characters) representation of the Issuer field in the encoded format specified via the MessageEncoding field.
107	SecurityDesc	N	
350	EncodedSecurityDescLen	N	Must be set if EncodedSecurityDesc field is specified and must immediately precede it.
351	EncodedSecurityDesc	N	Encoded (non-ASCII characters) representation of the SecurityDesc field in the encoded format specified via the MessageEncoding field.
53	Shares	Y	Total number of shares allocated to all accounts
30	LastMkt	N	Market of the executions.
336	TradingSessionID	N	
6	AvgPx	Y	For F/X orders, should be the "all-in" rate (spot rate adjusted for forward points).
15	Currency	N	Currency of AvgPx. Should be the currency of the local market or exchange where the trade was conducted.
74	AvgPrxPrecision	N	Absence of this field indicates that default precision arranged by the broker/institution is to be used
75	TradeDate	Y	
60	TransactTime	N	Date/time when allocation is generated
63	SettlmntTyp	N	Absence of this field is interpreted as Regular

64	FutSettDate	N	“Settlement Date”. Required with SettlmntTyp other than regular
381	GrossTradeAmt	N	Expressed in same currency as AvgPx. Sum of (AllocShares * AllocAvgPx or AllocPrice).
118	NetMoney	N	Expressed in same currency as AvgPx. Sum of AllocNetMoney.
77	OpenClose	N	
58	Text	N	
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
157	NumDaysInterest	N	Applicable for Convertible Bonds and fixed income
158	AccruedInterestRate	N	Applicable for Convertible Bonds and fixed income
78	NoAllocs	Y*	Indicates number of allocation groups to follow.
→	79	<i>AllocAccount</i>	Y* May be the same value as BrokerOfCredit if ProcessCode is step-out or soft-dollar step-out and Institution does not wish to disclose individual account breakdowns to the ExecBroker. Required if NoAllocs > 0. Must be first field in repeating group.
→	366	<i>AllocPrice</i>	N Used when performing “executed price” vs. “average price” allocations (e.g. Japan). AllocAccount plus AllocPrice form a unique Allocs entry. Used in lieu of AllocAvgPx.
→	80	<i>AllocShares</i>	Y
→	81	<i>ProcessCode</i>	N
→	92	<i>BrokerOfCredit</i>	N Required if ProcessCode is step-out or soft-dollar step-out
→	208	<i>NotifyBrokerOfCredit</i>	N
→	209	<i>AllocHandlInst</i>	N
→	161	<i>AllocText</i>	N Free format text field related to this AllocAccount
→	360	<i>EncodedAllocTextLen</i>	N Must be set if EncodedAllocText field is specified and must immediately precede it.
→	361	<i>EncodedAllocText</i>	N Encoded (non-ASCII characters) representation of the AllocText field in the encoded format specified via the MessageEncoding field.
→	76	<i>ExecBroker</i>	N Required for step-in and step-out trades
→	109	<i>ClientID</i>	N Used for firm identification in third-party transactions (should not be a substitute for OnBehalfOfCompID/DeliverToCompID).
→	12	<i>Commission</i>	N
→	13	<i>CommType</i>	N
→	153	<i>AllocAvgPx</i>	N AvgPx for this AllocAccount. For F/X orders, should be the “all-in” rate (spot rate adjusted for forward points) for this allocation.

→	154	<i>AllocNetMoney</i>		N	NetMoney for this AllocAccount ((AllocShares * AllocAvgPx) - Commission - sum of MiscFeeAmt + AccruedInterestAmt) if a Sell ((AllocShares * AllocAvgPx) + Commission + sum of MiscFeeAmt + AccruedInterestAmt) if a Buy
→	119	<i>SettlCurrAmt</i>		N	AllocNetMoney in SettlCurrency for this AllocAccount if SettlCurrency is different from “overall” Currency
→	120	<i>SettlCurrency</i>		N	SettlCurrency for this AllocAccount if different from “overall” Currency. Required if SettlCurrAmt is specified.
→	155	<i>SettlCurrFxRate</i>		N	Foreign exchange rate used to compute SettlCurrAmt from Currency to SettlCurrency
→	156	<i>SettlCurrFxRateCalc</i>		N	Specifies whether the SettlCurrFxRate should be multiplied or divided
→	159	<i>AccruedInterestAmt</i>		N	Applicable for Convertible Bonds and fixed income
→	160	<i>SettlInstMode</i>		N	Type of Settlement Instructions which will be provided via Settlement Instructions message (0=Default, 1=Standing Instructions, 2=Specific Allocation Account Overriding, 3=Specific Allocation Account Standing)
→	136	<i>NoMiscFees</i>		N	Required if any miscellaneous fees are reported. Indicates number of repeating entries. Repeating group within Alloc repeating group. ** Nested Repeating Group follows **
→	→	137	<i>MiscFeeAmt</i>	N	Required if NoMiscFees > 0
→	→	138	<i>MiscFeeCurr</i>	N	Required if NoMiscFees > 0
→	→	139	<i>MiscFeeType</i>	N	Required if NoMiscFees > 0
	<i>Standard Trailer</i>			Y	

Note: Req'd = “Y*” indicates that the field is not required for AllocTransTyp=Cancel

Allocation ACK -

The allocation ACK message is used to acknowledge the receipt and status of an allocation message received from the institution.

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

Allocation ACK

Tag	Field Name	Req'd	Comments
	<i>Standard Header</i>	Y	MsgType = P
109	ClientID	N	Used for firm identification in third-party transactions (should not be a substitute for OnBehalfOfCompID/DeliverToCompID).
76	ExecBroker	N	Used for firm identification in third-party transactions (should not be a substitute for OnBehalfOfCompID/DeliverToCompID).
70	AllocID	Y	
75	TradeDate	Y	
60	TransactTime	N	Date/Time AllocationACK generated
87	AllocStatus	Y	
88	AllocRejCode	N	Required for AllocStatus = 1 (rejected)
58	Text	N	Can include explanation for AllocRejCode = 7 (other)
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
	<i>Standard Trailer</i>	Y	

Settlement Instructions -

The Settlement Instructions message provides either the broker's or the institution's instructions for trade settlement. The SettlInstSource field indicates if the settlement instructions are the broker's or the institution's. This message has been designed so that it can be sent from the broker to the institution, from the institution to the broker, or from either to an independent "standing instructions" database or matching system.

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

- 1) To provide "standing instructions" for the settlement of trades occurring in the future, messages should include some combination of.
 - AllocAccount
 - LastMkt
 - Side
 - SecurityType
 - SettlLocation
 - SettlDeliveryType
 - EffectiveTime
- 2) To provide settlement instructions for a specific Allocation Account either as overriding or standing instructions to support matching. The following key should be used to tie the settlement instructions to the corresponding Allocation message.

(TradeDate + AllocID + AllocAccount)

The Settlement Instruction detail can be either explicitly specified (via SecuritySettl* and CashSettl* fields) or can exist on within an independent standing instructions database and can be referenced via the StandInstDbType, StandInstDbName, and StandInstDbID fields.

[See Appendix F – Settlement Instructions Field Usage Matrix](#)

Settlement Instructions

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = T
162	SettlInstID	Y	Unique message ID regardless of SettlInstMode
163	SettlInstTransType	Y	New, Replace, or Cancel
214	SettlInstRefID	Y	Required for Cancel and Replace SettlInstTransType messages
160	SettlInstMode	Y	1=Standing Instructions, 2=Specific Allocation Account Overriding, 3=Specific Allocation Account Standing
165	SettlInstSource	Y	1=Broker's Settlement Instructions, 2=Institution's Settlement Instructions
79	AllocAccount	Y	Required for SettlInstMode=1, 2, or 3

166	SettlLocation	N	Required for SettlInstMode=2 or 3, may be required for SettlInstMode=1 (i.e. may not be required if StandInstDbType and StandInstDbID are used)
75	TradeDate	N	Required for SettlInstMode=2 or 3
70	AllocID	N	Required for SettlInstMode=2 or 3
30	LastMkt	N	Required for SettlInstMode=2 or 3, May be required for SettlInstMode=1
336	TradingSessionID	N	
54	Side	N	Required for SettlInstMode=2 or 3, May be required for SettlInstMode=1
167	SecurityType	N	May be required for SettlInstMode=1
168	EffectiveTime	N	May be required for SettlInstMode=1 (timestamp when it goes in to effect)
60	TransactTime	Y	Date/Time Settlement Instructions were generated
109	ClientID	N	Used for firm identification in third-party transactions (should not be a substitute for OnBehalfOfCompID/DeliverToCompID).
76	ExecBroker	N	Used for firm identification in third-party transactions (should not be a substitute for OnBehalfOfCompID/DeliverToCompID).
169	StandInstDbType	N	1=DTC SID, 2=Thomson ALERT, 3=Global Custodian's, etc.
170	StandInstDbName	N	Name of StandInstDbType (i.e. DTC, Global Custodian's name)
171	StandInstDbID	N	Identifier used within the StandInstDbType
172	SettlDeliveryType	N	
173	SettlDepositoryCode	N	Applicable when SettlLocation is a depository
174	SettlBrkrCode	N	
175	SettlInstCode	N	
176	SecuritySettlAgentName	N	Applicable when settlement is being performed at a country vs. a depository
177	SecuritySettlAgentCode	N	Applicable when settlement is being performed at a country vs. a depository
178	SecuritySettlAgentAcctNum	N	Applicable when settlement is being performed at a country vs. a depository
179	SecuritySettlAgentAcctName	N	Applicable when settlement is being performed at a country vs. a depository
180	SecuritySettlAgentContactName	N	Applicable when settlement is being performed at a country vs. a depository
181	SecuritySettlAgentContactPhone	N	Applicable when settlement is being performed at a country vs. a depository
182	CashSettlAgentName	N	Applicable when SettlDeliveryType=Free

183	CashSettlAgentCode	N	Applicable when SettlDeliveryType=Free
184	CashSettlAgentAcctNum	N	Applicable when SettlDeliveryType=Free
185	CashSettlAgentAcctName	N	Applicable when SettlDeliveryType=Free
186	CashSettlAgentContactName	N	Applicable when SettlDeliveryType=Free
187	CashSettlAgentContactPhone	N	Applicable when SettlDeliveryType=Free
	<i>Standard Trailer</i>	Y	

Bid Request -

The BidRequest Message can be used in one of two ways depending on which market conventions are being followed.

In the “Non disclosed” convention (e.g. US/European model) the BidRequest message can be used to request a bid based on the sector, country, index and liquidity information contained within the message itself. In the “Non disclosed” convention the entry repeating group is used to define liquidity of the program. [See Appendix N – Program/Basket/List Trading for an example.](#)

In the “Disclosed” convention (e.g. Japanese model) the BidRequest message can be used to request bids based on the ListOrderDetail messages sent in advance of BidRequest message. In the “Disclosed” convention the list repeating group is used to define which ListOrderDetail messages a bid is being sort for and the directions of the required bids.

The pair of fields SideValue1 and SideValue2 are used to show the monetary total value in either direction (buy or sell) of the transaction without revealing whether it is the buy-side institution intention to buy or sell.

The two repeating groups, NoEntries and NoBidComponents are mutually exclusive and a function of which bidding model is being used. If the “Non Disclosure” method is being used the portfolio of stocks being traded is described by a number of “bid descriptors” entries. If the “Disclosure” Method is being used the portfolio is fully disclosed, except for side, by a number of “list” entries enumerating the lists that list the stocks to be traded.

A BidRequest message with BidRequestTransType cancel may be used to indicate to sell side firms that they no longer need to store details of the BidRequest as they have either lost the bid or the List has been canceled.

Tag	Field Name	Req'd	Comments
	<i>Standard Header</i>	Y	MsgType = k (lowercase)
390	BidID	N	Required to relate the bid response
391	ClientBidID	Y	
374	BidRequestTransType	Y	Identifies the Bid Request message transaction type
392	ListName	N	
393	TotalNumSecurities	Y	
394	BidType	Y	e.g. “Non Disclosed”, “Disclosed”, No Bidding Process
395	NumTickets	N	Total number of tickets/allocations assuming fully executed
15	Currency	N	Used to represent the currency of monetary amounts.
396	SideValue1	N	Expressed in Currency

397	SideValue2		N	Expressed in Currency
398	NoBidDescriptors		N	Used if BidType="Non Disclosed"
→	399	BidDescriptorType	N	Required if NoBidDescriptors > 0. Must be first field in repeating group.
→	400	BidDescriptor	N	
→	401	SideValueInd	N	Refers to the SideValue1 or SideValue2. These are used as opposed to Buy or Sell so that the basket can be quoted either way as Buy or Sell.
→	404	LiquidityValue	N	Value between LiquidityPctLow and LiquidityPctHigh in Currency
→	441	LiquidityNumSecurities	N	Number of Securites between LiquidityPctLow and LiquidityPctHigh in Currency
→	402	LiquidityPctLow	N	Liquidity indicator or lower limit if LiquidityNumSecurities > 1
→	403	LiquidityPctHigh	N	Upper liquidity indicator if LiquidityNumSecurities > 1
→	405	EFPTrackingError	N	Eg Used in EFP (Exchange For Physical) trades 12%
→	406	FairValue	N	Used in EFP trades
→	407	OutsideIndexPct	N	Used in EFP trades
→	408	ValueOfFutures	N	Used in EFP trades
420	NoBidComponents		N	Used if BidType="Disclosed"
→	66	ListID	N	Required if NoBidComponents > 0. Must be first field in repeating group.
→	54	Side	N	When used in request for a "Disclosed" bid indicates that bid is required on assumption that SideValue1 is Buy or Sell. SideValue2 can be derived by inference.
→	336	TradingSessionID	N	Indicates off-exchange type activities for Detail.
→	430	NetGrossInd	N	Indicates Net or Gross for selling Detail.
→	63	SettlmntTyp	N	Indicates order settlement period for Detail.
→	64	FutSettDate	N	
→	1	Account	N	
409	LiquidityIndType		N	
410	WtAverageLiquidity		N	Overall weighted average liquidity expressed as a % of average daily volume
411	ExchangeForPhysical		N	
412	OutMainCntryUIndex		N	% value of stocks outside main country in Currency
413	CrossPercent		N	% of program that crosses in Currency
414	ProgRptReqs		N	

415	ProgPeriodInterval	N	Time in minutes between each ListStatus report sent by SellSide. Zero means don't send status.
416	IncTaxInd	N	Net/Gross
121	ForexReq	N	Is foreign exchange required
417	NumBidders	N	Indicates the total number of bidders on the list
75	TradeDate	N	
418	TradeType	Y	
419	BasisPxType	Y	
443	StrikeTime	N	Used when BasisPxType = "C"
58	Text	N	
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
	<i>Standard Trailer</i>	Y	

Bid Response -

The Bid Response message can be used in one of two ways depending on which market conventions are being followed.

In the “Non disclosed” convention the Bid Response message can be used to supply a bid based on the sector, country, index and liquidity information contained within the corresponding bid request message.

[See Appendix N – Program/Basket/List Trading for an example.](#)

In the “Disclosed” convention the Bid Response message can be used to supply bids based on the List Order Detail messages sent in advance of the corresponding Bid Request message.

Tag	Field Name		Req'd	Comments
	<i>Standard Header</i>		Y	MsgType = 1 (lowercase L)
390	BidID		N	
391	ClientBidID		N	
420	NoBidComponents		Y	Number of bid repeating groups
→	12	Commission	Y	First element of price. Required if NoBidComponents > 0.
→	13	CommType	Y	
→	66	ListID	N	
→	421	Country	N	ISO Country Code
→	54	Side	N	When used in response to a “Disclosed” request indicates whether SideValue1 is Buy or Sell. SideValue2 can be derived by inference.
→	44	Price	N	Second element of price
→	423	PriceType	N	
→	406	FairValue	N	The difference between the value of a future and the value of the underlying equities after allowing for the discounted cash flows associated with the underlying stocks (E.g. Dividends etc).
→	430	NetGrossInd	N	Net/Gross
→	63	SettlmntTyp	N	Indicates order settlement period for Detail.
→	64	FutSettDate	N	
→	336	TradingSessionID	N	
→	58	Text	N	

→	354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.
→	355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
	<i>Standard Trailer</i>		Y	

New Order - List -

The NewOrderList Message can be used in one of two ways depending on which market conventions are being followed.

In the “Non disclosed” convention the New Order - List message is sent after the bidding process has been completed, by telephone or electronically. The New Order - List message enumerates the stocks, quantities, direction for the trade and may contain pre-allocation information.

This message may also be used as the first message for the transmission of a program trade where the bidding process has been done by means other than FIX. In this scenario the messages may either be used as a staging process, in which case the broker will start execution once either a ListExecute is received or for immediate execution, in which case the orders will be executed on receipt.

In the “Disclosed” convention the New Order - List message is sent before the bidding process is started, by telephone or electronically. The New Order - List message enumerates the stocks and quantities from the bidding process, and may contain pre-allocation information. The direction of the trade is disclosed after the bidding process is completed.

[See Appendix N – Program/Basket/List Trading for an example.](#)

The format for the New Order - List message is as follows:

New Order - List

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = E
66	ListID	Y	Must be unique, by customer, for the day
390	BidID	N	Should refer to an earlier program if bidding took place.
391	ClientBidID	N	
414	ProgRptReqs	N	
394	BidType	Y	e.g. Non Disclosed Model, Disclosed Model, No Bidding Process
415	ProgPeriodInterval	N	
433	ListExecInstType	N	Controls when execution should be begin.
69	ListExecInst	N	Free-form text.
352	EncodedListExecInstLen	N	Must be set if EncodedListExecInst field is specified and must immediately precede it.
353	EncodedListExecInst	N	Encoded (non-ASCII characters) representation of the ListExecInst field in the encoded format specified via the MessageEncoding field.

68	TotNoOrders			Y	Used to support fragmentation. Sum of NoOrders across all messages with the same ListID.
73	NoOrders			Y	Number of orders in this message (number of repeating groups to follow)
→	11	ClOrdID		Y	Must be the first field in the repeating group.
→	67	ListSeqNo		Y	Order number within the list
→	160	SettlInstMode		N	
→	109	ClientID		N	
→	76	ExecBroker		N	
→	1	Account		N	
→	78	NoAllocs		N	Indicates number of pre-trade allocation accounts to follow
→	→	79	AllocAccount	N	Required if NoAllocs > 0. Must be the first field in the repeating group.
→	→	80	AllocShares	N	
→	63	SettlmntTyp		N	
→	64	FutSettDate		N	
→	21	HandlInst		N	
→	18	ExecInst		N	Can contain multiple instructions, space delimited. If OrdType=P, exactly one of the following values (ExecInst = L, R, M, P, O, T, or W) must be specified.
→	110	MinQty		N	
→	111	MaxFloor		N	
→	100	ExDestination		N	
→	386	NoTradingSessions		N	
→	→	336	TradingSessionID	N	First field in repeating group. Required if NoTradingSessions > 0.
→	81	ProcessCode		N	
→	55	Symbol		Y	
→	65	SymbolSfx		N	Can be repeated multiple times if message is related to multiple symbols.
→	48	SecurityID		N	Can be repeated multiple times if message is related to multiple symbols.
→	22	IDSource		N	Can be repeated multiple times if message is related to multiple symbols.
→	167	SecurityType		N	Must be specified if a Future or Option. If a Future: Symbol, SecurityType, and MaturityMonthYear are required. If an Option: Symbol, SecurityType, MaturityMonthYear, PutOrCall, and StrikePrice are required.

→	200	<i>MaturityMonthYear</i>	N	Specifies the month and year of maturity. Required if MaturityDay is specified.
→	205	<i>MaturityDay</i>	N	Can be used in conjunction with MaturityMonthYear to specify a particular maturity date.
→	201	<i>PutOrCall</i>	N	For Options.
→	202	<i>StrikePrice</i>	N	For Options.
→	206	<i>OptAttribute</i>	N	For Options.
→	231	<i>ContractMultiplier</i>	N	For Fixed Income, Convertible Bonds, Derivatives, etc. Note: If used, quantities should be expressed in the "nominal" (e.g. contracts vs. shares) amount.
→	223	<i>CouponRate</i>	N	For Fixed Income.
→	207	<i>SecurityExchange</i>	N	Can be used to identify the security.
→	106	<i>Issuer</i>	N	Can be repeated multiple times if message is related to multiple symbols.
→	348	<i>EncodedIssuerLen</i>	N	Must be set if EncodedIssuer field is specified and must immediately precede it.
→	349	<i>EncodedIssuer</i>	N	Encoded (non-ASCII characters) representation of the Issuer field in the encoded format specified via the MessageEncoding field.
→	107	<i>SecurityDesc</i>	N	Can be repeated multiple times if message is related to multiple symbols.
→	350	<i>EncodedSecurityDescLen</i>	N	Must be set if EncodedSecurityDesc field is specified and must immediately precede it.
→	351	<i>EncodedSecurityDesc</i>	N	Encoded (non-ASCII characters) representation of the SecurityDesc field in the encoded format specified via the MessageEncoding field.
→	140	<i>PrevClosePx</i>	N	Useful for verifying security identification
→	54	<i>Side</i>	Y	Note: to indicate the side of SideValue1 or SideValue2, specify Side=Undisclosed and SideValueInd=either the SideValue1 or SideValue2 indicator.
→	401	<i>SideValueInd</i>	N	Refers to the SideValue1 or SideValue2. These are used as opposed to Buy or Sell so that the basket can be quoted either way as Buy or Sell.
→	114	<i>LocateReqd</i>	N	
→	60	<i>TransactTime</i>	N	
→	38	<i>OrderQty</i>	N	Either CashOrderQty or OrderQty is required. Note that either, but not both, CashOrderQty or OrderQty should be specified.

→	152	<i>CashOrderQty</i>	N	Either CashOrderQty or OrderQty is required. Note that either, but not both, CashOrderQty or OrderQty should be specified. Specifies the approximate "monetary quantity" for the order. Broker is responsible for converting and calculating OrderQty in shares for subsequent messages.
→	40	<i>OrdType</i>	N	
→	44	<i>Price</i>	N	
→	99	<i>StopPx</i>	N	
→	15	<i>Currency</i>	N	
→	376	<i>ComplianceID</i>	N	
→	377	<i>SolicitedFlag</i>	N	
→	23	<i>IOIid</i>	N	Required for Previously Indicated Orders (OrdType=E)
→	117	<i>QuoteID</i>	N	Required for Previously Quoted Orders (OrdType=D)
→	59	<i>TimeInForce</i>	N	
→	168	<i>EffectiveTime</i>	N	
→	432	<i>ExpireDate</i>	N	Conditionally required if TimeInForce = GTD and ExpireTime is not specified.
→	126	<i>ExpireTime</i>	N	Conditionally required if TimeInForce = GTD and ExpireDate is not specified.
→	427	<i>GTBookingInst</i>	N	States whether executions are booked out or accumulated on a partially filled GT order
→	12	<i>Commission</i>	N	
→	13	<i>CommType</i>	N	
→	47	<i>Rule80A</i> (aka <i>OrderCapacity</i>)	N	
→	121	<i>ForexReq</i>	N	
→	120	<i>SettlCurrency</i>	N	
→	58	<i>Text</i>	N	
→	354	<i>EncodedTextLen</i>	N	Must be set if EncodedText field is specified and must immediately precede it.
→	355	<i>EncodedText</i>	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
→	193	<i>FutSettDate2</i>	N	Can be used with OrdType = "Forex - Swap" to specify the "value date" for the future portion of a F/X swap.
→	192	<i>OrderQty2</i>	N	Can be used with OrdType = "Forex - Swap" to specify the order quantity for the future portion of a F/X swap.
→	77	<i>OpenClose</i>	N	

→	203	<i>CoveredOrUncovered</i>	N	
→	204	<i>CustomerOrFirm</i>	N	
→	210	<i>MaxShow</i>	N	
→	211	<i>PegDifference</i>	N	
→	388	<i>DiscretionInst</i>	N	Code to identify the price a DiscretionOffset is related to and should be mathematically added to. Required if DiscretionOffset is specified.
→	389	<i>DiscretionOffset</i>	N	Amount (signed) added to the “related to” price specified via DiscretionInst.
→	439	<i>ClearingFirm</i>	N	
→	440	<i>ClearingAccount</i>	N	
	<i>Standard Trailer</i>		Y	

List Strike Price -

The strike price message is used to exchange strike price information for principal trades. It can also be used to exchange reference prices for agency trades.

Tag	Field Name		Req'd	Comments
	<i>Standard Header</i>		Y	MsgType = m (lowercase)
66	ListID		Y	
422	TotNoStrikes		Y	Used to support fragmentation. Sum of NoStrikes across all messages with the same ListID.
428	NoStrikes		Y	Number of strike price entries
→	55	<i>Symbol</i>	Y	Required if NoStrikes > 0. Must be first field in repeating group.
→	65	<i>SymbolSfx</i>	N	
→	48	<i>SecurityID</i>	N	
→	22	<i>IDSource</i>	N	
→	167	<i>SecurityType</i>	N	Must be specified if a Future or Option. If a Future: Symbol, SecurityType, and MaturityMonthYear are required. If an Option: Symbol, SecurityType, MaturityMonthYear, PutOrCall, and StrikePrice are required.
→	200	<i>MaturityMonthYear</i>	N	Specifies the month and year of maturity. Required if MaturityDay is specified.
→	205	<i>MaturityDay</i>	N	Can be used in conjunction with MaturityMonthYear to specify a particular maturity date.
→	201	<i>PutOrCall</i>	N	For Options.
→	202	<i>StrikePrice</i>	N	For Options.
→	206	<i>OptAttribute</i>	N	For Options.
→	231	<i>ContractMultiplier</i>	N	For Fixed Income, Convertible Bonds, Derivatives, etc. Note: If used, quantities should be expressed in the "nominal" (e.g. contracts vs. shares) amount.
→	223	<i>CouponRate</i>	N	For Fixed Income.
→	207	<i>SecurityExchange</i>	N	Can be used to identify the security.
→	106	<i>Issuer</i>	N	
→	348	<i>EncodedIssuerLen</i>	N	Must be set if EncodedIssuer field is specified and must immediately precede it.
→	349	<i>EncodedIssuer</i>	N	Encoded (non-ASCII characters) representation of the Issuer field in the encoded format specified via the MessageEncoding field.
→	107	<i>SecurityDesc</i>	N	

→	350	<i>EncodedSecurityDescLen</i>	N	Must be set if EncodedSecurityDesc field is specified and must immediately precede it.
→	351	<i>EncodedSecurityDesc</i>	N	Encoded (non-ASCII characters) representation of the SecurityDesc field in the encoded format specified via the MessageEncoding field.
→	140	<i>PrevClosePx</i>	N	Useful for verifying security identification
→	11	<i>ClOrdID</i>	N	Can use client order identifier or the symbol and side to uniquely identify the stock in the list.
→	54	<i>Side</i>	N	
→	44	<i>Price</i>	Y	
→	15	<i>Currency</i>	N	
→	58	<i>Text</i>	N	
→	354	<i>EncodedTextLen</i>	N	Must be set if EncodedText field is specified and must immediately precede it.
→	355	<i>EncodedText</i>	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
	<i>Standard Trailer</i>		Y	

List Status -

The list status message is issued as the response to a List Status Request message sent in an unsolicited fashion by the sell-side. It indicates the current state of the orders within the list as they exists at the broker's site.

Orders within the list are statused at the summary level. Individual executions are not reported, rather, the current state of the order is reported.

The message contains repeating fields for each. The relative position of the repeating fields is important in this message, i.e. each instance of ClOrdID, CumQty, LeavesQty, CxlQty and AvgPx must be in the order shown below.

Description of ListOrderStatus field values:

- "InBiddingProcess": indicates that a list has been received and is being evaluated for pricing. It is envisaged that this status will only be used with the "Disclosed" List Order Trading model.
- "ReceivedForExecution": indicates that a list has been received and the sell side is awaiting the instruction to start working the trade. It is envisaged that this status will be used under both models.
- "Executing": indicates that a list has been received and the sell side is working it.
- "Canceling": indicates that a List Cancel Message has been received and the sell side is in the process of pulling any orders that were being worked. The status of individual order can be found out from the detail repeating group.
- "AllDone": indicates that a list has been executed as far as possible for the day.
- "Alert": used whenever any of the individual orders have a status that requires something to be done. For instance alert would be used when a buy-side firm has submitted a list that has individual stock reject that have not been addressed.
- "Rejected" used when a response cannot be generated. For example when the ListID is not recognised. The text field should include an explanation of why the Request s been rejected.

The list status message format is as follows:

List Status

Tag	Field Name	Req'd	Comments
	<i>Standard Header</i>	Y	MsgType = N
66	ListID	Y	
429	ListStatusType	Y	
82	NoRpts	Y	Total number of messages required to status complete list.
431	ListOrderStatus	Y	
83	RptSeq	Y	Sequence number of this report message.
444	ListStatusText	N	

445	EncodedListStatusTextLen		N	Must be set if EncodedListStatusText field is specified and must immediately precede it.
446	EncodedListStatusText		N	Encoded (non-ASCII characters) representation of the ListStatusText field in the encoded format specified via the MessageEncoding field.
60	TransactTime		N	
68	TotNoOrders		Y	Used to support fragmentation. Sum of NoOrders across all messages with the same ListID.
73	NoOrders		Y	Number of orders statused in this message, i.e. number of repeating groups to follow.
→	11	<i>ClOrdID</i>	Y	
→	14	<i>CumQty</i>	Y	
→	39	<i>OrdStatus</i>	Y	
→	151	<i>LeavesQty</i>	Y	Amount of shares open for further execution. LeavesQty = OrderQty - CumQty.
→	84	<i>CxlQty</i>	Y	
→	6	<i>AvgPx</i>	Y	
→	103	<i>OrdRejReason</i>	N	Used if the order is rejected
→	58	<i>Text</i>	N	
→	354	<i>EncodedTextLen</i>	N	Must be set if EncodedText field is specified and must immediately precede it.
→	355	<i>EncodedText</i>	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
	<i>Standard Trailer</i>		Y	

List Execute -

The list execute message type is used by institutions to instruct the broker to begin execution of a previously submitted list. This message may or may not be used, as it may be mirroring a phone conversation.

The format for the list execute message is as follows:

List Execute

<i>Tag</i>	<i>Field Name</i>	<i>Req'd</i>	<i>Comments</i>
	<i>Standard Header</i>	Y	MsgType = L
66	ListID	Y	Must be unique, by customer, for the day
391	ClientBidID	N	Used with BidType=Disclosed to provide the sell side the ability to determine the direction of the trade to execute.
390	BidID	N	
60	TransactTime	Y	Time this order request was initiated/released by the trader or trading system.
58	Text	N	
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
	<i>Standard Trailer</i>	Y	

List Cancel Request -

The list cancel request message type is used by institutions wishing to cancel previously submitted lists either before or during execution.

After the list has been staged with the broker, it can be canceled via the submission of the List Cancel message. If the list has not yet been submitted for execution, the List Cancel message will instruct the broker not to execute it, if the list is being executed, the List Cancel message should trigger the broker's system to generate cancel requests for the remaining quantities of each order within the list. Individual orders within the list can be canceled via the Order Cancel Request message.

The format for the list - cancel request message is as follows:

List Cancel Request

<i>Tag</i>	<i>Field Name</i>	<i>Req'd</i>	<i>Comments</i>
	<i>Standard Header</i>	Y	MsgType = K
66	ListID	Y	
60	TransactTime	Y	Time this order request was initiated/released by the trader or trading system.
58	Text	N	
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
	<i>Standard Trailer</i>	Y	

List Status Request -

The list status request message type is used by institutions to instruct the broker to generate status messages for a list.

The format for the list - status request message is as follows:

List Status Request

<i>Tag</i>	<i>Field Name</i>	<i>Req'd</i>	<i>Comments</i>
	<i>Standard Header</i>	Y	MsgType = M
66	ListID	Y	
58	Text	N	
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
	<i>Standard Trailer</i>	Y	

Fragmentation for List Order Messages

The messages used in program trading support fragmentation for the same reason and in the same way as some other FIX messages (e.g. Mass Quote). If there are too many entries within a repeating group to fit into one physical message, then the entries can be continued in another message by repeating all of the top level information and then specifying the number of entries in the continued message. A “Total Entries” field is provided to specify the total number of entries in a repeating group which is split over multiple messages. This permits, but does not require, a receiving application to react in a stateful manner where it can determine if it has received all entries in a repeating group before carrying out some action. However, the overall approach to fragmentation is to permit each message to be processed in a stateless manner as it is received. Each message should contain enough information to have the entries applied to a system without requiring the next message if fragmentation has occurred. Also, a continued message should not require any information from the previous message. The messages that support fragmentation and the repeating groups supporting it are listed in the table below.

Message	“Total Entries” field	Repeating group that may be fragmented
New Order - List	TotNoOrders	Orders repeating group following the NoOrders field in the message definition table
List Strike Price	TotNoStrikes	Strike price repeating group following the NoStrikes field in the message definition table
List Status	TotNoOrders	Status per order repeating group following the NoOrders field in the message definition table

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

Note: The TotNoOrders field has been added to the List Status message to support fragmentation in same way as other FIX messages. The NoRpts and RptSeq fields are preserved to backwards compatibility with previous versions of FIX which supported a stateful form of fragmentation.

Business Message Reject -

The Business Message Reject message can reject an application-level message which fulfills session-level rules and cannot be rejected via any other means. Note if the message fails a session-level rule (e.g. body length is incorrect), a session-level Reject message should be issued.

[See the session-level Reject message](#)

It should ***NOT*** be used in the following situations:

Session-level problem meeting the criteria of the session-level Reject message	Use the session-level Reject message (MsgType=3)
In response to <ul style="list-style-type: none">• New Order - Single• Order Status Request• New Order - List• List Execute	Use the Execution Report message
In response to: <ul style="list-style-type: none">• Order Cancel Request• Order Cancel/Replace Request• List Cancel Request	Use the Order Cancel Reject message
In response to: <ul style="list-style-type: none">• Execution Report	Use the Don't Know Trade (DK) message
In response to: <ul style="list-style-type: none">• Allocation	Use the Allocation ACK message
In response to: <ul style="list-style-type: none">• List Status Request	Use the List Status message
In response to: <ul style="list-style-type: none">• Quote Request• Quote• Mass Quote• Quote Cancel• Quote Status Request	Use the Quote Acknowledgment message
In response to: <ul style="list-style-type: none">• Market Data Request	Use the Market Data Request Reject message
In response to: <ul style="list-style-type: none">• Security Definition Request	Use the Security Definition message
In response to: <ul style="list-style-type: none">• Security Status Request	Use the Security Status message
In response to:	Use the Trading Session Status message

• Trading Session Status Request	
----------------------------------	--

Note the only exception to this rule is in the event a business message is received, fulfills session-level rules, however, the message cannot be communicated to the business-level processing system. In this situation a Business Message Reject with BusinessRejectReason = “Application not available at this time” can be issued if the the system is unable to send the specific “reject” message listed above due to this condition.

Messages which can be referenced via the Business Message Reject message are:

(the “ID” field BusinessRejectRefID refers to noted in [])
• Indication of Interest (IOI) [IOId]
• Advertisement [AdvId]
• News [Headline]
• Email [EmailThreadID]
• Order Cancel Reject [ClOrdID]
• Allocation ACK [AllocID]
• List Status [ListID]
• Don’t Know Trade (DK) – may respond with Order Cancel Reject if attempting to cancel order [ExecID]
• Settlement Instructions [SettlInstID]
• Market Data-Snapshot/Full Refresh [MDReqID]
• Market Data-Incremental Refresh [MDReqID]
• Market Data Request Reject [MDReqID]
• Quote Acknowledgment [QuoteID]
• Security Definition [SecurityResponseID]
• Security Status [SecurityStatusReqID]
• Trading Session Status [TradSesReqID]

Scenarios for Business Message Reject:

BusinessRejectReason
0 = Other
1 = Unkown ID
2 = Unknown Security
3 = Unsupported Message Type (receive a valid, but unsupported MsgType)
4 = Application not available

5 = Conditionally Required Field Missing
--

Whenever possible, it is strongly recommended that the cause of the failure be described in the Text field (e.g. “UNKNOWN SYBMOL: XYZ”).

The business message reject format is as follows:

Business Message Reject

<i>Tag</i>	<i>Field Name</i>	<i>Req'd</i>	<i>Comments</i>
	<i>Standard Header</i>	Y	MsgType = j (lowercase)
45	RefSeqNum	N	MsgSeqNum of rejected message
372	RefMsgType	Y	The MsgType of the FIX message being referenced.
379	BusinessRejectRefID	N	The value of the business-level “ID” field on the message being referenced. Required unless the corresponding ID field (see list above) was not specified.
380	BusinessRejectReason	Y	Code to identify reason for a Business Message Reject message.
58	Text	N	Where possible, message to explain reason for rejection
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
	<i>Standard Trailer</i>	Y	

Field Definitions

The following is a catalog of fields used to define the application and session protocol messages.

Please refer to the [“Data Types”](#) section for the definition and format of values within the “Format” column as well.

Field ID (TAG)	Field Name	Format	Description
1	Account	String	Account mnemonic as agreed between broker and institution.
2	AdvId	String	Unique identifier of advertisement message. (Prior to FIX 4.1 this field was of type int)
3	AdvRefID	String	Reference identifier used with CANCEL and REPLACE transaction types. (Prior to FIX 4.1 this field was of type int)
4	AdvSide	Char	Broker's side of advertised trade Valid values: B = Buy S = Sell X = Cross T = Trade
5	AdvTransType	String	Identifies advertisement message transaction type Valid values: N = New C = Cancel R = Replace
6	AvgPx	Price	Calculated average price of all fills on this order.
7	BeginSeqNo	int	Message sequence number of first message in range to be resent
8	BeginString	String	Identifies beginning of new message and protocol version. ALWAYS FIRST FIELD IN MESSAGE. (Always unencrypted) Valid values: FIX.4.2
9	BodyLength	int	Message length, in bytes, forward to the CheckSum field. ALWAYS SECOND FIELD IN MESSAGE. (Always unencrypted)

10	Checksum	String	Three byte, simple checksum (see Appendix B: CheckSum Calculation for description). ALWAYS LAST FIELD IN MESSAGE; i.e. serves, with the trailing <SOH>, as the end-of-message delimiter. Always defined as three characters. (Always unencrypted)
11	ClOrdID	String	Unique identifier for Order as assigned by institution (identified by SenderCompID or OnBehalfOfCompID as appropriate). Uniqueness must be guaranteed within a single trading day. Firms, particularly those which electronically submit multi-day orders, trade globally or throughout market close periods, should ensure uniqueness across days, for example by embedding a date within the ClOrdID field.
12	Commission	Amt	Commission. Note if CommType is percentage, Commission of 5% should be represented as .05.
13	CommType	char	Commission type Valid values: 1 = per share 2 = percentage 3 = absolute
14	CumQty	Qty	Total number of shares filled. (Prior to FIX 4.2 this field was of type int)
15	Currency	Currency	Identifies currency used for price. Absence of this field is interpreted as the default for the security. It is recommended that systems provide the currency value whenever possible. See Appendix A: Valid Currency Codes for information on obtaining valid values.
16	EndSeqNo	int	Message sequence number of last message in range to be resent. If request is for a single message BeginSeqNo = EndSeqNo. If request is for all messages subsequent to a particular message, EndSeqNo = "0" (representing infinity).
17	ExecID	String	Unique identifier of execution message as assigned by broker (will be 0 (zero) for ExecTransType=3 (Status)). Uniqueness must be guaranteed within a single trading day or the life of a multi-day order. Firms which accept multi-day orders should consider embedding a date within the ExecID field to assure uniqueness across days. (Prior to FIX 4.1 this field was of type int)

18	ExecInst	Multiple ValueString	<p>Instructions for order handling on exchange trading floor. If more than one instruction is applicable to an order, this field can contain multiple instructions separated by space.</p> <p>Valid values:</p> <ul style="list-style-type: none"> 1 = Not held 2 = Work 3 = Go along 4 = Over the day 5 = Held 6 = Participate don't initiate 7 = Strict scale 8 = Try to scale 9 = Stay on bidside 0 = Stay on offerside A = No cross (cross is forbidden) B = OK to cross C = Call first D = Percent of volume “(indicates that the sender does not want to be all of the volume on the floor vs. a specific percentage)” E = Do not increase - DNI F = Do not reduce - DNR G = All or none - AON I = Institutions only L = Last peg (last sale) M = Mid-price peg (midprice of inside quote) N = Non-negotiable O = Opening peg P = Market peg R = Primary peg (primary market - buy at bid/sell at offer) S = Suspend T = Fixed Peg to Local best bid or offer at time of order U = Customer Display Instruction (Rule 11Ac1-1/4) V = Netting (for Forex) W = Peg to VWAP
19	ExecRefID	String	<p>Reference identifier used with Cancel and Correct transaction types.</p> <p>(Prior to FIX 4.1 this field was of type int)</p>
20	ExecTransType	char	<p>Identifies transaction type</p> <p>Valid values:</p> <ul style="list-style-type: none"> 0 = New 1 = Cancel 2 = Correct 3 = Status

21	HandlInst	char	Instructions for order handling on Broker trading floor Valid values: 1 = Automated execution order, private, no Broker intervention 2 = Automated execution order, public, Broker intervention OK 3 = Manual order, best execution
22	IDSource	String	Identifies class of alternative SecurityID Valid values: 1 = CUSIP 2 = SEDOL 3 = QUIK 4 = ISIN number 5 = RIC code 6 = ISO Currency Code 7 = ISO Country Code 8 = Exchange Symbol 9 = Consolidated Tape Association (CTA) Symbol (SIAC CTS/CQS line format) 100+ are reserved for private security identifications
23	IOIid	String	Unique identifier of IOI message. (Prior to FIX 4.1 this field was of type int)
24	IOIOthSvc (no longer used)	char	No longer used as of FIX 4.2. Included here for reference to prior versions.
25	IOIQltyInd	char	Relative quality of indication Valid values: L = Low M = Medium H = High
26	IOIRefID	String	Reference identifier used with CANCEL and REPLACE, transaction types. (Prior to FIX 4.1 this field was of type int)
27	IOIShares	String	Number of shares in numeric or relative size. Valid values: 0 - 1000000000 S = Small M = Medium L = Large

28	IOITransType	char	Identifies IOI message transaction type Valid values: N = New C = Cancel R = Replace
29	LastCapacity	char	Broker capacity in order execution Valid values: 1 = Agent 2 = Cross as agent 3 = Cross as principal 4 = Principal
30	LastMkt	Exchange	Market of execution for last fill Valid values: <u>See Appendix C</u>
31	LastPx	Price	Price of this (last) fill. Field not required for ExecTransType = 3 (Status)
32	LastShares	Qty	Quantity of shares bought/sold on this (last) fill. Field not required for ExecTransType = 3 (Status) (Prior to FIX 4.2 this field was of type int)
33	LinesOfText	int	Identifies number of lines of text body
34	MsgSeqNum	int	Integer message sequence number.

35	MsgType	String	<p>Defines message type. ALWAYS THIRD FIELD IN MESSAGE. (Always unencrypted)</p> <p>Note: A "U" as the first character in the MsgType field (i.e. U1, U2, etc) indicates that the message format is privately defined between the sender and receiver.</p> <p>Valid values: *** Note the use of lower case letters ***</p> <ul style="list-style-type: none"> 0 = Heartbeat 1 = Test Request 2 = Resend Request 3 = Reject 4 = Sequence Reset 5 = Logout 6 = Indication of Interest 7 = Advertisement 8 = Execution Report 9 = Order Cancel Reject A = Logon B = News C = Email D = Order – Single E = Order – List F = Order Cancel Request G= Order Cancel/Replace Request H= Order Status Request J = Allocation K = List Cancel Request L = List Execute M = List Status Request N = List Status P = Allocation ACK Q = Don't Know Trade (DK) R = Quote Request S = Quote T = Settlement Instructions V = Market Data Request W = Market Data-Snapshot/Full Refresh X = Market Data-Incremental Refresh Y = Market Data Request Reject Z = Quote Cancel a = Quote Status Request b = Quote Acknowledgement c = Security Definition Request d = Security Definition e = Security Status Request f = Security Status g = Trading Session Status Request h = Trading Session Status i = Mass Quote j = Business Message Reject k = Bid Request l = Bid Response (lowercase L) m = List Strike Price
----	---------	--------	--

36	NewSeqNo	int	New sequence number
37	OrderID	String	Unique identifier for Order as assigned by broker. Uniqueness must be guaranteed within a single trading day. Firms which accept multi-day orders should consider embedding a date within the OrderID field to assure uniqueness across days.
38	OrderQty	Qty	Number of shares ordered. This represents the number of shares for equities or based on normal convention the number of contracts for options, futures, convertible bonds, etc. (Prior to FIX 4.2 this field was of type int)
39	OrdStatus	char	Identifies current status of order. Valid values: 0 = New 1 = Partially filled 2 = Filled 3 = Done for day 4 = Canceled 5 = Replaced 6 = Pending Cancel (e.g. result of Order Cancel Request) 7 = Stopped 8 = Rejected 9 = Suspended A = Pending New B = Calculated C = Expired D = Accepted for bidding E = Pending Replace (e.g. result of Order Cancel/Replace Request)

40	OrdType	char	Order type. Valid values: 1 = Market 2 = Limit 3 = Stop 4 = Stop limit 5 = Market on close 6 = With or without 7 = Limit or better 8 = Limit with or without 9 = On basis A = On close B = Limit on close C =Forex - Market D = Previously quoted E = Previously indicated F = Forex - Limit G = Forex - Swap H = Forex - Previously Quoted I = Funari (Limit Day Order with unexecuted portion handled as Market On Close. e.g. Japan) P = Pegged
41	OrigClOrdID	String	ClOrdID of the previous order (NOT the initial order of the day) as assigned by the institution, used to identify the previous order in cancel and cancel/replace requests.
42	OrigTime	UTCTimestamp	Time of message origination (always expressed in UTC (Universal Time Coordinated, also known as “GMT”))
43	PossDupFlag	Boolean	Indicates possible retransmission of message with this sequence number Valid values: Y = Possible duplicate N = Original transmission
44	Price	Price	Price per share
45	RefSeqNum	int	Reference message sequence number
46	RelatdSym	String	Symbol of issue related to story. Can be repeated within message to identify multiple companies.

47	Rule80A (aka OrderCapacity)	char	<p>Note that the name of this field is changing to “OrderCapacity” as Rule80A is a very US market-specific term. Other world markets need to convey similar information, however, often a subset of the US values. . See the <u>“Rule80A (aka OrderCapacity) Usage by Market” appendix</u> for market-specific usage of this field. Valid values:</p> <ul style="list-style-type: none"> A = Agency single order B = Short exempt transaction (refer to A type) C = Program Order, non-index arb, for Member firm/org D = Program Order, index arb, for Member firm/org E = Registered Equity Market Maker trades F = Short exempt transaction (refer to W type) H = Short exempt transaction (refer to I type) I = Individual Investor, single order J = Program Order, index arb, for individual customer K = Program Order, non-index arb, for individual customer L = Short exempt transaction for member competing market-maker affiliated with the firm clearing the trade (refer to P and O types) M = Program Order, index arb, for other member N = Program Order, non-index arb, for other member O = Competing dealer trades P = Principal R = Competing dealer trades S = Specialist trades T = Competing dealer trades U = Program Order, index arb, for other agency W = All other orders as agent for other member X = Short exempt transaction for member competing market-maker not affiliated with the firm clearing the trade (refer to W and T types) Y = Program Order, non-index arb, for other agency Z = Short exempt transaction for non-member competing market-maker (refer to A and R types)
48	SecurityID	String	CUSIP or other alternate security identifier
49	SenderCompID	String	Assigned value used to identify firm sending message.
50	SenderSubID	String	Assigned value used to identify specific message originator (desk, trader, etc.)
51	SendingDate (no longer used)	LocalMk tDate	No longer used. Included here for reference to prior versions.

52	SendingTime	UTCTim estamp	Time of message transmission (always expressed in UTC (Universal Time Coordinated, also known as "GMT"))
53	Shares	Qty	Number of shares (Prior to FIX 4.2 this field was of type int)
54	Side	char	Side of order Valid values: 1 = Buy 2 = Sell 3 = Buy minus 4 = Sell plus 5 = Sell short 6 = Sell short exempt 7 = Undisclosed (valid for IOI and List Order messages only) 8 = Cross (orders where counterparty is an exchange, valid for all messages except IOIs) 9 = Cross short
55	Symbol	String	Ticker symbol
56	TargetCompID	String	Assigned value used to identify receiving firm.
57	TargetSubID	String	Assigned value used to identify specific individual or unit intended to receive message. "ADMIN" reserved for administrative messages not intended for a specific user.
58	Text	String	Free format text string (Note: this field does not have a specified maximum length)
59	TimeInForce	char	Specifies how long the order remains in effect. Absence of this field is interpreted as DAY. Valid values: 0 = Day 1 = Good Till Cancel (GTC) 2 = At the Opening (OPG) 3 = Immediate or Cancel (IOC) 4 = Fill or Kill (FOK) 5 = Good Till Crossing (GTX) 6 = Good Till Date
60	TransactTime	UTCTim estamp	Time of execution/order creation (expressed in UTC (Universal Time Coordinated, also known as "GMT"))
61	Urgency	char	Urgency flag Valid values: 0 = Normal 1 = Flash 2 = Background

62	ValidUntilTime	UTCTim estamp	Indicates expiration time of indication message (always expressed in UTC (Universal Time Coordinated, also known as "GMT"))
63	SettlmntTyp	char	Indicates order settlement period. Absence of this field is interpreted as Regular. Regular is defined as the default settlement period for the particular security on the exchange of execution. Valid values: 0 = Regular 1 = Cash 2 = Next Day 3 = T+2 4 = T+3 5 = T+4 6 = Future 7 = When Issued 8 = Sellers Option 9 = T+ 5
64	FutSettDate	LocalMk tDate	Specific date of trade settlement (SettlementDate) in YYYYMMDD format. Required when SettlmntTyp = 6 (Future) or SettlmntTyp = 8 (Sellers Option). (expressed in local time at place of settlement)
65	SymbolSfx	String	Additional information about the security (e.g. preferred, warrants, etc.). Note also see SecurityType. Valid values: As defined in the NYSE Stock and bond Symbol Directory and in the AMEX Fitch Directory
66	ListID	String	Unique identifier for list as assigned by institution, used to associate multiple individual orders. Uniqueness must be guaranteed within a single trading day. Firms which generate multi-day orders should consider embedding a date within the ListID field to assure uniqueness across days.
67	ListSeqNo	int	Sequence of individual order within list (i.e. ListSeqNo of ListNoOrds, 2 of 25, 3 of 25, . . .)
68	TotNoOrders (formerly ListNoOrds) named:	int	Total number of list order entries across all messages. Should be the sum of all NoOrders in each message that has repeating list order entries related to the same ListID. Used to support fragmentation. (Prior to FIX 4.2 this field was named "ListNoOrds")
69	ListExecInst	String	Free format text message containing list handling and execution instructions.
70	AllocID	String	Unique identifier for allocation message. (Prior to FIX 4.1 this field was of type int)

71	AllocTransType	char	Identifies allocation transaction type Valid values: 0 = New 1 = Replace 2 = Cancel 3 = Preliminary (without MiscFees and NetMoney) 4 = Calculated (includes MiscFees and NetMoney) 5 = Calculated without Preliminary (sent unsolicited by broker, includes MiscFees and NetMoney)
72	RefAllocID	String	Reference identifier to be used with Replace, Cancel, and Calculated AllocTransType messages. (Prior to FIX 4.1 this field was of type int)
73	NoOrders	int	Indicates number of orders to be combined for average pricing and allocation.
74	AvgPrxPrecision	int	Indicates number of decimal places to be used for average pricing. Absence of this field indicates that default precision arranged by the broker/institution is to be used.
75	TradeDate	LocalMktDate	Indicates date of trade referenced in this message in YYYYMMDD format. Absence of this field indicates current day (expressed in local time at place of trade).
76	ExecBroker	String	Identifies executing / give-up broker. Standard NASD market-maker mnemonic is preferred.
77	OpenClose	char	Indicates whether the resulting position after a trade should be an opening position or closing position. Used for omnibus accounting - where accounts are held on a gross basis instead of being netted together. Valid Values: O=Open C=Close
78	NoAllocs	int	Number of repeating AllocAccount/AllocPrice entries.
79	AllocAccount	String	Sub-account mnemonic
80	AllocShares	Qty	Number of shares to be allocated to specific sub-account (Prior to FIX 4.2 this field was of type int)

81	ProcessCode	char	Processing code for sub-account. Absence of this field in AllocAccount / AllocPrice/AllocShares / ProcessCode instance indicates regular trade. Valid values: 0 = regular 1 = soft dollar 2 = step-in 3 = step-out 4 = soft-dollar step-in 5 = soft-dollar step-out 6 = plan sponsor
82	NoRpts	int	Total number of reports within series.
83	RptSeq	int	Sequence number of message within report series.
84	CxlQty	Qty	Total number of shares canceled for this order. (Prior to FIX 4.2 this field was of type int)
85	NoDlvyInst (no longer used)	int	Number of delivery instruction fields to follow No longer used. Included here for reference to prior versions.
86	DlvyInst (no longer used)	String	Free format text field to indicate delivery instructions No longer used. Included here for reference to prior versions.
87	AllocStatus	int	Identifies status of allocation. Valid values: 0 = accepted (successfully processed) 1 = rejected 2 = partial accept 3 = received (received, not yet processed)
88	AllocRejCode	int	Identifies reason for rejection. Valid values: 0 = unknown account(s) 1 = incorrect quantity 2 = incorrect average price 3 = unknown executing broker mnemonic 4 = commission difference 5 = unknown OrderID 6 = unknown ListID 7 = other
89	Signature	data	Electronic signature
90	SecureDataLen	int	Length of encrypted message
91	SecureData	data	Actual encrypted data stream
92	BrokerOfCredit	String	Broker to receive trade credit.
93	SignatureLength	int	Number of bytes in signature field.

94	EmailType	char	Email message type. Valid values: 0 = New 1 = Reply 2 = Admin Reply
95	RawDataLength	int	Number of bytes in raw data field.
96	RawData	data	Unformatted raw data, can include bitmaps, word processor documents, etc.
97	PossResend	Boolean	Indicates that message may contain information that has been sent under another sequence number. Valid Values: Y=Possible resend N=Original transmission
98	EncryptMethod	int	Method of encryption. Valid values: 0 = None / other 1 = PKCS (proprietary) 2 = DES (ECB mode) 3 = PKCS/DES (proprietary) 4 = PGP/DES (defunct) 5 = PGP/DES-MD5 (see app note on FIX web site) 6 = PEM/DES-MD5 (see app note on FIX web site)
99	StopPx	Price	Price per share
100	ExDestination	Exchange	Execution destination as defined by institution when order is entered. Valid values: <u>See Appendix C</u>
101	(Not Defined)	n/a	This field has not been defined.
102	CxlRejReason	int	Code to identify reason for cancel rejection. Valid values: 0 = Too late to cancel 1 = Unknown order 2 = Broker Option 3 = Order already in Pending Cancel or Pending Replace status

103	OrdRejReason	int	Code to identify reason for order rejection. Valid values: 0 = Broker option 1 = Unknown symbol 2 = Exchange closed 3 = Order exceeds limit 4 = Too late to enter 5 = Unknown Order 6 = Duplicate Order (e.g. dupe ClOrdID) 7 = Duplicate of a verbally communicated order 8 = Stale Order
104	IOIQualifier	char	Code to qualify IOI use. Valid values: A = All or none C = At the close I = In touch with L = Limit M = More behind O = At the open P = Taking a position Q = At the Market (previously called Current Quote) R = Ready to trade S = Portfolio show-n T = Through the day V = Versus W = Indication - Working away X = Crossing opportunity Y = At the Midpoint Z = Pre-open
105	WaveNo	String	Identifier to aid in the management of multiple lists derived from a single, master list.
106	Issuer	String	Company name of security issuer (e.g. International Business Machines)
107	SecurityDesc	String	Security description.
108	HeartBtInt	int	Heartbeat interval (seconds)
109	ClientID	String	Firm identifier used in third party-transactions (should not be a substitute for OnBehalfOfCompID/DeliverToCompID).
110	MinQty	Qty	Minimum quantity of an order to be executed. (Prior to FIX 4.2 this field was of type int)

111	MaxFloor	Qty	Maximum number of shares within an order to be shown on the exchange floor at any given time. (Prior to FIX 4.2 this field was of type int)
112	TestReqID	String	Identifier included in Test Request message to be returned in resulting Heartbeat
113	ReportToExch	Boolean	Identifies party of trade responsible for exchange reporting. Valid values: Y = Indicates that party receiving message must report trade N = Indicates that party sending message will report trade
114	LocateReqd	Boolean	Indicates whether the broker is to locate the stock in conjunction with a short sell order. Valid values: Y = Indicates the broker is responsible for locating the stock N = Indicates the broker is not required to locate
115	OnBehalfOfCompID	String	Assigned value used to identify firm originating message if the message was delivered by a third party i.e. the third party firm identifier would be delivered in the SenderCompID field and the firm originating the message in this field.
116	OnBehalfOfSubID	String	Assigned value used to identify specific message originator (i.e. trader) if the message was delivered by a third party
117	QuoteID	String	Unique identifier for quote
118	NetMoney	Amt	Total amount due as the result of the transaction (e.g. for Buy order - principal + commission + fees) reported in currency of execution.
119	SettlCurrAmt	Amt	Total amount due expressed in settlement currency (includes the effect of the forex transaction)
120	SettlCurrency	Currency	Currency code of settlement denomination.
121	ForexReq	Boolean	Indicates request for forex accommodation trade to be executed along with security transaction. Valid values: Y = Execute Forex after security trade N = Do not execute Forex after security trade
122	OrigSendingTime	UTCTim estamp	Original time of message transmission (always expressed in UTC (Universal Time Coordinated, also known as "GMT") when transmitting orders as the result of a resend request.

123	GapFillFlag	Boolean	Indicates that the Sequence Reset message is replacing administrative or application messages which will not be resent. Valid values: Y = Gap Fill message, MsgSeqNum field valid N = Sequence Reset, ignore MsgSeqNum
124	NoExecs	int	No of execution repeating group entries to follow.
125	CxlType (no longer used)	char	No longer used. Included here for reference to prior versions.
126	ExpireTime	UTCTim estamp	Time/Date of order expiration (always expressed in UTC (Universal Time Coordinated, also known as "GMT"))
127	DKReason	char	Reason for execution rejection. Valid values: A = Unknown symbol B = Wrong side C = Quantity exceeds order D = No matching order E = Price exceeds limit Z = Other
128	DeliverToCompID	String	Assigned value used to identify the firm targeted to receive the message if the message is delivered by a third party i.e. the third party firm identifier would be delivered in the TargetCompID field and the ultimate receiver firm ID in this field.
129	DeliverToSubID	String	Assigned value used to identify specific message recipient (i.e. trader) if the message is delivered by a third party
130	IOINaturalFlag	Boolean	Indicates that IOI is the result of an existing agency order or a facilitation position resulting from an agency order, not from principal trading or order solicitation activity. Valid values: Y = Natural N = Not natural
131	QuoteReqID	String	Unique identifier for quote request
132	BidPx	Price	Bid price/rate
133	OfferPx	Price	Offer price/rate
134	BidSize	Qty	Quantity of bid (Prior to FIX 4.2 this field was of type int)
135	OfferSize	Qty	Quantity of offer (Prior to FIX 4.2 this field was of type int)

136	NoMiscFees	int	Number of repeating groups of miscellaneous fees
137	MiscFeeAmt	Amt	Miscellaneous fee value
138	MiscFeeCurr	Currency	Currency of miscellaneous fee
139	MiscFeeType	char	Indicates type of miscellaneous fee. Valid values: 1 = Regulatory (e.g. SEC) 2 = Tax 3 = Local Commission 4 = Exchange Fees 5 = Stamp 6 = Levy 7 = Other 8 = Markup 9 = Consumption Tax
140	PrevClosePx	Price	Previous closing price of security.
141	ResetSeqNumFlag	Boolean	Indicates that the both sides of the FIX session should reset sequence numbers. Valid values: Y = Yes, reset sequence numbers N = No
142	SenderLocationID	String	Assigned value used to identify specific message originator's location (i.e. geographic location and/or desk, trader)
143	TargetLocationID	String	Assigned value used to identify specific message destination's location (i.e. geographic location and/or desk, trader)
144	OnBehalfOfLocationID	String	Assigned value used to identify specific message originator's location (i.e. geographic location and/or desk, trader) if the message was delivered by a third party
145	DeliverToLocationID	String	Assigned value used to identify specific message recipient's location (i.e. geographic location and/or desk, trader) if the message was delivered by a third party
146	NoRelatedSym	int	Specifies the number of repeating symbols specified.
147	Subject	String	The subject of an Email message
148	Headline	String	The headline of a News message
149	URLLink	String	A URL (Uniform Resource Locator) link to additional information (i.e. http://www.XYZ.com/research.html)

150	ExecType	char	<p>Describes the specific ExecutionRpt (i.e. Pending Cancel) while OrdStatus will always identify the current order status (i.e. Partially Filled)</p> <p>Valid values:</p> <ul style="list-style-type: none"> 0 = New 1 = Partial fill 2 = Fill 3 = Done for day 4 = Canceled 5 = Replace 6 = Pending Cancel (e.g. result of Order Cancel Request) 7 = Stopped 8 = Rejected 9 = Suspended A = Pending New B = Calculated C = Expired D = Restated (ExecutionRpt sent unsolicited by sellside, with ExecRestatementReason set) E = Pending Replace (e.g. result of Order Cancel/Replace Request)
151	LeavesQty	Qty	<p>Amount of shares open for further execution. If the OrdStatus is Canceled, DoneForTheDay, Expired, Calculated, or Rejected (in which case the order is no longer active) then LeavesQty could be 0, otherwise $\text{LeavesQty} = \text{OrderQty} - \text{CumQty}$.</p> <p>(Prior to FIX 4.2 this field was of type int)</p>
152	CashOrderQty	Qty	Specifies the approximate order quantity desired in total monetary units vs. as a number of shares. The broker would be responsible for converting and calculating a share quantity (OrderQty) based upon this amount to be used for the actual order and subsequent messages.
153	AllocAvgPx	Price	AvgPx for a specific AllocAccount
154	AllocNetMoney	Amt	NetMoney for a specific AllocAccount
155	SettlCurrFxRate	float	Foreign exchange rate used to compute SettlCurrAmt from Currency to SettlCurrency
156	SettlCurrFxRateCalc	char	<p>Specifies whether or not SettlCurrFxRate should be multiplied or divided.</p> <ul style="list-style-type: none"> M=Multiply D=Divide
157	NumDaysInterest	int	Number of Days of Interest for convertible bonds and fixed income
158	AccruedInterestRate	float	Accrued Interest Rate for convertible bonds and fixed income

159	AccruedInterestAmt	Amt	Amount of Accrued Interest for convertible bonds and fixed income
160	SettlInstMode	char	Indicates mode used for Settlement Instructions Valid values: 0 = Default 1 = Standing Instructions Provided 2 = Specific Allocation Account Overriding 3 = Specific Allocation Account Standing
161	AllocText	String	Free format text related to a specific AllocAccount.
162	SettlInstID	String	Unique identifier for Settlement Instructions message.
163	SettlInstTransType	char	Settlement Instructions message transaction type Valid values: N = New C = Cancel R = Replace
164	EmailThreadID	String	Unique identifier for an email thread (new and chain of replies)
165	SettlInstSource	char	Indicates source of Settlement Instructions Valid values: 1 = Broker's Instructions 2 = Institution's Instructions
166	SettlLocation	String	Identifies Settlement Depository or Country Code (ISITC spec) Valid values: CED = CEDEL DTC = Depository Trust Company EUR = Euroclear FED = Federal Book Entry PNY= Physical PTC = Participant Trust Company ISO Country Code = Local Market Settle Location

167	SecurityType	String	<p>Indicates type of security (ISITC spec)</p> <p>Valid values:</p> <p>BA = Bankers Acceptance</p> <p>CB = Convertible Bond (Note not part of ISITC spec)</p> <p>CD = Certificate Of Deposit</p> <p>CMO = Collateralize Mortgage Obligation</p> <p>CORP = Corporate Bond</p> <p>CP = Commercial Paper</p> <p>CPP = Corporate Private Placement</p> <p>CS = Common Stock</p> <p>FHA = Federal Housing Authority</p> <p>FHL = Federal Home Loan</p> <p>FN = Federal National Mortgage Association</p> <p>FOR = Foreign Exchange Contract</p> <p>FUT = Future</p> <p>GN = Government National Mortgage Association</p> <p>GOVT = Treasuries + Agency Debenture</p> <p>IET Mortgage IOETTE</p> <p>MF = Mutual Fund</p> <p>MIO = Mortgage Interest Only</p> <p>MPO = Mortgage Principal Only</p> <p>MPP = Mortgage Private Placement</p> <p>MPT = Miscellaneous Pass-Thru</p> <p>MUNI = Municipal Bond</p> <p>NONE = No ISITC Security Type</p> <p>OPT = Option</p> <p>PS = Preferred Stock</p> <p>RP = Repurchase Agreement</p> <p>RVRP = Reverse Repurchase Agreement</p> <p>SL = Student Loan Marketing Association</p> <p>TD = Time Deposit</p> <p>USTB = US Treasury Bill</p> <p>WAR = Warrant</p> <p>ZOO = Cats, Tigers & Lions (a real code: US Treasury Receipts)</p> <p>? = "Wildcard" entry (used on Security Definition Request message)</p>
168	EffectiveTime	UTCTim estamp	Time the details within the message should take effect (always expressed in UTC (Universal Time Coordinated, also known as "GMT"))
169	StandInstDbType	int	<p>Identifies the Standing Instruction database used</p> <p>Valid values:</p> <p>0 = Other</p> <p>1 = DTC SID</p> <p>2 = Thomson ALERT</p> <p>3 = A Global Custodian (StandInstDbName must be provided)</p>

170	StandInstDbName	String	Name of the Standing Instruction database represented with StandInstDbType (i.e. the Global Custodian's name).
171	StandInstDbID	String	Unique identifier used on the Standing Instructions database for the Standing Instructions to be referenced.
172	SettlDeliveryType	int	Identifies type of settlement 0 = "Versus. Payment": Deliver (if Sell) or Receive (if Buy) vs. (Against) Payment 1 = "Free": Deliver (if Sell) or Receive (if Buy) Free
173	SettlDepositoryCode	String	Broker's account code at the depository (i.e. CEDEL ID for CEDEL, FINS for DTC, or Euroclear ID for Euroclear) if SettlLocation is a depository
174	SettlBrkrCode	String	BIC (Bank Identification Code—Swift managed) code of the broker involved (i.e. for multi-company brokerage firms)
175	SettlInstCode	String	BIC (Bank Identification Code—Swift managed) code of the institution involved (i.e. for multi-company institution firms)
176	SecuritySettlAgentName	String	Name of SettlInstSource's local agent bank if SettlLocation is not a depository
177	SecuritySettlAgentCode	String	BIC (Bank Identification Code--Swift managed) code of the SettlInstSource's local agent bank if SettlLocation is not a depository
178	SecuritySettlAgentAccountNum	String	SettlInstSource's account number at local agent bank if SettlLocation is not a depository
179	SecuritySettlAgentAccountName	String	Name of SettlInstSource's account at local agent bank if SettlLocation is not a depository
180	SecuritySettlAgentContactName	String	Name of contact at local agent bank for SettlInstSource's account if SettlLocation is not a depository
181	SecuritySettlAgentContactPhone	String	Phone number for contact at local agent bank if SettlLocation is not a depository
182	CashSettlAgentName	String	Name of SettlInstSource's local agent bank if SettlDeliveryType=Free
183	CashSettlAgentCode	String	BIC (Bank Identification Code--Swift managed) code of the SettlInstSource's local agent bank if SettlDeliveryType=Free
184	CashSettlAgentAccountNum	String	SettlInstSource's account number at local agent bank if SettlDeliveryType=Free
185	CashSettlAgentAccountName	String	Name of SettlInstSource's account at local agent bank if SettlDeliveryType=Free
186	CashSettlAgentContactName	String	Name of contact at local agent bank for SettlInstSource's account if SettlDeliveryType=Free

187	CashSettlAgentContactPhone	String	Phone number for contact at local agent bank for SettlInstSource's account if SettlDeliveryType=Free
188	BidSpotRate	Price	Bid F/X spot rate.y vary and not limited to four)
189	BidForwardPoints	PriceOffset	Bid F/X forward points added to spot rate. May be a negative value.
190	OfferSpotRate	Price	Offer F/X spot rate.
191	OfferForwardPoints	PriceOffset	Offer F/X forward points added to spot rate. May be a negative value.
192	OrderQty2	Qty	OrderQty of the future part of a F/X swap order.
193	FutSettDate2	LocalMktDate	FutSettDate of the future part of a F/X swap order.
194	LastSpotRate	Price	F/X spot rate.
195	LastForwardPoints	PriceOffset	F/X forward points added to LastSpotRate. May be a negative value.
196	AllocLinkID	String	Can be used to link two different Allocation messages (each with unique AllocID) together, i.e. for F/X "Netting" or "Swaps". Should be unique.
197	AllocLinkType	int	Identifies the type of Allocation linkage when AllocLinkID is used. Valid values: 0 = F/X Netting 1 = F/X Swap
198	SecondaryOrderID	String	Assigned by the party which accepts the order. Can be used to provide the OrderID used by an exchange or executing system.
199	NoIOIQualifiers	int	Number of repeating groups of IOIQualifiers.
200	MaturityMonthYear	month-year	Month and Year of the maturity for SecurityType=FUT or SecurityType=OPT. Required if MaturityDay is specified. Format: YYYYMM (i.e. 199903)
201	PutOrCall	int	Indicates whether an Option is for a put or call. Valid values: 0 = Put 1 = Call
202	StrikePrice	Price	Strike Price for an Option.
203	CoveredOrUncovered	int	Used for options Valid values: 0 = Covered 1 = Uncovered

204	CustomerOrFirm	int	<p>Used for options when delivering the order to an execution system/exchange to specify if the order is for a customer or the firm placing the order itself.</p> <p>Valid values: 0 = Customer 1 = Firm</p>
205	MaturityDay	day-of-month	<p>Day of month used in conjunction with MaturityMonthYear to specify the maturity date for SecurityType=FUT or SecurityType=OPT.</p> <p>Valid values: 1-31</p>
206	OptAttribute	char	<p>Can be used for SecurityType=OPT to identify a particular security.</p> <p>Valid values vary by SecurityExchange:</p> <p>For Exchange: MONEP (Paris) L = Long (a.k.a. "American") S = Short (a.k.a. "European")</p> <p>For Exchanges: DTB (Frankfurt), HKSE (Hong Kong), and SOFFEX (Zurich) 0-9 = single digit "version" number assigned by exchange following capital adjustments (0=current, 1=prior, 2=prior to 1, etc).</p>
207	SecurityExchange	Exchange	<p>Market used to help identify a security.</p> <p>Valid values: <u>See Appendix C</u></p>
208	NotifyBrokerOfCredit	Boolean	<p>Indicates whether or not details should be communicated to BrokerOfCredit (i.e. step-in broker).</p> <p>Valid values: Y = Details should be communicated N = Details should not be communicated</p>
209	AllocHandlInst	int	<p>Indicates how the receiver (i.e. third party) of Allocation message should handle/process the account details.</p> <p>Valid values: 1 = Match 2 = Forward 3 = Forward and Match</p>
210	MaxShow	Qty	<p>Maximum number of shares within an order to be shown to other customers (i.e. sent via an IOI).</p> <p>(Prior to FIX 4.2 this field was of type int)</p>

211	PegDifference	PriceOffset	Amount (signed) added to the price of the peg for a pegged order.
212	XmlDataLen	int	Length of the XmlData data block.
213	XmlData	data	Actual XML data stream (e.g. FIXML). See appropriate XML reference (e.g. FIXML). Note: may contain embedded SOH characters.
214	SettlInstRefID	String	Reference identifier for the SettlInstID with Cancel and Replace SettlInstTransType transaction types.
215	NoRoutingIDs	int	Number of repeating groups of RoutingID and RoutingType values. <u>See Appendix L – Pre-Trade Message Targeting/Routing</u>
216	RoutingType	int	Indicates the type of RoutingID specified. Valid values: 1 = Target Firm 2 = Target List 3 = Block Firm 4 = Block List
217	RoutingID	String	Assigned value used to identify a specific routing destination.
218	SpreadToBenchmark	PriceOffset	For Fixed Income. Basis points relative to a benchmark. To be expressed as "count of basis points" (vs. an absolute value). E.g. High Grade Corporate Bonds may express price as basis points relative to benchmark (the Benchmark field). Note: Basis points can be negative.
219	Benchmark	char	For Fixed Income. Identifies the benchmark (e.g. used in conjunction with the SpreadToBenchmark field). Valid values: 1 = CURVE 2 = 5-YR 3 = OLD-5 4 = 10-YR 5 = OLD-10 6 = 30-YR 7 = OLD-30 8 = 3-MO-LIBOR 9 = 6-MO-LIBOR
220-222	Reserved/Allocated to the Fixed Income proposal		
223	CouponRate	float	For Fixed Income. Coupon rate of the bond. Will be zero for step-up bonds.

224-230	Reserved/Allocated to the Fixed Income proposal		
231	ContractMultiplier	float	Specifies the ratio or multiply factor to convert from contracts to shares (e.g. 1.0, 100, 1000, etc). Applicable For Fixed Income, Convertible Bonds, Derivatives, etc. Note: If used, quantities should be expressed in the "nominal" (e.g. contracts vs. shares) amount.
232-261	Reserved/Allocated to the Fixed Income proposal		
262	MDReqID	String	Unique identifier for Market Data Request
263	SubscriptionRequestType	char	Subscription Request Type Valid values: 0 = Snapshot 1 = Snapshot + Updates (Subscribe) 2 = Disable previous Snapshot + Update Request (Unsubscribe)
264	MarketDepth	int	Depth of market for Book Snapshot Valid values: 0 = Full Book 1 = Top of Book N>1 = Report best N price tiers of data
265	MDUpdateType	int	Specifies the type of Market Data update. Valid values: 0 = Full Refresh 1 = Incremental Refresh
266	AggregatedBook	Boolean	Specifies whether or not book entries should be aggregated. Valid values: Y = one book entry per side per price N = Multiple entries per side per price allowed (Not specified) = broker option
267	NoMDEntryTypes	int	Number of MDEntryType fields requested.
268	NoMDEntries	int	Number of entries in Market Data message.

269	MDEntryType	char	Type Market Data entry. Valid values: 0 = Bid 1 = Offer 2 = Trade 3 = Index Value 4 = Opening Price 5 = Closing Price 6 = Settlement Price 7 = Trading Session High Price 8 = Trading Session Low Price 9 = Trading Session VWAP Price
270	MDEntryPx	Price	Price of the Market Data Entry.
271	MDEntrySize	Qty	Number of shares represented by the Market Data Entry.
272	MDEntryDate	UTCDate	Date of Market Data Entry.
273	MDEntryTime	UTCTimeOnly	Time of Market Data Entry.
274	TickDirection	char	Direction of the "tick". Valid values: 0 = Plus Tick 1 = Zero-Plus Tick 2 = Minus Tick 3 = Zero-Minus Tick
275	MDMkt	Exchange	Market posting quote / trade. Valid values: <u>See Appendix C</u>
276	QuoteCondition	Multiple ValueString	Space-delimited list of conditions describing a quote. Valid values: A = Open / Active B = Closed / Inactive C = Exchange Best D = Consolidated Best E = Locked F = Crossed G = Depth H = Fast Trading I = Non-Firm

277	TradeCondition	Multiple ValueString	Space-delimited list of conditions describing a trade Valid values: A = Cash (only) Market B = Average Price Trade C = Cash Trade (same day clearing) D = Next Day (only) Market E = Opening / Reopening Trade Detail F = Intraday Trade Detail G = Rule 127 Trade (NYSE) H = Rule 155 Trade (Amex) I = Sold Last (late reporting) J = Next Day Trade (next day clearing) K = Opened (late report of opened trade) L = Seller M = Sold (out of sequence) N = Stopped Stock (guarantee of price but does not execute the order)
278	MDEntryID	String	Unique Market Data Entry identifier.
279	MDUpdateAction	char	Type of Market Data update action. Valid values: 0 = New 1 = Change 2 = Delete
280	MDEntryRefID	String	Refers to a previous MDEntryID.
281	MDReqRejReason	char	Reason for the rejection of a Market Data request. Valid values: 0 = Unknown symbol 1 = Duplicate MDReqID 2 = Insufficient Bandwidth 3 = Insufficient Permissions 4 = Unsupported SubscriptionRequestType 5 = Unsupported MarketDepth 6 = Unsupported MDUpdateType 7 = Unsupported AggregatedBook 8 = Unsupported MDEntryType
282	MDEntryOriginator	String	Originator of a Market Data Entry
283	LocationID	String	Identification of a Market Maker's location
284	DeskID	String	Identification of a Market Maker's desk
285	DeleteReason	char	Reason for deletion. Valid values: 0 = Cancellation / Trade Bust 1 = Error

286	OpenCloseSettleFlag	char	Flag that identifies a price. Valid values: 0 = Daily Open / Close / Settlement price 1 = Session Open / Close / Settlement price 2 = Delivery Settlement price
287	SellerDays	int	Specifies the number of days that may elapse before delivery of the security
288	MDEntryBuyer	String	Buying party in a trade
289	MDEntrySeller	String	Selling party in a trade
290	MDEntryPositionNo	int	Display position of a bid or offer, numbered from most competitive to least competitive, per market side, beginning with 1.
291	FinancialStatus	char	Identifies a firm's financial status. Valid values: 1 = Bankrupt
292	CorporateAction	char	Identifies the type of Corporate Action. Valid values: A = Ex-Dividend B = Ex-Distribution C = Ex-Rights D = New E = Ex-Interest
293	DefBidSize	Qty	Default Bid Size.
294	DefOfferSize	Qty	Default Offer Size.
295	NoQuoteEntries	int	The number of quote entries for a QuoteSet.
296	NoQuoteSets	int	The number of sets of quotes in the message.
297	QuoteAckStatus	int	Identifies the status of the quote acknowledgement. Valid values: 0-Accepted 1-Canceled for Symbol(s) 2-Canceled for Security Type(s) 3-Canceled for Underlying 4-Canceled All 5-Rejected
298	QuoteCancelType	int	Identifies the type of quote cancel. Valid Values: 1 – Cancel for Symbol(s) 2 – Cancel for Security Type(s) 3 – Cancel for Underlying Symbol 4 – Cancel All Quotes
299	QuoteEntryID	String	Uniquely identifies the quote as part of a QuoteSet.

300	QuoteRejectReason	int	Reason Quote was rejected: Valid Values: 1 = Unknown symbol (Security) 2 = Exchange(Security) closed 3 = Quote Request exceeds limit 4 = Too late to enter 5 = Unknown Quote 6 = Duplicate Quote 7 = Invalid bid/ask spread 8 = Invalid price 9 = Not authorized to quote security
301	QuoteResponseLevel	int	Level of Response requested from receiver of quote messages. Valid Values: 0 – No Acknowledgement (Default) 1 – Acknowledge only negative or erroneous quotes 2 – Acknowledge each quote messages
302	QuoteSetID	String	Unique id for the Quote Set.
303	QuoteRequestType	int	Indicates the type of Quote Request being generated Valid values: 1-Manual 2-Automatic
304	TotQuoteEntries	int	Total number of quotes for the quote set across all messages. Should be the sum of all NoQuoteEntries in each message that has repeating quotes that are part of the same quote set.
305	UnderlyingIDSource	String	Underlying security's IDSource. Valid values: see IDSource field
306	UnderlyingIssuer	String	Underlying security's Issuer. See Issuer field for description
307	UnderlyingSecurityDesc	String	Underlying security's SecurityDesc. See SecurityDesc field for description
308	UnderlyingSecurityExchange	Exchange	Underlying security's SecurityExchange. Can be used to identify the underlying security. Valid values: see SecurityExchange
309	UnderlyingSecurityID	String	Underlying security's SecurityID. See SecurityID field for description
310	UnderlyingSecurityType	String	Underlying security's SecurityType. Valid values: see SecurityType field
311	UnderlyingSymbol	String	Underlying security's Symbol. See Symbol field for description

312	UnderlyingSymbolSfx	String	Underlying security's SymbolSfx. See SymbolSfx field for description
313	UnderlyingMaturityMonthYear	month-year	Underlying security's MaturityMonthYear. Required if UnderlyingMaturityDay is specified. See MaturityMonthYear field for description
314	UnderlyingMaturityDay	day-of-month	Underlying security's MaturityDay. See MaturityDay field for description
315	UnderlyingPutOrCall	int	Underlying security's PutOrCall. See PutOrCall field for description
316	UnderlyingStrikePrice	Price	Underlying security's StrikePrice. See StrikePrice field for description
317	UnderlyingOptAttribute	char	Underlying security's OptAttribute. See OptAttribute field for description
318	Underlying Currency	Currency	Underlying security's Currency. See Currency field for description and valid values
319	RatioQty	Quantity	Quantity of a particular leg in the security.
320	SecurityReqID	String	Unique ID of a Security Definition Request.
321	SecurityRequestType	int	Type of Security Definition Request. Valid values: 0 = Request Security identity and specifications 1 = Request Security identity for the specifications provided (Name of the security is not supplied) 2 = Request List Security Types 3 = Request List Securities (Can be qualified with Symbol, SecurityType, TradingSessionID, SecurityExchange is provided then only list Securities for the specific type)
322	SecurityResponseID	String	Unique ID of a Security Definition message.
323	SecurityResponseType	int	Type of Security Definition message response. Valid values: 1 = Accept security proposal as is 2 = Accept security proposal with revisions as indicated in the message 3 = List of security types returned per request 4 = List of securities returned per request 5 = Reject security proposal 6 = Can not match selection criteria
324	SecurityStatusReqID	String	Unique ID of a Security Status Request message.

325	UnsolicitedIndicator	Boolean	<p>Indicates whether or not message is being sent as a result of a subscription request or not.</p> <p>Valid values: Y = Message is being sent unsolicited N = Message is being sent as a result of a prior request</p>
326	SecurityTradingStatus	int	<p>Identifies the trading status applicable to the transaction.</p> <p>Valid values: 1 = Opening Delay 2 = Trading Halt 3 = Resume 4 = No Open/No Resume 5 = Price Indication 6 = Trading Range Indication 7 = Market Imbalance Buy 8 = Market Imbalance Sell 9 = Market On Close Imbalance Buy 10 = Market On Close Imbalance Sell 11 = (not assigned) 12 = No Market Imbalance 13 = No Market On Close Imbalance 14 = ITS Pre-Opening 15 = New Price Indication 16 = Trade Dissemination Time 17 = Ready to trade (start of session) 18 = Not Available for trading (end of session) 19 = Not Traded on this Market 20 = Unknown or Invalid</p>
327	HaltReason	char	<p>Denotes the reason for the Opening Delay or Trading Halt.</p> <p>Valid values: I = Order Imbalance X = Equipment Changeover P = News Pending D = News Dissemination E = Order Influx M = Additional Information</p>
328	InViewOfCommon	Boolean	<p>Indicates whether or not the halt was due to Common Stock trading being halted.</p> <p>Valid values: Y = Halt was due to common stock being halted N = Halt was not related to a halt of the common stock</p>

329	DueToRelated	Boolean	Indicates whether or not the halt was due to the Related Security being halted. Valid values: Y = Halt was due to related security being halted N = Halt was not related to a halt of the related security
330	BuyVolume	Qty	Number of shares bought.
331	SellVolume	Qty	Number of shares sold.
332	HighPx	Price	Represents an indication of the high end of the price range for a security prior to the open or reopen
333	LowPx	Price	Represents an indication of the low end of the price range for a security prior to the open or reopen
334	Adjustment	int	Identifies the type of adjustment. Valid values: 1 = Cancel 2 = Error 3 = Correction
335	TradSesReqID	String	Unique ID of a Trading Session Status message.
336	TradingSessionID	String	Identifier for Trading Session Can be used to represent a specific market trading session (e.g. "PRE-OPEN", "CROSS_2", "AFTER-HOURS", "TOSTNET1", "TOSTNET2", etc). Values should be bi-laterally agreed to between counterparties.
337	ContraTrader	String	Identifies the trader (e.g. "badge number") of the ContraBroker.
338	TradSesMethod	int	Method of trading Valid values: 1 = Electronic 2 = Open Outcry 3 = Two Party
339	TradSesMode	int	Trading Session Mode Valid values: 1 = Testing 2 = Simulated 3 = Production

340	TradSesStatus	int	State of the trading session. Valid values: 1 = Halted 2 = Open 3 = Closed 4 = Pre-Open 5 = Pre-Close
341	TradSesStartTime	UTCTim estamp	Starting time of the trading session
342	TradSesOpenTime	UTCTim estamp	Time of the opening of the trading session
343	TradSesPreCloseTime	UTCTim estamp	Time of the pre-closed of the trading session
344	TradSesCloseTime	UTCTim estamp	Closing time of the trading session
345	TradSesEndTime	UTCTim estamp	End time of the trading session
346	NumberOfOrders	int	Number of orders in the market.
347	MessageEncoding	String	Type of message encoding (non-ASCII (non-English) characters) used in a message's "Encoded" fields. Valid values: ISO-2022-JP (for using JIS) EUC-JP (for using EUC) Shift_JIS (for using SJIS) UTF-8 (for using Unicode)
348	EncodedIssuerLen	int	Byte length of encoded (non-ASCII characters) EncodedIssuer field.
349	EncodedIssuer	data	Encoded (non-ASCII characters) representation of the Issuer field in the encoded format specified via the MessageEncoding field. If used, the ASCII (English) representation should also be specified in the Issuer field.
350	EncodedSecurityDesc Len	int	Byte length of encoded (non-ASCII characters) EncodedSecurityDesc field.
351	EncodedSecurityDesc	data	Encoded (non-ASCII characters) representation of the SecurityDesc field in the encoded format specified via the MessageEncoding field. If used, the ASCII (English) representation should also be specified in the SecurityDesc field.
352	EncodedListExecInstLen	int	Byte length of encoded (non-ASCII characters) EncodedListExecInst field.

353	EncodedListExecInst	data	Encoded (non-ASCII characters) representation of the ListExecInst field in the encoded format specified via the MessageEncoding field. If used, the ASCII (English) representation should also be specified in the ListExecInst field.
354	EncodedTextLen	int	Byte length of encoded (non-ASCII characters) EncodedText field.
355	EncodedText	data	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field. If used, the ASCII (English) representation should also be specified in the Text field.
356	EncodedSubjectLen	int	Byte length of encoded (non-ASCII characters) EncodedSubject field.
357	EncodedSubject	data	Encoded (non-ASCII characters) representation of the Subject field in the encoded format specified via the MessageEncoding field. If used, the ASCII (English) representation should also be specified in the Subject field.
358	EncodedHeadlineLen	int	Byte length of encoded (non-ASCII characters) EncodedHeadline field.
359	EncodedHeadline	data	Encoded (non-ASCII characters) representation of the Headline field in the encoded format specified via the MessageEncoding field. If used, the ASCII (English) representation should also be specified in the Headline field.
360	EncodedAllocTextLen	int	Byte length of encoded (non-ASCII characters) EncodedAllocText field.
361	EncodedAllocText	data	Encoded (non-ASCII characters) representation of the AllocText field in the encoded format specified via the MessageEncoding field. If used, the ASCII (English) representation should also be specified in the AllocText field.
362	EncodedUnderlyingIssuerLen	int	Byte length of encoded (non-ASCII characters) EncodedUnderlyingIssuer field.
363	EncodedUnderlyingIssuer	data	Encoded (non-ASCII characters) representation of the UnderlyingIssuer field in the encoded format specified via the MessageEncoding field. If used, the ASCII (English) representation should also be specified in the UnderlyingIssuer field.
364	EncodedUnderlyingSecurityDescLen	int	Byte length of encoded (non-ASCII characters) EncodedUnderlyingSecurityDesc field.

365	EncodedUnderlyingSecurityDesc	data	Encoded (non-ASCII characters) representation of the UnderlyingSecurityDesc field in the encoded format specified via the MessageEncoding field. If used, the ASCII (English) representation should also be specified in the UnderlyingSecurityDesc field.
366	AllocPrice	Price	Executed price for an AllocAccount entry used when using “executed price” vs. “average price” allocations (e.g. Japan).
367	QuoteSetValidUntilTime	UTCTimestamp	Indicates expiration time of this particular QuoteSet (always expressed in UTC (Universal Time Coordinated, also known as “GMT”))
368	QuoteEntryRejectReason	int	Reason Quote Entry was rejected: Valid values: 1 = Unknown symbol (Security) 2 = Exchange(Security) closed 3 = Quote exceeds limit 4 = Too late to enter 5 = Unknown Quote 6 = Duplicate Quote 7 = Invalid bid/ask spread 8 = Invalid price 9 = Not authorized to quote security
369	LastMsgSeqNumProcessed	int	The last MsgSeqNum value received and processed. Can be specified on every message sent. Useful for detecting a backlog with a counterparty.
370	OnBehalfOfSendingTime	UTCTimestamp	Used when a message is sent via a “hub” or “service bureau”. If A sends to Q (the hub) who then sends to B via a separate FIX session, then when Q sends to B the value of this field should represent the SendingTime on the message A sent to Q. (always expressed in UTC (Universal Time Coordinated, also known as “GMT”))
371	RefTagID	int	The tag number of the FIX field being referenced.
372	RefMsgType	String	The MsgType of the FIX message being referenced.

373	SessionRejectReason	int	<p>Code to identify reason for a session-level Reject message.</p> <p>Valid values:</p> <ul style="list-style-type: none"> 0 = Invalid tag number 1 = Required tag missing 2 = Tag not defined for this message type 3 = Undefined Tag 4 = Tag specified without a value 5 = Value is incorrect (out of range) for this tag 6 = Incorrect data format for value 7 = Decryption problem 8 = Signature problem 9 = CompID problem 10 = SendingTime accuracy problem 11 = Invalid MsgType
374	BidRequestTransType	char	<p>Identifies the Bid Request message type.</p> <p>Valid values:</p> <ul style="list-style-type: none"> N = New C = Cancel
375	ContraBroker	String	Identifies contra broker. Standard NASD market-maker mnemonic is preferred.
376	ComplianceID	String	ID used to represent this transaction for compliance purposes (e.g. OATS reporting).
377	SolicitedFlag	Boolean	<p>Indicates whether or not the order was solicited.</p> <p>Valid values:</p> <ul style="list-style-type: none"> Y = Was solicited N = Was not solicited
378	ExecRestatementReason	int	<p>Code to identify reason for an ExecutionRpt message sent with ExecType=Restated or used when communicating an unsolicited cancel.</p> <p>Valid values:</p> <ul style="list-style-type: none"> 0 = GT Corporate action 1 = GT renewal / restatement (no corporate action) 2 = Verbal change 3 = Repricing of order 4 = Broker option 5 = Partial decline of OrderQty (e.g. exchange-initiated partial cancel)
379	BusinessRejectRefID	String	The value of the business-level "ID" field on the message being referenced.

380	BusinessRejectReason	int	Code to identify reason for a Business Message Reject message. Valid values: 0 = Other 1 = Unkown ID 2 = Unknown Security 3 = Unsupported Message Type 4 = Application not available 5 = Conditionally Required Field Missing
381	GrossTradeAmt	Amt	Total amount traded (e.g. CumQty * AvgPx) expressed in units of currency.
382	NoContraBrokers	int	The number of ContraBroker entries.
383	MaxMessageSize	int	Maximum number of bytes supported for a single message.
384	NoMsgTypes	int	Number of MsgTypes in repeating group.
385	MsgDirection	char	Specifies the direction of the message. Valid values: S = Send R = Receive
386	NoTradingSessions	int	Number of TradingSessionIDs in repeating group.
387	TotalVolumeTraded	Qty	Total volume (quantity) traded.
388	DiscretionInst	char	Code to identify the price a DiscretionOffset is related to and should be mathematically added to. Valid values: 0 = Related to displayed price 1 = Related to market price 2 = Related to primary price 3 = Related to local primary price 4 = Related to midpoint price 5 = Related to last trade price
389	DiscretionOffset	PriceOffset	Amount (signed) added to the “related to” price specified via DiscretionInst.
390	BidID	String	Unique identifier for Bid Response as assigned by broker. Uniqueness must be guaranteed within a single trading day.
391	ClientBidID	String	Unique identifier for a Bid Request as assigned by institution. Uniqueness must be guaranteed within a single trading day.
392	ListName	String	Descriptive name for list order.
393	TotalNumSecurities	int	Total number of securities.

394	BidType	int	Code to identify the type of Bid Request. Valid values: 1 – “Non Disclosed” Style (e.g. US/European) 2 – “Disclosed” Style (e.g. Japanese) 3 – No Bidding Process
395	NumTickets	int	Total number of tickets.
396	SideValue1	Amt	Amounts in currency
397	SideValue2	Amt	Amounts in currency
398	NoBidDescriptors	int	Number of BidDescriptor entries.
399	BidDescriptorType	int	Code to identify the type of BidDescriptor. Valid values: 1 – Sector 2 – Country 3 – Index
400	BidDescriptor	String	BidDescriptor value. Usage depends upon BidDescriptorType. If BidDescriptorType =1 Industrials etc – Free text If BidDescriptorType =2 "FR" etc – ISO Country Codes If BidDescriptorType =3 FT100, FT250, STOXX – Free text
401	SideValueInd	int	Code to identify which "SideValue" the value refers to. SideValue1 and SideValue2 are used as opposed to Buy or Sell so that the basket can be quoted either way as Buy or Sell. Valid values: 1 – SideValue1 2 – SideValue 2
402	LiquidityPctLow	float	Liquidity indicator or lower limit if TotalNumSecurities > 1. Represented as a percentage.
403	LiquidityPctHigh	float	Upper liquidity indicator if TotalNumSecurities > 1. Represented as a percentage.
404	LiquidityValue	Amt	Value between LiquidityPctLow and LiquidityPctHigh in Currency
405	EFPTrackingError	float	Eg Used in EFP trades 12% (EFP – Exchange for Physical). Represented as a percentage.
406	FairValue	Amt	Used in EFP trades
407	OutsideIndexPct	float	Used in EFP trades. Represented as a percentage.
408	ValueOfFutures	Amt	Used in EFP trades

409	LiquidityIndType	int	Code to identify the type of liquidity indicator. Valid values: 1 – 5day moving average 2 – 20 day moving average 3 – Normal Market Size 4 – Other
410	WtAverageLiquidity	float	Overall weighted average liquidity expressed as a % of average daily volume. Represented as a percentage.
411	ExchangeForPhysical	Boolean	Indicates whether or not to exchange for physical. Valid values: Y = True N = False
412	OutMainCntryUIndex	Amt	Value of stocks in Currency
413	CrossPercent	float	Percentage of program that crosses in Currency. Represented as a percentage.
414	ProgRptReqs	int	Code to identify the desired frequency of progress reports. Valid values: 1 – BuySide explicitly requests status using StatusRequest (Default) The sell-side firm can however, send a DONE status List Status Response in an unsolicited fashion 2 – SellSide periodically sends status using ListStatus. Period optionally specified in ProgressPeriod 3 – Real-time execution reports (to be discouraged)
415	ProgPeriodInterval	int	Time in minutes between each ListStatus report sent by SellSide. Zero means don't send status.
416	IncTaxInd	int	Code to represent whether value is net (inclusive of tax) or gross. Valid values: 1 - Net 2 - Gross
417	NumBidders	int	Indicates the total number of bidders on the list
418	TradeType	char	Code to represent the type of trade. Valid values: R: Risk Trade G: VWAP Guarantee A: Agency J: Guaranteed Close

419	BasisPxType	char	Code to represent the basis price type. Valid values: 2: Closing Price at morning session 3: Closing Price 4: Current price 5: SQ 6: VWAP through a day 7: VWAP through a morning session 8: VWAP through an afternoon session 9: VWAP through a day except <i>YORI</i> A: VWAP through a morning session except <i>YORI</i> B: VWAP through an afternoon session except <i>YORI</i> C: Strike D: Open Z: Others
420	NoBidComponents	int	Indicates the number of list entries.
421	Country	String	ISO Country Code in field
422	TotNoStrikes	int	Total number of strike price entries across all messages. Should be the sum of all NoStrikes in each message that has repeating strike price entries related to the same ListID. Used to support fragmentation.
423	PriceType	int	Code to represent the price type. Valid values: 1 - Percentage 2 - per share (e.g. cents per share) 3 - Fixed Amount (absolute value)
424	DayOrderQty	Qty	For GT orders, the OrderQty less all shares (adjusted for stock splits) that traded on previous days. $\text{DayOrderQty} = \text{OrderQty} - (\text{CumQty} - \text{DayCumQty})$
425	DayCumQty	Qty	The number of shares on a GT order that have traded today.
426	DayAvgPx	Price	The average price of shares on a GT order that have traded today.
427	GTBookingInst	int	Code to identify whether to book out executions on a part-filled GT order on the day of execution or to accumulate. Valid values: 0 = book out all trades on day of execution 1 = accumulate executions until order is filled or expires 2 = accumulate until verbally notified otherwise
428	NoStrikes	int	Number of list strike price entries.

429	ListStatusType	int	Code to represent the price type. Valid values: 1 – Ack 2 – Response 3 – Timed 4 – ExecStarted 5 – AllDone 6 – Alert
430	NetGrossInd	int	Code to represent whether value is net (inclusive of tax) or gross. Valid values: 1 - Net 2 - Gross
431	ListOrderStatus	int	Code to represent the status of a list order. Valid values: 1 – InBiddingProcess 2 – ReceivedForExecution 3 – Executing 4 – Canceling 5 – Alert 6 – All Done 7 – Reject
432	ExpireDate	LocalMktDate	Date of order expiration (last day the order can trade), always expressed in terms of the local market date. The time at which the order expires is determined by the local market's business practices
433	ListExecInstType	char	Identifies the type of ListExecInst. Valid values: 1 - Immediate 2 - Wait for Execute Instruction (e.g. a List Execute message or phone call before proceeding with execution of the list)
434	CxlRejResponseTo	char	Identifies the type of request that a Cancel Reject is in response to. Valid values: 1 - Order Cancel Request 2 - Order Cancel/Replace Request
435	UnderlyingCouponRate	float	Underlying security's CouponRate. See CouponRate field for description
436	UnderlyingContractMultiplier	float	Underlying security's ContractMultiplier. See ContractMultiplier field for description
437	ContraTradeQty	Qty	Quantity traded with the ContraBroker.

438	ContraTradeTime	UTCTimestamp	Identifies the time of the trade with the ContraBroker. (always expressed in UTC (Universal Time Coordinated, also known as “GMT”))
439	ClearingFirm	String	Firm that will clear the trade. Used if different from the executing firm.
440	ClearingAccount	String	Supplemental accounting information forwarded to clearing house/firm.
441	LiquidityNumSecurities	int	Number of Securities between LiquidityPctLow and LiquidityPctHigh in Currency.
442	MultiLegReportingType	char	Used to indicate what an Execution Report represents (e.g. used with multi-leg securities, such as option strategies, spreads, etc.). Valid Values: 1 - Single Security (default if not specified) 2 - Individual leg of a multi-leg security 3 - Multi-leg security
443	StrikeTime	UTCTimestamp	The time at which current market prices are used to determine the value of a basket.
444	ListStatusText	String	Free format text string related to List Status.
445	EncodedListStatusTextLen	int	Byte length of encoded (non-ASCII characters) EncodedListStatusText field.
446	EncodedListStatusText	data	Encoded (non-ASCII characters) representation of the ListStatusText field in the encoded format specified via the MessageEncoding field. If used, the ASCII (English) representation should also be specified in the ListStatusText field.

FIX Field Index sorted by tag number:

1	Account
2	AdvId
3	AdvRefID
4	AdvSide
5	AdvTransType
6	AvgPx
7	BeginSeqNo
8	BeginString
9	BodyLength
10	Checksum
11	ClOrdID
12	Commission
13	CommType
14	CumQty
15	Currency
16	EndSeqNo
17	ExecID
18	ExecInst
19	ExecRefID
20	ExecTransType
21	HandlInst
22	IDSSource
23	IOId
24	IOIOthSvc (no longer used)
25	IOIQltyInd
26	IOIRefID
27	IOIShares
28	IOITransType
29	LastCapacity
30	LastMkt
31	LastPx
32	LastShares
33	LinesOfText
34	MsgSeqNum
35	MsgType

36	NewSeqNo
37	OrderID
38	OrderQty
39	OrdStatus
40	OrdType
41	OrigClOrdID
42	OrigTime
43	PossDupFlag
44	Price
45	RefSeqNum
46	RelatdSym
47	Rule80A (aka OrderCapacity)
48	SecurityID
49	SenderCompID
50	SenderSubID
51	SendingDate (no longer used)
52	SendingTime
53	Shares
54	Side
55	Symbol
56	TargetCompID
57	TargetSubID
58	Text
59	TimeInForce
60	TransactTime
61	Urgency
62	ValidUntilTime
63	SettlmntTyp
64	FutSettDate
65	SymbolSfx
66	ListID
67	ListSeqNo
68	TotNoOrders (formerly ListNoOrds) named:

69	ListExecInst
70	AllocID
71	AllocTransType
72	RefAllocID
73	NoOrders
74	AvgPrxPrecision
75	TradeDate
76	ExecBroker
77	OpenClose
78	NoAllocs
79	AllocAccount
80	AllocShares
81	ProcessCode
82	NoRpts
83	RptSeq
84	CxlQty
85	NoDlvlyInst (no longer used)
86	DlvlyInst (no longer used)
87	AllocStatus
88	AllocRejCode
89	Signature
90	SecureDataLen
91	SecureData
92	BrokerOfCredit
93	SignatureLength
94	EmailType
95	RawDataLength
96	RawData
97	PossResend
98	EncryptMethod
99	StopPx
100	ExDestination
101	(Not Defined)
102	CxlRejReason

103	OrdRejReason
104	IOIQualifier
105	WaveNo
106	Issuer
107	SecurityDesc
108	HeartBtInt
109	ClientID
110	MinQty
111	MaxFloor
112	TestReqID
113	ReportToExch
114	LocateReqd
115	OnBehalfOfComplID
116	OnBehalfOfSubID
117	QuoteID
118	NetMoney
119	SettlCurrAmt
120	SettlCurrency
121	ForexReq
122	OrigSendingTime
123	GapFillFlag
124	NoExecs
125	CxlType (no longer used)
126	ExpireTime
127	DKReason
128	DeliverToComplID
129	DeliverToSubID
130	IOINaturalFlag
131	QuoteReqID
132	BidPx
133	OfferPx
134	BidSize
135	OfferSize
136	NoMiscFees
137	MiscFeeAmt
138	MiscFeeCurr
139	MiscFeeType
140	PrevClosePx

141	ResetSeqNumFlag
142	SenderLocationID
143	TargetLocationID
144	OnBehalfOfLocationID
145	DeliverToLocationID
146	NoRelatedSym
147	Subject
148	Headline
149	URLLink
150	ExecType
151	LeavesQty
152	CashOrderQty
153	AllocAvgPx
154	AllocNetMoney
155	SettlCurrFxRate
156	SettlCurrFxRateCalc
157	NumDaysInterest
158	AccruedInterestRate
159	AccruedInterestAmt
160	SettlInstMode
161	AllocText
162	SettlInstID
163	SettlInstTransType
164	EmailThreadID
165	SettlInstSource
166	SettlLocation
167	SecurityType
168	EffectiveTime
169	StandInstDbType
170	StandInstDbName
171	StandInstDbID
172	SettlDeliveryType
173	SettlDepositoryCode
174	SettlBrkrCode
175	SettlInstCode
176	SecuritySettlAgentName
177	SecuritySettlAgentCode

178	SecuritySettlAgentAcctNum
179	SecuritySettlAgentAcctName
180	SecuritySettlAgentContactName
181	SecuritySettlAgentContactPhone
182	CashSettlAgentName
183	CashSettlAgentCode
184	CashSettlAgentAcctNum
185	CashSettlAgentAcctName
186	CashSettlAgentContactName
187	CashSettlAgentContactPhone
188	BidSpotRate
189	BidForwardPoints
190	OfferSpotRate
191	OfferForwardPoints
192	OrderQty2
193	FutSettDate2
194	LastSpotRate
195	LastForwardPoints
196	AllocLinkID
197	AllocLinkType
198	SecondaryOrderID
199	NoIOIQualifiers
200	MaturityMonthYear
201	PutOrCall
202	StrikePrice
203	CoveredOrUncovered
204	CustomerOrFirm
205	MaturityDay
206	OptAttribute
207	SecurityExchange
208	NotifyBrokerOfCredit
209	AllocHandlInst
210	MaxShow
211	PegDifference
212	XmlDataLen
213	XmlData

214	SettlInstRefID
215	NoRoutingIDs
216	RoutingType
217	RoutingID
218	SpreadToBenchmark
219	Benchmark
220 - 222	Reserved/Allocated to the Fixed Income proposal
223	CouponRate
224 - 230	Reserved/Allocated to the Fixed Income proposal
231	ContractMultiplier
232 - 261	Reserved/Allocated to the Fixed Income proposal
262	MDReqID
263	SubscriptionRequestType
264	MarketDepth
265	MDUpdateType
266	AggregatedBook
267	NoMDEntryTypes
268	NoMDEntries
269	MDEntryType
270	MDEntryPx
271	MDEntrySize
272	MDEntryDate
273	MDEntryTime
274	TickDirection
275	MDMkt
276	QuoteCondition
277	TradeCondition
278	MDEntryID
279	MDUpdateAction
280	MDEntryRefID
281	MDReqRejReason
282	MDEntryOriginator
283	LocationID
284	DeskID
285	DeleteReason
286	OpenCloseSettleFlag

287	SellerDays
288	MDEntryBuyer
289	MDEntrySeller
290	MDEntryPositionNo
291	FinancialStatus
292	CorporateAction
293	DefBidSize
294	DefOfferSize
295	NoQuoteEntries
296	NoQuoteSets
297	QuoteAckStatus
298	QuoteCancelType
299	QuoteEntryID
300	QuoteRejectReason
301	QuoteResponseLevel
302	QuoteSetID
303	QuoteRequestType
304	TotQuoteEntries
305	UnderlyingIDSource
306	UnderlyingIssuer
307	UnderlyingSecurityDesc
308	UnderlyingSecurityExchange
309	UnderlyingSecurityID
310	UnderlyingSecurityType
311	UnderlyingSymbol
312	UnderlyingSymbolSfx
313	UnderlyingMaturityMonthYear
314	UnderlyingMaturityDay
315	UnderlyingPutOrCall
316	UnderlyingStrikePrice
317	UnderlyingOptAttribute
318	UnderlyingCurrency
319	RatioQty
320	SecurityReqID
321	SecurityRequestType
322	SecurityResponseID
323	SecurityResponseType
324	SecurityStatusReqID

325	UnsolicitedIndicator
326	SecurityTradingStatus
327	HaltReason
328	InViewOfCommon
329	DueToRelated
330	BuyVolume
331	SellVolume
332	HighPx
333	LowPx
334	Adjustment
335	TradSesReqID
336	TradingSessionID
337	ContraTrader
338	TradSesMethod
339	TradSesMode
340	TradSesStatus
341	TradSesStartTime
342	TradSesOpenTime
343	TradSesPreCloseTime
344	TradSesCloseTime
345	TradSesEndTime
346	NumberOfOrders
347	MessageEncoding
348	EncodedIssuerLen
349	EncodedIssuer
350	EncodedSecurityDescLen
351	EncodedSecurityDesc
352	EncodedListExecInstLen
353	EncodedListExecInst
354	EncodedTextLen
355	EncodedText
356	EncodedSubjectLen
357	EncodedSubject
358	EncodedHeadlineLen
359	EncodedHeadline
360	EncodedAllocTextLen
361	EncodedAllocText
362	EncodedUnderlyingIssuerLen

363	EncodedUnderlyingIssuer
364	EncodedUnderlyingSecurityDescLen
365	EncodedUnderlyingSecurityDesc
366	AllocPrice
367	QuoteSetValidUntilTime
368	QuoteEntryRejectReason
369	LastMsgSeqNumProcessed
370	OnBehalfOfSendingTime
371	RefTagID
372	RefMsgType
373	SessionRejectReason
374	BidRequestTransType
375	ContraBroker
376	ComplianceID
377	SolicitedFlag
378	ExecRestatementReason
379	BusinessRejectRefID
380	BusinessRejectReason
381	GrossTradeAmt
382	NoContraBrokers
383	MaxMessageSize
384	NoMsgTypes
385	MsgDirection
386	NoTradingSessions
387	TotalVolumeTraded
388	DiscretionInst
389	DiscretionOffset
390	BidID
391	ClientBidID
392	ListName
393	TotalNumSecurities
394	BidType
395	NumTickets
396	SideValue1
397	SideValue2
398	NoBidDescriptors
399	BidDescriptorType

400	BidDescriptor
401	SideValueInd
402	LiquidityPctLow
403	LiquidityPctHigh
404	LiquidityValue
405	EFPTTrackingError
406	FairValue
407	OutsideIndexPct
408	ValueOfFutures
409	LiquidityIndType
410	WtAverageLiquidity
411	ExchangeForPhysical
412	OutMainCntryUIndex
413	CrossPercent
414	ProgRptReqs
415	ProgPeriodInterval
416	IncTaxInd
417	NumBidders
418	TradeType
419	BasisPxType
420	NoBidComponents
421	Country
422	TotNoStrikes
423	PriceType
424	DayOrderQty
425	DayCumQty
426	DayAvgPx
427	GTBookingInst
428	NoStrikes
429	ListStatusType
430	NetGrossInd
431	ListOrderStatus
432	ExpireDate
433	ListExecInstType
434	CxlRejResponseTo
435	UnderlyingCouponRate
436	UnderlyingContractMultiplier
437	ContraTradeQty

438	ContraTradeTime
439	ClearingFirm
440	ClearingAccount
441	LiquidityNumSecurities
442	MultiLegReportingType
443	StrikeTime
444	ListStatusText
445	ListStatusEncodedTextLen
446	ListStatusEncodedText

FIX Field Index sorted by field name:

101	(Not Defined)
220 - 222	Reserved/Allocated to the Fixed Income proposal
224 - 230	Reserved/Allocated to the Fixed Income proposal
232 - 261	Reserved/Allocated to the Fixed Income proposal
1	Account
159	AccruedInterestAmt
158	AccruedInterestRate
334	Adjustment
2	AdvId
3	AdvRefID
4	AdvSide
5	AdvTransType
266	AggregatedBook
79	AllocAccount
153	AllocAvgPx
209	AllocHandlInst
70	AllocID
196	AllocLinkID
197	AllocLinkType
154	AllocNetMoney
366	AllocPrice
88	AllocRejCode
80	AllocShares
87	AllocStatus
161	AllocText
71	AllocTransType
74	AvgPrxPrecision
6	AvgPx
419	BasisPxType
7	BeginSeqNo
8	BeginString
219	Benchmark
400	BidDescriptor
399	BidDescriptorType

189	BidForwardPoints
390	BidID
132	BidPx
374	BidRequestTransType
134	BidSize
188	BidSpotRate
394	BidType
9	BodyLength
92	BrokerOfCredit
380	BusinessRejectReason
379	BusinessRejectRefID
330	BuyVolume
152	CashOrderQty
185	CashSettlAgentAcctName
184	CashSettlAgentAcctNum
183	CashSettlAgentCode
186	CashSettlAgentContactName
187	CashSettlAgentContactPhone
182	CashSettlAgentName
10	Checksum
440	ClearingAccount
439	ClearingFirm
391	ClientBidID
109	ClientID
11	ClOrdID
12	Commission
13	CommType
376	ComplianceID
375	ContraBroker
231	ContractMultiplier
437	ContraTradeQty
337	ContraTrader
438	ContraTradeTime
292	CorporateAction
421	Country
223	CouponRate

203	CoveredOrUncovered
413	CrossPercent
14	CumQty
15	Currency
204	CustomerOrFirm
84	CxlQty
102	CxlRejReason
434	CxlRejResponseTo
125	CxlType (no longer used)
426	DayAvgPx
425	DayCumQty
424	DayOrderQty
293	DefBidSize
294	DefOfferSize
285	DeleteReason
128	DeliverToCompID
145	DeliverToLocationID
129	DeliverToSubID
284	DeskID
388	DiscretionInst
389	DiscretionOffset
127	DKReason
86	DlvlyInst (no longer used)
329	DueToRelated
168	EffectiveTime
405	EFPTackingError
164	EmailThreadID
94	EmailType
361	EncodedAllocText
360	EncodedAllocTextLen
359	EncodedHeadline
358	EncodedHeadlineLen
349	EncodedIssuer
348	EncodedIssuerLen
353	EncodedListExeclInst
352	EncodedListExeclInstLen

351	EncodedSecurityDesc
350	EncodedSecurityDescLen
357	EncodedSubject
356	EncodedSubjectLen
355	EncodedText
354	EncodedTextLen
363	EncodedUnderlyingIssuer
362	EncodedUnderlyingIssuerLen
365	EncodedUnderlyingSecurityDesc
364	EncodedUnderlyingSecurityDescLen
98	EncryptMethod
16	EndSeqNo
411	ExchangeForPhysical
100	ExDestination
76	ExecBroker
17	ExecID
18	ExecInst
19	ExecRefID
378	ExecRestatementReason
20	ExecTransType
150	ExecType
432	ExpireDate
126	ExpireTime
406	FairValue
291	FinancialStatus
121	ForexReq
64	FutSettDate
193	FutSettDate2
123	GapFillFlag
381	GrossTradeAmt
427	GTBookingInst
327	HaltReason
21	HandlInst
148	Headline
108	HeartBtInt
332	HighPx
22	IDSrc

416	IncTaxInd
328	InViewOfCommon
23	IOIId
130	IOINaturalFlag
24	IOIOthSvc (no longer used)
25	IOIQltyInd
104	IOIQualifier
26	IOIRefID
27	IOIShares
28	IOITransType
106	Issuer
29	LastCapacity
195	LastForwardPoints
30	LastMkt
369	LastMsgSeqNumProcessed
31	LastPx
32	LastShares
194	LastSpotRate
151	LeavesQty
33	LinesOfText
409	LiquidityIndType
441	LiquidityNumSecurities
403	LiquidityPctHigh
402	LiquidityPctLow
404	LiquidityValue
69	ListExecInst
433	ListExecInstType
66	ListID
392	ListName
431	ListOrderStatus
67	ListSeqNo
446	ListStatusEncodedText
445	ListStatusEncodedTextLen
429	ListStatusType
444	ListStatusText
114	LocateReqd
283	LocationID

333	LowPx
264	MarketDepth
205	MaturityDay
200	MaturityMonthYear
111	MaxFloor
383	MaxMessageSize
210	MaxShow
288	MDEntryBuyer
272	MDEntryDate
278	MDEntryID
282	MDEntryOriginator
290	MDEntryPositionNo
270	MDEntryPx
280	MDEntryRefID
289	MDEntrySeller
271	MDEntrySize
273	MDEntryTime
269	MDEntryType
275	MDMkt
262	MDReqID
281	MDReqRejReason
279	MDUpdateAction
265	MDUpdateType
347	MessageEncoding
110	MinQty
137	MiscFeeAmt
138	MiscFeeCurr
139	MiscFeeType
385	MsgDirection
34	MsgSeqNum
35	MsgType
442	MultiLegReportingType
430	NetGrossInd
118	NetMoney
36	NewSeqNo
78	NoAllocs
420	NoBidComponents
398	NoBidDescriptors
382	NoContraBrokers

85	NoDlvInst (no longer used)
124	NoExecs
199	NoIOQualifiers
268	NoMDEntries
267	NoMDEntryTypes
136	NoMiscFees
384	NoMsgTypes
73	NoOrders
295	NoQuoteEntries
296	NoQuoteSets
146	NoRelatedSym
215	NoRoutingIDs
82	NoRpts
428	NoStrikes
208	NotifyBrokerOfCredit
386	NoTradingSessions
346	NumberOfOrders
417	NumBidders
157	NumDaysInterest
395	NumTickets
191	OfferForwardPoints
133	OfferPx
135	OfferSize
190	OfferSpotRate
115	OnBehalfOfCompID
144	OnBehalfOfLocationID
370	OnBehalfOfSendingTime
116	OnBehalfOfSubID
77	OpenClose
286	OpenCloseSettleFlag
206	OptAttribute
37	OrderID
38	OrderQty
192	OrderQty2
103	OrdRejReason
39	OrdStatus
40	OrdType
41	OrigClOrdID

122	OrigSendingTime
42	OrigTime
412	OutMainCntryUIndex
407	OutsideIndexPct
211	PegDifference
43	PossDupFlag
97	PossResend
140	PrevClosePx
44	Price
423	PriceType
81	ProcessCode
415	ProgPeriodInterval
414	ProgRptReqs
201	PutOrCall
297	QuoteAckStatus
298	QuoteCancelType
276	QuoteCondition
299	QuoteEntryID
368	QuoteEntryRejectReason
117	QuoteID
300	QuoteRejectReason
131	QuoteReqID
303	QuoteRequestType
301	QuoteResponseLevel
302	QuoteSetID
367	QuoteSetValidUntilTime
319	RatioQty
96	RawData
95	RawDataLength
72	RefAllocID
372	RefMsgType
45	RefSeqNum
371	RefTagID
46	RelatdSym
113	ReportToExch
141	ResetSeqNumFlag
217	RoutingID
216	RoutingType
83	RptSeq

47	Rule80A (aka OrderCapacity)
198	SecondaryOrderID
91	SecureData
90	SecureDataLen
107	SecurityDesc
207	SecurityExchange
48	SecurityID
320	SecurityReqID
321	SecurityRequestType
322	SecurityResponseID
323	SecurityResponseType
179	SecuritySettlAgentAcctName
178	SecuritySettlAgentAcctNum
177	SecuritySettlAgentCode
180	SecuritySettlAgentContactName
181	SecuritySettlAgentContactPhone
176	SecuritySettlAgentName
324	SecurityStatusReqID
326	SecurityTradingStatus
167	SecurityType
287	SellerDays
331	SellVolume
49	SenderCompID
142	SenderLocationID
50	SenderSubID
51	SendingDate (no longer used)
52	SendingTime
373	SessionRejectReason
174	SettlBrkrCode
119	SettlCurrAmt
120	SettlCurrency
155	SettlCurrFxRate
156	SettlCurrFxRateCalc
172	SettlDeliveryType
173	SettlDepositoryCode

175	SettlInstCode
162	SettlInstID
160	SettlInstMode
214	SettlInstRefID
165	SettlInstSource
163	SettlInstTransType
166	SettlLocation
63	SettlMntTyp
53	Shares
54	Side
396	SideValue1
397	SideValue2
401	SideValueInd
89	Signature
93	SignatureLength
377	SolicitedFlag
218	SpreadToBenchmark
171	StandInstDbID
170	StandInstDbName
169	StandInstDbType
99	StopPx
202	StrikePrice
443	StrikeTime
147	Subject
263	SubscriptionRequestType
55	Symbol
65	SymbolSfx

56	TargetCompID
143	TargetLocationID
57	TargetSubID
112	TestReqID
58	Text
274	TickDirection
59	TimeInForce
393	TotalNumSecurities
387	TotalVolumeTraded
68	TotNoOrders (formerly ListNoOrds) named:
422	TotNoStrikes
304	TotQuoteEntries
277	TradeCondition
75	TradeDate
418	TradeType
336	TradingSessionID
344	TradSesCloseTime
345	TradSesEndTime
338	TradSesMethod
339	TradSesMode
342	TradSesOpenTime
343	TradSesPreCloseTime
335	TradSesReqID
341	TradSesStartTime
340	TradSesStatus
60	TransactTime
318	Underlying Currency

436	UnderlyingContractMultiplier
435	UnderlyingCouponRate
305	UnderlyingIDSource
306	UnderlyingIssuer
314	UnderlyingMaturityDay
313	UnderlyingMaturityMonthYear
317	UnderlyingOptAttribute
315	UnderlyingPutOrCall
307	UnderlyingSecurityDesc
308	UnderlyingSecurityExchange
309	UnderlyingSecurityID
310	UnderlyingSecurityType
316	UnderlyingStrikePrice
311	UnderlyingSymbol
312	UnderlyingSymbolSfx
325	UnsolicitedIndicator
61	Urgency
149	URLLink
62	ValidUntilTime
408	ValueOfFutures
105	WaveNo
410	WtAverageLiquidity
213	XmlData
212	XmlDataLen

Appendix A

Valid Currency Codes

Currency codes used in FIX are those defined in ISO 4217 standard. To obtain the current valid list please contact the ISO 4217 secretariat at +44-181-996-9000.

Note: Prices defined in FIX messages should be made consistent with the currency code used. In some markets, prices are quoted as multiples or fractions of the currency, FIX messages should normalize the amount to coincide with the indicated code (e.g. UK securities are quoted in pence but must be represented in FIX messages as pounds).

Appendix B

Checksum Calculation

The checksum of a FIX message is calculated by summing every byte of the message up to but not including the checksum field itself. This checksum is then transformed into a modulo 256 number for transmission and comparison. The checksum is calculated after all encryption is completed, i.e. the message as transmitted between parties is processed.

For transmission, the checksum must be sent as printable characters, so the checksum is transformed into three ASCII digits.

For example, if the checksum has been calculated to be 274 then the modulo 256 value is 18 ($256 + 18 = 274$). This value would be transmitted as `|10=018|` where "10=" is the tag for the checksum field.

A sample code fragment to generate the checksum field is as follows:

```
char *GenerateChecksum( char *buf, long bufLen )
{
    static char tmpBuf[ 4 ];
    long idx;
    unsigned int cks;

    for( idx = 0L, cks = 0; idx < bufLen; cks += (unsigned int)buf[ idx++ ] );
    sprintf( tmpBuf, "%03d", (unsigned int)( cks % 256 ) );
    return( tmpBuf );
}
```

Appendix C

Reuters Exchange Mnemonics

Abidjan Stock Exchange	CI	Korea Stock Exchange	KS
American Stock Exchange	A	Kuala Lumpur Stock Exchange	KL
Amman Stock Exchange	AM	Kuwait Stock Exchange	KW
AEX Options and Futures Exchange	E	Kyoto Stock Exchange	KY
AEX Stock Exchange	AS	Lagos Stock Exchange	LG
Australian Stock Exchange	AX	Latin American Market	LA
Bahrain Stock Exchange	BH	in Spain (LATIBEX)	
Barcelona Stock Exchange -	BC	Le Nouveau Marche	LN
Floor Trading		Lima Stock Exchange	LM
Barcelona Stock Exchange -	MC	Lisbon Stock Exchange (Portugal)	LS
CATS Feed		London Stock Exchange	L
Beirut Stock Exchange	BY	Lusaka Stock Exchange	LZ
Belfox	b	Luxembourg Stock Exchange	LU
Berlin Stock Exchange	BE	Madras Stock Exchange	MD
Berne Stock Exchange	BN	Madrid Stock Exchange -	MA
Bilbao Stock Exchange	BI	Floor Trading	
		Madrid Stock Exchange -	MC
Bombay Stock Exchange	BO	CATS Feed	
Boston Stock Exchange	B	Malta Stock Exchange	MT
Botswana Share Market	BT	Mauritius Stock Exchange	MZ
Bremen Stock Exchange	BM	Medellin Stock Exchange	ML
Brussels Stock Exchange	BR	Mexican Stock Exchange	MX
Calcutta Stock Exchange	CL	Milan Stock Exchange	MI
Canadian Ventures Exchange	V	MONEP Paris Stock Options	p
Channel Islands	CH	Montreal Exchange	M
Chicago Board Options	W	Moscow Inter Bank Currency Exchange	MM
Exchange		Moscow Stock Exchange	MO
Chicago Stock Exchange	MW	Munich Stock Exchange	MU
Chile Electronic Exchange	CE	Muscat Stock Exchange	OM
Cincinnati Stock Exchange	C	Namibia Stock Exchange	NM
Colombo Stock Exchange	CM	Nagoya Stock Exchange	NG
Copenhagen Stock Exchange	CO	Nairobi Stock Exchange	NR
Dehli Stock Exchange	DL	NASDAQ	O
		NASDAQ Dealers -	OB
Doha Securities Market	QA	Bulletin Board	
Dubai Financial Market	DU	NASDAQ Japan	OJ
Dusseldorf Stock Exchange	D	National Stock Exchange of India	NS
Electronic Stock Exchange of	EB	New York Stock Exchange	N
Venezuela		New Zealand Stock Exchange	NZ
Eurex Germany (DTB)	d	NewEx (Austria)	NW
Eurex Switzerland (SFF)	Z	Occidente Stock Exchange	OD
Frankfurt Stock Exchange	F	Osaka Stock Exchange	OS
Fukuoka Stock Exchange	FU	Oslo Stock Exchange	OL
Ghana Stock Exchange	GH	Pacific Stock Exchange	P
Hamburg Stock Exchange	H	Paris Stock Exchange	PA
Hanover Stock Exchange	HA	Philadelphia Stock Exchange	PH
Helsinki Stock Exchange	HE	Philadelphia Stock Exchange	X
Hong Kong Stock Exchange	HK	Options	
Iceland Stock Exchange	IC	Philippine Stock Exchange	PS
Interbolsa (Portugal)	IN	Prague Stock Exchange	PR
Irish Stock Exchange	I	Pink Sheets (National	PNK
Istanbul Stock Exchange	IS	Quotation Bureau)	
Jakarta Stock Exchange	JK	RASDAQ (Romania)	RQ
Japanese Securities Dealers	Q	Riga Stock Exchange	RI
Association (JASDAQ)		Rio de Janeiro OTC Stock	SO
Johannesburg Stock Exchange	J	Exchange (SOMA)	
Karachi Stock Exchange	KA	Russian Trading System	RTS
KASDAQ (Korea)	KQ	Santiago Stock Exchange	SN
Kazakhstan Stock Exchange	KZ	Sao Paulo Stock Exchange	SA

Sapporo Stock Exchange	SP
Saudi Stock Exchange	SE
SBI Stock Exchange (Sweden)	SBI
Singapore Stock Exchange	SI
Shanghai Stock Exchange	SS
Shenzhen Stock Exchange	SZ
Stockholm Stock Exchange	ST
Stuttgart Stock Exchange	SG
St. Petersburg Stock Exchange	PE
Surabaya Stock Exchange	SU
SWX Swiss Exchange	S
Taiwan OTC Securities Exchange	TWO
Taiwan Stock Exchange	TW
Tallinn Stock Exchange	TL
Tel Aviv Stock Exchange	TA
Thailand Stock Exchange	BK
Third Market	TH
Tokyo Stock Exchange	T
Toronto Options Exchange	K
Toronto Stock Exchange	TO
Tradepoint Stock Exchange	TP
Tunis Stock Exchange	TN
Ukraine PFTS	PFT
Valencia Stock Exchange	VA
Vilnius Stock Exchange	VL
virt-x	VX
Vienna Stock Exchange	VI

Zimbabwe Stock Exchange	ZI	
Other (assigned numeric values):		
American Stock Exchange Options	1	
Chicago Mercantile Exchange (CME)	2	
Futures Exchange (LIFFE)		
Jiway	14	
International Securities Market Association (ISMA)	15	
London International Financial	3	
London Traded Options Market	5	
MEFF Renta Variable	16	
Montreal Exchange Options (MOE)	6	
New York Mercantile Exchange (NYMEX)	12	
None	0	
Non-exchange-based Over The Counter Market	11	
NYFIX Millennium	13	
NYSE BBSS (broker booth system)	10	
Pacific Stock Exchange Options (PAO)		8
POSIT	4	
Stockholm Options Market	17	
Vancouver Options Exchange (VAO)	9	
MEFF Renta Variable	??	
Stockholm Options Market	??	

Appendix D

Order State Change Matrices

The following matrices are included to clarify the sequence of messages and the status of orders involved in the submission and processing of new orders, executions, cancel requests, cancel/replace requests and order status requests. The matrices have been arranged in groups as follows:

Ref	Group	Description
D1	Vanilla	Filled order
D2	Vanilla	Part-filled day order, done for day
D3	Cancel	Cancel request issued for a zero-filled order
D4	Cancel	Cancel request issued for a part-filled order – executions occur whilst cancel request is active
D5	Cancel	Cancel request issued for an order that becomes filled before cancel request can be accepted
D6	Replace to increase qty	Zero-filled order, cancel/replace request issued to increase order qty
D7	Replace to increase qty	Part-filled order, followed by cancel/replace request to increase order qty, execution occurs whilst order is pending replace
D8	Replace to increase qty	Filled-order followed by cancel/replace request to increase order quantity
D9	Replace not for qty change	Cancel/replace request (not for quantity change) is rejected as a fill has occurred
D10	Replace to decrease qty	Cancel/replace request sent whilst execution is being reported – the requested order qty exceeds the cum qty . Order is replaced then filled
D11	Replace to decrease qty	Cancel/replace request sent whilst execution is being reported – the requested order qty equals the cum qty – order qty is amended to cum qty
D12	Replace to decrease qty	Cancel/replace request sent whilst execution is being reported – the requested order qty is below cum qty – order qty is amended to cum qty
D13	Replace - sequence	One cancel/replace request is issued which is accepted – another one is issued which is also accepted
D14	Replace - sequence	One cancel/replace request is issued which is rejected before order becomes pending replace – then another one is issued which is accepted
D15	Replace - sequence	One cancel/replace request is issued which is rejected after it is in pending replace – then another one is issued which is accepted
D16	Replace - chaining	One cancel/replace request is issued followed immediately by another – broker processes sequentially
D17	Replace - chaining	One cancel/replace request is issued followed immediately by another – broker rejects the second as order is pending replace
D18	Unsolicited reports	Telephoned order
D19	Unsolicited reports	Unsolicited cancellation of a part-filled order
D20	Unsolicited reports	Unsolicited replacement of a part-filled order
D21	Unsolicited reports	Unsolicited reduction of order quantity by sell side
D22	Order reject	Order rejected due to duplicate ClOrdID
D23	Order reject	Order rejected because the order has already been verbally submitted
D24	Status	Order status request rejected for unknown order
D25	Status	Status request followed by "Nothing done".
D26	Status	Order sent, immediately followed by a status request. Subsequent status requests sent
D27	GT	GTC order partially filled, restated (renewed) and partially filled the following day

D28	GT	GTC order with partial fill, a 2:1 stock split then a partial fill and fill the following day
D29	GT	GTC order partially filled, restated(renewed) and canceled the following day
D30	GT	GTC order partially filled, restated(renewed) followed by replace request to increase quantity
D31	Resend	Poss resend
D32	TIF	Fill or kill order that cannot be filled
D33	TIF	Immediate or Cancel order that cannot be immediately hit
D34	Execution correct/cancel	Filled order, followed by correction and cancellation of executions
D35	Execution correct/cancel	A cancel of a partially filled order followed by an execution cancel(bust) and new execution.
D36	Execution correct/cancel	GTC order partially filled, restated (renewed) and partially filled the following day, with corrections of quantity on both executions.
D37	Stopped/Guarantee	A stopped (execution price guarantee) report followed by execution.

The Table below shows which state transitions have been illustrated by the matrices in this Appendix (marked with an asterisk). The row represents the current value of OrdStatus and the column represents the next value as reported back to the buy-side via an execution report or order cancel reject message. Next to each OrdStatus value is its precedence – this is used when the order exists in a number of states simultaneously to determine the value that should be reported back. Note that absence of a scenario should not necessarily be interpreted as meaning that the state transition is not allowed:

<i>OrdStatus (precedence value)</i>	New (2)	Partially Filled (4)	Filled (8)	Done For Day (10)	Pending Cancel (12)	Pending Replace (11)	Replaced (3)	Canceled (5)	Rejected (2)	Stopped (7)
Pending New (2)	*								*	
New (2)	*	*	*	*	*	*	*		*	*
Partially Filled (4)		*	*	*	*	*		*		
Filled (8)		*	*			*				
Done for Day (10)		*								
Pending Cancel (12)	*	*	*		*			*		
Pending Replace (11)	*	*	*			*	*	*		
Replaced (3)		*								
Canceled (5)										
Rejected (2)										
Stopped (7)		*								

How to read the Order State Change Matrices:

- The ‘Execution Report’ message is referred to simply as ‘Execution’
- The ‘Order Cancel/Replace Request’ and ‘Order Cancel Request’ messages are referred to as ‘Replace Request’ and ‘Cancel Request’ respectively
- The shaded rows represent messages sent from buy-side to the sell-side
- In general where two lines of a matrix share the same time, this means either
 - that there are two possible paths (e.g. a request is accepted or rejected) – in this case the first row of the two possible paths is the reject case which is italicized. The non-italicized row is the path that is continued by the remainder of the matrix
 - that two messages are being sent at the same time but in different directions such that the messages cross on the connection (e.g. a cancel request is sent at the same time as the sell-side is sending an execution) – in this case both lines have bold text

- For scenarios involving cancel requests or cancel/replace requests ‘X’ refers to the original order, ‘Y’ refers to the cancel/replacing order. A similar convention is used for corrections or cancels to executions

D1 - Filled order

<u>Time</u>	<u>Message Received</u> (CLOrdID, OrigCLOrdID)	<u>Message Sent</u> (CLOrdID, OrigCLOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Comment</u>
1	New Order(X)					10000				
2		Execution(X)	Rejected	Rejected	New	10000	0	0	0	If order is rejected by sales
2		Execution(X)	New	New	New	10000	0	10000	0	
3		Execution(X)	Rejected	Rejected	New	10000	0	0	0	If order is rejected by trader/exchange
3		Execution(X)	Partial Fill	Partially Filled	New	10000	2000	8000	2000	Execution of 2000
4		Execution(X)	Partial Fill	Partially Filled	New	10000	3000	7000	1000	Execution of 1000
5		Execution(X)	Fill	Filled	New	10000	10000	0	7000	Execution of 7000

D2 – Part-filled day order, done for day

<u>Time</u>	<u>Message Received</u> (CLOrdID, OrigCLOrdID)	<u>Message Sent</u> (CLOrdID, OrigCLOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Comment</u>
1	New Order(X)					10000				
2		Execution(X)	Rejected	Rejected	New	10000	0	0	0	If order is rejected
2		Execution(X)	New	New	New	10000	0	10000	0	
3		Execution(X)	Partial Fill	Partially Filled	New	10000	2000	8000	2000	Execution of 2000
4		Execution(X)	Partial Fill	Partially Filled	New	10000	3000	7000	1000	Execution of 1000
5		Execution(X)	Done for Day	Done for Day	New	10000	3000	0	0	Assuming day order. See other examples which cover GT orders

D3 – Cancel request issued for a zero-filled order

<u>Time</u>	<u>Message Received</u> (CfOrdID, OrigCfOrdID)	<u>Message Sent</u> (CfOrdID, OrigCfOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Comment</u>
1	New Order(X)					10000				
2		Execution(X)	Rejected	Rejected	New	10000	0	0	0	If order is rejected
2		Execution(X)	New	New	New	10000	0	10000	0	
3	Cancel Request(Y,X)					10000				
4		Cancel Reject (Y,X)		New		10000				If rejected by salesperson
4		Execution (Y,X)	Pending Cancel	Pending Cancel	New	10000	0	10000	0	
5		Cancel Reject (Y,X)		New		10000				If rejected by trader/exchange
5		Execution (Y,X)	Canceled	Canceled	New	10000	0	0	0	

D4 – Cancel request issued for a part-filled order – executions occur whilst cancel request is active

<u>Time</u>	<u>Message Received</u> (CfOrdID, OrigCfOrdID)	<u>Message Sent</u> (CfOrdID, OrigCfOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Comment</u>
1	New Order(X)					10000				
2		Execution(X)	Rejected	Rejected	New	10000	0	0	0	If order is rejected
2		Execution(X)	New	New	New	10000	0	10000	0	
3		Execution(X)	Partial Fill	Partially Filled	New	10000	2000	8000	2000	Execution for 2000
4	Cancel Request(Y,X)					10000				
4		Execution(X)	Partial Fill	Partially Filled	New	10000	5000	5000	3000	Execution for 3000. This execution passes the cancel request on the connection

5		Cancel Reject (Y,X)		Partially Filled		10000				If request is rejected
5		Execution (Y,X)	Pending Cancel	Pending Cancel	New	10000	5000	5000	0	'Pending cancel' order status takes precedence over 'partially filled' order status
6		Execution(X)	Partial Fill	Pending Cancel	New	10000	6000	4000	1000	Execution for 1000 whilst order is pending cancel – 'pending cancel' order status takes precedence over 'partially filled' order status
7		Cancel Reject (Y,X)		Partially Filled		10000				If request is rejected
7		Execution (Y,X)	Canceled	Canceled	New	10000	6000	0	0	'Canceled' order status takes precedence over 'partially filled' order status

D5 – Cancel request issued for an order that becomes filled before cancel request can be accepted

<u>Time</u>	<u>Message Received</u> (CfOrdID, OrigCfOrdID)	<u>Message Sent</u> (CfOrdID, OrigCfOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Comment</u>
1	New Order(X)					10000				
2		Execution(X)	Rejected	Rejected	New	10000	0	0	0	If order is rejected
2		Execution(X)	New	New	New	10000	0	10000	0	
3		Execution(X)	Partial Fill	Partially Filled	New	10000	2000	8000	2000	Execution for 2000
4	Cancel Request(Y,X)					10000				
4		Execution(X)	Partial Fill	Partially Filled	New	10000	5000	5000	3000	Execution for 3000. This execution passes the cancel request on the connection
5		Cancel Reject (Y,X)		Partially Filled		10000				If request is rejected
5		Execution (Y,X)	Pending Cancel	Pending Cancel	New	10000	5000	5000	0	'Pending cancel' order status takes precedence over 'partially filled' order status

6		Execution(X)	Fill	Pending Cancel	New	10000	10000	0	5000	Execution for 5000 whilst order is pending cancel. 'Pending cancel' order status takes precedence over 'filled' order status
7		Cancel Reject (Y,X)		Filled		10000				Cancel request rejected – CxlRejectReason = 0 (too late to cancel)

D6 – Zero-filled order, cancel/replace request issued to increase order qty

<u>Time</u>	<u>Message Received</u> (ClOrdID, OrigClOrdID)	<u>Message Sent</u> (ClOrdID, OrigClOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Comment</u>
1	New Order(X)					10000				
2		Execution(X)	Rejected	Rejected	New	10000	0	0	0	If order is rejected by broker
2		Execution(X)	New	New	New	10000	0	10000	0	
3	Replace Request(Y,X)					11000				Request to increase order qty to 11000
4		Cancel Reject (Y,X)		New		10000				If request is rejected by salesperson
4		Execution (Y,X)	Pending Replace	Pending Replace	New	10000	0	10000	0	
5		Cancel Reject (Y,X)		New		10000				If rejected by trader/exchange
5		Execution (Y,X)	Replace	Replaced	New	11000	0	11000	0	'Replaced' order status takes precedence over 'new' order status
6		Execution (Y)	Partial Fill	Partially Filled	New	11000	1000	10000	1000	Execution for 1000
7		Execution (Y)	Partial Fill	Partially Filled	New	11000	3000	8000	2000	Execution for 2000

D7 – Part-filled order, followed by cancel/replace request to increase order qty, execution occurs whilst order is pending replace

<u>Time</u>	<u>Message Received</u> (ClOrdID, OrigClOrdID)	<u>Message Sent</u> (ClOrdID, OrigClOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Comment</u>
1	New Order(X)					10000				
2		Execution(X)	Rejected	Rejected	New	10000	0	0	0	If order is rejected by broker
2		Execution(X)	New	New	New	10000	0	10000	0	
3		Execution(X)	Partial Fill	Partially Filled	New	10000	1000	9000	1000	Execution for 1000
4	Replace Request(Y,X)					12000				Request increase in order quantity to 12000
5		Cancel Reject (Y,X)		Partially Filled		10000				If request is rejected
5		Execution (Y,X)	Pending Replace	Pending Replace	New	10000	1000	9000	0	'Pending replace' order status takes precedence over 'partially filled' order status
6		Execution(X)	Partial Fill	Pending Replace	New	10000	1100	8900	100	Execution for 100 before cancel/replace request is responded to
7		Cancel Reject (Y,X)		Partially Filled		10000				If request is rejected
7		Execution (Y,X)	Replace	Partially Filled	New	12000	1100	10900	0	'Partially filled'' order status takes precedence over 'replaced' order status
8		Execution(Y)	Fill	Filled	New	12000	12000	0	10900	Execution for 10900

D8 – Filled order, followed by cancel/replace request to increase order quantity

<u>Time</u>	<u>Message Received</u> (ClOrdID, OrigClOrdID)	<u>Message Sent</u> (ClOrdID, OrigClOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Comment</u>
1	New Order(X)					10000				
2		Execution(X)	Rejected	Rejected	New	10000	0	0	0	If order is rejected by broker
2		Execution(X)	New	New	New	10000	0	10000	0	
3		Execution(X)	Fill	Filled	New	10000	10000	0	10000	Execution for 10000

4	Replace Request(Y,X)					12000				Request increase in order quantity to 12000
5		Cancel Reject (Y,X)		Filled		10000				If request is rejected
5		Execution (Y,X)	Pending Replace	Pending Replace	New	10000	10000	0	0	'Pending replace' order status takes precedence over 'partially filled' order status
6		Cancel Reject (Y,X)		Filled		10000				If request is rejected
6		Execution (Y,X)	Replace	Partially Filled	New	12000	10000	2000	0	'Partially filled' order status takes precedence over 'replaced' order status.
7		Execution(Y)	Fill	Filled	New	12000	12000	0	2000	Execution for 2000

D9 – Cancel/replace request (not for quantity change) is rejected as a fill has occurred

<u>Time</u>	<u>Message Received</u> (ClOrdID, OrigClOrdID)	<u>Message Sent</u> (ClOrdID, OrigClOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Comment</u>
1	New Order(X)					10000				
2		Execution(X)	Rejected	Rejected	New	10000	0	0	0	If order is rejected by broker
2		Execution(X)	New	New	New	10000	0	10000	0	
3		Execution(X)	Partial Fill	Partially Filled	New	10000	1000	9000	1000	Execution for 1000
4	Replace Request(Y,X)					10000				Assume in this scenario that client does not wish to increase qty (e.g. client wants to amend limit price)
4		Execution (X)	Fill	Filled	New	10000	10000	0	9000	Execution for 9000 – the replace request message and this execution report pass each other on the connection
5		Cancel Reject (Y,X)		Filled		10000				CxlRejectReason = 0 (too late to cancel)

D10 – Cancel/replace request sent whilst execution is being reported – the requested order qty exceeds the cum qty. Order is replaced then filled

<u>Time</u>	<u>Message Received</u> (ClOrdID, OrigClOrdID)	<u>Message Sent</u> (ClOrdID, OrigClOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Comment</u>
1	New Order(X)					10000				
2		Execution(X)	Rejected	Rejected	New	10000	0	0	0	If order is rejected
2		Execution(X)	New	New	New	10000	0	10000	0	
3		Execution(X)	Partial Fill	Partially Filled	New	10000	1000	9000	1000	Execution for 1000
4	Replace Request(Y,X)					8000				Request a decrease order quantity to 8000 (leaving 7000 open)
4		Execution(X)	Partial Fill	Partially Filled	New	10000	1500	8500	500	Execution for 500 sent. Replace request and this execution report pass each other on the connection
5		Cancel Reject (Y,X)		Partially Filled		10000				If request is rejected by salesperson
5		Execution (Y,X)	Pending Replace	Pending Replace	New	10000	1500	8500	0	'Pending replace' order status takes precedence over 'partially filled' order status
6		Execution(X)	Partial Fill	Pending Replace	New	10000	1600	8400	100	Execution for 100 occurs before cancel/replace request is accepted
7		Cancel Reject (Y,X)		Partially Filled		10000				If request is rejected by trader/exchange
7		Execution (Y,X)	Replace	Partially Filled	New	8000	1600	6400	0	'Partially filled' order status takes precedence over 'replaced' order status. Replace is accepted as requested order qty exceeds cum qty
8		Execution (Y)	Fill	Filled	New	8000	8000	0	6400	Execution for 6400.

D11 – Cancel/replace request sent whilst execution is being reported – the requested order qty equals the cum qty – order qty is amended to cum qty

<u>Time</u>	<u>Message Received</u> (CfOrdID, OrigCfOrdID)	<u>Message Sent</u> (CfOrdID, OrigCfOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Comment</u>
1	New Order(X)					10000				
2		Execution(X)	Rejected	Rejected	New	10000	0	0	0	If order is rejected by broker
2		Execution(X)	New	New	New	10000	0	10000	0	
3	Replace Request(Y,X)					7000				Client wishes to amend order qty to 7000 shares
3		Execution(X)	Partial Fill	Partially Filled	New	10000	7000	3000	7000	Execution for 7000 - the replace message and this execution report pass each other on the connection
4		Execution (Y,X)	Replace	Filled	New	7000	7000	0	0	The replace request is interpreted as requiring the balance of the order to be canceled – the ‘filled’ order status takes precedence over ‘canceled’ or ‘replaced’

D12 – Cancel/replace request sent whilst execution is being reported – the requested order qty is below cum qty – order qty is amended to cum qty

<u>Time</u>	<u>Message Received</u> (CfOrdID, OrigCfOrdID)	<u>Message Sent</u> (CfOrdID, OrigCfOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Comment</u>
1	New Order(X)					10000				
2		Execution(X)	Rejected	Rejected	New	10000	0	0	0	If order is rejected by broker
2		Execution(X)	New	New	New	10000	0	10000	0	
3	Replace Request(Y,X)					7000				Client wishes to amend order qty to 7000 shares
3		Execution(X)	Partial Fill	Partially Filled	New	10000	8000	2000	8000	Execution for 8000 - the replace message and this execution report pass each other on the connection
4		Execution (Y,X)	Replace	Filled	New	8000	8000	0	0	The replace request is interpreted as requiring the balance of the order to be canceled – the ‘filled’ order status takes precedence over ‘canceled’ or ‘replaced’

D13 – One cancel/replace request is issued which is accepted – another one is issued which is also accepted

<u>Time</u>	<u>Message Received</u> (CfOrdID, OrigCfOrdID)	<u>Message Sent</u> (CfOrdID, OrigCfOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Comment</u>
1	New Order(X)					10000				
2		Execution(X)	Rejected	Rejected	New	10000	0	0	0	If order is rejected by broker
2		Execution(X)	New	New	New	10000	0	10000	0	
3		Execution(X)	Partial Fill	Partially Filled	New	10000	1000	9000	1000	Execution for 1000
4	Replace Request(Y,X)					8000				Request decrease in order quantity to 8000, leaving 7000 open
5		Execution (Y,X)	Pending Replace	Pending Replace	New	10000	1000	9000	0	'Pending replace' order status takes precedence over 'partially filled' order status
6		Execution(X)	Partial Fill	Pending Replace	New	10000	1500	8500	500	Execution for 500
7		Execution (Y,X)	Replace	Partially Filled	New	8000	1500	6500	0	'Partially filled' order status takes precedence over 'replaced' order status
8		Execution (Y)	Partial Fill	Partially Filled	New	8000	3500	4500	2000	Execution for 2000
9	Replace Request(Z,Y)					6000				Request decrease in order quantity to 6000, leaving 2500 open
10		Execution (Z,Y)	Pending Replace	Pending Replace	New	8000	3500	4500	0	
11		Execution (Z,Y)	Replace	Partially Filled	New	6000	3500	2500	0	'Partially filled' order status takes precedence over 'replaced' order status
12		Execution(Z)	Fill	Filled	New	6000	6000	0	2500	Execution for 2500

D14 – One cancel/replace request is issued which is rejected before order becomes pending replace – then another one is issued which is accepted

<u>Time</u>	<u>Message Received</u> (CfOrdID, OrigCfOrdID)	<u>Message Sent</u> (CfOrdID, OrigCfOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Comment</u>
1	New Order(X)					10000				
2		Execution(X)	Rejected	Rejected	New	10000	0	0	0	If order is rejected by broker
2		Execution(X)	New	New	New	10000	0	10000	0	
3		Execution(X)	Partial Fill	Partially Filled	New	10000	1000	9000	1000	Execution for 1000
4	Replace Request(Y,X)					8000				Request decrease in order quantity to 8000, leaving 7000 open
5		Cancel Reject (Y,X)		Partially Filled		10000				Request is rejected
6		Execution(X)	Partial Fill	Partially Filled	New	10000	1500	8500	500	Execution for 500
7		Execution(X)	Partial Fill	Partially Filled	New	10000	3500	6500	2000	Execution for 2000
8	Replace Request(Z,X)					6000				Request decrease in order quantity to 6000, leaving 2500 open. Note that OrigCfOrdID = X
9		Execution (Z,X)	Pending Replace	Pending Replace	New	10000	3500	6500	0	Note that OrigCfOrdID = X
10		Execution (Z,X)	Replace	Partially Filled	New	6000	3500	2500	0	Note that OrigCfOrdID = X
11		Execution(Z)	Partial Fill	Partially Filled	New	6000	5000	1000	1500	Execution for 1500

D15 One cancel/replace request is issued which is rejected after it is in pending replace – then another one is issued which is accepted

<u>Time</u>	<u>Message Received</u> (CfOrdID, OrigCfOrdID)	<u>Message Sent</u> (CfOrdID, OrigCfOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Comment</u>
1	New Order(X)					10000				
2		Execution(X)	Rejected	Rejected	New	10000	0	0	0	If order is rejected by broker
2		Execution(X)	New	New	New	10000	0	10000	0	
3		Execution(X)	Partial Fill	Partially Filled	New	10000	1000	9000	1000	Execution for 1000
4	Replace Request(Y,X)					8000				Request decrease in order quantity to 8000, leaving 7000 open
5		Execution (Y,X)	Pending Replace	Pending Replace		10000	1000	9000	0	
6		Execution(X)	Partial Fill	Pending Replace	New	10000	1500	8500	500	Execution for 500. 'Pending replace' order status takes precedence over 'partially filled' order status
7		Cancel Reject (Y,X)		Partially Filled		10000				Request is rejected (e.g. by trader/exchange)
8		Execution(X)	Partial Fill	Partially Filled	New	10000	3500	6500	2000	Execution for 2000
9	Replace Request(Z,X)					6000				Request decrease in order quantity to 6000, leaving 2500 open. Note that OrigCfOrdID = X
10		Execution (Z,X)	Pending Replace	Pending Replace	New	10000	3500	6500	0	
11		Execution (Z,X)	Replace	Partially Filled	New	6000	3500	2500	0	
12		Execution(Z)	Partial Fill	Partially Filled	New	6000	5000	1000	1500	Execution for 1500

D16– One cancel/replace request is issued followed immediately by another – broker processes sequentially

<u>Time</u>	<u>Message Received</u> (CfOrdID, OrigCfOrdID)	<u>Message Sent</u> (CfOrdID, OrigCfOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Comment</u>
1	New Order(X)					10000				
2		Execution(X)	New	New	New	10000	0	10000	0	
3		Execution(X)	Partial Fill	Partially Filled	New	10000	1000	9000	1000	Execution for 1000
4	Replace Request(Y,X)					8000				Request decrease in order quantity to 8000, leaving 7000 open
5	Replace Request(Z,Y)					7000				Request decrease in order quantity to 7000, leaving 6000 open
6		Execution (Y,X)	Pending Replace	Pending Replace	New	10000	1000	9000	0	Broker processes Replace (Y,X) first
7		Execution (Y,X)	Replace	Partially Filled	New	8000	1000	7000	0	Broker processes Replace (Y,X) first
8		Execution (Z,Y)	Pending Replace	Pending Replace	New	8000	1000	7000	0	Broker then processes Replace (Z,Y)
9		Execution (Z,Y)	Replace	Partially Filled	New	7000	1000	6000	0	Broker then processes Replace (Z,Y)
10		Execution(Z)	Fill	Filled	New	7000	7000	0	6000	Execution for 6000

D17– One cancel/replace request is issued followed immediately by another – broker rejects the second as order is pending replace

<u>Time</u>	<u>Message Received</u> (CfOrdID, OrigCfOrdID)	<u>Message Sent</u> (CfOrdID, OrigCfOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Comment</u>
1	New Order(X)					10000				
2		Execution(X)	New	New	New	10000	0	10000	0	
3		Execution(X)	Partial Fill	Partially Filled	New	10000	1000	9000	1000	Execution for 1000
4	Replace Request(Y,X)					8000				Request decrease in order quantity to 8000, leaving 7000 open
5	Replace Request(Z,Y)					7000				Request decrease in order quantity to 7000, leaving 6000 open

6		Execution (Y,X)	Pending Replace	Pending Replace	New	10000	1000	9000	0	
7		Cancel Reject (Z,Y)		Pending Replace		10000				Rejected because broker does not support processing of order cancel replace request whilst order is pending cancel. CxlRejReason = 'Order already in pending cancel or pending replace status'
8		Execution (Y,X)	Replace	Partially Filled	New	8000	1000	7000	0	'Partially filled' order status takes precedence over 'replaced' order status
9		Execution (Y)	Partial Fill	Partially Filled	New	8000	3000	5000	2000	Execution for 2000

This matrix illustrates the case where the broker does not support multiple outstanding order cancel or order cancel/replace requests

D18 – Telephoned order

<u>Time</u>	<u>Message Received</u> (CfOrdID, OrigCfOrdID)	<u>Message Sent</u> (CfOrdID, OrigCfOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Comment</u>
1										Order for 10000 shares phoned to broker
2		Execution	New	New	New	10000	0	0	0	Confirm that the broker has accepted the order – note that broker does not need to capture a CfOrdID
3		Execution	Partial Fill	Partially Filled	New	10000	2000	8000	2000	Execution of 2000
4		Execution	Partial Fill	Partially Filled	New	10000	3000	7000	1000	Execution of 1000
5		Execution	Fill	Filled	New	10000	10000	0	7000	Execution of 7000

D19 – Unsolicited cancel of a part-filled order

<u>Time</u>	<u>Message Received</u> (CfOrdID, OrigCfOrdID)	<u>Message Sent</u> (CfOrdID, OrigCfOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Comment</u>
1	New Order(X)					10000				
2		Execution(X)	Rejected	Rejected	New	10000	0	0	0	If order is rejected by broker

2		Execution(X)	New	New	New	10000	0	10000	0	
3		Execution(X)	Partial Fill	Partially Filled	New	10000	1000	9000	1000	Execution for 1000
4										Broker verbally agrees to cancel order
5		Execution(X)	Canceled	Canceled	New	10000	1000	0	0	Broker signifies that order has been canceled - ExecRestatementReason = Verbal change

This scenario might occur if the buy-side has not implemented order cancel requests or alternatively there is an electronic communication problem at the point that the buy-side wishes to send a cancel request.

D20 – Unsolicited replacement of a part-filled order

<u>Time</u>	<u>Message Received</u> (CfOrdID, OrigCfOrdID)	<u>Message Sent</u> (CfOrdID, OrigCfOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Comment</u>
1	New Order(X)					10000				
2		Execution(X)	Rejected	Rejected	New	10000	0	0	0	If order is rejected by broker
2		Execution(X)	New	New	New	10000	0	10000	0	
3										Broker verbally agrees to increase order quantity to 11000
4		Execution(X)	Restated	New	New	11000	0	0	0	Broker signifies that order has been replaced ExecRestatementReason = Verbal
5		Execution(X)	Partial Fill	Partially Filled	New	11000	1000	10000	1000	Execution for 1000
6										Broker verbally agrees to increase order quantity to 12000
7		Execution(X)	Restated	Partially Filled	New	12000	1000	11000	0	Broker signifies that order has been replaced ExecRestatementReason = Verbal change

This scenario might occur if the buy-side has not implemented order cancel/replace requests or alternatively there is an electronic communication problem at the point that the buy-side wishes to send a cancel replace request

D21 - Unsolicited reduction of order quantity by sell side (e.g. for US ECNs to communicate Nasdaq SelectNet declines)

<u>Time</u>	<u>Message Received</u> (CfOrdID, OrigCfOrdID)	<u>Message Sent</u> (CfOrdID, OrigCfOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Comment</u>
1	New Order(X)					10000				
2		Execution(X)	Rejected	Rejected	New	10000	0	0	0	If order is rejected by broker
2		Execution(X)	New	New	New	10000	0	10000	0	
3		Execution(X)	Restated	New	New	9000	0	9000	0	ExecRestatementReason="Partial Decline of OrderQty"
4		Execution(X)	Fill	Filled	New	9000	9000	0	9000	

D22– Order rejected due to duplicate ClOrdID

<u>Time</u>	<u>Message Received</u> (ClOrdID, OrigClOrdID)	<u>Message Sent</u> (ClOrdID, OrigClOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Comment</u>
1	New Order(X)					10000				
2		Execution(X)	New	New	New	10000	0	10000	0	
3		Execution(X)	Partial Fill	Partially Filled	New	10000	1000	9000	1000	Execution for 1000
4	New Order(X)					10000				Order submitted with the same order id
5		Execution(X)	Rejected	Partially Filled	New	10000	1000	9000	0	OrdRejReason = duplicate order

D23 - Order rejected because the order has already been verbally submitted

<u>Time</u>	<u>Message Received</u> (ClOrdID, OrigClOrdID)	<u>Message Sent</u> (ClOrdID, OrigClOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Comment</u>
1	New Order(X)					10000				Order for 10000 sent electronically
2										Order passed verbally as there is communication problem and order does not arrive. The verbally passed order starts getting executed
3		Execution(X)	Rejected	Rejected	New	10000	0	0	0	Order finally arrives and is detected as a duplicate of a verbal order and is therefore rejected. OrdRejReason = duplicate of a verbal order

Note that the sell-side may employ a number of mechanisms to detect that the electronic order is potentially a duplicate of a verbally passed order, e.g. :

- Checking the possdup flag on the order message header
- Checking the incoming order details against other orders from the same client (e.g. side, quantity)
- Looking at the transact time on the order as a guide to 'staleness'

D24 - Order status request rejected for unknown order

<u>Time</u>	<u>Message Received</u> (ClOrdID, OrigClOrdID)	<u>Message Sent</u> (ClOrdID, OrigClOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Comment</u>
1	New Order(X)					10000				
2		Execution(X)	New	New	New	10000	0	10000	0	
3		Execution(X)	Partial Fill	Partially Filled	New	10000	1000	9000	1000	Execution for 1000
4	Status Request (Y)									
5		Execution(Y)	Rejected	Rejected	Status	0	0	0		OrdRejReason = unknown order LastShares not required when ExecTransType=Status

D25– Transmitting a CMS-style "Nothing Done" in response to a status request

<u>Time</u>	<u>Message Received</u> (ClOrdID, OrigClOrdID)	<u>Message Sent</u> (ClOrdID, OrigClOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Comment</u>
1	New Order(X)					10000				
2		Execution(X)	Rejected	Rejected	New	10000	0	0	0	If order is rejected by broker
2		Execution(X)	New	New	New	10000	0	10000	0	
3	Status Request(X)									
4		Execution(X)	New	New	Status	10000	0	10000	0	Text="Nothing Done"

D26 - Order sent, immediately followed by a status request. Subsequent status requests sent during life of order

<u>Time</u>	<u>Message Received</u> (ClOrdID, OrigClOrdID)	<u>Message Sent</u> (ClOrdID, OrigClOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Comment</u>
1	New Order(X)					10000				
2	Status Request (X)									
3		Execution(X)	Pending New	Pending New	Status	10000	0	10000		Sent in response to status request. LastShares not required when ExecTransType=status
4		Execution(X)	Rejected	Rejected	New	10000	0	0	0	If order is rejected
4		Execution(X)	New	New	New	10000	0	10000	0	
5	Status Request (X)									
6		Execution(X)	New	New	Status	10000	0	10000		Sent in response to status request
7		Execution(X)	Partial Fill	Partially Filled	New	10000	2000	8000	2000	Execution for 2000
8	Status Request (X)									
9		Execution(X)	Partial Fill	Partially Filled	Status	10000	2000	8000		Sent in response to status request

10		Execution(X)	Fill	Filled	New	10000	1000 0	0	8000	Execution for 8000
11	Status Request (X)									
12		Execution(X)	Fill	Filled	Status	10000	1000 0	0		Sent in response to status request
13	Replace Request(Y,X)					12000				Request to increase order qty
14		Execution (Y,X)	Pending Replace	Pending Replace	New	10000	1000 0	0	0	
15		Execution (Y,X)	Replace	Partially Filled	New	12000	1000 0	2000	0	
16	Status Request (X)									
17		Execution (Y,X)	Partial Fill	Partially Filled	Status	12000	1000 0	2000		Sent in response to status request. Note reference to X to allow tie back of execution report to status request
18	Status Request (Y)									
19		Execution(Y)	Partial Fill	Partially Filled	Status	12000	1000 0	2000		Sent in response to status request

D27 - GTC order partially filled, restated (renewed) and partially filled the following day

<u>Time</u>	<u>Message Received</u> (ClOrdID, OrigClOrdID)	<u>Message Sent</u> (ClOrdID, OrigClOrdID)	<u>Exec Type</u>	<u>Ord Status</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Day Order Qty</u>	<u>Day Cum Qty</u>	<u>Comment</u>
Day 1,1	New Order(X)					10000						
Day 1,2		Execution(X)	New	New	New	10000	0	10000	0			
Day 1,3		Execution(X)	Partial Fill	Partially Filled	New	10000	2000	8000	2000			Execution for 2000
Day 1,4		Execution(X)	Done for Day	Done for Day	New	10000	2000	8000	0			Optional at end of trading day
Day 2,1		Execution(X)	Restated	Partially Filled	New	10000	2000	8000	0	8000	0	ExecRestatementReason = GTC renewal/restatement (no change) – optionally sent the following morning
Day 2,2		Execution(X)	Partial Fill	Partially Filled	New	10000	3000	7000	1000	8000	1000	Execution for 1000

D28 - GTC order with partial fill, a 2:1 stock split then a partial fill and fill the following day

<u>Time</u>	<u>Message Received</u> (ClOrdID, OrigClOrdID)	<u>Message Sent</u> (ClOrdID, OrigClOrdID)	<u>Exec Type</u>	<u>Ord Status</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Day Order Qty</u>	<u>Day Cum Qty</u>	<u>Comment</u>
Day 1,1	New Order(X)					10000						
Day 1,2		Execution(X)	New	New	New	10000	0	10000	0			
Day 1,3		Execution(X)	Partial Fill	Partially Filled	New	10000	2000	8000	2000			Execution for 2000 @ 50
Day 1,4		Execution(X)	Done for Day	Done for Day	New	10000	2000	8000	0			Optional at end of trading day

Day 2,1		Execution(X)	Restated	Partially Filled	New	20000	4000	16000	0	16000	0	Sent the following morning after the split ExecRestatementReason = GTC corporate action. AvgPx=25, DayAvgPx=0
Day 2,2		Execution(X)	Partial Fill	Partially Filled	New	20000	9000	11000	5000	16000	5000	Execution for 5000
Day 2,3		Execution(X)	Fill	Filled	New	20000	20000	0	11000	16000	16000	Execution for 11000

D29 - GTC order partially filled, restated(renewed) and canceled the following day

<u>Time</u>	<u>Message Received</u> (CfOrdID, OrigCfOrdID)	<u>Message Sent</u> (CfOrdID, OrigCfOrdID)	<u>Exec Type</u>	<u>Ord Status</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Day Order Qty</u>	<u>Day Cum Qty</u>	<u>Comment</u>
Day 1,1	New Order(X)					10000						
Day 1,2		Execution(X)	New	New	New	10000	0	10000	0			
Day 1,3		Execution(X)	Partial Fill	Partially Filled	New	10000	2000	8000	2000			Execution for 2000
Day 1,4		Execution(X)	Done for Day	Done for Day	New	10000	2000	8000	0			Optional at end of trading day
Day 2,1		Execution(X)	Restated	Partially Filled	New	10000	2000	8000	0	8000	0	ExecRestatementReason = GTC renewal/restatement (no change) – optionally sent the following morning
Day 2,2	Cancel Request (Y,X)					10000						
Day 2,3		Cancel Reject (Y,X)		Partially Filled		10000						If rejected by salesperson
Day 2,3		Execution (Y,X)	Pending Cancel	Pending Cancel		10000	2000	8000	0	8000	0	
Day 2,4		Cancel Reject (Y,X)		Partially Filled		10000						If rejected by trader/exchange
Day 2,4		Execution (Y,X)	Canceled	Canceled		10000	2000	0	0	8000	0	

D30 - GTC order partially filled, restated(renewed) followed by replace request to increase quantity

<u>Time</u>	<u>Message Received</u> (ClOrdID, OrigClOrdID)	<u>Message Sent</u> (ClOrdID, OrigClOrdID)	<u>Exec Type</u>	<u>Ord Status</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Day Order Qty</u>	<u>Day Cum Qty</u>	<u>Comment</u>
Day 1,1	New Order(X)					10000						
Day 1,2		Execution(X)	New	New	New	10000	0	10000	0			
Day 1,3		Execution(X)	Partial Fill	Partially Filled	New	10000	2000	8000	2000			Execution for 2000
Day 1,4		Execution(X)	Done for Day	Done for Day	New	10000	2000	8000	0			Optional at end of trading day
Day 2,1		Execution(X)	Restated	Partially Filled	New	10000	2000	8000	0	8000	0	ExecRestatementReason = GTC renewal/restatement (no change) – optionally sent the following morning
Day 2,2	Replace Request(Y,X)					15000						Increasing qty
Day 2,3		Cancel Reject (Y,X)		Partially Filled		10000						If rejected by salesperson
Day 2,3		Execution (Y,X)	Pending Replace	Pending Replace		10000	2000	8000	0	8000	0	
Day 2,4		Execution (X)	Partial Fill	Pending Replace		10000	3000	7000	1000	8000	1000	Execution for 1000
Day 2,5		Cancel Reject (Y,X)		Partially Filled		10000						If rejected by trader/exchange
Day 2,5		Execution (Y,X)	Replace	Partially Filled		15000	3000	12000	0	13000	1000	

D31 - Poss resend order

<u>Time</u>	<u>Message Received</u> (ClOrdID, OrigClOrdID)	<u>Message Sent</u> (ClOrdID, OrigClOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Comment</u>
-------------	--	--	------------------	------------------	------------------------	------------------	----------------	-------------------	--------------------	----------------

1	New Order(X)					10000				
2		Execution(X)	New	New	New	10000	0	10000	0	
3	New Order(X)					10000				PossResend=Y
4		Execution(X)	New	New	Status	10000	0	10000		Because order X has already been received, confirm back the current state of the order. Last shares not required when ExecTransType = Status
5	New Order(Y)					15000				PossResend=Y
6		Execution(Y)	New	New	New	15000	0	15000	0	Because order Y has not been received before, confirm back as a new order.

D32 – Fill or Kill order cannot be filled

<u>Time</u>	<u>Message Received</u> (CfOrdID, OrigCfOrdID)	<u>Message Sent</u> (CfOrdID, OrigCfOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Comment</u>
1	New Order(X)					10000				Order is FOK
2		Execution(X)	Rejected	Rejected	New	10000	0	0	0	If order is rejected by broker
2		Execution(X)	New	New	New	10000	0	10000	0	
3		Execution(X)	Canceled	Canceled	New	10000	0	0	0	If order cannot be immediately filled

D33 – Immediate or Cancel order that cannot be immediately hit

<u>Time</u>	<u>Message Received</u> (CfOrdID, OrigCfOrdID)	<u>Message Sent</u> (CfOrdID, OrigCfOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Comment</u>
1	New Order(X)					10000				Order is IOC
2		Execution(X)	Rejected	Rejected	New	10000	0	0	0	If order is rejected by broker
2		Execution(X)	New	New	New	10000	0	10000	0	
3		Execution(X)	Partial Fill	Partially Filled	New	10000	1000	9000	1000	Execution for 1000
4		Execution(X)	Canceled	Canceled	New	10000	1000	0	0	If order cannot be immediately hit

D34 – Filled order, followed by correction and cancellation of executions

<u>Time</u>	<u>Message Received</u> (ClOrdID, OrigClOrdID)	<u>Message Sent</u> (ClOrdID, OrigClOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>AvgPx</u>	<u>Last Shares</u>	<u>Last Px</u>	<u>ExecId (ExecRefID)</u>	<u>Comment</u>
1	New Order(X)					10000							
2		Execution(X)	Rejected	Rejected	New	10000	0	0		0		A	If order is rejected by broker
2		Execution(X)	New	New	New	10000	0	10000	0	0		B	
3		Execution(X)	Partial Fill	Partially Filled	New	10000	1000	9000	100	1000	100	C	Execution for 1000 @ 100
4		Execution(X)	Fill	Filled	New	10000	1000 0	0	109	9000	110	D	Execution for 9000 @ 110
5		Execution(X)	Partial Fill	Partially Filled	Cancel	10000	9000	1000	110	0	0	E (C)	Cancel execution for 1000
6		Execution(X)	Partial Fill	Partially Filled	Correct	10000	9000	1000	100	9000	100	F (D)	Correct price on execution for 9000 to 100
7		Execution(X)	Fill	Filled	New	10000	1000 0	0	102	1000	120	G	Execution for 1000 @ 120
8		Execution(X)	Fill	Filled	Correct	10000	1000 0	0	120	9000	120	H(F)	Correct price on execution for 9000 to 120
9	Replace Request (Y,X)					12000							Request to increase order qty
10		Execution (Y,X)	Pending Replace	Pending Replace	New	10000	1000 0	0	120	0	0	I	
11		Execution (Y,X)	Replace	Partially Filled	New	12000	1000 0	2000	120	0	0	J	
12		Execution(Y)	Partial Fill	Partially Filled	Correct	12000	1050 0	1500	120	9500	120	K(H)	Correct execution of 9000 @ 120 to 9500 @ 120

D35 - A canceled order followed by a busted execution and a new execution

<u>Time</u>	<u>Message Received</u> (ClOrdID, OrigClOrdID)	<u>Message Sent</u> (ClOrdID, OrigClOrdID)	<u>Exec Type</u>	<u>Ord Status</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>ExecID (ExecRefID)</u>	<u>Comment</u>
-------------	--	--	------------------	-------------------	------------------------	------------------	----------------	-------------------	--------------------	---------------------------	----------------

1	New Order(X)					10000					
2		Execution(X)	New	New	New	10000	0	10000	0	A	
3		Execution(X)	Partial Fill	Partially Filled	New	10000	5000	5000	5000	B	LastPx=50
4	Cancel Request(Y,X)					10000					
5		Execution (Y,X)	Pending Cancel	Pending Cancel	New	10000	5000	5000	0	C	
6		Execution (Y,X)	Canceled	Canceled	New	10000	5000	0	0	D	
7		Execution(Y)	Partial Fill	Canceled	Cancel	10000	0	0	0	E(B)	Cancel of the execution. 'Canceled' order status takes precedence over 'New'
8		Execution(Y)	Partial Fill	Canceled	New	10000	4000	0	4000	F	Fill for 4000. LastPx=51

D36 - GTC order partially filled, restated (renewed) and partially filled the following day, with corrections of quantity on both executions

<u>Time</u>	<u>Message Received</u> (CfOrdID, OrigCfOrdID)	<u>Message Sent</u> (CfOrdID, OrigCfOrdID)	<u>Exec Type</u>	<u>Ord Status</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Day Order Qty</u>	<u>Day Cum Qty</u>	<u>ExecID (ExecRefID)</u>	<u>Comment</u>
Day 1,1	New Order(X)					10000							
Day 1,2		Execution(X)	New	New	New	10000	0	10000	0			A	
Day 1,3		Execution(X)	Partial Fill	Partially Filled	New	10000	2000	8000	2000			B	Execution for 2000
Day 1,4		Execution(X)	Done for Day	Done for Day	New	10000	2000	8000	0			C	Optional at end of trading day
Day 2,1		Execution(X)	Restated	Partially Filled	New	10000	2000	8000	0	8000	0	D	ExecRestatementReason = GTC renewal/restatement (no change) – optionally sent the following morning
Day 2,2		Execution(X)	Partial Fill	Partially Filled	New	10000	3000	7000	1000	8000	1000	E	Execution for 1000
Day 2,3		Execution(X)	Partial Fill	Partially Filled	Correct	10000	2500	7500	1500	8500	1000	F (B)	Correct quantity on previous day's execution from 2000 to 1500

Day 2,4		Execution(X)	Partial Fill	Partially Filled	Correct	10000	2000	8000	500	8500	500	G (E)	Correct quantity on today's execution from 1000 to 500
------------	--	--------------	-----------------	---------------------	---------	-------	------	------	-----	------	-----	-------	---

D37– Transmitting a guarantee of execution prior to execution

<u>Time</u>	<u>Message Received</u> (CtOrdID, OrigCtOrdID)	<u>Message Sent</u> (CtOrdID, OrigCtOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Exec Trans Type</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u>Last Shares</u>	<u>Comment</u>
1	New Order(X)					10000				
2		Execution(X)	Rejected	Rejected	New	10000	0	0	0	If order is rejected by broker
2		Execution(X)	New	New	New	10000	0	10000	0	
3		Execution(X)	Partial Fill	Stopped	New	10000	0	10000	1000	Text="You are guaranteed to buy 1000 at 50.10"; LastPx=50.10. This is similar to the concept of a 'protected' trade
4		Execution(X)	Partial Fill	Stopped	New	10000	1000	9000	1000	LastPx=50 * executed price is better than guaranteed

Appendix E

Order Handling and Instruction Semantics

London SETS Order Types Matrix

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

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

Asia/Pacific Regional Order Handling

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

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

Handling Instructions (HandlInst) field

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

1 = Automated execution order, private, no Broker

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

intervention

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

Notes:

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

2 = Automated execution order, public, Broker intervention OK

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

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

3 = Manual order, best execution

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

Notes:

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

Pegged Orders

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

L = Last peg (last sale)

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

O = Opening peg

P = Market peg

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

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

W = Peg to VWAP

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

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

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

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

“Reserve Quantity” Orders

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

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

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

Appendix F

Settlement Instructions Field Usage Matrix

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

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

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

	SettlBrkrCode SettlInstCode SecuritySettlAgentCode SecuritySettlAgentAcctNum CashSettlAgentCode CashSettlAgentAcctNum	

Appendix G

Rule80A (aka OrderCapacity) Usage by Market

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

United States Listed Equity Markets:

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

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

Valid values:

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

Japanese Equity Markets:

Used to specify whether order is Agency or Principal.

Valid values:

- A = Agency single order

P = Principal

Other Markets:

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

Appendix H

Mass Quote Message Scenarios

Unsolicited quote(s) no response requested

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

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

Unsolicited quote(s) negative response only requested

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

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

Unsolicited quote(s) full response requested

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

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

Cancel All Quotes

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

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

Appendix I

Security Definition, Security Status, and Trading Session Message Scenarios

Overview

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

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

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

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

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

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

Background

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

Definitions

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

Examples:

- Vertical Spread
- Butterfly Spread

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

Approach

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

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

The *Security Definition* message is used to:

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

Extensions to other messages

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

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

For an example:

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

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

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

Rules

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

Specifying Derivative Trading Strategies using the Security Definition message

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

Two additional pieces are needed specify the strategy.

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

Example using *RatioQty* and *Side*:

A Butterfly strategy consists of simultaneously:

Buying 1 Call at Strike Price #1

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

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

The Legs that would describe this strategy are as follows:

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

Scenarios

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

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

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

Scenario 2 - Inquire Securities Types Available

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

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

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

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

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

Scenario 4 - Inquire all securities traded by a trading party

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

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

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

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

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

Scenario 6 - Inquire list of option series for a class

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

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

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

Appendix J

Example Usage of Encoded Fields For Japanese Language Support

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

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

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

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

Precautions when using UNICODE

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

Appendix K

Example Usage of Allocations

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

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

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

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

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

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

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

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

Post-Trade Allocation supports three different message flows:

1. **Buyside initiated without Misc Fees (see examples 1-1 and 1-2)**

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

	→	Allocation (AllocTransTyp=New)	
--	---	--------------------------------	--

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

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

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

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

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

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

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

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

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

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

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

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

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

Allocation Msg



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

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

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

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



Allocation Msg

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

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

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

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

Allocation Msg



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

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

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

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

Allocation Msg



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

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

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

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

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

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

Allocation Msg



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

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

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

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

Allocation Msg

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

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

Appendix L

Pre-Trade Message Targeting/Routing

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

Targeting

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

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

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

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

Note: The ^ character represents the SOH delimiter.

Tag Explanation

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

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

Blocking

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

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

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

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

Note: The ^ character represents the SOH delimiter.

Tag Explanation

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

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

Other Issues

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

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

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

Appendix M

FIXML Support

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

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

The addition of these new fields will allow users to:

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

Appendix N

Program/Basket/List Trading

Overview

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

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

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

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

“Disclosed” model (e.g. Japanese)

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

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

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

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

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

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

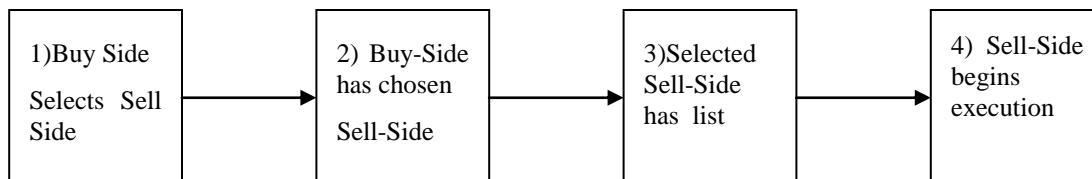
Pre-trade allocation is much more common place in the program trading community than it is in block trading. For the purposes of pre-trade allocation, a repeating group covering account and number of shares has been added to the order message. It is assumed that participants will use the either FIX allocation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

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

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

Message Flow Diagrams

Overview of logical stages



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

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

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

Order Details sent via FIX (Bidding Outside of FIX)

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

“Disclosed” Bid and Program Trade

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

“Non Disclosed” Bid and Program Trade

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

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

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

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

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

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

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

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

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

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

Buy Side Institution	←	1) Execution Reports (if requested)		Sell Side Broker Dealer
---------------------------------	---	-------------------------------------	--	------------------------------------

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

Buy Side Institution		1) List Status Request message	→	Sell Side Broker Dealer
	←	2) List Status message		

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

Buy Side Institution	←	2) List Status message		Sell Side Broker Dealer
---------------------------------	---	------------------------	--	------------------------------------

Scenario 1 Bidding performed by Telephone and List provided via FIX

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

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

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

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

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

Message	Description	Purpose
may be	Bid Request from B/S to	Details the liquidity information about the stocks that an

omitted done by phone	S/S	institution wishes to trade. It does not identify the stocks in the program.
may be omitted done by phone	Bid Response Message from S/S to B/S	Details the bid response for a program
	List Message Detail Message from B/S to S/S	Details the list of stocks that an institution wishes to trade including the stock, quantity, and direction for each order.
Required if previous provided	List Status Response	Details the status of each order in the list. The status should be <i>awaiting execution, executing or rejected</i> for each order.
may be omitted done by phone	List Execute Message from B/S to S/S	Details the bid for a program
required if previous provided	List Status Response	Details the status of each order in the list. The status should be <i>executing</i> for each order.

Illustration of liquidity indicator fields usage

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

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

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

				\$9000000
Hotels	2	7	5	1 Sec @60%, \$3000000

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

Illustration of liquidity indicator fields usage

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

List liquidity information by country.

% columns refer to percentage of average daily volume.

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

List liquidity information by Security Sector.

% columns refer to percentage of average daily volume.

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

Would be represented by the following BidRequest Message.

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

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

Notes

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

Appendix O

Foreign Exchange (F/X) Trading

Notes:

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

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

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

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

Glossary

Business Terms

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

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

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