

# FINANCIAL INFORMATION EXCHANGE PROTOCOL (FIX)

Version 4.3 with Errata 20020920

## *VOLUME 1 –INTRODUCTION TO THE FIX PROTOCOL*

Includes Errata adjustments as of September 20, 2002

### **Errata Purpose:**

This document includes a list of minor adjustments to the FIX 4.3 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 4.3 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 4.3 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 4.3 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.

~~August 24, 2001~~September 20, 2002

## **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-~~2002~~ FIX Protocol Limited, all rights reserved

## PREFACE

The Financial Information 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, brokers, and other market participants 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. FIX has grown from its original buy-side-to-sell-side equity trading roots. It is now used by markets (exchanges, "ECNs", etc) and other market participants. In addition to equities, FIX currently supports four other products: Collective Investment Vehicles (CIVs), Derivatives, Fixed Income, and Foreign Exchange. The process for modifications to the specification is very open with input and feedback encouraged from the community. Those interested in providing input to the protocol are encouraged use the FIX website Discussion section or contact the Technical Committee Chairpersons, Scott Atwell, American Century Investments, (US) 816-340-7053 ([scott\\_atwell@americancentury.com](mailto:scott_atwell@americancentury.com)) or [Dean Kauffman, TradeWeb LLC, \(US\) 201-536-5827 \(dean.kauffman@tradeweb.com\)](mailto:dean.kauffman@tradeweb.com). ~~Peter White, Goldman, Sachs & Co., (UK) 011 44 207 552 1315 (peter.white@gs.com)~~. The FIX website at <http://www.fixprotocol.org> is the main source of information, discussion, and notification of FIX-related events.

We look forward to your participation.

FIX Protocol Ltd

August 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	Baillie Gifford	Goldman Sachs (Japan) Ltd.	Capital Research

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

# VOLUME INDEX

## VOLUME 1 - INTRODUCTION

VOLUME INDEX  
INTRODUCTION  
DOCUMENT NAVIGATION  
FIX PROTOCOL SYNTAX  
COMMON COMPONENTS OF APPLICATION MESSAGES  
COMMON APPLICATION MESSAGES  
GLOSSARY

## VOLUME 2 - FIX SESSION PROTOCOL

TRANSMITTING FIXML OR OTHER XML-BASED CONTENT  
FIX MESSAGE DELIVERY  
SESSION PROTOCOL  
ADMINISTRATIVE MESSAGES  
CHECKSUM CALCULATION  
FIX SESSION USING A MULTICAST TRANSPORT  
FIX SESSION-LEVEL TEST CASES AND EXPECTED BEHAVIORS

## VOLUME 3 -FIX APPLICATION MESSAGES: PRE-TRADE

CATEGORY: INDICATION  
CATEGORY: EVENT COMMUNICATION  
CATEGORY: QUOTATION  
CATEGORY: MARKET DATA  
CATEGORY: SECURITY AND TRADING SESSION DEFINITION/STATUS

## VOLUME 4 -FIX APPLICATION MESSAGES: ORDERS AND EXECUTIONS (TRADE)

CATEGORY: SINGLE/GENERAL ORDER HANDLING  
CATEGORY: CROSS ORDERS  
CATEGORY: MULTILEG ORDERS (SWAPS, OPTION STRATEGIES, ETC)  
CATEGORY: LIST/PROGRAM/BASKET TRADING

## **VOLUME 5 - FIX APPLICATION MESSAGES: POST-TRADE**

**CATEGORY: ALLOCATION AND READY-TO-BOOK**

**CATEGORY: SETTLEMENT INSTRUCTIONS**

**CATEGORY: TRADE CAPTURE ("STREETSIDE") REPORTING**

**CATEGORY: REGISTRATION INSTRUCTIONS**

## **VOLUME 6 - FIX DATA DICTIONARY**

**FIELD DEFINITIONS**

**APPENDIX 6-A - VALID CURRENCY CODES**

**APPENDIX 6-B - FIX FIELDS BASED UPON OTHER STANDARDS**

**APPENDIX 6-C - EXCHANGE CODES - ISO 10383 MARKET IDENTIFIER CODE (MIC)**

**APPENDIX 6-D - CFICODE USAGE - ISO 10962 CLASSIFICATION OF FINANCIAL INSTRUMENTS (CFI CODE)**

**APPENDIX 6-E - DEPRECATED (PHASED-OUT) FEATURES AND SUPPORTED APPROACH**

**APPENDIX 6-F - REPLACED FEATURES AND SUPPORTED APPROACH**

**APPENDIX 6-G - USE OF <PARTIES> COMPONENT BLOCK**

## **VOLUME 7 - FIX USAGE BY PRODUCT**

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

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

**PRODUCT: EQUITIES**

**PRODUCT: FIXED INCOME**

**PRODUCT: FOREIGN EXCHANGE**

# Contents – Volume 1

<b>PREFACE</b>	<b>3</b>
<b>VOLUME INDEX</b>	<b>5</b>
<u>VOLUME 1 - INTRODUCTION</u>	5
VOLUME INDEX	5
INTRODUCTION	5
DOCUMENT NAVIGATION	5
FIX PROTOCOL SYNTAX	5
COMMON COMPONENTS OF APPLICATION MESSAGES	5
COMMON APPLICATION MESSAGES	5
GLOSSARY	5
<u>VOLUME 2 - FIX SESSION PROTOCOL</u>	5
TRANSMITTING FIXML OR OTHER XML-BASED CONTENT	5
FIX MESSAGE DELIVERY	5
SESSION PROTOCOL	5
ADMINISTRATIVE MESSAGES	5
CHECKSUM CALCULATION	5
FIX SESSION USING A MULTICAST TRANSPORT	5
FIX SESSION-LEVEL TEST CASES AND EXPECTED BEHAVIORS	5
<u>VOLUME 3 - FIX APPLICATION MESSAGES: PRE-TRADE</u>	5
CATEGORY: INDICATION	5
CATEGORY: EVENT COMMUNICATION	5
CATEGORY: QUOTATION	5
CATEGORY: MARKET DATA	5
CATEGORY: SECURITY AND TRADING SESSION DEFINITION/STATUS	5
<u>VOLUME 4 - FIX APPLICATION MESSAGES: ORDERS AND EXECUTIONS (TRADE)</u>	5
CATEGORY: SINGLE/GENERAL ORDER HANDLING	5
CATEGORY: CROSS ORDERS	5
CATEGORY: MULTILEG ORDERS (SWAPS, OPTION STRATEGIES, ETC)	5
CATEGORY: LIST/PROGRAM/BASKET TRADING	5
<u>VOLUME 5 - FIX APPLICATION MESSAGES: POST-TRADE</u>	6
CATEGORY: ALLOCATION AND READY-TO-BOOK	6
CATEGORY: SETTLEMENT INSTRUCTIONS	6
CATEGORY: TRADE CAPTURE ("STREETSIDE") REPORTING	6
CATEGORY: REGISTRATION INSTRUCTIONS	6
<u>VOLUME 6 - FIX DATA DICTIONARY</u>	6
FIELD DEFINITIONS	6
APPENDIX 6-A - VALID CURRENCY CODES	6
APPENDIX 6-B - FIX FIELDS BASED UPON OTHER STANDARDS	6
APPENDIX 6-C - EXCHANGE CODES - ISO 10383 MARKET IDENTIFIER CODE (MIC)	6
APPENDIX 6-D - CFICODE USAGE - ISO 10962 CLASSIFICATION OF FINANCIAL INSTRUMENTS (CFI CODE)	6
APPENDIX 6-E - DEPRECATED (PHASED-OUT) FEATURES AND SUPPORTED APPROACH	6
APPENDIX 6-F - REPLACED FEATURES AND SUPPORTED APPROACH	6
APPENDIX 6-G - USE OF <PARTIES> COMPONENT BLOCK	6
<u>VOLUME 7 - FIX USAGE BY PRODUCT</u>	6
PRODUCT: COLLECTIVE INVESTMENT VEHICLES (CIV)	6
PRODUCT: DERIVATIVES (FUTURES & OPTIONS)	6

PRODUCT: EQUITIES	6
PRODUCT: FIXED INCOME	6
PRODUCT: FOREIGN EXCHANGE	6
<b>INTRODUCTION</b>	<b>9</b>
<b>DOCUMENT NAVIGATION</b>	<b>9</b>
<b>FIX PROTOCOL SYNTAX</b>	<b>10</b>
COMMON FIX SYNTAX RULES	10
Data Types:	10
Required Fields:	12
FIX “Tag=Value” SYNTAX	13
Message Format	13
Field Delimiter:	13
Repeating Groups:	13
User Defined Fields:	15
Example Usage of Encoded Fields For Japanese Language Support	16
FIXML SYNTAX	17
Background	17
FIXML Highlights	17
FIXML Design Rules	18
<b>COMMON COMPONENTS OF APPLICATION MESSAGES (Included in pre-trade, trade, and post-trade messages)</b>	<b>24</b>
Instrument (symbology) component block -	24
Examples using Alternative Security Ids	27
UnderlyingInstrument (underlying instrument) component block -	28
<b>Instrument Leg Component Block</b>	<b>30</b>
Instrument Leg (symbology) component block -	30
OrderQtyData component block -	32
CommissionData component block -	33
Parties component block -	34
NestedParties component block -	35
SpreadOrBenchmarkCurveData component block -	36
Stipulations component block -	37
YieldData component block -	38
<b>COMMON APPLICATION MESSAGES (Apply to pre-trade, trade, and post-trade)</b>	<b>39</b>
Business Message Reject -	39
<b>Glossary</b>	<b>43</b>
Business Terms	43

# 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 cross-border trading, derivatives, fixed income, and other products. 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.

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 divided into volumes and organized to reflect the distinction.

## DOCUMENT NAVIGATION

One useful tip when navigating within a volume is to take advantage of the fact that each document contains “bookmarks” to its main sections. You can use the word processor’s “Goto” function (e.g. Ctrl-G) to quickly navigate from one key section or appendix to another.

Third parties or volunteers have historically built useful utilities “generated” using the specification document as their basis which provide cross-reference and lookup capabilities. Such free utilities are available via the FIX website

## FIX PROTOCOL SYNTAX

The FIX Protocol currently exists in two syntaxes:

1. "Tag=Value" syntax
2. FIXML syntax

The same business message flow applies to either syntax. A specific syntax is simply a slightly different way to represent the same thing in much the same way that "3" and "three" represent the same thing.

## COMMON FIX SYNTAX RULES

The following section summarizes general specifications for constructing FIX messages which are applicable to both "Tag=Value" and FIXML syntaxes.

### 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|

- 
- **Length:** int field (see definition of "int" above) representing the length in bytes. Value must be positive.
- **NumInGroup:** int field (see definition of "int" above) representing the number of entries in a repeating group. Value must be positive.
- **SeqNum:** int field (see definition of "int" above) representing a message sequence number. Value must be positive.
- **TagNum:** int field (see definition of "int" above) representing a field's tag number when using FIX "Tag=Value" syntax. Value must be positive and may **not** contain leading zeros.
- **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" (securities denominated in whole units) or a decimal value containing decimal places for non-share quantity asset classes (securities denominated in fractional units).

- **Price:** float field (see definition of “float” above) representing a price. Note the number of decimal places may vary. For certain asset classes prices may be negative values. For example, options strategies can be negative under certain market conditions. Refer to Volume 7: FIX Usage by Product for asset classes that support negative price values.
  - **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.
  - **Percentage:** float field (see definition of “float” above) representing a percentage (e.g. .05 represents 5% and .9525 represents 95.25%). Note the number of decimal places may vary.
- **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.
    - **Country:** String field (see definition of “String” above) representing a country using ISO 3166 Country code (2 character) values.
 

*Valid values:*

      - *See "Appendix 6-B - FIX Fields Based Upon Other Standards"*
  - **Currency:** String field (see definition of “String” above) representing a currency type using ISO 4217 Currency code (3 character) values.
 

*Valid values:*

    - *See "Appendix 6-A - Currency Codes - ISO 4217 Currency codes"*
  - **Exchange:** String field (see definition of “String” above) representing a market or exchange.
 

*Valid values:*

    - *See "Appendix 6-C - Exchange Codes - ISO 10383 Market Identifier Code (MIC)"*
  - **month-year:** String field representing month of a year in YYYYMM format. Valid values: YYYY = 0000-9999, MM = 01-12.
  - **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".*

**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.

## FIX “Tag=Value” SYNTAX

The following section summarizes general specifications for constructing FIX messages in “Tag=Value” syntax.

### 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. Tags are of data type *TagNum*. **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 MessageType (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 first field listed after the NoXXX, then becomes conditionally required if the NoXXX field is greater than zero.

- The NoXXX field (for example: NoTradingSessions, NoAllocs) which specifies the number of repeating group instances occurs once for a repeating group and must immediately precede the repeating group contents.
- The NoXXX field is required if one of the fields in the repeating group is required. If all members of a repeating group are optional, then the NoXXX field should also be optional.
- 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

Example of a repeating group:

<i>Part of message</i>				
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.
<i>Rest of the message not shown</i>				

-Example of nested repeating group

<i>Portion of New Order - List message showing a nested repeating group for allocations for each order. Note the NoAllocs repeating group is nested within the NoOrders repeating group and as such each instance of the orders repeating group may contain a repeating group of allocations.</i>				
73	NoOrders		Y	Number of orders in this message (number of repeating groups to follow)
→	11	<i>ClOrdID</i>	Y	Must be the first field in the repeating group.
→	526	<i>SecondaryClOrdID</i>	N	
→	67	<i>ListSeqNo</i>	Y	Order number within the list
→	583	<i>ClOrdLinkID</i>	N	
→	160	<i>SettInstMode</i>	N	

→	<u><i>component block &lt;Parties&gt;</i></u>		<u>N</u>	<u>Insert here the set of "Parties" (firm identification) fields defined in "COMMON COMPONENTS OF APPLICATION MESSAGES"</u>
→	<u>229</u>	<u><i>TradeOriginationDate</i></u>	<u>N</u>	
→	<u>1</u>	<u><i>Account</i></u>	<u>N</u>	
→	<u>581</u>	<u><i>AccountType</i></u>	<u>N</u>	
→	<u>589</u>	<u><i>DayBookingInst</i></u>	<u>N</u>	
→	<u>590</u>	<u><i>BookingUnit</i></u>	<u>N</u>	
→	<u>591</u>	<u><i>PreallocMethod</i></u>	<u>N</u>	
→	<u>78</u>	<u><i>NoAllocs</i></u>	<u>N</u>	<u>Indicates number of pre-trade allocation accounts to follow</u>
→	→	<u>79</u> <u><i>AllocAccount</i></u>	<u>N</u>	<u>Required if NoAllocs &gt; 0. Must be the first field in the repeating group.</u>
→	→	<u>467</u> <u><i>IndividualAllocID</i></u>	<u>N</u>	
→	→	<u><i>component block &lt;NestedParties&gt;</i></u>	<u>N</u>	<u>Insert here the set of "Nested Parties" (firm identification "nested" within additional repeating group) fields defined in "COMMON COMPONENTS OF APPLICATION MESSAGES"</u>
→	→	<u>80</u> <u><i>AllocQty</i></u>	<u>N</u>	
→	<u>63</u>	<u><i>SettlmntTyp</i></u>	<u>N</u>	
→	<u>64</u>	<u><i>FutSettDate</i></u>	<u>N</u>	<u>Takes precedence over SettlmntTyp value and conditionally required/omitted for specific SettlmntTyp values.</u>
<u><i>Rest of the message not shown</i></u>				

#### **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.

## 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
<i>... Other Standard Header fields</i>		
347	MessageEncoding	Shift_JIS
<i>... Other Standard Header fields</i>		
<i>... Other Message Body fields</i>		
106	Issuer	HITACHI
348	EncodedIssuerLen	10
349	EncodedIssuer	日立製作所
<i>... Other Message Body fields</i>		

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

### 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.

## **FIXML SYNTAX**

### **Background**

The FPL FIXML Working Group began investigating the XML format in 1998 and published a White Paper supporting an evolutionary approach to migrating the FIX Protocol to an XML format. The working group released an initial version of the FIXML DTDs on January 15th, 1999. There are currently DTDs based on FIX Protocol versions 4.1 and 4.2.

### **FIXML Highlights**

- FIXML is the XML vocabulary for creating FIX messages.
- Uses the same FIX data dictionary and business logic.
- Focuses primarily on the FIX Application Messages and does not provide a session layer.
- Can be encapsulated within the FIX Session Protocol or within another protocol like SOAP.

This document incorporates FIXML in two distinct ways:

- 1) A corresponding DTD fragment supports each message definition
- 2) Each item in the data dictionary has a corresponding DTD equivalent.

Note: while this document and the DTD are relatively in sync, the DTD will contain the full FIXML definitions.

FIXML 4.3 will also eventually be supplemented by an XML Schema version, which has recently been approved as a W3C Recommendation.

## FIXML Design Rules

- 1) Elements can contain other Elements, EMPTY content or PCDATA (text) content.
- 2) FIXML uses camel case notation in which elements and attributes may be made up of multiple words with each word beginning with a capital letter.
- 3) Certain commonly used and well-known acronyms like IOI and DK are capitalized and separated from the rest of the tag by an underscore. (i.e. IOI\_Qty).
- 4) FIXML requires ordered content model. This differs from the traditional FIX approach (“tag=value” syntax) which allows fields to be in any order (other than the first couple and last).
- 5) FIXML supports conditionally required content models. Options must contain a Strike Price.

```
<!ELEMENT Option (StrikePrice, OptAttribute?)>
<!ATTLIST Option FIXTag CDATA #FIXED '167'
           DataType CDATA #FIXED 'String'
           Value CDATA #FIXED 'OPT' >
```

- 6) Content models of business messages contain entities that allow for customization. For example, all application messages have a custom entity that can be redefined to extend the content model of the particular message. The following illustrates the ListExecute message:

```
<!ENTITY % ListExecuteCustom "">
<!ENTITY % ListExecuteContent "ListID,ClientBidID?,BidID?,TransactTime,Text?,EncodedTextGroup?
%ListExecuteCustom;" >
<!ELEMENT ListExecute (%ListExecuteContent);>
<!ATTLIST ListExecute FIXTag CDATA #FIXED '35'
           DataType CDATA #FIXED 'String'
           Value CDATA #FIXED 'L' >
```

To extend the content model of the ListExecute message, add the following to the internal subset of a FIXML message.

```
<!DOCTYPE fixml SYSTEM "fixmlmain.dtd" [
  <!ENTITY % ListExecuteCustom ", InternalTransNumber?">
  <!ELEMENT InternalTransNumber (#PCDATA)>
```

]>

After entity reference resolution the Indication content model will look like:

```
<!ELEMENT ListExecute (ListID,ClientBidID?,BidID?,TransactTime,Text?,EncodedTextGroup?,
InternalTransNumber?)>
```

instead of

```
<!ELEMENT ListExecute (ListID,ClientBidID?,BidID?,TransactTime,Text?,EncodedTextGroup? )>
```

- 7) FIXML elements have attributes, which contain referential information related to the FIX Field ID, Data type, and numeric constraints. **Validation of these attributes must happen at the application level.**

**FIXTag** - contains the FIX Protocol Field ID (Tag).

**DataType** - reflects data types (char, int, float, month-year, day-of-month, time, date) from the FIX specification.

Example:

```
<!ELEMENT ForexReq EMPTY>
<!ATTLIST ForexReq FIXTag CDATA #FIXED '121'
          DataType CDATA #FIXED 'Boolean'
          Value (Y|N) #REQUIRED
          SDValue (Yes|No) #IMPLIED >
```

- 8) FIX defines message types with the **MsgType** field (tag "35"). Since the existence of a particular element indicates the message type (ie <ExecutionReport>), **MsgType** is reflected as meta-data information. Each FIX message contains the attribute **FIXTag** with a fixed value equal to "35" and a **Value** attribute equal to the corresponding **MsgType** value.

```
<!ELEMENT QuoteReq (%QuoteReqContent; )>
```

```

<!ATTLIST QuoteReq
  FIXTag CDATA #FIXED "35"
  DataType CDATA #FIXED "char"
  Value CDATA #FIXED "R"
>

```

- 9) FIXML allows for the XML parser to validate enumerations from the FIX Specification. These elements are defined with EMPTY content models and an attribute called Value. The acceptable values for FIXML attribute enumerations come from the FIX Specification. **An optional attribute list call SDValue (SelfDescribingValue) contains the human-readable equivalent of the FIX specification values.**

```

<!ELEMENT ProcessCode EMPTY>
<!ATTLIST ProcessCode FIXTag CDATA #FIXED '81'
  DataType CDATA #FIXED 'char'
  Value (0|1|2|3|4|5|6) #REQUIRED
  SDValue (Regular |
           SoftDollar |
           StepIn |
           StepOut |
           StepInSoft |
           StepOutSoft |
           PlanSponsor) #IMPLIED >

```

The linkage between Value and SDValue cannot be validated.

- 10) When fields are conditionally required based on the value of other fields, the Tag=Value pair becomes an element. For example, ExecRefID is required when ExecTransType = Cancel. The attribute **Value** is added and contains the valid FIX Specification value.

```

<!ELEMENT ExecTransType (ExecNew | ExecCancel | ExecCorrect | ExecStatus)>

<!ELEMENT ExecCancel (ExecRefID, LastQty, LastPx)>
  <!ATTLIST ExecCancel
    FIXTag CDATA #FIXED "20"

```

Value CDATA #FIXED "1">

>

20=1 (ExecTransType=Cancel)

becomes

<ExecTransType><ExecNew FIXTag="20" Value="1"> ... </ExecTransType>

Applies to:

ExecNew, ExecCancel, ExecCorrect, ExecStatus, AllocStatusAccept, AllocStatusReject, AllocPartialAccept, AllocStatusReceived, AdvNew, AdvCancel, AdvReplace, IOINew, IOICancel, IOIReplace

11) FIXML has elements that serve as containers and do not map directly to FIX tag=value pairs.

<!ELEMENT MiscFeeList (NoMiscFees? , MiscFeeGroup+)>

12) Special containers are used when enumeration values of a FIX field must be split into two elements to handle conditionally required elements.

<!ELEMENT OrderDuration (TimeInForce | GTDTimeInForce)>

<!ELEMENT TimeInForce EMPTY>

<!ATTLIST TimeInForce

FIXTag CDATA #FIXED "59"

DataType CDATA #FIXED "char"

Value (0|1|2|3|4|5) #REQUIRED

SDValue (Day|GoodTillCancel|AtTheOpening|ImmediateOrCancel|FillOrKill|

GoodTillCrossing) #IMPLIED

>

<!ELEMENT GTDTimeInForce (ExpireTime)>

<!ATTLIST GTDTimeInForce

```
FIXTag CDATA #FIXED "59"  
DataType CDATA #FIXED "char"  
Value CDATA #FIXED "6"  
SDValue CDATA #FIXED "GoodTillDate" >
```

- 13) Certain FIX Fields are grouped into parent/child relationships. Referential information is contained in two places. The attribute **FIXTags** contains a list of valid tags in the content model and each field has its own attribute.

```
<!ELEMENT Sender (CompID, SubID?, LocationID?)>  
<!ELEMENT CompID (#PCDATA)>  
<!ATTLIST CompID  
FIXTag CDATA #FIXED "49-56-115-128"  
SenderFIXTag CDATA #FIXED "49"  
TargetFIXTag CDATA #FIXED "56"  
OnBehalfOfFIXTag CDATA #FIXED "115"  
DeliverToFIXTag CDATA #FIXED "128"  
DataType CDATA #FIXED "char">
```

For example:

49=ssmb

becomes

```
<Sender><CompID SenderFIXTag="49">ssmb</CompID></Sender>
```

Applies to:

Sender, Target, Location, OnBehalfOf, DeliverTo

- 14) FIX repeating groups are supported through the use of collection elements. To support conversions between FIX and FIXML, fields that identify the number of repeating elements are contained in the content model of the collection element.

```
<!ELEMENT BidComponentList (NoBidComponents? , BidComponentGroup+)>  
<!ELEMENT BidComponentGroup ( ListID?, Side?,TradingSessionID?,  
TradingSessionSubID?,NetGrossInd?, Settlement?, Account? )>
```



## COMMON COMPONENTS OF APPLICATION MESSAGES (Included in pre-trade, trade, and post-trade messages)

Many of the FIX Application Messages are composed of common "building blocks" or sets of data fields. For instance, almost every FIX Application Message has the set of symbology-related fields used to define the "Instrument": Symbol, SymbolSfx, SecurityIDSource, SecurityID..... EncodedSecurityDesc. Rather than replicate a common group of fields, the FIX specification specifies several key component blocks below which are simply referenced by component name within each Application Message which uses them. Thus when reviewing a specific message definition, the appropriate group of fields should be expanded and used whenever a component block is identified.

Note that some component blocks may be part of repeating groups thus if the component block is denoted as part of a repeating group, then the entire group of fields representing the component block are to be specified at the component block's repeating group "level" in the message definition and follow repeating group rules concerning field order. See "Repeating Groups" for more details.

### Instrument (symbology) component block -

| <Instrument> |                  |                            |       |  |
|--------------|------------------|----------------------------|-------|--|
| Tag          | Field Name       |                            | Req'd | Comments   |
| 55           | Symbol           |                            | ***   | Common, "human understood" representation of the security. SecurityID value can be specified if no symbol exists (e.g. non-exchange traded Collective Investment Vehicles)   |
| 65           | SymbolSfx        |                            | N     |  |
| 48           | SecurityID       |                            | N     | Takes precedence in identifying security to counterparty over SecurityAltID block. Requires SecurityIDSource if specified.   |
| 22           | SecurityIDSource |                            | N     | Required if SecurityID is specified.   |
| 454          | NoSecurityAltID  |                            | N     | Number of alternate Security Identifiers   |
| →            | 455              | <i>SecurityAltID</i>       | N     | Security Alternate identifier for this security<br>First member of repeating group - must be specified if NoSecurityAltID > 0<br>The Security Alternative identifier block should not be populated unless SecurityID and SecurityIDSource are populated and should not duplicate the SecurityID and SecurityIDSource values contained in the SecurityID/SecurityIDSource tags. Use of SecurityAltID may be used if bilaterally agreed to assist in security identification, and does not imply an obligation on the receiver of the message to ensure validity or consistency with the SecurityID and SecurityIDSource fields which take precedence. |
| →            | 456              | <i>SecurityAltIDSource</i> | N     | Source of SecurityAltID. Required if SecurityAltID is specified.   |
| 460          | Product          |                            | N     | Indicates the type of product the security is associated with (high-level category)  |

|     |                            |   |   |
|-----|----------------------------|---|---|
| 461 | CFICode                    | N | Indicates the type of security using ISO 10962 standard, Classification of Financial Instruments (CFI code) values. It is recommended that CFICode be used instead of SecurityType for non-Fixed Income instruments.  |
| 167 | SecurityType               | N | It is recommended that CFICode be used instead of SecurityType for non-Fixed Income instruments.<br><br>Futures and Options should be specified using the CFICode[461] field instead of SecurityType[167] (Refer to Volume 7 – Recommendations and Guidelines for Futures and Options Markets.”)  |
| 200 | MaturityMonthYear          | N | Specifies the month and year of maturity. Applicable for standardized derivatives which are typically only referenced by month and year (e.g. S&P futures). Note MaturityDate (a full date) can also be specified.  |
| 541 | MaturityDate               | N | Specifies date of maturity (a full date). Note that standardized derivatives which are typically only referenced by month and year (e.g. S&P futures).may use MaturityMonthYear and/or this field.<br><br>When using MaturityMonthYear, it is recommended that markets and sell sides report the MaturityDate on all outbound messages as a means of data enrichment. |
|     |                            |   |   |
| 224 | CouponPaymentDate          | N | Date interest is to be paid. Used in identifying Corporate Bond issues.   |
| 225 | IssueDate                  | N | Date instrument was issued. For Fixed Income IOIs for new issues, specifies the issue date.   |
| 239 | RepoCollateralSecurityType | N | Identifies the collateral used in the transaction. For Fixed Income, required for RP and RVRP security types.   |
| 226 | RepurchaseTerm             | N | Number of business days before repurchase of a repo.  |
| 227 | RepurchaseRate             | N | Percent of par at which a Repo will be repaid. Represented as a percent, e.g. .9525 represents 95-1/4 percent of par.   |
| 228 | Factor                     | N | Fraction for deriving Current face from Original face for TIPS, ABS or MBS Fixed Income securities. Note the fraction may be greater than, equal to or less than 1.   |
| 255 | CreditRating               | N |   |
| 543 | InstrRegistry              | N | The location at which records of ownership are maintained for this instrument, and at which ownership changes must be recorded. Can be used in conjunction with ISIN to address ISIN uniqueness issues.   |
| 470 | CountryOfIssue             | N | ISO Country code of instrument issue (e.g. the country portion typically used in ISIN). Can be used in conjunction with non-ISIN SecurityID (e.g. CUSIP for Municipal Bonds without ISIN) to provide uniqueness.  |
| 471 | StateOrProvinceOfIssue     | N | A two-character state or province abbreviation.   |

|               |                        |   |  |
|---------------|------------------------|---|--|
| 472           | LocaleOfIssue          | N | The three-character IATA code for a locale (e.g. airport code for Municipal Bonds).  |
| 240           | RedemptionDate         | N | Return of investor's principal in a security. Bond redemption can occur before maturity date.  |
|               |                        |   |  |
| 202           | StrikePrice            | N | Used for derivatives, such as options and covered warrants   |
| 206           | OptAttribute           | N | Used for derivatives, such as options and covered warrants to indicate a versioning of the contract when required due to corporate actions to the underlying. Should not be used to indicate type of option – use the CFICode[461] for this purpose. |
| 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.   |
| </Instrument> |                        |   |  |

\*\*\* = Required status should match "Req'd" setting for <Instrument> component block in the message definition

### Examples using Alternative Security Ids

The first example is from an order for shares in Daimler Chrysler, which has an ISIN DE0007100000, a CUSIP D1668R123, and a Sedol 5529027

| Field (tag)                 | Value        | Explanation                           |
|-----------------------------|--------------|---------------------------------------|
| Symbol (55)                 | DCX          | Symbol = DCX (Daimler Chrysler)       |
| SecurityID (48)             | DE0007100000 |                                       |
| SecurityIDSource (22)       | 4            | ID Type is ISIN                       |
| NoSecurityAltID (454)       | 2            | Two additional security IDs specified |
| → SecurityAltID (455)       | D1668R123    |                                       |
| → SecurityAltIDSource (456) | 1            | SecurityID type is Cusip              |
| → SecurityAltID (455)       | 5529027      |                                       |
| → SecurityAltIDSource (456) | 2            | SecurityID type is Sedol              |

The second example is from an order for shares in IBM, which has an ISIN US4592001014, and a QUICK (Japanese) code of 000006680

| Field (tag)                 | Value        | Explanation                                    |
|-----------------------------|--------------|--|
| Symbol (55)                 | IBM          | Symbol = IBM (International Business Machines) |
| SecurityID (48)             | US4592001014 |  |
| SecurityIDSource (22)       | 4            | ID Type is ISIN                                |
| NoSecurityAltID (454)       | 1            | One additional security ID specified           |
| → SecurityAltID (455)       | 000006680    |  |
| → SecurityAltIDSource (456) | 3            | SecurityID type is Quick                       |

## UnderlyingInstrument (underlying instrument) component block -

*Refer to the Instrument component block comments as this component block mirrors Instrument.*

| <UnderlyingInstrument> |                                      |       |          |
|------------------------|--------------------------------------|-------|----------|
| Tag                    | Field Name                           | Req'd | Comments |
| 311                    | UnderlyingSymbol                     | ***   |          |
| 312                    | UnderlyingSymbolSfx                  | N     |          |
| 309                    | UnderlyingSecurityID                 | N     |          |
| 305                    | UnderlyingSecurityIDSource           | N     |          |
| 457                    | NoUnderlyingSecurityAltID            | N     |          |
| →                      | 458 UnderlyingSecurityAltID          | N     |          |
| →                      | 459 UnderlyingSecurityAltIDSource    | N     |          |
| 462                    | UnderlyingProduct                    | N     |          |
| 463                    | UnderlyingCFICode                    | N     |          |
| 310                    | UnderlyingSecurityType               | N     |          |
| 313                    | UnderlyingMaturityMonthYear          | N     |          |
| 542                    | UnderlyingMaturityDate               | N     |          |
|                        |                                      |       |          |
| 241                    | UnderlyingCouponPaymentDate          | N     |          |
| 242                    | UnderlyingIssueDate                  | N     |          |
| 243                    | UnderlyingRepoCollateralSecurityType | N     |          |
| 244                    | UnderlyingRepurchaseTerm             | N     |          |
| 245                    | UnderlyingRepurchaseRate             | N     |          |
| 246                    | UnderlyingFactor                     | N     |          |
| 256                    | UnderlyingCreditRating               | N     |          |
| 595                    | UnderlyingInstrRegistry              | N     |          |
| 592                    | UnderlyingCountryOfIssue             | N     |          |
| 593                    | UnderlyingStateOrProvinceOfIssue     | N     |          |
| 594                    | UnderlyingLocaleOfIssue              | N     |          |

|                           |                                  |              |  |
|---------------------------|----------------------------------|--------------|--|
| 247                       | UnderlyingRedemptionDate         | N            |  |
| <del>315</del>            | <del>UnderlyingPutOrCall</del>   | <del>N</del> |  |
| 316                       | UnderlyingStrikePrice            | N            |  |
| 317                       | UnderlyingOptAttribute           | N            |  |
| 436                       | UnderlyingContractMultiplier     | N            |  |
| 435                       | UnderlyingCouponRate             | N            |  |
| 308                       | UnderlyingSecurityExchange       | N            |  |
| 306                       | UnderlyingIssuer                 | N            |  |
| 362                       | EncodedUnderlyingIssuerLen       | N            |  |
| 363                       | EncodedUnderlyingIssuer          | N            |  |
| 307                       | UnderlyingSecurityDesc           | N            |  |
| 364                       | EncodedUnderlyingSecurityDescLen | N            |  |
| 365                       | EncodedUnderlyingSecurityDesc    | N            |  |
| </ UnderlyingInstrument > |                                  |              |  |

\*\*\* = Required status should match "Req'd" setting for <UnderlyingInstrument> component block in the message definition

## Instrument Leg Component Block

*Refer to the Instrument component block comments as this component block mirrors Instrument.*

Several multileg-oriented messages specify an instrument leg component block. An instrument can have zero or more instrument legs. The fundamental business rule that applies to the multileg instrument is that the multileg instrument is defined as the combination of instrument legs. The multileg instrument must be able to be traded atomically – that all instrument legs are traded or none are traded.

The LegRatioQty[623] is used to define the quantity of the leg that makes up a single unit of the multileg instrument. An option butterfly strategy is made up of three option legs.

### Instrument Leg (symbology) component block -

| <InstrumentLeg> |  |       |          |
|-----------------|--|-------|----------|
| Tag             | Field Name                               | Req'd | Comments |
| 600             | LegSymbol                                | ***   |          |
| 601             | LegSymbolSfx                             | N     |          |
| 602             | LegSecurityID                            | N     |          |
| 603             | LegSecurityIDSource                      | N     |          |
| 604             | NoLegSecurityAltID                       | N     |          |
| →               | <b>605</b> <i>LegSecurityAltID</i>       | N     |          |
| →               | <b>606</b> <i>LegSecurityAltIDSource</i> | N     |          |
| 607             | LegProduct                               | N     |          |
| 608             | LegCFIcode                               | N     |          |
| 609             | LegSecurityType                          | N     |          |
| 610             | LegMaturityMonthYear                     | N     |          |
| 611             | LegMaturityDate                          | N     |          |
| 248             | LegCouponPaymentDate                     | N     |          |
| 249             | LegIssueDate                             | N     |          |
| 250             | LegRepoCollateralSecurityType            | N     |          |
| 251             | LegRepurchaseTerm                        | N     |          |
| 252             | LegRepurchaseRate                        | N     |          |
| 253             | LegFactor                                | N     |          |
| 257             | LegCreditRating                          | N     |          |
| 599             | LegInstrRegistry                         | N     |          |
| 596             | LegCountryOfIssue                        | N     |          |

|                  |                           |   |   |
|------------------|---------------------------|---|---|
| 597              | LegStateOrProvinceOfIssue | N |   |
| 598              | LegLocaleOfIssue          | N |   |
| 254              | LegRedemptionDate         | N |   |
| 612              | LegStrikePrice            | N |   |
| 613              | LegOptAttribute           | N |   |
| 614              | LegContractMultiplier     | N |   |
| 615              | LegCouponRate             | N |   |
| 616              | LegSecurityExchange       | N |   |
| 617              | LegIssuer                 | N |   |
| 618              | EncodedLegIssuerLen       | N |   |
| 619              | EncodedLegIssuer          | N |   |
| 620              | LegSecurityDesc           | N |   |
| 621              | EncodedLegSecurityDescLen | N |   |
| 622              | EncodedLegSecurityDesc    | N |   |
| 623              | LegRatioQty               | N | Specific to the <InstrumentLeg> (not in <Instrument>) |
| 624              | LegSide                   | N | Specific to the <InstrumentLeg> (not in <Instrument>) |
| </InstrumentLeg> |                           |   |   |

\*\*\* = Required status should match "Req'd" setting for <OrderQtyData> component block in message definition

**OrderQtyData component block -**

| <i>&lt;OrderQtyData&gt;</i>  |                   |              |  |
|------------------------------|-------------------|--------------|--|
| <i>Tag</i>                   | <i>Field Name</i> | <i>Req'd</i> | <i>Comments</i>  |
| 38                           | OrderQty          | N            | One of CashOrderQty, OrderQty, or (for CIV only) OrderPercent is required. Note that unless otherwise specified, only one of CashOrderQty, OrderQty, or OrderPercent should be specified.  |
| 152                          | CashOrderQty      | N            | One of CashOrderQty, OrderQty, or (for CIV only) OrderPercent is required. Note that unless otherwise specified, only one of CashOrderQty, OrderQty, or OrderPercent should be specified. Specifies the approximate "monetary quantity" for the order. Broker is responsible for converting and calculating OrderQty in tradeable units (e.g. shares) for subsequent messages. |
| 516                          | OrderPercent      | N            | For CIV - Optional. One of CashOrderQty, OrderQty or (for CIV only) OrderPercent is required. Note that unless otherwise specified, only one of CashOrderQty, OrderQty, or OrderPercent should be specified.   |
| 468                          | RoundingDirection | N            | For CIV – Optional   |
| 469                          | RoundingModulus   | N            | For CIV – Optional   |
| <i>&lt;/OrderQtyData&gt;</i> |                   |              |  |

**\*\*\* = Required status should match "Req'd" setting for <OrderQtyData> component block in message definition**

**CommissionData component block -**

| <CommissionData>  |                   |              |                    |
|-------------------|-------------------|--------------|--------------------|
| <i>Tag</i>        | <i>Field Name</i> | <i>Req'd</i> | <i>Comments</i>    |
| 12                | Commission        | N            |                    |
| 13                | CommType          | N            |                    |
| 479               | CommCurrency      | N            | For CIV - Optional |
| 497               | FundRenewWaiv     | N            | For CIV - Optional |
| </CommissionData> |                   |              |                    |

**\*\*\* = Required status should match "Req'd" setting for <CommissionData> component block in message definition**

**Parties component block -**

*See “Volume 6 - APPENDIX 6-G - USE OF <PARTIES> COMPONENT BLOCK”.*

| <Parties>  |                          |       |  |
|------------|--------------------------|-------|--|
| Tag        | Field Name               | Req'd | Comments   |
| 453        | NoPartyIDs               | N     | Repeating group below should contain unique combinations of PartyID, PartyIDSource, and PartyRole                        |
| →          | 448 <i>PartyID</i>       | N     | Used to identify source of PartyID. Required if PartyIDSource is specified. Required if NoPartyIDs > 0.                  |
| →          | 447 <i>PartyIDSource</i> | N     | Used to identify class source of PartyID value (e.g. BIC). Required if PartyID is specified. Required if NoPartyIDs > 0. |
| →          | 452 <i>PartyRole</i>     | N     | Identifies the type of PartyID (e.g. Executing Broker). Required if NoPartyIDs > 0.                                      |
| →          | 523 <i>PartySubID</i>    | N     | Sub-identifier (e.g. Clearing Acct for PartyID=Clearing Firm) if applicable  |
| </Parties> |                          |       |  |

**\*\*\* = Required status should match "Req'd" setting for <Parties> component block in message definition**

## NestedParties component block -

| <NestedParties>  |                  |                            |       |  |
|------------------|------------------|----------------------------|-------|--|
| Tag              | Field Name       |                            | Req'd | Comments   |
| 539              | NoNestedPartyIDs |                            | N     | Repeating group below should contain unique combinations of NestedPartyID, NestedPartyIDSource, and NestedPartyRole                        |
| →                | 524              | <i>NestedPartyID</i>       | N     | Used to identify source of NestedPartyID. Required if NestedPartyIDSource is specified. Required if NoNestedPartyIDs > 0.                  |
| →                | 525              | <i>NestedPartyIDSource</i> | N     | Used to identify class source of NestedPartyID value (e.g. BIC). Required if NestedPartyID is specified. Required if NoNestedPartyIDs > 0. |
| →                | 538              | <i>NestedPartyRole</i>     | N     | Identifies the type of NestedPartyID (e.g. Executing Broker). Required if NoNestedPartyIDs > 0.  |
| →                | 545              | <i>NestedPartySubID</i>    | N     | Sub-identifier (e.g. Clearing Acct for NestedPartyID=Clearing Firm) if applicable  |
| </NestedParties> |                  |                            |       |  |

\*\*\* = Required status should match "Req'd" setting for <NestedParties> component block in message definition

**SpreadOrBenchmarkCurveData component block -**

| <SpreadOrBenchmarkCurveData>  |                        |              |                  |
|-------------------------------|------------------------|--------------|------------------|
| <i>Tag</i>                    | <i>Field Name</i>      | <i>Req'd</i> | <i>Comments</i>  |
| 218                           | Spread                 | N            | For Fixed Income |
| 220                           | BenchmarkCurveCurrenty | N            |                  |
| 221                           | BenchmarkCurveName     | N            |                  |
| 222                           | BenchmarkCurvePoint    | N            |                  |
| </SpreadOrBenchmarkCurveData> |                        |              |                  |

**\*\*\* = Required status should match "Req'd" setting for <SpreadOrBenchmarkCurveData> component block in message definition**

**Stipulations component block -**

| <Stipulations>  |                   |                         |              |                               |
|-----------------|-------------------|-------------------------|--------------|-------------------------------|
| <i>Tag</i>      | <i>Field Name</i> |                         | <i>Req'd</i> | <i>Comments</i>               |
| 232             | NoStipulations    |                         | N            |                               |
| →               | 233               | <i>StipulationType</i>  | N            | Required if NoStipulations >0 |
| →               | 234               | <i>StipulationValue</i> | N            |                               |
| </Stipulations> |                   |                         |              |                               |

**\*\*\* = Required status should match "Req'd" setting for <Stipulations> component block in message definition**

**YieldData component block -**

| <i>&lt;YieldData&gt;</i>  |                   |              |                 |
|---------------------------|-------------------|--------------|-----------------|
| <i>Tag</i>                | <i>Field Name</i> | <i>Req'd</i> | <i>Comments</i> |
| 235                       | YieldType         | N            |                 |
| 236                       | Yield             | N            |                 |
| <i>&lt;/YieldData&gt;</i> |                   |              |                 |

**\*\*\* = Required status should match "Req'd" setting for <YieldData> component block in message definition**

## COMMON APPLICATION MESSAGES (Apply to pre-trade, trade, and post-trade)

### 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:

| Situation   | Appropriate Response                             |
|---|--|
| 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"> <li>Quote Request</li> </ul>   | Use the Quote Request Reject message             |
| In response to: <ul style="list-style-type: none"> <li>Quote</li> <li>Quote Cancel</li> <li>Quote Status Request</li> </ul> | Use the Quote Status Report message              |
| In response to: <ul style="list-style-type: none"> <li>Mass Quote</li> <li></li> </ul>                                      | Use the Mass Quote Acknowledgment message        |
| In response to: <ul style="list-style-type: none"> <li>Market Data Request</li> </ul>                                       | Use the Market Data Request Reject message       |
| In response to: <ul style="list-style-type: none"> <li>Security Definition Request</li> </ul>                               | Use the Security Definition message              |
| In response to: <ul style="list-style-type: none"> <li>Security Type Request</li> </ul>                                     | Use the SecurityTypes message                    |
| In response to: <ul style="list-style-type: none"> <li>Security List Request</li> </ul>                                     | Use the Security List message                    |
| In response to: <ul style="list-style-type: none"> <li>Derivative Security List Request</li> </ul>                          | Use the Derivative Security List message         |
| In response to: <ul style="list-style-type: none"> <li>Security Status Request</li> </ul>                                   | Use the Security Status message                  |
| In response to: <ul style="list-style-type: none"> <li>Trading Session Status Request</li> </ul>                            | Use the Trading Session Status message           |
| In response to: <ul style="list-style-type: none"> <li>New Order - Single</li> </ul>  | Use the Execution Report message                 |

|   |  |
|---|--|
| <ul style="list-style-type: none"> <li>• Order Status Request</li> <li>• Order Mass Status Request</li> <li>• New Order – Cross</li> <li>• New Order – Multileg</li> <li>• New Order – List</li> <li>• List Execute</li> </ul>  |  |
| <p>In response to:</p> <ul style="list-style-type: none"> <li>• Order Cancel Request</li> <li>• Order Cancel/Replace Request</li> <li>• Cross Order Cancel Request</li> <li>• Cross Order Cancel/Replace Request</li> <li>• Multileg Order Cancel/Replace Request</li> <li>• List Cancel Request</li> </ul> | Use the Order Cancel Reject message                |
| <p>In response to:</p> <ul style="list-style-type: none"> <li>• Execution Report</li> </ul>   | Use the Don't Know Trade (DK) message              |
| <p>In response to:</p> <ul style="list-style-type: none"> <li>• Order Mass Cancel Request</li> </ul>  | Use the Order Mass Cancel Report message           |
| <p>In response to:</p> <ul style="list-style-type: none"> <li>• List Status Request</li> </ul>  | Use the List Status message                        |
| <p>In response to:</p> <ul style="list-style-type: none"> <li>• Allocation</li> </ul>   | Use the Allocation ACK message                     |
| <p>In response to:</p> <ul style="list-style-type: none"> <li>• Registration Instructions</li> </ul>  | Use the Registration Instructions Response message |
| <p>In response to:</p> <ul style="list-style-type: none"> <li>• Trade Capture Report Request</li> </ul>   | Use the Trade Capture Report Request message       |

**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:

|  |
|--|
| <p>(the “ID” field BusinessRejectRefID refers to noted in [ ])</p> <ul style="list-style-type: none"> <li>• Indication of Interest (IOI) [IOIid]</li> <li>• Advertisement [AdvId]</li> </ul> |
|--|

- News [Headline]
- Email [EmailThreadID]
- Order Cancel Reject [COrdID]
- 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]
- Order Mass Cancel Report [OrderID]
- Security Types [SecurityResponseID]
- Security List [SecurityResponseID]
- Derivative Security List [SecurityResponseID]
- Quote Request Reject [QuoteReqID]
- RFQ Request [RFQReqID]
- Quote Status Report [QuoteID]
- Registration Instructions Response [RegistID]
- Trade Capture Report [TradeReportID]

Scenarios for Business Message Reject:

| <b>BusinessRejectReason</b>   |
|---|
| 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:

| <b>Business Message Reject</b> |                   |              |                 |
|--------------------------------|-------------------|--------------|-----------------|
| <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 |  |

## 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.

| Term                                  | Definition   | Field where used |
|---------------------------------------|--|------------------|
| 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]       |
| As defined                            | Sides of the legs are the same as defined in the multileg instrument.  | [Side]           |
| At the close                          | Indicated price is to be around the closing price, however, not held to the closing price.   | [IOIQualifier]   |
| At the Opening                        | A market or limit-price order to be executed at the opening of the stock or not at all; all or part of any order not executed at the opening is treated as canceled.   | [TimeInForce]    |
| Basis Price                           | A price established by joint agreement of odd-lot dealers in 100-share-unit stocks when: <ul style="list-style-type: none"> <li>- no round-lot has occurred during the trading session,</li> <li>- the spread between the closing bid and offer is two points or more, and</li> <li>- on odd-lot the dealer has been given a "basis-price" order.</li> </ul>   | [OrdType]        |
| Broker of Credit                      | Broker to receive trade credit.  | [PartyRole]      |
| Buy Minus                             | A round-lot market order to buy "minus" is an order to buy a stated amount of a stock provided that its price is: <ul style="list-style-type: none"> <li>- not higher than the last sale if the last sale was a "minus" or "zero minus" tick and</li> <li>- not higher than the last sale minus the minimum fractional change in the stock if the last sale was a "plus" or "zero plus" tick.</li> </ul> A limit price order to buy "minus" also states the highest price at which it can be executed. | [Side]           |
| Call First                            | Refer to client before trading.  | [ExecInst]       |
| Cancel on System Failure              | If a system failure interrupts trading or order routing, attempt to cancel this order. Note that depending on the type and severity of the failure, this might not be possible.  | [ExecInst]       |
| Cancel on Trading Halt                | If trading in this instrument is halted, cancel this order and do not reinstate it when/if trading resumes.  | [ExecInst]       |
| CIV ("Collective Investment Vehicle") | Collective investment vehicle ("CIV") are set up for the purposes of collecting and pooling investor funds and issuing shares (or their  | [CFICode]<br>and |

|                             |  |                             |
|-----------------------------|--|-----------------------------|
|                             | <p>equivalent). "Open-ended" CIVs entitle the holder to receive, on demand, an amount in value which is proportionate to the whole net asset value of the vehicle. Conversely "Closed-ended" CIVs do not grant this right to investors.</p> <p>CIVs are more commonly known as Mutual Funds, Unit Trusts, OEICS (Open Ended Investment Companies), SICAVs etc.</p> <p>A CIV may be legally constituted as a Limited Company with variable capital, a Trust or a Limited Partnership - depending on local legislation &amp; tax regimes.</p> <p>CIVs typically invest in equities, bonds, derivatives etc. - as described in their prospectus. Other CIVs are Umbrella Fund (made up of sub-funds investing in equities, gilts etc), Fund Of Funds (invests only in other funds), Master-Feeder Fund (marketed to a specific group for investment in a central master fund), Multi-Manager Fund (whose asset management is divided between several managers), Side By Side (onshore and offshore funds with the same investment strategy)</p> | a "Product" within Volume 7 |
| Clearing Firm               | Firm that will clear the trade. Used if different from the executing firm.   | [PartyRole]                 |
| Client ID                   | Firm identifier used in third party-transactions or for investor identification in intermediary transactions. (should not be a substitute for OnBehalfOfCompID/DeliverToCompID).   | [PartyRole]                 |
| Contra Firm                 | The broker or other firm which is the contra side of the trade.  | [PartyRole]                 |
| Contra Clearing Firm        | Clearing firm of the broker or other firm which is the contra side of the trade.   | [PartyRole]                 |
| Correspondent Clearing Firm | ClearingFirm that is going to carry the position on their books at another clearing house (exchanges).   | [PartyRole]                 |
| Cross                       | Client sends Broker a buy or sell order. Broker wishes to take the other side and cross with the client. Broker sends an order with Side=Cross to an exchange.   | [OrdType]                   |
| Cross Short                 | Client wants to establish a short position, and so sends a Sell Short to Broker. Broker wants to cross with the Client, so Broker sends a Cross Short order to an exchange. Cross Short is crucial here because many exchanges have tick rules needing to be enforced, and the order getting converted from Sell Short to Cross (instead of Cross Short) could result in an illegal short sell.  | [OrdType]                   |
| Cross Short Exempt          | Client wants to establish a short position, and is exempt from the uptick restriction. So Client sends Sell Short Exempt to Broker. Broker wants to cross with the Client, so Broker needs a way to send "Cross Short Exempt" to the exchange so that an audit trail traces back indicating that the party selling short was exempt from the uptick rule.  | [OrdType]                   |
| 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  | [ExecInst]                  |

|                              |  |                                  |
|------------------------------|--|----------------------------------|
|                              | result of a stock dividend or distribution.  |                                  |
| 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]                       |
| Entering Firm                | Broker who has recorded or reported an execution. This field is particularly useful where the trade is entered into a trade recording system by a broker who is not a party to the trade, as it allows any inquiries or problem resolution to be directed to the appropriate source.   | [PartyRole]                      |
| Executing Firm               | Identifies executing / give-up broker.   | [PartyRole]                      |
| Executing System             | System Identifier where execution took place (e.g. some markets have multiple execution location such as an electronic book or automatic execution system).  | [PartyRole]                      |
| Executing Trader             | Trader or broker id associated with Executing Firm who actually executes the trade.  | [PartyRole]                      |
| 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]                    |
| <b><u>FIX Connection</u></b> | <b><u>A FIX Connection is comprised of three parts: logon, message exchange, and logout.</u></b>   |                                  |
| <b><u>FIX Session</u></b>    | <b><u>A FIX Session is comprised of one or more FIX Connections, meaning that a FIX Session spans multiple logins.</u></b>   |                                  |
| Forex - Limit                | A "Limit" order for Foreign Exchange (currency trading).   | [OrdType]<br><b>(Deprecated)</b> |
| Forex - Market               | A "Market" order for Foreign Exchange (currency trading).  | [OrdType]<br><b>(Deprecated)</b> |
| Forex - Previously Quoted    | A "Previously Quoted" order for Foreign Exchange (currency trading).   | [OrdType]<br><b>(Deprecated)</b> |
| Forex - Swap                 | A "Swap" order for Foreign Exchange (currency trading).  | [OrdType]                        |
| Funari                       | Japanese term for an order to buy or sell a stated amount of a security at the specified limit price with any unexecuted (leftover) quantity becoming a Market On Close order.   | [OrdType]                        |
| Fund manager Client ID       | <b><i>For CIV:</i></b><br>An identifier for an Investor or a broker or funds supermarket's nominee/custodian company which is recognized by the Fund manager.  | [PartyRole]                      |
| Giveup Clearing Firm         | Firm to which the trade is given up (carries the position that results from a trade).  | [PartyRole]                      |
| Good Till Canceled           | An order to buy or sell that remains in effect until it is either  | [TimeInForce]                    |

|                               |  |                                  |
|-------------------------------|--|----------------------------------|
|                               | executed or canceled; sometimes called an "open order".  |                                  |
| Held                          | The firm executing the order is held to best execution requirements, and may not make discretionary decisions. Opposite of Not Held  | [ExecInst]                       |
| 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]                    |
| Institutions Only             | Broker is restricted to dealing with other buy side firms.   | [ExecInst]                       |
| Introducing Firm              | The broker or other intermediary with the closest association with the investor.   | [PartyRole]                      |
| Investor ID                   | <b>For Equities:</b><br>Identifies beneficiary or broker acting on behalf of beneficiary. This field is mandatory for various exchanges either pre or post trade.<br>Numerical entry containing no dashes.                     | [PartyRole]                      |
|                               | <b>For CIV:</b><br>An Investor identifier such as a taxpayer reference (NINO, NPN, DSS, SSN number etc) for an individual investor or a registration number (EIN, etc.) for a company.<br>May contain alphanumeric and dashes. | [PartyRole]                      |
| Limit                         | An order to buy a security at or <del>above</del> <u>below</u> a stated price, or to sell a security at or <del>below</del> <u>above</u> a stated price.   | [OrdType]                        |
| Limit or Better               | Indicates an order to<br>- buy a security at the indicated limit price or lower, or to<br>- 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]                        |
| Locate/Lending Firm           | Identity of the firm which is loaning the security in a short sale.  | [PartyRole]                      |
| 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 If Touched             | Indicates an order to buy or sell a stated amount of a security or commodity as soon as a preset market price is reached, at which point it becomes a Market order.  | [OrdType]                        |
| Market On Close               | Indicated price is held to the closing price ("firm" instruction).   | [IOIQualifier]                   |
| Market On Close               | An order to be executed at the close of the market. Depending on Exchange / ECN / ATS rules, this may be at an exact official market closing price, or may be near the closing price.  | [OrdType]<br><b>(Deprecated)</b> |
| Market Or Better              | Indicates an order to buy or sell a stated amount of a security at the quoted market or better.  | [OrdType]                        |
| Market with Leftover as Limit | Indicates an order to buy or sell a stated amount of a security at the prevailing market price with any unexecuted (leftover) quantity becoming a Limit order at the last executed price.                                      | [OrdType]                        |

|                               |  |                                  |
|-------------------------------|--|----------------------------------|
| Next Fund Valuation Point     | For CIV orders - indicates that the Investor wishes the order to be dealt at the unit price determined at the next Valuation Point, a.k.a. a Forward price.  | [OrdType]                        |
| No Cross                      | The broker executing this trade is forbidden from taking the other side of the trade. Opposite of OK to Cross.   | [ExecInst]                       |
| Not Held                      | The firm executing the order is not held to best execution requirements, and may be able to make some discretionary decisions. Opposite of Held.   | [ExecInst]                       |
| OK to Cross                   | The broker executing this trade is allowed to take the other side of the trade. Opposite of No Cross.  | [ExecInst]                       |
| Opposite                      | Sides of the legs are the opposite of their definition in the multileg instrument.   | [Side]                           |
| On Basis                      | An order to buy or sell at the basis price. The basis price is established by joint agreement of odd lot dealers in 100 share unit stocks when no round lot sale has occurred during the trading session, the spread between the closing bid and offer is two points or more, and an odd lot dealer has been given a basis price order. (e.g. NYSE order type)                       | [OrdType]                        |
| On Close                      | An odd-lot order to buy or sell to be filled at the price of the closing round-lot offer<br>- plus the differential, for a buy order, or<br>- minus the differential, for a sell order,<br>or<br>A crossing session order to buy or sell at the closing price.   | [OrdType]<br><b>(Deprecated)</b> |
| Order Origination Firm        | Buy-side firm associated with Order Origination Firm which originates/submits the order.   | [PartyRole]                      |
| Order Origination Trader      | Buy-side trader id associated with Order Origination Firm which originates/submits the order.  | [PartyRole]                      |
| Par                           | Equal to the face value (nominal) of a security, i.e. A bond selling at par is worth an equivalent to its original issue value, typically \$1000/bond.   | [QuantityType]                   |
| Participate Don't Initiate    | An order that may participate in a transaction initiated by another party, but may not initiate a transaction. For example, on US ECNs / Exchanges, this may represent an order that will be quoted to the marketplace and will trade if another party initiates a trade (i.e. hits the posted quote), but cannot be routed to initiate a trade with another market or market maker. | [ExecInst]                       |
| Percent of Volume             | The sender does not want to be all of the volume on the floor.   | [ExecInst]                       |
| Previous Fund Valuation Point | For CIV orders - indicates that the Investor prefers that the order be dealt at the unit price determined at the immediately preceding Valuation Point, a.k.a. a Historic price. (This can be overridden by the constitution of the fund or, in certain circumstances, by the Fund Manager.)   | [OrdType]                        |

|                             |  |                 |
|-----------------------------|--|-----------------|
| Order Originator            | ID of the party entering the trade into the system (data entry, userid, buy side trader, etc.).  | [PartyRole]     |
| Previously indicated        | An order sent in response to an Indication of Interest message.  | [OrdType]       |
| Previously quoted           | An order sent in response to a Quote message.  | [OrdType]       |
| Reinstate on System Failure | If a system failure interrupts trading or order routing, attempt to reinstate this order, subject to time in force limitations. Note that depending on the type and severity of the failure, this might not be possible.   | [ExecInst]      |
| Reinstate on Trading Halt   | If trading in this instrument is halted, reinstate this order when/if trading resumes, subject to time in force limitations.   | [ExecInst]      |
| Riskless Principal          | "Riskless" principal transactions are generally described as trades in which, after receiving an order to buy (or sell) from a customer, the broker-dealer purchases (or sells) the security from (or to) another person in a contemporaneous offsetting transaction.<br><br>Above from the SEC web-site <a href="http://www.sec.gov/rules/final/34-44291.htm">http://www.sec.gov/rules/final/34-44291.htm</a><br><br>See Exchange Act Rule 10b-10(a)(2)(ii)(A) [17 CFR 240. 10b-10(a)(2)(ii)(A)]; Exchange Act Rel. No. 33743 (Mar. 9, 1994) at n.11. | [OrderCapacity] |
| 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:<br><br>- not lower than the last sale if the last sale was a "plus" or "zero plus" tick and<br><br>- 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.<br><br>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]       |
| Settlement Location         | Identifies Settlement Depository or, if local settlement, the ISO Country Code.  | [PartyRole]     |
| Sponsoring Firm             | A member of the exchange that is sponsoring an Entering Entity to send orders to the exchange. The Sponsoring Member Firm permits sponsorees (i.e. Entering Entities) to trade thereby allowing them to enter orders directly to the exchange via automated means. (e.g. NYSE allowing direct access via Anonymous DOT service).   | [PartyRole]     |
| 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]                 |
| Stopped                            | A trade is guaranteed for the order, usually at a stated price or better, but has not yet occurred. For example, a specialist on an exchange may "stop" an order while searching for a better price.  | [OrdStatus]               |
| Streetside Trade Capture Reporting | Reporting of completed trades for clearance and settlement or compliance purposes. Reports may be originated by Exchanges or by clearing firms and sent to clearing firms directly or via a clearing corporation or central counterparty such as DTCC in the US.  | A "Section" in "Volume 5" |
| Suspended                          | The order is not eligible for trading. This usually happens as a result of a verbal or otherwise out of band request to suspend the order, or because the order was submitted, or modified via a Cancel/Replace Request, with ExecInst=Suspended.   | [OrdStatus]               |
| TED Price                          | The price spread between the active 3 month treasury bill futures contract and the 3 month Eurodollar futures contract. Used as an indicator of investor confidence in the U.S. markets.  | [PriceType]               |
| TED Yield                          | The difference in basis points between the yield-to-maturity of the bond / note and the yield-to-maturity of a Hypothetical Euromarket bond with identical coupon and maturity.   | [PriceType]               |
| Trade Along                        | Clients who specify "Trade Along" give brokers permission to handle and place their order in the market even if the broker already has its own proprietary orders for the same security placed in the market.   | [ExecInst]                |
| Try to Stop                        | Used in specialist-driven markets to direct the specialist to try and stop the order.   | [ExecInst]                |
| Underlying Contra Firm             | The broker or other firm which is the contra side of the trade for the underlying security.   | [PartyRole]               |
| With or Without                    | An odd lot order filled on an effective round lot transaction, or on an effective bid or offer, whichever occurs first after the specialist receives the order. (e.g. NYSE order type)  | [OrdType]                 |