

Global Fixed Income Committee

Best Practices for Trading Fixed Income Instruments

CDS & IRS

VOLUME 2 – Common Workflows

Editor(s):	Produced by: FIX Trading Community Global Fixed Income Committee
Date:	Thursday, 08 May 2014
Version:	5.0
Paper designation:	Analysis

DISCLAIMER

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

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

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

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

Copyright 2014 FIX Protocol Ltd, all rights reserved

Table of Contents

DISCLAIMER	2
DOCUMENT HISTORY	6
VOLUME INDEX	7
1 INTRODUCTION	8
2 FIX COMMON INFRASTRUCTURE MESSAGES	9
2.1 Session Management Best practices	9
2.2 Business Reject Message Best practices	9
2.3 Parties Component Best Practices.....	9
3 INSTRUMENT IDENTIFIERS	11
3.1 Scope.....	11
3.2 Out of Scope	11
3.3 Best Practices Instrument Identifiers Recommendations	11
3.4 Instrument Identifier Attributes	12
3.5 Trading Types	17
3.6 Reference Data Messages	19
3.7 Market Data Messages	20
3.8 Quote Contribution Messages.....	20
3.9 Quotation/Negotiation and Quote Order Based Messages	21
3.10 Central Order Book Messages	23
4 MARKET CONVENTIONS.....	24
4.1 Side/Direction of Trade.....	24
4.2 LastQty(32) of multi-leg instrument	24
4.3 Default structure for IRS vs. treasury multi-leg instrument.....	24
4.4 Price type convention.....	25
4.5 Factor based on DV01	25
4.6 Scenario MC1 – Example of IRS Gap Spread Definition and Usage	26
4.7 Scenario MC2 - Example of Duration Neutral IRS Butterfly	29
5 REGULATORY CHANGES	32
5.1 Unique Swap Identifier (USI).....	32
5.2 Mid-Market Mark	33
5.3 US vs. Non-US Person.....	34
5.4 Required vs. Permitted Transaction	34
5.5 Block Trade.....	34
5.6 Reporting Party Indicator.....	35
5.7 Intermediary Reporting to CCP	35
5.8 Scenario RC1 – Example of Flow for ‘Required Transaction’	36
5.9 Scenario RC2 – Example of Flow for ‘Permitted Transaction’	36
6 PRE-TRADE - REFERENCE DATA.....	37

6.1 Overview diagram.....	37
6.2 Message Flows Summary	38
6.3 Scenario SL1 – Dealer Requests Snapshot of Instruments, Execution Venue Returns Snapshot	39
6.4 Scenario SL2 – Dealer Requests Snapshot of Instruments, Execution Venue Returns Snapshot (fragmented)	41
6.5 Scenario SL3 – Dealer Subscribes to Instruments Reference Data, Execution Venue Returns Snapshot and Updates	43
6.6 Scenario SL4 – Dealer Cancels Instrument Reference Data Subscription	45
6.7 Scenario SL5 – Dealer Requests Snapshot or Subscription of Instruments Reference Data, Execution Venue Returns Error	46
6.8 Scenario SL6 – Dealer Cancels Instrument Reference Data Subscription, Execution Venue Rejects	47
6.9 Scenario SS1 – Dealer Requests Instrument Status, Execution Venue Returns Status	48
6.10 Scenario SS2 – Dealer Subscribes to Instrument Status, Execution Venue Returns Snapshot and Updates	49
6.11 Scenario SS3 – Dealer Requests Instrument Status, Execution Venue Returns Error	50
6.12 Scenario SD1 – Dealer Sends Instrument Definition Request, Execution Venue Confirms	51
6.13 Scenario SD2 – Dealer Sends Instrument Definition Request, Execution Venue Rejects	52
6.14 Scenario SD3 – Execution Venue Publishes Instrument Definition	53
7 PRE-TRADE – PRICE CONTRIBUTION	54
7.1 Scenario PP1 – Dealer Sends Different Prices for the Same Instrument to Multiple Customers – Dealer Amends Trade Information (Price & type)	55
8 PRE-TRADE – PRICE SUBSCRIPTION	58
8.1 Overview diagram.....	58
8.2 Message Flows Summary	58
8.3 Scenario MDS1 – Dealer Requests/Receives Market Data.....	59
8.4 Scenario MDS2 – Dealer Unsubscribes Market Data	60
8.5 Scenario MDS3 – Dealer Requests Market Data, Execution Venue Rejects	61
9 PRE-TRADE – QUOTE CONTRIBUTION	62
9.1 Overview diagram.....	62
9.2 Message Flows Summary	62
9.3 Scenario PC1 – Dealer Sends Quote to Market	63
9.4 Scenario PC2 – Dealer Cancels Quote on Market	64
9.5 Scenario PC3 – Dealer Cancels All Quotes	65
9.6 Scenario PC4 – Dealer Sends Quote to Market, Execution Venue Rejects	66
9.7 Scenario PC5 – Dealer Cancels a Quote, Execution Venue Rejects	67
9.8 Scenario MQ1 – Dealer Sends Mass Quote to Execution Venue	68
9.9 Scenario MQ2 – Dealer Cancels a Single Quote Sent Within a Mass Quote	70
9.10 Scenario MQ3 – Dealer Cancels All Quotes Sent within a Mass Quote.....	72
10 MESSAGE DETAIL	73
10.1 Tag PartyIDSource(447) values	74
10.2 Tag PartyRole(452) values.....	75
10.3 Instrument Component.....	77

10.4 Security List Request (MsgType=x)	80
10.5 Security List (MsgType=y)	82
10.6 Security List Update Report (MsgType=BK)	83
10.7 Security Definition Request (MsgType=c)	84
10.8 Security Definition (MsgType=d)	85
10.9 Security Status Request (MsgType=e)	86
10.10 Security Status (MsgType=f)	87
10.11 Market Data Request (MsgType=V)	90
10.12 Market Data Snapshot Full Refresh (MsgType=W)	91
10.13 Market Data Incremental Refresh (MsgType=X)	92
10.14 Market Data Request Reject (MsgType=Y)	94
10.15 Quote (MsgType=S)	95
10.16 Quote Cancel (MsgType=Z)	97
10.17 Quote Status Report (MsgType=AI)	98
10.18 Mass Quote (MsgType=i)	100
10.19 Mass Quote Ack (MsgType=b)	102
10.20 Quote Ack (MsgType=CW)	104
APPENDIX I - DOCUMENT FIGURES	105
APPENDIX II - DOCUMENT TABLES	106

Document History

Date	Change	Version
23 September 2011	<ul style="list-style-type: none"> Split the original document into 4 volumes <ul style="list-style-type: none"> Volume 1 – Introduction Volume 2 – Common Workflows (this document) Volume 3 – Quote-Driven Workflows Volume 4 – Central Order Book Workflows This volume is based on chapters 2-5 of the un-split <i>Best Practices: FIX Message Flows and Usage for Interest Rate Swaps (IRS) and Credit Default Swaps (CDS) version 2.3.</i> Added Scenario SD3 Update Pre-Trade Quote Contribution Overview diagram change message name MassQuoteAcknowledge to MassQuoteAcknowledgement Update Pre-Trade Price Subscription add MarketDataRequestReject Update Scenario SD3 - rename to “Execution Venue Publishes Instrument Definition” and add Sequence Diagram Link scenarios in Pre-Trade Quote Contribution section to scenarios in Volume 3 Restore note on QuoteAcknowledgement message in Pre-Trade Quote Contribution section Added Instrument identifier approach 	3.0
18 October 2011	<ul style="list-style-type: none"> Merged the errata into this document Corrected the Overview diagram in “Pre-trade – Quote Contribution” section 	3.01
31 October 2011	<ul style="list-style-type: none"> Removed the Instrument identifier proposal and updated that section with a note to that effect Merged the errata into this document 	3.02
25 November 2011	<ul style="list-style-type: none"> Updated QuoteAcknowledgement with QuoteAckStatus Added narratives for MassQuoteAcknowledgement 	3.03
14 February 2012	<ul style="list-style-type: none"> Merged Instrument Identifier documentation 	3.04
27 February 2012	<ul style="list-style-type: none"> Merge GFIC LegSecurityXML Proposal Merge GFIC QuoteAck Message Proposal 	3.05
5 March 2012	<ul style="list-style-type: none"> Added explanations about the Message Details tables Fixed copyrights years 	3.06
31 July 2012	<ul style="list-style-type: none"> Merged with cash bonds version Reference the pre-trade FpML payload schema Added narratives to explain parties component 	3.1
10 August 2012	<ul style="list-style-type: none"> Modifications after Lisa Taikitsadaporn review 	3.2
28 August 2012	<ul style="list-style-type: none"> Updated link to pre-trade schema definition (Section 3.3.1) 	3.3
3 September 2012	<ul style="list-style-type: none"> Added examples for switch and butterfly LastQuantity 	3.4
24 October 2012	<ul style="list-style-type: none"> Merged with Cash Bonds Best Practices 	4.0
04 January 2013	<ul style="list-style-type: none"> Applied recommendations of FIX-FpML Collaboration Working Group 	4.01
11 July 2013	<ul style="list-style-type: none"> Document updated to include regulatory changes emerging from the finalised SEF rules 	4.02
07 April 2014	<ul style="list-style-type: none"> Document ready for FIX Trading Community member firm review followed by public release (Phase 3) 	5.0

VOLUME INDEX

VOLUME 1 - INTRODUCTION

VOLUME INDEX
INTRODUCTION
ORGANIZATION OF DOCUMENT
OVERVIEW DIAGRAM
FIX VERSIONS SUPPORTED & BACKWARD COMPATIBILITY
GLOSSARY

VOLUME 2 - COMMON WORKFLOWS

INTRODUCTION
FIX COMMON INFRASTRUCTURE MESSAGES
INSTRUMENT IDENTIFIERS
MARKET CONVENTIONS
REGULATORY CHANGES
PRE-TRADE – REFERENCE DATA
PRE-TRADE – PRICE SUBSCRIPTION
PRE-TRADE – QUOTE CONTRIBUTION
MESSAGE DETAIL
APPENDIX

VOLUME 3 - QUOTE- DRIVEN WORKFLOWS

INTRODUCTION
QUOTE DRIVEN MODEL OVERVIEW
QUOTE REQUEST WORKFLOWS
TRADEABLE QUOTE WITH DEALER'S LAST LOOK WORKFLOWS
TRADEABLE QUOTE WITHOUT DEALER'S LAST LOOK WORKFLOWS
INDICATIVE QUOTE WORKFLOWS
MULTI-DEALER WORKFLOWS
VOICE TRADING
AXE INDICATION
MESSAGE DETAIL
APPENDIX

VOLUME 4 - CENTRAL ORDER BOOK WORKFLOWS

INTRODUCTION
TRADING-CENTRAL LIMIT ORDER BOOK
QUOTE CONTRIBUTION TO CENTRAL LIMIT ORDER BOOK
VOICE TRADING
AUCTIONS AND WORKUPS
STOP ORDERS, ICEBERG ORDERS & PEG ORDERS
MESSAGE DETAIL
APPENDIX

VOLUME 5 – POST-TRADE WORKFLOWS

INTRODUCTION
ALLOCATIONS
CONFIRMATIONS
MESSAGE DETAIL
APPENDIX

1 Introduction

This document represents volume 2 of the *Best Practices for Trading Fixed Income Instruments - CDS & IRS* document suite. This volume describes the best practices that are applicable to common workflows which are used in trading scenarios. The topics covered in this volume are:

- FIX Session
- Instrument Identifiers
- Pre-Trade – Reference Data
- Pre-Trade – Price Subscription
- Pre-Trade – Quote Contribution

Overview of this document and document conventions are explained in *Best Practices for Trading Fixed Income Instruments - Volume 1*. Reading of *Best Practices for Trading Fixed Income Instruments - Volume 1* is taken as a prerequisite to understand this volume.

2 FIX Common Infrastructure Messages

2.1 Session Management Best practices

This section describes the connectivity between Execution Venue and Dealer. The messages within are used to initiate sessions, terminate sessions, synchronize sessions and keep all parties aware of the status of FIX sessions.

- Logon message is initiated by Dealer (i.e. FIX session initiator). The Execution Venue receives the Logon message (i.e. FIX session listener)
- Immediately after logon, it is expected that both the Dealer and the Execution Venue will synchronize the messages sequence and send 'ResendRequest' message (35=2) to receive missing messages if necessary
- The standard heartbeat interval is determined by the Execution Venue. In most cases the interval should be between 5 – 30 seconds.
- TestRequest (35=1), ResendRequest(35=2), Reject (35=3), SequenceReset (35=4) and Logout (35=5) messages may be sent by both the Dealer and the Execution Venue. Both sides should expect to receive them and be able to handle them.
- All TransactTime(60) fields should have a resolution of 1ms or better

All FIX session handling will follow standard FIX session behaviour as documented by the FIX Protocol standard.

2.2 Business Reject Message Best practices

The BusinessMessageReject(MsgTyp=j) message can reject an application-level message which fulfils session-level rules and cannot be rejected via any other means. One example usage of this message is where the receiving system (either FIX Gateway at Dealer or at Execution Venue) is able to receive the message but unable to pass it on to the intended upstream application. In this case the receiving system (FIX Gateway) should use BusinessMessageReject message to reject the message.

FIX specifications Version 5.0 Service Pack 2 – Volume 1, chapter “Common Infrastructure Messages” describes the usage of this message in detail. Best practice recommends that usage of this message should adhere to the guidelines set out in the FIX Specification document.

2.3 Parties Component Best Practices

The Parties component can be used in most FIX messages.

Amongst others it can be contained in each one of the messages that are referred to in the following sections:

- Volume 2 – Common Workflows
 - Price subscription workflows
 - Quote contribution workflows
- Volume 3 – Quote Negotiation
 - Quote request workflows
 - Tradeable quotes without dealer's last look workflows
 - Tradeable quotes with dealer's last look workflows
 - Indicative quotes workflows
 - Multi-dealer workflows
- Volume 4 – Central Limit Order Book
 - Trading - central limit order book workflows
 - Quote contribution to central limit order book workflows

The following table summarizes the main PartyRole(452) values that are expected to be exchanged during typical trading workflows. The complete set of values for PartyRole(452) can be found in the [Tag PartyRole\(452\) values](#) section.

Role Represented	PartyRole Values
Execution Venue	One of: <ul style="list-style-type: none"> Execution Venue(73)¹ Exchange(22)
Central Counterparty Clearing House	<ul style="list-style-type: none"> Clearing Organization(21)¹
Dealer	One of: <ul style="list-style-type: none"> Executing Firm(1)¹ Market Maker(66)
Counterparty (i.e. customer or another dealer)	One of: <ul style="list-style-type: none"> Client ID(3)¹ Order Origination Firm(13)

2.3.1 Legal Entity Identifier

The Legal Entity Identifier (LEI) originated as an initiative under the USA Dodd-Frank legislation to create an identifier for legal entities. The LEI is to be used to identify legal entities across financial products and services in order correlate financial information for the purposes of analyzing systemic risk. The LEI has undergone a significant global development effort since its inception. A new ISO standard, ISO 17442 Legal Entity Identifier was recently approved. Globally the LEI will be managed by the Financial Stability Board of the G20 countries.

Source: [EP156 Legal Entity Identifier \(LEI\) Extension](#)

Recently, the Legal Entity Identifier (LEI) has been added as an enumeration to the tag PartyIDSource(447)=N.

¹ Suggested best practices value. We could not identify a unanimous market convention.

3 Instrument Identifiers

3.1 Scope

3.1.1 Instrument Types

- Standard Instrument - definition supplied prior to trading in a Securities Reference Data message [i.e. SecurityList or SecurityDefinition]
- Non-Standard Instrument - definition supplied during trading

3.1.2 Trading Types

- Outright Instruments (IRS/CDS Instruments)
- Multi-leg Instruments Consisting of Outright Instruments
 - Spreads consisting of Outright Instruments
 - Butterflies consisting of Outright Instruments
 - > 3 leg trades consisting of Outright Instruments
- Nested Multi-leg Instruments
 - Spread trades where some leg(s) are themselves Multi-Leg Instruments
 - > 2 leg trades where some leg(s) are themselves Multi-Leg Instruments

3.1.3 Strategy Types: Notional Amount Calculation for Multi-Leg Instruments

- Pre-defined notional amount ratio between legs
- Overridden (fixed) notional amounts

3.1.4 Workflows

- Reference Data
- Market Data
- Quote Contribution
- Quotation Negotiation
- Order Quote Based
- Central Order Book

3.2 Out of Scope

- Definition of notional amounts for Multi-leg Instruments where the notional amount is not a fixed ratio between the legs
- Non-Standard Nested Multi-leg Instruments in Quotation/Negotiation and Quotes/Orders scenarios

3.3 Best Practices Instrument Identifiers Recommendations

3.3.1 General Recommendations:

- The definition of a standard outright instrument should be supplied within a SecurityXML component block of a security's reference data message using an FpML payload
- The FpML payload schema can be found at: <http://www.fpml.org/spec/fpml-5-5-1-wd-1/>
- The definition of a standard multi-leg instrument should be supplied within the InstrumentLeg component. The reference to each leg of the instrument needs to be supplied in LegSecurityID(602).
- The definition of a non-standard instrument should be supplied in one of two message types:
 - For central limit order books, the definition should be supplied in reference data messages (e.g. SecurityDefinition message as used in scenario SD3)
 - For quote driven models, see details in [Non Standard Instrument in QuoteRequest Message](#) and [Quotation/Negotiation and Quote Order Based Messages](#)

- Once an instrument definition is supplied, the instrument should be referenced in all subsequent messages using the attributes defined in [Common-Instrument-Attributes](#) (see below)
- The associated SecurityID(48) should be assigned by the execution venue, until such a time as an industry standard is established
- The default value for SecurityIDSource(22) where the execution venue assigns the SecurityIDs is MarketplaceAssignedIdentifier(M)
- Transmission of the full instrument definition repeatedly in trading messages should be avoided, as this may considerably increase the size of the FIX messages and the latency in trading flows
- Defining a multi-leg instrument within the FpML payload should be avoided. The individual legs of each multi-leg instrument should instead be described using the appropriate FIX component block (e.g. InstrumentLeg component).

3.3.2 Instrument Identifiers and Attributes in Central Order Book Workflow Recommendation:

- When a Multi-leg Instrument is executed there maybe one or more Execution Reports
 - for each Outright Instrument (IRS/CDS) that is part of the traded Instrument
 - for the Instrument as a whole

3.4 Instrument Identifier Attributes

3.4.1 Common Instrument Identifier Attributes

- The following minimum set of attributes is used in all FIX messages to refer to an Instrument (some FIX messages will require additional attributes).

Common Instrument FIX Attributes (Common-Instrument-Attributes)
Symbol(55)=<Required human readable name of the Instrument or [N/A]>
SecurityID(48)=<Required unique ID for the Instrument>
SecurityIDSource(22)=<Required, default value: Marketplace-assigned Identifier(M)>
SecurityType(167)=<Required>

This set of FIX attributes is referred to in this document as the [Common-Instrument-Attributes](#).

- The SecurityID(48) uniquely identifies a single Instrument
- The value of SecurityType(167) is one of 'CDS', 'IRS' or 'MLEG'

3.4.2 Standard outright CDS Instrument Identifier Attributes

- CDS instruments which are based on Markit REDCode should be identified by the following attributes:
 - Symbol(55) = [N/A]
 - SecurityID(48) = contains the Markit REDCode 9 alphanumeric characters
 - SecurityIDSource(22) = P (Markit RED pair CLIP)
 - SecurityType(167) = CDS
- In addition to the above fields, the following attributes may also be used in the Instrument component:

Standard CDS Instrument FIX Attributes

CouponRate(223)
CouponPaymentDate(224)
IssueDate(225)
MaturityDate(541)
PriceUnitofMeasure(1191)
ValuationMethod(1197)
RestructuringType(1449)
Seniority(1450)
NotionalPercentageOutstanding(1451)
OriginalNotionalPercentageOutstanding(1452)
ObligationType(1739)

3.4.3 Standard Instrument Identifier Attributes

Bespoke instruments, which cannot be identified using the regular FIX attributes, are defined using an FpML pre-trade payload. The FpML payload was defined by ISDA for IRS and CDS.

Such an FpML payload is sent from the Execution Venue to the dealers in the SecurityXML FIX Component.

3.4.3.1 Standard Outright Instrument Identifier Attributes

- The definition of the Standard Outright Instrument is supplied in response to a reference data request.
- The definition contains the following FIX attributes:

Standard Outright Instrument FIX Attributes ([Outright-Instrument-Attributes](#))

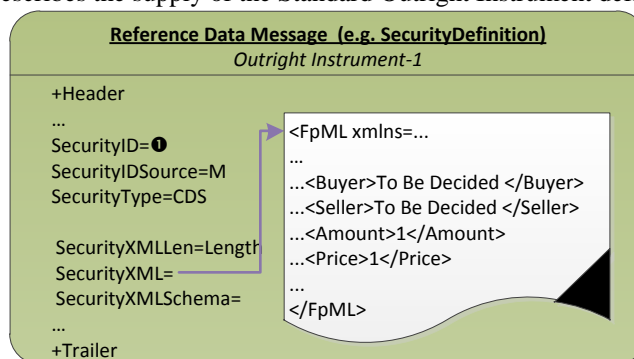
Symbol(55)=<Required human readable name of the Instrument or [N/A]>
SecurityID(48)=<Required unique ID for the Instrument>
SecurityIDSource(22)=<Required, default value: Marketplace-assigned Identifier(M)>
SecurityType(167)=<Required>

SecurityXMLLen(1184)=Length
SecurityXML(1185)=<Required>
SecurityXMLSchema(1186)=<Optional>

This set of FIX attributes is referred to in this document as the [Outright-Instrument-Attributes](#).

- The Securities Reference Data message contains the XML payload that provides the Instrument definition in the SecurityXML(1185) attribute. The format and precise content of the XML payload is out of scope for this document
- The SecurityType(167) value is one of ‘CDS’ or ‘IRS’

The following Diagram describes the supply of the Standard Outright Instrument definition:



- Once a Standard Outright Instrument definition is supplied, the Instrument should be referenced in all subsequent messages using the [Common-Instrument-Attributes](#)

3.4.3.2 Standard Multi-leg Instrument Identifier Attributes

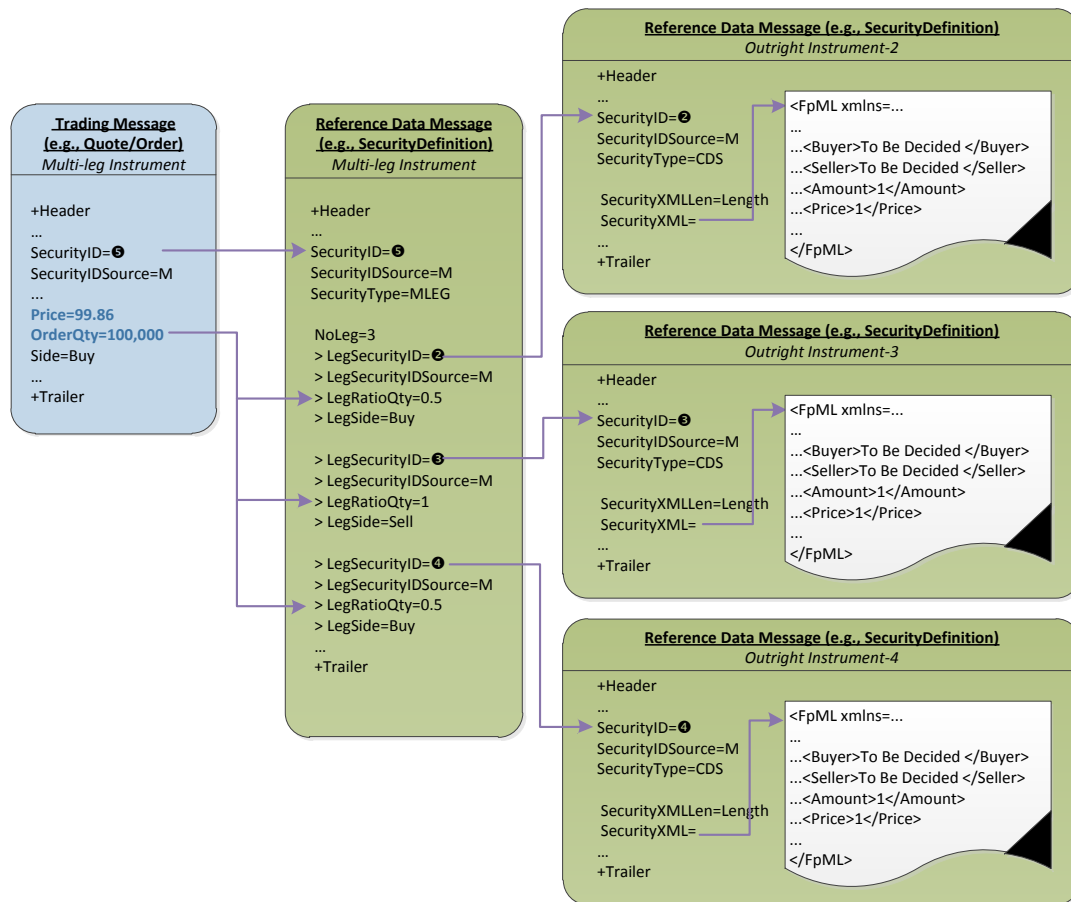
- The definition of a Standard Multi-leg Instrument is supplied in response to a reference data request.
- The Instrument definition contains the following FIX attributes:

Standard Multi-leg Instrument FIX Attributes (Multi-Leg-Instrument-Attributes)
Symbol(55)=<Required human readable name of the Instrument or [N/A]> SecurityID(48)=<Required unique ID for the Instrument> SecurityIDSource(22)=<Required, default value: Marketplace-assigned Identifier(M)> SecurityType(167)=<Required> NoLegs(555)= <Number of Legs> > LegSymbol(600)=<Required> > LegSecurityID(602)=<Required> > LegSecurityIDSource(603)=<Required, default: MarketplaceAssignedIdentifier(M)> > LegSecurityType(609)=<Required> > LegRatioQty(623)=<Conditionally required> > LegSide(624)=<Required>

This set of FIX attributes is referred to in this document as the [Multi-Leg-Instrument-Attributes](#)

- The SecurityType(167) value is MLEG
- The values of { LegSymbol, LegSecurityID, LegSecurityIDSource and LegSecurityType } point to the leg Instrument
- LegRatioQty is conditionally required and used to calculate the notional amount of each leg. See further description in [Multi-leg Instrument types](#)
- The side of each leg specified in LegSide(624) is represented from the Customer's perspective
- As with a Standard Outright Instrument, once a Standard Multi-leg Instrument definition is supplied, the Instrument should be referenced in all subsequent messages using the [Common-Instrument-Attributes](#)

The following Diagram describes the structure of a Standard Multi-leg Instrument definition:



3.4.4 Non Standard Instrument Identifier Attributes

3.4.4.1 Non-Standard Instrument Identifier Attributes in a Securities Reference Data Message

A Non-Standard Instrument definition supplied in a Securities Reference Data message has the same attributes as a [Standard Instrument](#). The only difference lies in the time when the Instruments are published:

- The Standard Instrument definition is published at the start of the day upon Reference Data Request
- The Non-Standard Instrument definition is an unsolicited message published during the trading hours

3.4.4.2 Non-Standard Instrument Identifier Attributes in a QuoteRequest Message

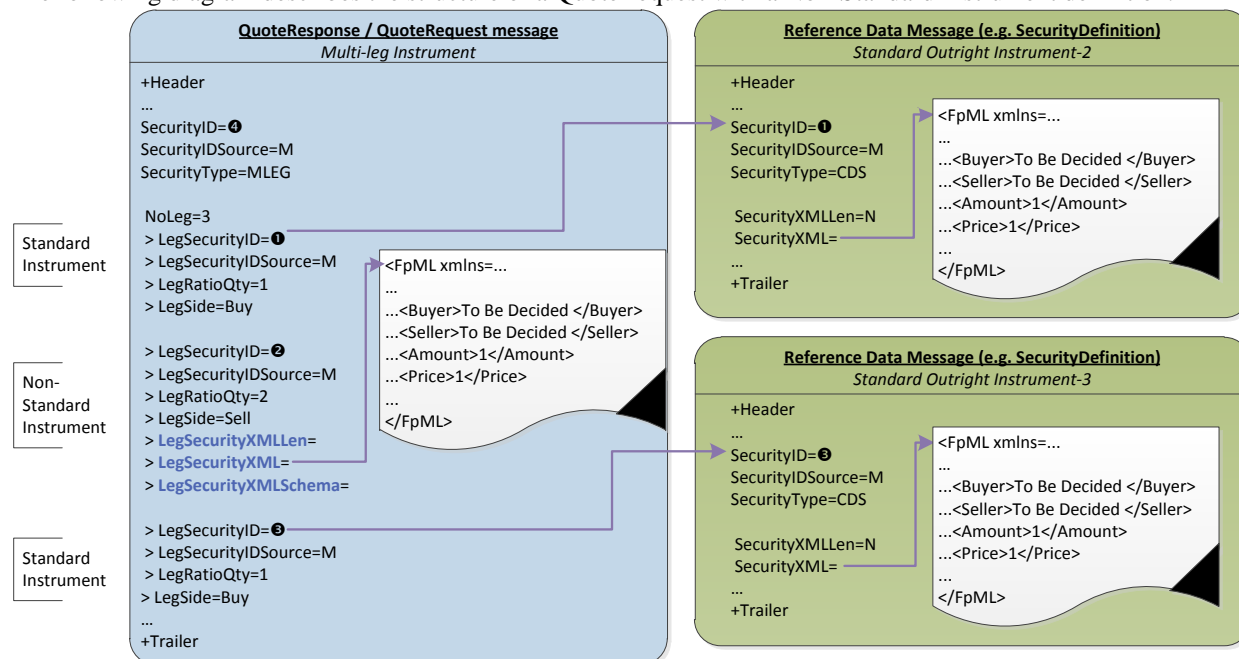
- When an Execution Venue sends the Dealer a QuoteRequest for a Non-Standard Instrument, the Instrument definition must be embedded in the QuoteRequest message itself.
- The definition for a Non-Standard Outright Instrument contains the [Outright-Instrument-Attributes](#)
- The definition for a Non-Standard Multi-leg Instrument contains the following FIX attributes:

Non Standard Multi-leg Instrument in QuoteRequest Message FIX Attributes

[Common-Instrument-Attributes](#)

NoLegs(555)= <Number of Legs>
> LegSymbol(600)=<optional>
> LegSecurityID(602)=<required>
> LegSecurityIDSource(603)=<required>
> LegSecurityType(609)=<required>
> LegRatioQty(623)=<conditionally required>
> LegSide(624)=<required>
> [LegSecurityXMLLen\(1871\)](#)=<optional>
> [LegSecurityXML\(1872\)](#)=<optional>
> [LegSecurityXMLSchema\(1873\)](#)=<optional>

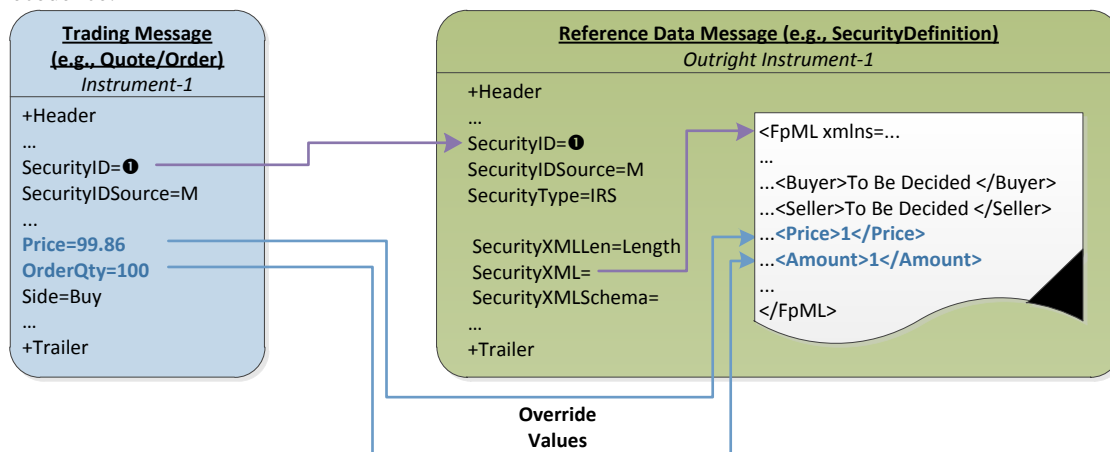
The following diagram describes the structure of a QuoteRequest with a Non-Standard Instrument definition:



3.5 Trading Types

3.5.1 Outright Instrument

In the case where a trading attribute exists both as a FIX attribute within one or more trading message as well in the XML payload as one of the attributes in the Instrument definition, the FIX attribute in the trading message takes precedence.



3.5.2 Multi-leg Instrument Types

“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 is, all instrument legs are traded or none are traded.” FIX version 5.0 Service Pack 2 Volume 1.

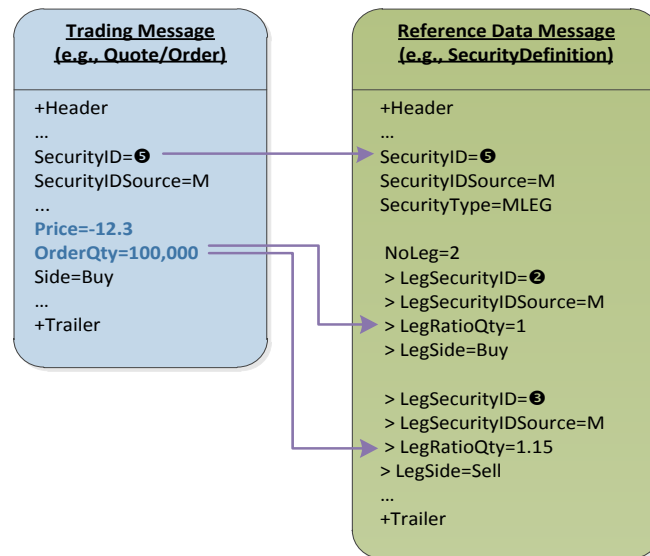
Multi-leg Instruments, such as spread and butterfly strategies, allow the buying and/or selling simultaneously of a number of different outright contracts (that traditionally could only be achieved by placing separate orders).

3.5.2.1 Multi-leg Instrument with Pre-defined Notional Amount Ratio between Legs

For Multi-leg Instruments where the notional amount ratio between each leg is fixed, the LegRatioQty(623) attribute should be used to define this ratio.

The following diagram describes the definition and usage of a Multi-leg Instrument with a pre-defined notional amount ratio between legs. In this diagram:

- Buyer buys 100,000 * 1 (OrderQty * LegRatioQty) notional amount of leg ②
- Buyer sells 100,000 * 1.15 (OrderQty * LegRatioQty) notional amount of leg ③



3.5.2.2 Multi-leg Instrument with Fixed Notional Amount for each Leg

For Quote Driven Models, the FIX attribute LegOrderQty(685) may be used in orders and quote requests to supply the notional amount of a leg in a Multi-leg Instrument.

If the LegOrderQty is present, it should be present in each one of the legs in the Multi-leg Instrument.

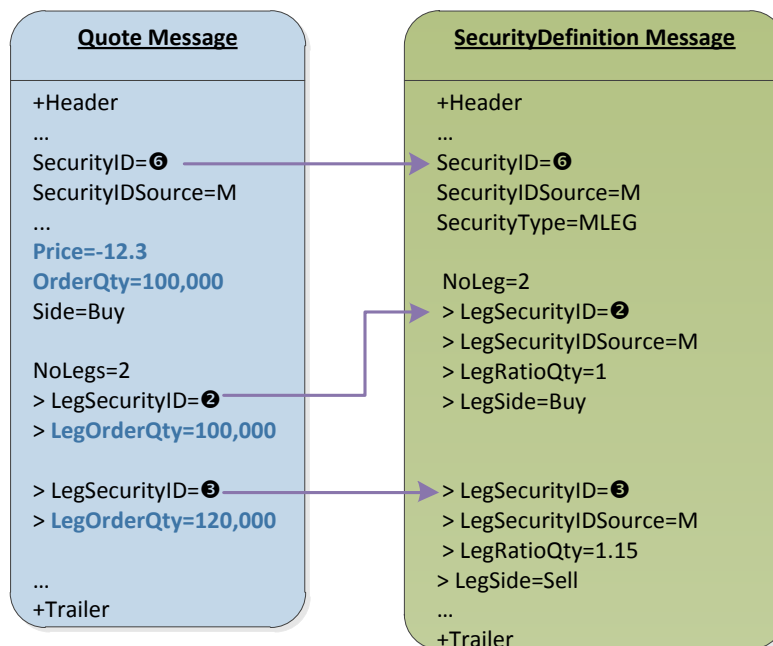
If the LegOrderQty value is present, its value takes precedence over calculations made using the LegRatioQty value (for further explanation of *notional amount calculations made using the LegRatioQty* see [Multi-leg Instrument with Pre-defined Notional Amount Ratio between Legs](#))

This LegOrderQty will typically be used in the following messages:

- QuoteRequest (MsgType=R)
- QuoteResponse (MsgType=AJ)

The following diagram describes the messages used for supplying the notional amount in a QuoteRequest message. In this diagram:

- Buyer wishes to buy 100,000 notional amount of leg ②
- Buyer wishes to sell 120,000 notional amount of leg ③ (Overrides: $100,000 * 1.15 \text{ OrderQty} * \text{LegRatioQty}$)



3.6 Reference Data Messages

This section describes how the Instrument identifier attributes are used in the FIX messages associated with the Reference Data section of the Best Practices document.

3.6.1 SecurityListRequest

SecurityListRequest (MsgTyp=x)	
Common-Instrument-Attributes	Required if SecurityListRequestType(559)=Symbol(0)

3.6.2 SecurityList , SecurityListUpdateReport

SecurityList (MsgTyp=y)	
SecurityListUpdateReport (MsgTyp=BK)	
> Common-Instrument-Attributes	
OR	
> Outright-Instrument-Attributes	
OR	
> Multi-Leg-Instrument-Attributes	

3.6.3 SecurityDefinition

SecurityList (MsgTyp=y)	
SecurityListUpdateReport (MsgTyp=BK)	
SecurityDefinition (MsgTyp=d)	
> Outright-Instrument-Attributes	
OR	
> Multi-Leg-Instrument-Attributes	

3.6.4 SecurityDefinitionRequest

SecurityDefinitionRequest (MsgTyp=c)	
Symbol(55)=<Required human readable name of the Instrument> SecurityType(167)=CDS/IRS SecurityXMLLen(1184)=<length> SecurityXML(1185)=<FpML/XML Payload> SecurityXMLSchema(1186)=<URI/URL of the schema>	Required for Non-Standard Outright Instrument
OR	
Symbol(55)=<Required human readable name of the Instrument> SecurityType(167)=MLEG NoLegs(555)= <Number of Legs> > LegSecurityID(602) > LegSecurityIDSource(603) <required> > LegSide(624)=< Buy Sell>	Required for Non-Standard Multi-leg The Reference IDs of all legs of a Multi-Leg Instrument in a Security Definition Request were exchanged in preceding message(s)

3.6.5 SecurityStatusRequest

SecurityStatusRequest Request (MsgTyp=e)	
Common-Instrument-Attributes	Required

3.6.6 SecurityStatus

SecurityStatusRequest (MsgTyp=f)	
Common-Instrument-Attributes	Required

3.7 Market Data Messages

This section describes how the Instrument identifier attributes are used in the FIX messages associated with the Market Data section of the Best Practices document.

3.7.1 MarketDataRequest

MarketDataRequest (MsgTyp=e)	
> Common-Instrument-Attributes	Required

3.7.2 MarketDataSnapshotFullRefresh

MarketDataSnapshotFullRefresh (MsgTyp=W)	
Common-Instrument-Attributes	Required

3.7.3 MarketDataIncrementalFullRefresh

MarketDataIncrementalFullRefresh (MsgTyp=X)	
> Common-Instrument-Attributes	Required

3.7.4 MarketDataRequestReject

MarketDataRequestReject (MsgTyp=Y)	
None	

3.8 Quote Contribution Messages

This section describes how the Instrument identifier attributes are used in the FIX messages associated with the Quote Contribution section of the Best Practices document.

3.8.1 Quote, QuoteCancel

Quote (MsgTyp=S)	
QuoteCancel (MsgTyp=Z)	
Common-Instrument-Attributes	Required
NoLegs(555)= <Number of Legs> > LegSymbol(600)	Optional

> LegSecurityID= < required if NoLegs exist> > LegSecurityIDSource < required if NoLegs exist > > LegOrderQty(685)=Quantity < required if NoLegs exist >	
--	--

3.8.2 QuoteAck, QuoteCancel

QuoteAck (MsgTyp=CW) QuoteCancel (MsgTyp=Z)	
Common-Instrument-Attributes	Required

3.8.3 MassQuote

MassQuote (MsgTyp=i)	
> Common-Instrument-Attributes	Required
> NoLegs(555)= <Number of Legs> >> LegSymbol(600) >> LegSecurityID= < required if NoLegs exist> >> LegSecurityIDSource < required if NoLegs exist > >> LegOrderQty(685)=Quantity < required if NoLegs exist >	Optional

3.8.4 MassQuoteAck

MassQuoteAck (MsgTyp=b)	
> Common-Instrument-Attributes	Required
> NoLegs(555)= <Number of Legs> >> LegSymbol(600) >> LegSecurityID= < required if NoLegs exist> >> LegSecurityIDSource < required if NoLegs exist > >> LegOrderQty(685)=Quantity < required if NoLegs exist >	Optional

3.9 Quotation/Negotiation and Quote Order Based Messages

This section describes how the Instrument identifier attributes are used in the FIX messages associated with the Quotation/Negotiation and the Quote Order Based sections of the Best Practices document.

3.9.1 QuoteRequest

QuoteRequest (MsgTyp=R)	
Common-Instrument-Attributes	Required for Standard Instrument
OR	
Common-Instrument-Attributes SecurityXMLLen(1184)=<Length> SecurityXML(1185)=<FpML/XML Payload> SecurityXMLSchema(1186)=<URI/URL of the schema>	Required for Non-Standard Outright Instrument
OR	
Common-Instrument-Attributes NoLegs(555)= <Number of Legs> > LegSymbol(600)	Required for Non-Standard Multi-leg Instruments Optional for Standard Multi-leg Instruments

<ul style="list-style-type: none"> > LegSecurityID(602) > LegSecurityIDSource(603) > LegOrderQty(685) <Optional> > LegRatioQty(623) <Optional> 	
<ul style="list-style-type: none"> > LegSecurityXMLLen(1871)=<optional> > LegSecurityXML(1872)=<optional> > LegSecurityXMLSchema(1873)=<optional> 	Required for Non-Standard Multi-leg Instruments where the leg is a Non-Standard Outright Instrument

3.9.2 Quote

Quote (MsgTyp=S)	
Common-Instrument-Attributes	Required
NoLegs(555)= <Number of Legs> > LegSymbol(600) > LegSecurityID(602) > LegSecurityIDSource(603) > LegOrderQty(685)	Required if the associated Quote Request contains NoLegs(555)

3.9.3 QuoteRequestReject, QuoteStatusReport

QuoteRequestReject (MsgTyp=AG) QuoteStatusReport (MsgTyp=AI)	
Common-Instrument-Attributes	Required
NoLegs(555)= <Number of Legs> > LegSymbol(600) > LegSecurityID(602) > LegSecurityIDSource(603) > LegOrderQty(685)	Optional

3.9.4 QuoteAck, QuoteCancel, Execution Ack

QuoteAck (MsgTyp=CW) QuoteCancel (MsgTyp=Z) ExecutionAck (MsgTyp=BN)	
Common-Instrument-Attributes	Required

3.9.5 QuoteResponse

QuoteResponse (MsgTyp=AJ)	
Common-Instrument-Attributes	Required for Standard Instrument
OR	
Common-Instrument-Attributes SecurityXMLLen(1184)=<Length> SecurityXML(1185)=<FpML/XML Payload> SecurityXMLSchema(1186)=<URI/URL of the schema>	Required for Non-Standard Outright Instrument
OR	
Common-Instrument-Attributes NoLegs(555)= <Number of Legs>	Required for Non-Standard Multi-leg Instruments

<ul style="list-style-type: none"> > LegSymbol(600) > LegSecurityID(602) > LegSecurityIDSource(603) > LegOrderQty(685) <Optional> > LegRatioQty(623) <Optional> 	Optional for Standard Multi-leg Instruments
<ul style="list-style-type: none"> > LegSecurityXMLLen(1871)=<optional> > LegSecurityXML(1872)=<optional> > LegSecurityXMLSchema(1873)=<optional> 	Required for Non-Standard Multi-leg Instruments where the leg is a Non-Standard Outright Instrument

3.9.6 ExecutionReport

ExecutionReport (MsgTyp=8)	
Common-Instrument-Attributes	Required
NoLegs(555)=<Number of Legs> > LegSymbol(600) > LegSecurityID(602) > LegSecurityIDSource(603)	Required for Multi-leg Instruments

3.10 Central Order Book Messages

This section describes how the Instrument identifier attributes are used in the FIX messages associated with the Reference Data section of the Best Practices document.

- The messages in this section do not contain the Instrument definition. Instead they refer to the instrument using the SecurityID attribute. In cases where an order is placed for a Non-Standard Instrument the workflow should contain a SecurityDefinitionRequest followed by a SecurityDefinition prior to submitting the order.

3.10.1 NewOrderSingle, OrderCancelRequest, OrderMassCancelRequest, OrderCancelReplaceRequest, OrderStatusRequest

NewOrderSingle (MsgTyp=D) OrderCancelRequest (MsgTyp=F) OrderMassCancelRequest (MsgTyp=q) OrderCancelReplaceRequest (MsgTyp=G) OrderStatusRequest (MsgTyp=H)	
Common-Instrument-Attributes	Required

3.10.2 OrderCancelReject, OrderMassCancelReport

OrderCancelReject (MsgTyp=9) OrderMassCancelReport (MsgTyp=r)	
Common-Instrument-Attributes	Required

3.10.3 ExecutionReport

- There should be individual Execution Reports for each of the Outright Instruments nested within the traded Multi-leg Instrument
- There should be an overall Execution Report to describe the traded Multi-leg Instrument as a whole
- For Nested Multi-leg Instruments, it is optional whether there are Execution Reports for the intermediate strategies

ExecutionReport (MsgTyp=8)	
Common-Instrument-Attributes	Required

4 Market Conventions

4.1 Side/Direction of Trade

4.1.1 Side/Direction of IRS Trade (Outright Instrument)

- For IRS based on fixed rate vs. floating rate:
 - The buyer of an IRS is the fixed rate payer (as defined in the IRS contract)
 - The seller of an IRS is the fixed rate receiver (as defined in the IRS contract)

4.1.2 Side/Direction of CDS Trade (Outright Instrument)

- For a typical (“default”) CDS outright instrument:
 - The buyer of a CDS is the protection buyer
 - The seller of a CDS is the protection seller
- The FpML pre-trade instrument payload contains an element called: BuyerConvention, this element may have the values of ‘Protection’ or ‘Risk’ to indicate the attribute which is being bought
- The FIX Side(54) tag determines the side of the trade whilst the FpML pre-trade instrument payload ‘BuyerConvention’ attribute determines Risk or Protection of the product
- The following table lists the different combinations of the above fields and indicates the attribute that is being bought:

FIX Side(54)	FpML: BuyerConvention	Result
Buy	Protection (default)	Buy protection (sell risk)
Sell	Protection (default)	Sell protection (buy risk)
Buy	Risk	Buy risk (sell protection)
Sell	Risk	Sell risk (buy protection)

4.2 LastQty(32) of multi-leg instrument

- When trading multi-leg instruments based on an IRS, there is a market convention to populate LastQty(32) in the ExecutionReport(8) message as follows:
 - For switches: $\text{LastQty}(32) = \text{LegLastQty}(1418) / \text{LegRatioQty}(623)$ of the far leg
 - For butterflies: $\text{LastQty}(32) = \text{LegLastQty}(1418) / \text{LegRatioQty}(623)$ of the body
- When trading multi-leg instruments based on an IRS and treasury, where there is a single IRS leg involved, there is a market convention to populate LastQty(32) in the ExecutionReport(8) message as follows:
 - $\text{LastQty}(32) = \text{LegLastQty}(1418) / \text{LegRatioQty}(623)$ of the IRS leg
- Some multi-leg instruments may have additional values calculated from the duration (DV01) that is involved in deriving the notional amount of each leg. Such values may or may not be included in the FIX messages. (See scenarios [MC1](#) and [MC2](#))

4.3 Default structure for IRS vs. treasury multi-leg instrument

- When trading multi-leg instruments based on an IRS and treasury the common convention is that the buyer buys both the IRS and the treasury, whilst the seller sells both the IRS and the treasury

4.4 Price type convention

4.4.1 Price Type of IRS Trade

- When trading an outright IRS the default value of PriceType(423)=Yield(9)
- When trading multi-leg instruments based on an IRS the default value of PriceType(423) = Spread (basis points spread)(6)

4.4.2 Price Type of CDS Trade

The PriceType(423) of an outright CDS is one of the following:

- Percentage(1) – percentage of PAR or “dollar price”
- Spread(6) – basis points spread
- Yield(9)

When trading multi-leg instruments based on an CDS the default value of PriceType(423) = Spread (basis points spread)(6)

4.5 Factor based on DV01

- In the case where the DV01 changes intraday, the factor based on DV01 may be supplied as part of market data in either MarketDataSnapshotFullRefresh(W) or MarketDataIncrementalRefresh(X) message where MDEntryType(269)=’S’ [Swap Value Factor (SVP) for swaps cleared through a central counterparty (CCP)].
- In the case where the DV01 does not change intraday, the factor based on DV01 should be supplied in one or more of the following messages:
 - As part of the Instrument component in tag: Factor(228)
 - As part of the InstrumentLeg component in tag: LegFactor(253)
- The product of (Quantity * Price * Factor) is sometimes denoted as the nominal value of an instrument

4.6 Scenario MC1 – Example of IRS Gap Spread Definition and Usage

This scenario demonstrates the usage of the Factor(228) based on DV01 which is provided in a reference data message. The scenario starts when the Execution Venue sends a new switch instrument definition to the Dealer having LegFactor(253) for each of the legs. The Dealer places an order for the switch, which is partially filled, then filled. The scenario concentrates on the values and relationship between the order quantity, the last traded quantities of the switch level and of each of the legs, the LegRatioQty(623) and the LegFactor(253) of each of the legs.

The switch instrument in this scenario is a standard 2Yx5Y LIBOR USD 3M spread that combines the two legs:

Attribute	LEG1 – Far leg	LEG2 – Near leg
LegSecurityID(602)	②	③
Description	5 year standard IRS USD LIBOR	2 year standard IRS USD LIBOR
LegSide(624)	Buy(1)	Sell(2)
LegRatioQty(623)	1.0	1.0
LegFactor(253)	1.0	0.4195804196
LegLastQty(1418)	$LegLastQty(1418) = \frac{LastQty(32) * LegRatio(623)}{LegFactor(253)}$	

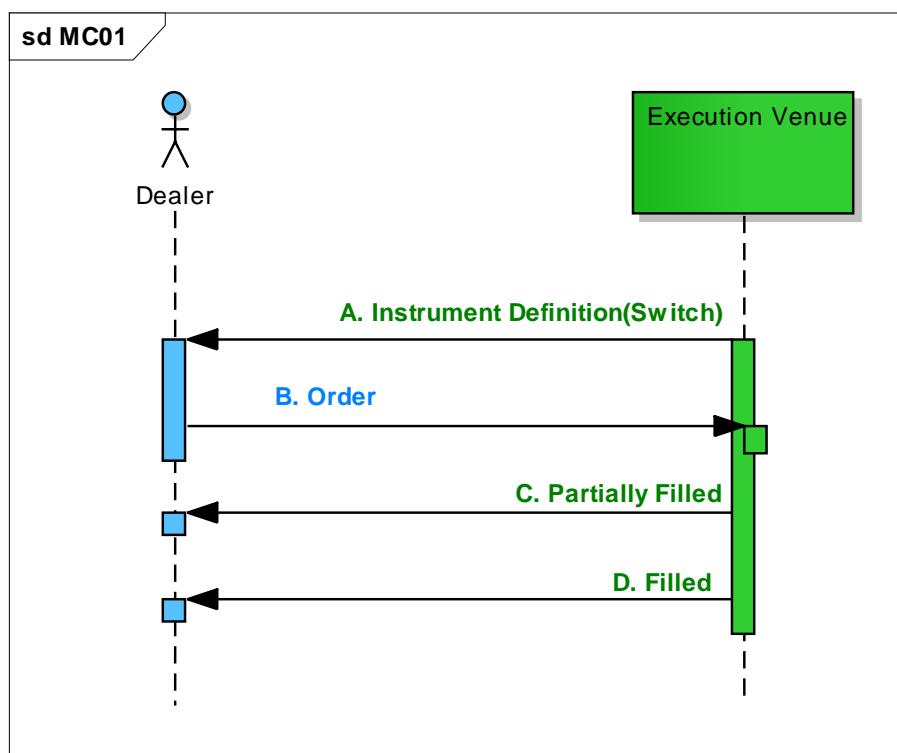


Figure 1: Scenario MC1- Example of IRS Gap Spread Definition and Usage

Model Flow

The following table illustrates the flows expected when communicating with an Execution Venue implementing FIX 5.0 SP2. In order to simplify the scenario, some required FIX messages and attributes have been omitted.

Model FIX 5.0				
Pre-condition: The definition of outright IRS instruments ❷ ["5Y USD LIBOR"] and ❸ ["2Y USD LIBOR"] have been sent and can be referenced using tag LegSecurityID(602)				
(A) Execution Venue sends switch definition	Dealer	←	d – SecurityDefinition Symbol(55)="[N/A]" SecurityID(48)= ❶ SecurityType(167)=MLEG NoLegs(555)=2 <hr/> LEG1 LegSymbol (600)= "5Y USD LIBOR" LegSecurityID(602)= ❷ LegRatioQty(623)=1 LegSide (624)=Buy(1) LegFactor(253)=1.0 <hr/> LEG2 LegSymbol (600)= "2Y USD LIBOR" LegSecurityID(602)= ❸ LegRatioQty(623)=1 LegSide (624)=Sell(2) LegFactor(253)= 0.4195804196	Execution Venue
(B) Dealer place an an order		→	D – NewOrderSingle ClOrdId(11)=❹ OrderQty(38)=1000000 SecurityID(48)= ❶	
(C) Partially filled		←	8 – ExecutionReport ClOrdID(11)=❹ OrderId(37)=❺ SecurityID(48)= ❶ ExecID=❻ OrderQty(38)=1000000 LastQty(32)=300000 CumQty(14)=300000 LeavesQty(151)=700000 ExecType(150)=Trade(F) OrdStatus(39)=PartiallyFilled(1) NoLegs(555)=2 <hr/> LEG1 LegSymbol (600)= "5Y USD LIBOR" LegSecurityID(602)= ❷ LegRatioQty(623)=1 LegSide (624)=Buy(1) LegFactor(253)=1.0 LegLastQty(1418) =300000 <hr/> LEG2 LegSymbol (600)= "2Y USD LIBOR" LegSecurityID(602)= ❸ LegRatioQty(623)=1 LegSide (624)=Sell(2) LegFactor(253)= 0.4195804196 LegLastQty(1418) =715000 [based on: 300000*1.0/0.4195804196]	

Model FIX 5.0				
Pre-condition: The definition of outright IRS instruments ❷ ["5Y USD LIBOR"] and ❸ ["2Y USD LIBOR"] have been sent and can be referenced using tag LegSecurityID(602)				
(D) Filled		←	8 – ExecutionReport ClOrdID(11) = ❹ OrderID(37)=❺ SecurityID(48)= ❶ ExecID=❽ OrderQty(38)=1000000 LastQty(32)=700000 CumQty(14)= 1000000 LeavesQty(151)=0 ExecType(150)=Trade(F) OrdStatus(39)= Filled(2) NoLegs(555)=2	
			LEG1	LegSymbol (600)= "5Y USD LIBOR" LegSecurityID(602)= ❷ LegRatioQty(623)=1 LegSide (624)=Buy(1) LegFactor(253)=1.0 LegLastQty(1418) =700000
			LEG2	LegSymbol (600)= "2Y USD LIBOR" LegSecurityID(602)= ❸ LegRatioQty(623)=1 LegSide (624)=Sell(2) LegFactor(253)= 0.4195804196 LegLastQty(1418) = 1668333 [based on: 700000*1.0/0.4195804196]

Table 1: Scenario MC1- Example of IRS Gap Spread Definition and Usage

4.7 Scenario MC2 - Example of Duration Neutral IRS Butterfly

This scenario demonstrates the usage of the Factor(228) based on DV01 which is provided in market data messages. The scenario starts after the butterfly has been defined. The dealer places an order for the butterfly, which is partially filled, then filled. The scenario concentrates on the values and relationship between the order quantity, the last traded quantities of the parent level and of each of the legs, the LegRatioQty(623) and the LegFactor(253) of each of the legs.

The instrument in this scenario is a standard 2Yx5Yx10Y USD LIBOR 3M duration neutral butterfly that combines a body leg and two wings:

Attribute	Wing1	Body	Wing2
ID	③	②	④
Description	2 year standard IRS USD LIBOR	5 year standard IRS USD LIBOR	10 year standard IRS USD LIBOR
LegSide	Sell(2)	Buy(1)	Sell(2)
LegRatioQty(623)	0.5	1.0	0.5
Factor value at time of execution	0.4195804196	1.0	1.5464172605
LegLastQty	$LegLastQty(1418) = \frac{LastQty(32) * LegRatio(623)}{Factor}$		

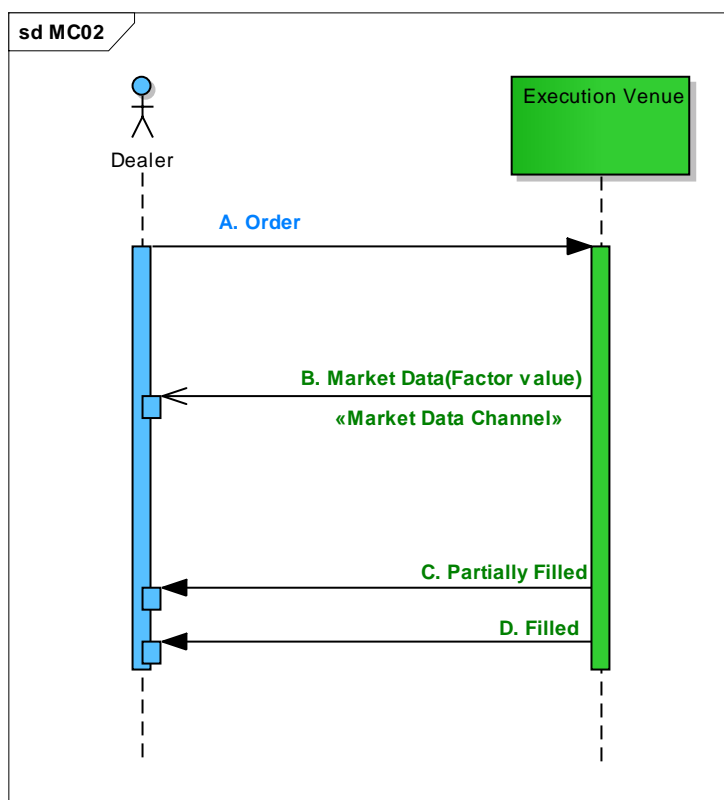


Figure 2: Scenario MC2 - Example of Duration Based IRS Butterfly

Model Flow

The following table illustrates the flows expected when communicating with an Execution Venue implementing FIX 5.0 SP2. In order to simplify the scenario, some required FIX messages and attributes have been omitted.

Model FIX 5.0				
Pre-condition: The definition of the butterfly ❶, as well as the outright IRS instruments ❷ ["5Y USD LIBOR"], ❸ ["2Y USD LIBOR"] and ❹ ["10Y USD LIBOR"] have been sent and can be referenced using tag LegSecurityID(602)				
(A) Dealer place an an order	Dealer	→	D – NewOrderSingle ClOrdId(11)=❶ OrderQty(38)=1000000 SecurityID(48)= ❶	Execution Venue
(B) Market data values Note: This message may be sent on a different channel		←	X – MarketDataIncrementalRefresh MDReqID(262) = ❸ NoMDEntires(268)=3 MDUpdateAction(279)=New(0) Symbol(55)= "2Y USD LIBOR" (WING1) SecurityID(48)= ❸ MDEntryType(269)=Swap Value Factor(S) MDEntryPx(270)= 0.4195804196	
			MDUpdateAction(279)=New(0) Symbol(55)= "5Y USD LIBOR" (BODY) SecurityID(48)= ❷ MDEntryType(269)=Swap Value Factor(S) MDEntryPx(270)= 1.0	
			MDUpdateAction(279)=New(0) Symbol(55)= "10Y USD LIBOR" (WING2) SecurityID(48)= ❹ MDEntryType(269)=Swap Value Factor(S) MDEntryPx(270)= 1.5463917526	
(C) Partially filled		←	8 – ExecutionReport ClOrdID(11) =❹ OrderID(37)=❶ SecurityID(48)= ❶ ExecID=❷ OrderQty(38)=1000000 LastQty(32)=300000 CumQty(14)=300000 LeavesQty(151)=700000 ExecType(150)=Trade(F) OrdStatus(39)=PartiallyFilled(1) NoLegs(555)=3	
			WING1 LegSymbol (600)= "2Y USD LIBOR" LegSecurityID(602)= ❸ LegRatioQty(623)=0.5 LegSide (624)= Sell(2) LegLastQty(1418) = 357500 [based on: 300000*0.5/0.4195804196]	
			BODY LegSymbol (600)= "5Y USD LIBOR" LegSecurityID(602)= ❷ LegRatioQty(623)=1 LegSide (624)=Buy(1) LegFactor(253)= 1.0 LegLastQty(1418) = 300000	

Model FIX 5.0				
Pre-condition: The definition of the butterfly ❶, as well as the outright IRS instruments ❷ ["5Y USD LIBOR"], ❸ ["2Y USD LIBOR"] and ❹ ["10Y USD LIBOR"] have been sent and can be referenced using tag LegSecurityID(602)				
(D) Filled		←	WING2	LegSymbol (600)= "10Y USD LIBOR" LegSecurityID(602)= ❹ LegRatioQty(623)=0.5 LegSide (624)=Sell(2) LegLastQty(1418) =97000 [based on: 300000*0.5/1.5463917526]
			8 – ExecutionReport ClOrdID(11) =❹ OrderId(37)=❷ SecurityID(48)= ❶ ExecID=❸ OrderQty(38)=1000000 LastQty(32)=700000 CumQty(14)= 1000000 LeavesQty(151)=0 ExecType(150)=Trade(F) OrdStatus(39)= Filled(2) NoLegs(555)=2	
			WING1	LegSymbol (600)= "2Y USD LIBOR" LegSecurityID(602)= ❸ LegRatioQty(623)=0.5 LegSide (624)= Sell(2) LegLastQty(1418) = 1668334 [based on: 700000*0.5/1.5463917526]
			BODY	LegSymbol (600)= "5Y USD LIBOR" LegSecurityID(602)= ❷ LegRatioQty(623)=1.0 LegSide (624)= Buy(1) LegLastQty(1418) =700000
			WING2	LegSymbol (600)= "10Y USD LIBOR" LegSecurityID(602)= ❹ LegRatioQty(623)=0.5 LegSide (624)=Sell(2) LegLastQty(1418) =226333 [based on: 700000*0.5/1.5463917526]

Table 2: Scenario MC2 - Example of Duration Based IRS Butterfly

5 Regulatory Changes

The sections below detail a number of regulatory requirements mandated under Dodd-Frank and associated best practices recommendations for supporting these regulatory requirements using the FIX Protocol.

In June 2013, the US Commodity Futures Trading Commission (CFTC) published the ‘Core Principals and Other Requirements for Swap Execution Facilities’. These final rules provide some additional information relevant to the topics discussed in this section and can be found at the following links:

- <http://www.cftc.gov/ucm/groups/public/@lrfederalregister/documents/file/2013-12242a.pdf>
- <http://www.cftc.gov/ucm/groups/public/@lrfederalregister/documents/file/2013-12250a.pdf>

5.1 Unique Swap Identifier (USI)

According to the CFTC, Dodd-Frank Act’s 17 CFR Part 45: “A new USI is assigned by the Swap Dealer (SD), Major Swap Participant (MSP) or Swap Execution Facility (SEF) at the time of execution for the trade”

- This means the USI is a required field in the ExecutionReport(35=8), ExecutionAcknowledgement(35=BN) and TradeCaptureReport(35=AE) messages

The CFTC have published [technical standards for the format of USI](#):

- Namespace – consists of 10 alpha-numeric (uppercase) characters [3 character prefix, followed by 7 character identifier]
- Transaction Identifier - consists of maximum 32 alpha-numeric (uppercase) characters

Tag	FieldName		Data Type	FIXML	Req'd	Description
RegulatoryTradeIDGrp component				<RegTrdID>		
1907	NoRegulatoryTradeIDs		NumInGroup	-		Number of trade identifiers in the repeating group.
→	1903	RegulatoryTradeID	String	@ID	Y	Trade identifier required by government regulators or other regulatory organizations for regulatory reporting purposes. For example, unique swap identifier (USI) as required by the U.S. Commodity Futures Trading Commission.
→	1905	RegulatoryTradeIDSource	String	@Src	Y	Identifies the reporting entity that originated the value in RegulatoryTradeID(1903) (“Namespace”). The reporting entity identifier may be a regulator assigned value for the entity.
→	1904	RegulatoryTradeIDEvent	Int	@Evnt	N	Identifies the event which caused the origination of the identifier. When more than one event is the cause, use the higher enumeration value. For example, if the identifier is originated due to an allocated trade which was cleared and reported, use the enumeration value 2(Clearing). Values: 0 = Initial block trade 1 = Allocation (or determination that the block trade will not be further allocated) 2 = Clearing 3 = Compression 4 = Novation 5 = Termination
→	1906	RegulatoryTradeIDType	Int	@Typ	N	Specifies the type of trade identifier provided in RegulatoryTradeID(1903), within the context of the hierarchy of

						trade events. Values: 0 = Current (default if not specified) 1 = Previous (The previous trade's identifier when reporting a cleared trade or novation of a previous trade.) 2 = Block (The block trade's identifier when reporting an allocated subtrade) 3 = Related (The related trade identifier when reporting a mixed swap.)
--	--	--	--	--	--	--

Additionally FPL have published a [white paper to support the USI](#)

- Based on the recommendations of this paper, we will propose a gap analysis to the FPL Global Technical Committee (GTC) to add the following repeating component to the ExecutionReport(35=8), ExecutionAcknowledgement(35=BN) and TradeCaptureReportAck(35=AR) messages. This component already exist in the TradeCaptureReport(35=AE) message.
- Information on the related Legal Entity Identifier (LEI) is available in [Section 2.3.1](#)

For an example of usage see [Scenario RC1 – Example of Flow for ‘Required Transaction’](#) and [Scenario RC2 – Example of Flow for ‘Permitted Transaction’](#)

5.2 Mid-Market Mark

On May 1st 2013, the Business Conduct Standards for swap dealers (SD) and major swap participants (MSP) went into effect. The official document is available to view at the link below. One of the requirements resulting from this document is the need for swap dealers and major swap participants to provide counterparties with pre-trade mid-market marks.

<http://www.cftc.gov/ucm/groups/public/@lrfederalregister/documents/file/2012-1244a.pdf>

For uncleared swaps, a SD/MSP must provide daily marks:

- These must be calculated at mid-market, exclusive of amounts for profit, credit reserve, funding, liquidity or any other costs or adjustments
- A SD/MSP must also disclose the methodology and assumptions used to prepare the mark, and other information to ensure fair and balanced communications
- Marks must be “as of the close of business” or as otherwise agreed in writing

The mid-market price should be sent in the MidPx(631) field. We suggest adding the MidPx(631) field to the following messages:

- ExecutionReport(35=8)
- TradeCaptureReport(35=AE)

We also suggest adding the MidPx(631) field to the following component:

- QuotReqGrp

Note that this field already exists in the Quote(35=S), QuoteResponse(35=AJ) and QuoteStatusReport(35=AI) messages. Additionally, the market data messages also provide support for the mid-price.

We also suggest adding the new field, **LegMidPx(2346)** to the following components:

- InstrmtLegExecGrp
- LegQuotGrp
- LegQuotStatGrp
- QuotReqLegsGrp
- TrdInstrmtLegGrp

Note that for uncleared OTC swaps, the LegMidPx(2346) and MidPx(631) fields are mutually exclusive, if the LegMidPx(2346) field is populated then the MidPx(631) field is not required.

For an example of usage see [Scenario RC2 – Example of Flow for ‘Permitted Transaction’](#)

5.3 US vs. Non-US Person

It is important to know if the person executing the trade is considered a US or Non-US person. A person located outside of the US may be considered a US person for regulatory purposes.

In the ‘Parties’ component, we recommend using PartySubIDType(803) = U.S. person(48). This information, which identifies the counterparty doing the trade with the dealer as either a US or Non-US person, is typically sent from the execution venue to the dealer in a QuoteRequest(35=R) or ExecutionReport(35=8) message.

For an example of usage see [Scenario RC1 – Example of Flow for ‘Required Transaction’](#)

5.4 Required vs. Permitted Transaction

The regulations make a distinction between:

- Required Transactions - transactions subject to the trade execution mandate and not block trades
- Permitted Transactions - transactions not subject to the clearing and trade execution mandates, illiquid or bespoke swaps, or block trades

Execution venues may need to communicate whether a transaction is ‘required’ or ‘permitted’. This information is typically sent from the execution venue to the dealer in either the QuoteRequest(35=R) or ExecutionReport(35=8) message.

We recommend adding a new optional field, **RegulatoryTransactionType(2347)** to the core FIX specification with the following enumerations:

0 = None (default if not specified)

The transaction does not fall under any special regulatory rule or mandate.

1 = Swap Execution Facility (SEF) required transaction

The transaction is a "Required transaction" under Dodd-Frank Act SEF Rules. "Required" transactions are subject to the trade execution mandate under section 2(h)(8) of the CEA and are not block trades.

2 = Swap Execution Facility (SEF) permitted transaction

The transaction is a "Permitted transaction" under Dodd-Frank Act SEF Rules. "Permitted" transactions are not subject to the clearing and trade execution mandates, illiquid or bespoke swaps, or block trades.

We suggest adding this new field to the following message:

- ExecutionReport(35=8)
- QuoteRequest(35=R)
- QuoteResponse(35=AJ)
- TradeCaptureReport(35=AE)

For an example of usage see [Scenario RC1 – Example of Flow for ‘Required Transaction’](#) and [Scenario RC2 – Example of Flow for ‘Permitted Transaction’](#)

5.5 Block Trade

The regulations mention different trading methods and reporting requirements for block trades of swaps. It is important for the execution venues to identify block trades and communicate this information to the dealer. The execution venue will typically send this information pre-trade in a QuoteRequest(35=R) message.

The best practices recommend using the field TrdType(828)= Block swap trade(58) to indicate that the message refers to a block trade. As such, we recommend adding TrdType(828) to the following messages:

- ExecutionReport(35=8)
- QuoteRequest(35=R)
- QuoteResponse(35=AJ)
- TradeCaptureReport(35=AE)

For an example of usage see [Scenario RC2 – Example of Flow for ‘Permitted Transaction’](#)

5.6 Reporting Party Indicator

Of the two parties involved in a trade, one will be considered the reporting party who is responsible for providing reporting updates throughout the life of the trade. In most cases the reporting party will be the sell-side but may also be the buy-side. This determination is made by the execution venue and the information is typically sent in a QuoteRequest(35=R) or ExecutionReport(35=8) message.

The best practices recommends using PartySubIDType(803)=Reporting entity indicator(49) to identify the reporting party.

For an example of usage see [Scenario RC1 – Example of Flow for ‘Required Transaction’](#)

5.7 Intermediary Reporting to CCP

According to the regulations, most swaps are cleared through a Central Counter Party (CCP). The CCP can be identified in the ‘Parties’ component with PartyRole(452)=Clearing Organization(21).

Some execution venues and swap dealers, report their trades to the CCP via an intermediary. In such a case there is a requirement for the venue to communicate the intermediary to the dealer. The best practices recommends identifying the intermediary in the ‘Parties’ component with PartyRole(452)= Reporting Intermediary(72) [as well as PartyID(448)=“name of intermediary”]. This information will typically be sent from the venue to the dealer in a QuoteRequest(35=R), ExecutionReport(35=8) or TradeCaptureReport(35=AE) message.

For an example of usage see [Scenario RC1 – Example of Flow for ‘Required Transaction’](#)

5.8 Scenario RC1 – Example of Flow for ‘Required Transaction’

The following diagram describes the flow of a ‘Required’ transaction (see [Required vs. Permitted Transactions](#) above).

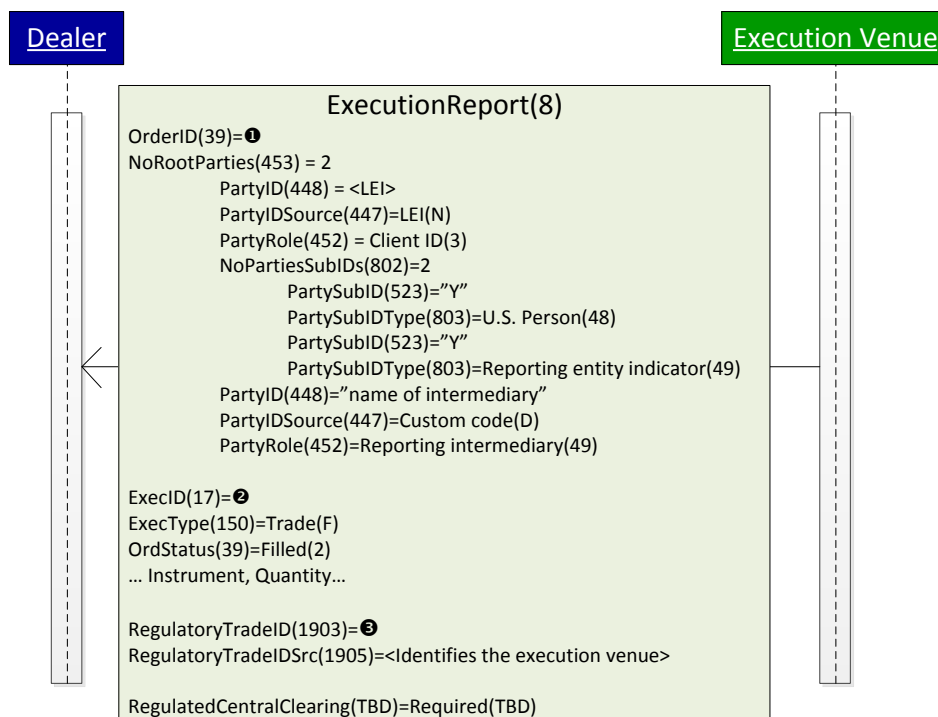


Figure 3: Flow of ‘Required Transaction’

5.9 Scenario RC2 – Example of Flow for ‘Permitted Transaction’

The following diagram describes the flow of a ‘Permitted’ transaction (see [Required vs. Permitted Transactions](#) above).

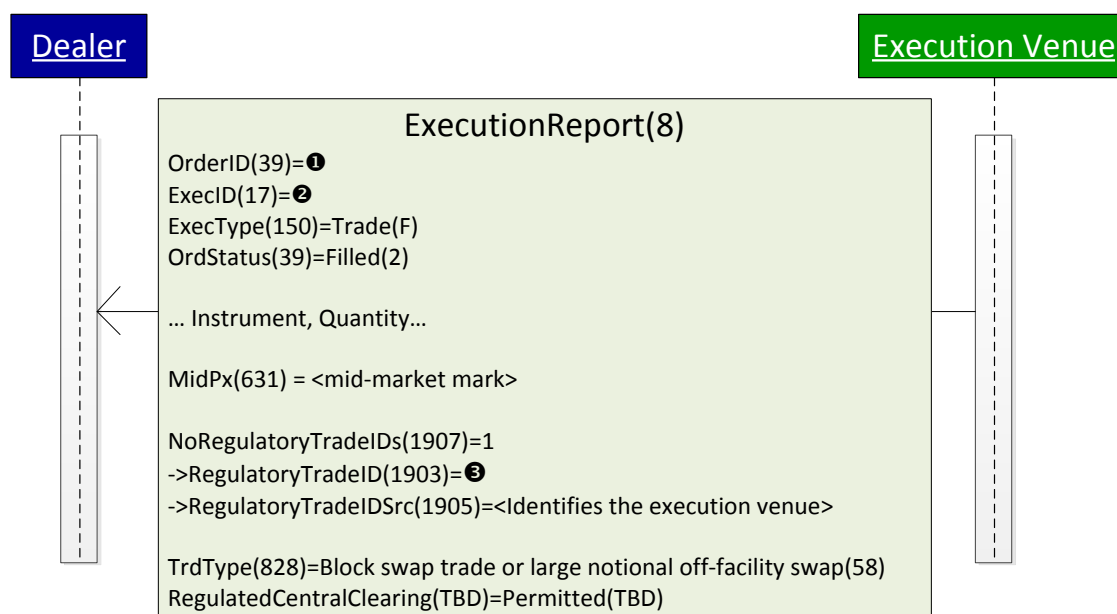


Figure 4: Flow of ‘Permitted Transaction’

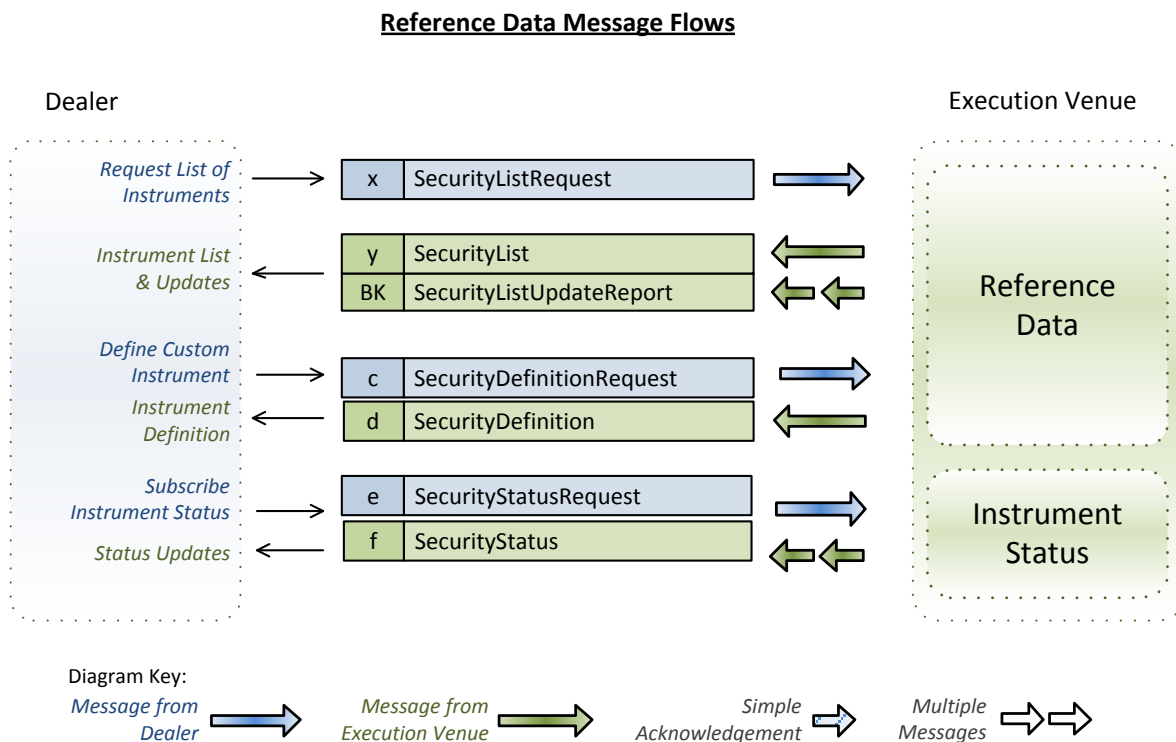
6 Pre-Trade - Reference Data

This section describes messages relevant for access to reference data to enable trading. There are 3 scenarios covered:

- Querying information about available instruments
- Querying status of instruments
- Definition of custom instruments (e.g. spread trades between two instruments)

6.1 Overview diagram

The following diagram illustrates the FIX messages and the Workflows described in this chapter.



6.2 Message Flows Summary

The following scenarios illustrate the use of these messages.

Query Reference Data

Scenario	Description
SL1	Dealer Requests Snapshot of Instruments, Execution Venue Returns Snapshot
SL2	Dealer Requests Snapshot of Instruments, Execution Venue Returns Snapshot (fragmented)
SL3	Dealer Subscribes to Instruments Reference Data, Execution Venue Returns Snapshot and Updates
SL4	Dealer Cancels Instrument Reference Data Subscription
SL5	Dealer Requests Snapshot or Subscription of Instruments Reference Data, Execution Venue Returns Error
SL6	Dealer Cancels Instrument Reference Data Subscription, Execution Venue Rejects

Query Status

Scenario	Description
SS1	Dealer Requests Instrument Status, Execution Venue Returns Status
SS2	Dealer Subscribes to Instrument Status, Execution Venue Returns Snapshot and Updates
SS3	Dealer Requests Instrument Status, Execution Venue Returns Error

Instrument Definition

Scenario	Description
SD1	Dealer Sends Instrument Definition Request, Execution Venue Confirms
SD2	Dealer Sends Instrument Definition Request, Execution Venue Rejects
SD3	Execution Venue Publishes Instrument Definition

6.3 Scenario SL1 – Dealer Requests Snapshot of Instruments, Execution Venue Returns Snapshot

This scenario occurs when the Dealer requests a snapshot of the instruments known to the Execution Venue and the Execution Venue returns the data. A snapshot is a one-time response with the available data at the time of the response. Note that the returned instruments are not necessarily tradable at the time of the response. Secondly note in most cases, the Dealer will request the instrument snapshot at the beginning of the trading day and will cache the instrument list. The data is returned in a single SecurityList message. See [Scenario SL2](#) for a fragmented SecurityList message where the SecurityList message size may exceed the maximum size of a message. In the SecurityList message in this scenario, the LastFragment(893) tag should not exist, whilst in SL2 the LastFragment(893) tag is required in each SecurityList message.

The Dealer may add filtering criteria to his request. The result may be an empty security list which indicated by SecurityRequestResult(560) = NoInstrumentFound(2) .

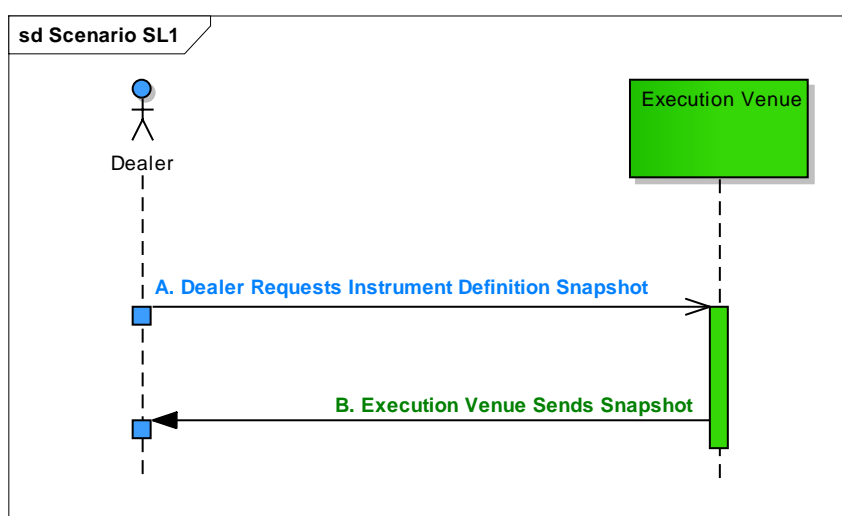


Figure 5: Scenario SL1 – Dealer Requests Snapshot of Instruments, Execution Venue Returns Snapshot

Model Flow

The following table illustrates the flows expected when communicating with an Execution Venue implementing FIX 5.0 SP2.

Model FIX 5.				
(A) Dealer Requests Instrument Snapshot	Dealer	→	x – SecurityListRequest SecurityReqID(320)= ❶ SecurityListRequestType(559)=required e.g. AllSecurities(4) SubscriptionRequestType(263)=Snapshot(0)	Execution Venue
(B) Execution Venue Sends Snapshot		←	y – SecurityList SecurityReqID(320)= ❶ SecurityResponseID(322)=<Execution Venue assigned ID> SecurityRequestResult(560)=ValidRequest(0) NoRelatedSym(146)=<count> Symbol(55)=< human readable name of the instrument > SecurityID(48)=<ID > SecurityIDSource(22)=<required> SecurityType(167)=<required> <i>Additional instrument attributes</i>	

Table 3: Scenario SL1 – Dealer Requests Snapshot of Instruments, Execution Venue Returns Snapshot

Note: It is expected that the Dealer's SecurityReqID is unique throughout the session. Best practice is that a new SecurityReqID should be used for subsequent request throughout the lifetime of the FIX session. If a second request using the SecurityReqID is made while the previous request is still in active the request will be rejected.

6.4 Scenario SL2 – Dealer Requests Snapshot of Instruments, Execution Venue Returns Snapshot (fragmented)

This scenario occurs when the Dealer requests a snapshot of the instruments known to the Execution Venue and the Execution Venue returns the data. The data is returned as a number of fragmented messages (for example to avoid exceeding the maximum size of a message). [See Scenario SL1.](#)

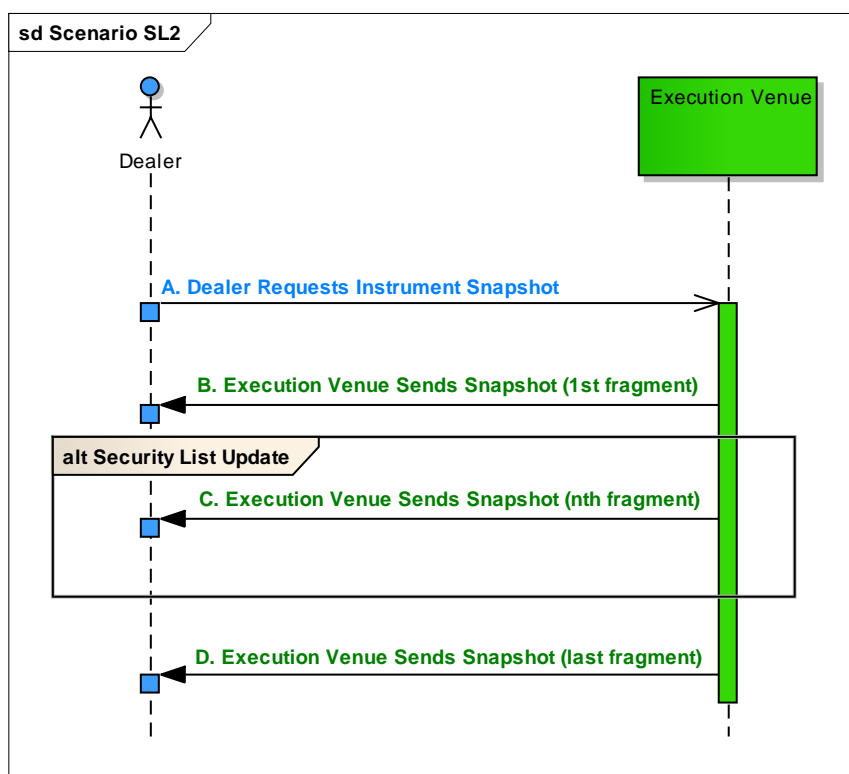


Figure 6: Scenario SL2 – Dealer Requests Snapshot of Instruments, Execution Venue Returns Snapshot (fragmented)

Model Flow

The following table illustrates the flows expected when communicating with an Execution Venue implementing FIX 5.0 SP2.

Model FIX 5.0				
(A) Dealer Requests Instrument Snapshot	Dealer	→	x – SecurityListRequest SecurityReqID(320)= ❶ SecurityListRequestType(559)=AllSecurities(4) SubscriptionRequestType(263)=Snapshot(0)	Execution Venue
(B) Execution Venue Sends Snapshot (1st fragment)		←	y – SecurityList SecurityReqID(320)= ❶ SecurityResponseID(322)=<Execution Venue assigned ID. In the event of fragmented SecurityList messages all fragmented messages us this sameSecurityResponseID> SecurityRequestResult(560)=ValidRequest(0) TotNoRelatedSym(393)=< Total count of securities> LastFragment(893)= NotLastMessage(N) NoRelatedSym(146)=<count> Symbol(55)=< human readable name of the instrument > SecurityID(48)=<ID > SecurityIDSource(22)=<required> SecurityType(167)=<required> <i>Additional instrument attributes</i>	
(C) Execution Venue Sends Snapshot (nth fragment)		↑ ↑ ↑	y – SecurityList SecurityReqID(320)= ❶ SecurityResponseID(322)=<Execution Venue assigned ID shared by all Fragemented messages for a request – see above> SecurityRequestResult(560)=ValidRequest(0) TotNoRelatedSym(393)=<Total count of securities> LastFragment(893)= NotLastMessage(N) NoRelatedSym(146)=<count> Symbol(55)=< human readable name of the instrument > SecurityID(48)=<ID > SecurityIDSource(22)=<required> SecurityType(167)=<required> <i>Additional instrument attributes</i>	
(D) Execution Venue Sends Snapshot (last fragment)		←	y – SecurityList SecurityReqID(320)= ❶ SecurityResponseID(322)=<Execution Venue assigned ID shared by all Fragemented messages for a request – see above> SecurityRequestResult(560)=ValidRequest(0) TotNoRelatedSym(393)=<Total count of securities> LastFragment(893)= LastMessage(Y) NoRelatedSym(146)=<count> Symbol(55)=< human readable name of the instrument > SecurityID(48)=<ID > SecurityIDSource(22)=<required> SecurityType(167)=<required> <i>Additional instrument attributes</i>	

Table 4: Scenario SL2 – Dealer Requests Snapshot of Instruments, Execution Venue Returns Snapshot (fragmented)

6.5 Scenario SL3 – Dealer Subscribes to Instruments Reference Data, Execution Venue Returns Snapshot and Updates

This scenario occurs when the Dealer requests to subscribe to the set of instruments known to the Execution Venue. A snapshot and updates are returned.

The scenario is common for Execution Venues that support instrument list updates during their sessions or during their trading day. [See notes in Scenario SL1.](#)

The snapshot results may be fragmented [see notes in Scenario SL2.](#)

Each SecurityListUpdateReport (SLUR) may contain a list of instruments to be updated. Each instrument in the SLUR should have the '[ListUpdateAction](#)' field associated.

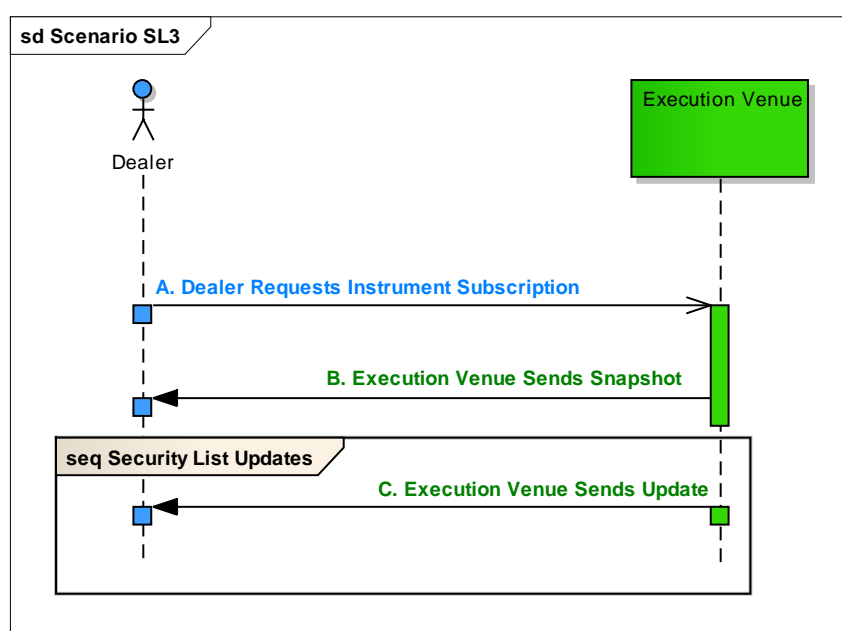


Figure 7: Scenario SL3 – Dealer Subscribes to Instruments Reference Data, Execution Venue Returns Snapshot and Updates

Model Flow

The following table illustrates the flows expected when communicating with an Execution Venue implementing FIX 5.0 SP2.

Model FIX 5.0				
(A) Dealer Requests Instrument Subscription	Dealer	→	x – SecurityListRequest SecurityReqID(320)= ❶ SecurityListRequestType(559)= required e.g. AllSecurities(4) SubscriptionRequestType(263)= SnapshotAndUpdates(1)	Execution Venue
(B) Execution Venue Sends Snapshot		←	y – SecurityList SecurityReqID(320)= ❶ SecurityResponseID(322)=<Execution Venue assigned ID, Same ID for ALL the messages of this request> SecurityRequestResult(560)=ValidRequest(0) NoRelatedSym(146)=<count> Symbol(55)=< human readable name of the instrument > SecurityID(48)=<ID > SecurityIDSource(22)=<required> SecurityType(167)=<required> <i>Additional instrument attributes</i>	
(C) Execution Venue Sends Updates		←	BK – SecurityListUpdateReport SecurityReqID(320)= ❶ SecurityResponseID(322)=<Execution Venue assigned ID, Same ID for ALL the messages of this request> SecurityRequestResult(560)=ValidRequest(0) NoRelatedSym(146)=<count> ListUpdateAction(1324)=Add(A) Delete(D) Modify(M) NoRelatedSym(146)=<count> Symbol(55)=< human readable name of the instrument > SecurityID(48)=<ID > SecurityIDSource(22)=<required> SecurityType(167)=<required> <i>Additional instrument attributes</i>	

Table 5: Scenario SL3 – Dealer Subscribes to Instruments Reference Data, Execution Venue Returns Snapshot and Updates

6.6 Scenario SL4 – Dealer Cancels Instrument Reference Data Subscription

This scenario is where the Dealer cancels a previous subscription request.

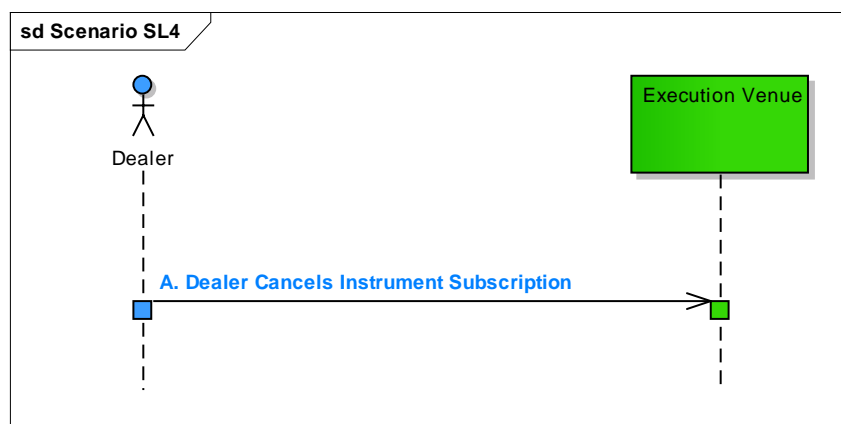


Figure 8: Scenario SL4 – Dealer Cancels Instrument Reference Data Subscription

Model Flow

The following table illustrates the flows expected when communicating with an Execution Venue implementing FIX 5.0 SP2.

Model FIX 5.0				
(A) Dealer Cancels Instrument Subscription	Dealer	→	x – SecurityListRequest SecurityReqID(320)= ❶ Same ID as specified in the request SubscriptionRequestType(263)=DisablePreviousSnapshot(2)	Execution Venue

Table 6: Scenario SL4 – Dealer Cancels Instrument Reference Data Subscription

6.7 Scenario SL5 – Dealer Requests Snapshot or Subscription of Instruments Reference Data, Execution Venue Returns Error

This scenario occurs when the Dealer requests a snapshot or subscription of the instruments that the Execution Venue cannot provide. The Execution Venue rejects the request.

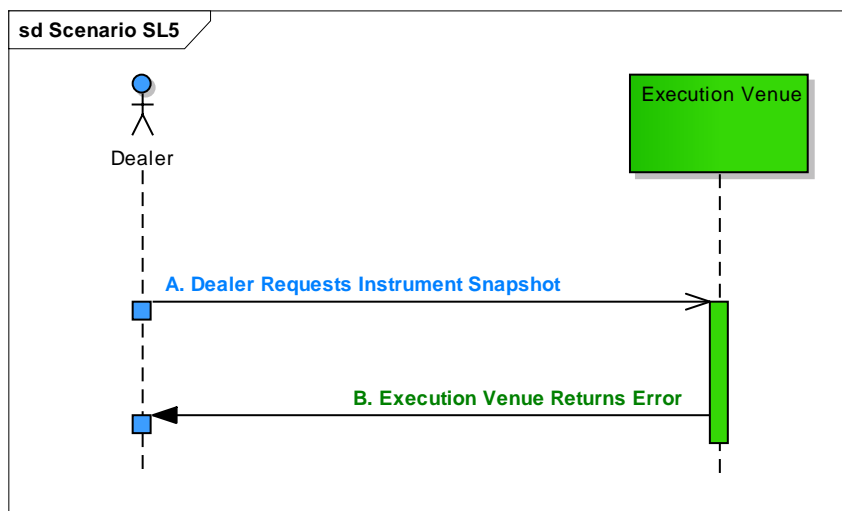


Figure 9: Scenario SL5 – Dealer Requests Snapshot or Subscription of Instruments Reference Data, Execution Venue Returns Error

Model Flow

The following table illustrates the flows expected when communicating with an Execution Venue implementing FIX 5.0 SP2.

Model FIX 5.0				
(A) Dealer Requests Instrument Snapshot	Dealer	→	x – SecurityListRequest SecurityReqID(320)= ❶ SecurityListRequestType(559)=required e.g. AllSecurities(4) SubscriptionRequestType(263)=Snapshot(0) or SnapshotAndUpdates(1)	Execution Venue
(B) Execution Venue Returns Error		←	y – SecurityList SecurityReqID(320)= ❶ SecurityResponseID(322)=<Execution Venue assigned ID> SecurityRequestResult(560)=Required e.g. InvalidOrUnsupportedRequest(1)	

Table 7: Scenario SL5 – Dealer Requests Snapshot or Subscription of Instruments Reference Data, Execution Venue Returns Error

This scenario is applicable to either snapshot or subscription requests.

6.8 Scenario SL6 – Dealer Cancels Instrument Reference Data Subscription, Execution Venue Rejects

This scenario is where the Dealer unsubscribes to a previous subscription but the Execution Venue rejects.

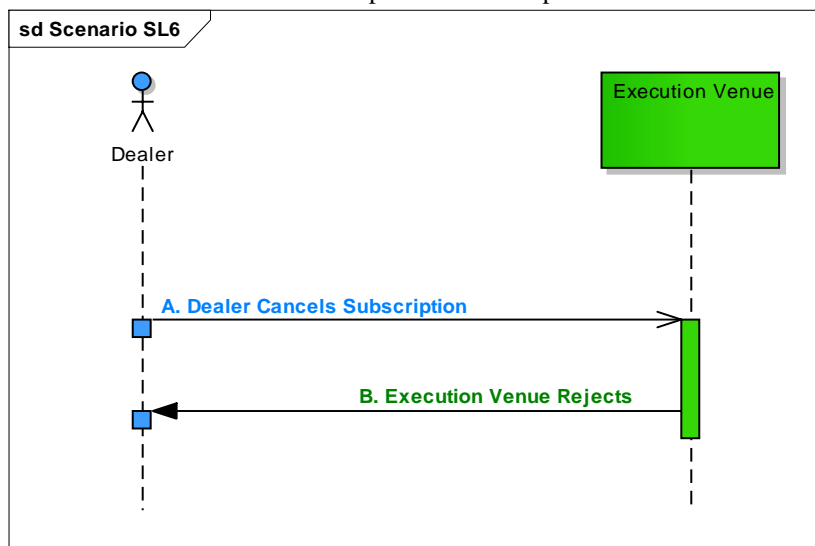


Figure 10: Scenario SL6 – Dealer Cancels Instrument Reference Data Subscription, Execution Venue Rejects

Model Flow

The following table illustrates the flows expected when communicating with an Execution Venue implementing FIX 5.0 SP2.

Model FIX 5.0				
(A) Dealer Cancel Subscription	Dealer	→	x – SecurityListRequest SecurityReqID(320)= ❶ SubscriptionRequestType(263)=DisablePreviousSnapshot(2)	Execution Venue
(B) Execution Venue Rejects		←	BK – SecurityListUpdateReport SecurityReqID(320)= ❶ SecurityRequestResult(560)=required e.g. InvalidOrUnsupportedRequest(1)	

Table 8: Scenario SL6 – Dealer Cancels Instrument Reference Data Subscription, Execution Venue Rejects

6.9 Scenario SS1 – Dealer Requests Instrument Status, Execution Venue Returns Status

This scenario is where the Dealer requests the current status of a single instrument which the Execution Venue returns.

The Execution Venue responds with the appropriate status for the instrument. Symbology fields in the Instrument component block are required to identify the instrument.

For tradable instrument the SecurityTradingStatus is 'ReadyToTrade'. Other status may be set as appropriate, by the Execution Venue.

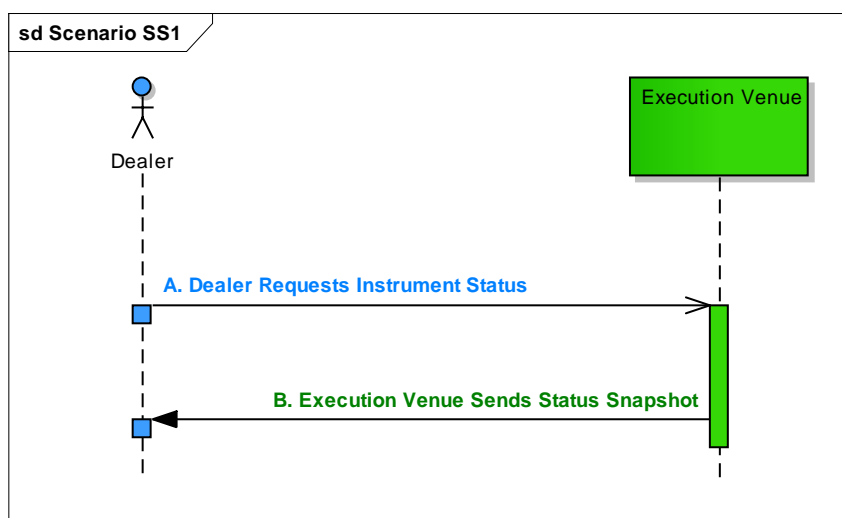


Figure 11: Scenario SS1 – Dealer Requests Instrument Status, Execution Venue Returns Status

Model Flow

The following table illustrates the flows expected when communicating with an Execution Venue implementing FIX 5.0 SP2.

Model FIX 5.0				
(A) Dealer Requests current instrument status	Dealer	→	e – SecurityStatusRequest SecurityStatusReqID(324)= ❶ SecurityID(48)=<ID > SubscriptionRequestType(263)= Snapshot(0)	Execution Venue
(B) Execution Venue Sends Status Snapshot		←	f – SecurityStatus SecurityStatusReqID(324)= ❶ SecurityID(48)=<ID > UnsolicitedIndicator(325)= Message is being sent as a result of a prior request(N) SecurityTradingStatus(326)= required e.g. ReadyToTrade(17)	

Table 9: Scenario SS1 – Dealer Requests Instrument Status, Execution Venue Returns Status

6.10 Scenario SS2 – Dealer Subscribes to Instrument Status, Execution Venue Returns Snapshot and Updates

This scenario is where the Dealer subscribes to status of instrument. Execution Venue returns current state and updates. [See notes in scenario SS1](#)

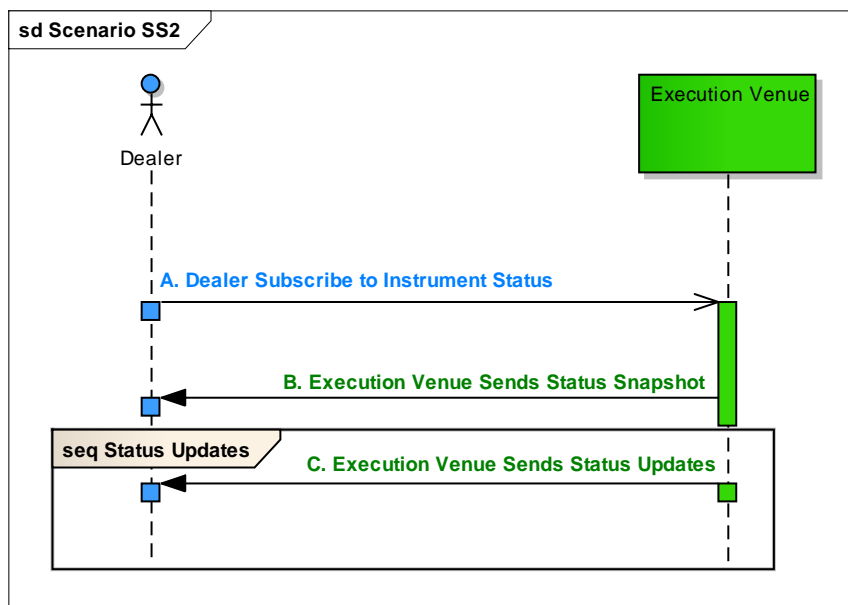


Figure 12: Scenario SS2 – Dealer Subscribes to Instrument Status, Execution Venue Returns Snapshot and Updates

Model Flow

The following table illustrates the flows expected when communicating with an Execution Venue implementing FIX 5.0 SP2.

Model FIX 5.0				
(A) Dealer Subscribes to Instrument Status	Dealer	→	e – SecurityStatusRequest SecurityStatusReqID(324)= ❶ Symbol(55)=<Ticker symbol, part of repeated block> SubscriptionRequestType(263)= SnapshotAndUpdates(1)	Execution Venue
(B) Execution Venue Sends Status Snapshot		←	f – SecurityStatus SecurityStatusReqID(324)= ❶ Symbol(55)=<Ticker symbol, part of repeated block> UnsolicitedIndicator(325)= Message is being sent as a result of a prior request(N) SecurityTradingStatus(326)=required – appropriate status	
(C) Execution Venue Sends status Updates		←	f – SecurityStatus SecurityStatusReqID(324)= ❶ Symbol(55)=<Ticker symbol, part of repeated block> UnsolicitedIndicator(325)= Message is being sent unsolicited (Y) SecurityTradingStatus(326)= required – appropriate status	

Table 10: Scenario SS2 – Dealer Subscribes to Instrument Status, Execution Venue Returns Snapshot and Updates

6.11 Scenario SS3 – Dealer Requests Instrument Status, Execution Venue Returns Error

This scenario occurs when a Dealer requests the current instrument status of an instrument which is unknown to the Execution Venue. [See notes in scenario SS1](#)

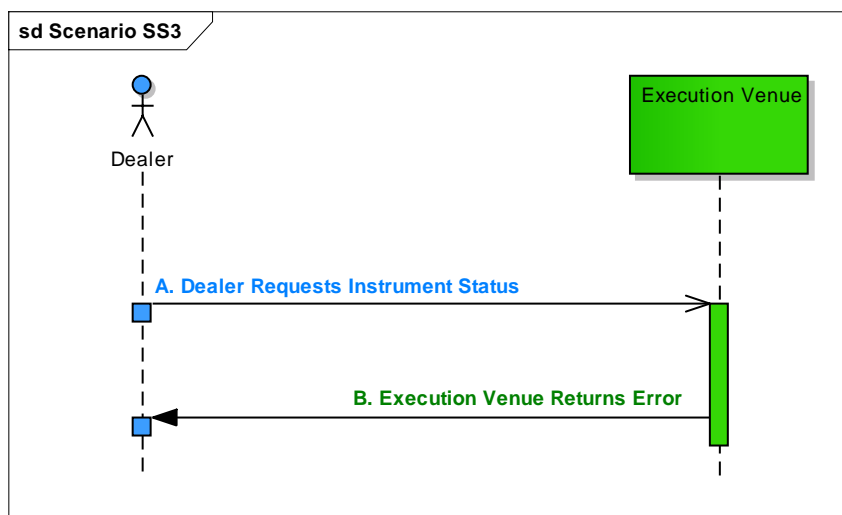


Figure 13: Scenario SS3 – Dealer Requests Instrument Status, Execution Venue Returns Error

Model Flow

The following table illustrates the flows expected when communicating with an Execution Venue implementing FIX 5.0 SP2.

Model FIX 5.0				
(A) Dealer Requests current instrument status	Dealer	→	e – SecurityStatusRequest SecurityStatusReqID(324)= ❶ Symbol(55)=<Ticker symbol, part of repeated block> SubscriptionRequestType(263)= required e.g. Snapshot(0)	Execution Venue
(B) Execution Venue Returns Error		←	f – SecurityStatus SecurityStatusReqID(324)= ❶ SecurityTradingStatus(326)= Unknown or Invalid (20)	

Table 11: Scenario SS3 – Dealer Requests Instrument Status, Execution Venue Returns Error

6.12 Scenario SD1 – Dealer Sends Instrument Definition Request, Execution Venue Confirms

This scenario occurs when the Dealer specifies a custom strategy instrument definition which the Execution Venue confirms.

In most cases, the new strategy is based on instruments and other strategies that are already defined by the Execution Venue. These instruments are listed in the legs of the newly defined strategy.

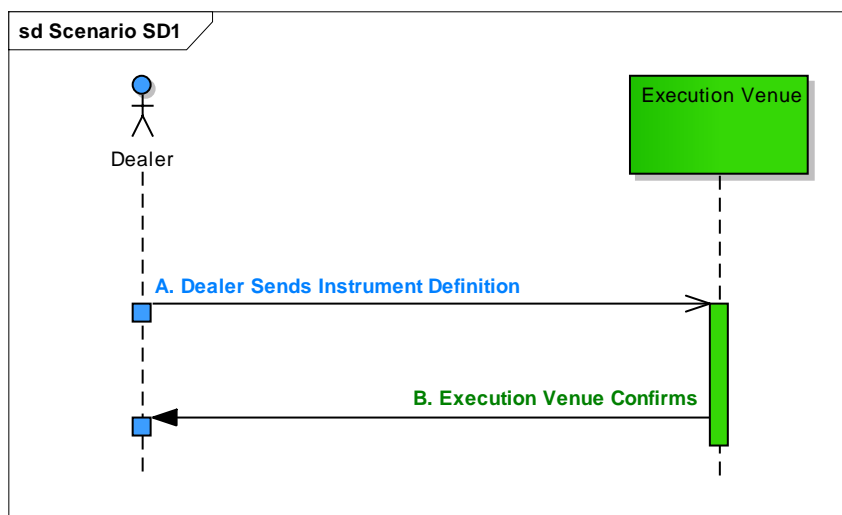


Figure 14: Scenario SD1 – Dealer Sends Instrument Definition Request, Execution Venue Confirms

Model Flow

The following table illustrates the flows expected when communicating with an Execution Venue implementing FIX 5.0 SP2.

Model FIX 5.0				
(A) Dealer Sends Instrument Definition	Dealer	→	c – SecurityDefinitionRequest SecurityReqID(320)= ❶ SecurityType(167)=MLEG SecuritySubType(762)=<sub type> NoLegs(555)=<count> > LegSecurityID(602) > LegSecurityIDSource(603)=8/M (Exchange Symbol/Marketplace assigned identifier) > LegRatioQty(623) > LegSide(624)=< enum value >	Execution Venue
(B) Execution Venue Confirms		←	d – SecurityDefinition SecurityReqID(320)= ❶ SecurityResponseType(323)=Accept security proposal as-is(1) Symbol(55)=< human readable name of the instrument SecurityID(48)=<ID > SecurityType(167)=MLEG SecuritySubType(762)=<sub type> NoLegs(555)=<count> > LegSecurityID(602) > LegSecurityIDSource(603)=8/M (Exchange Symbol/Marketplace assigned identifier) > LegRatioQty(623) > LegSide(624)=< enum value >	

Table 12: Scenario SD1 – Dealer Sends Instrument Definition Request, Execution Venue Confirms

6.13 Scenario SD2 – Dealer Sends Instrument Definition Request, Execution Venue Rejects

This scenario occurs when the Dealer provides a custom instrument definition which the Execution Venue rejects.

The Execution Venues that support the instrument definition messages will have different restrictions on the security definitions that are supported. The reject message is sent whenever the Security Request Definition does not meet these restrictions.

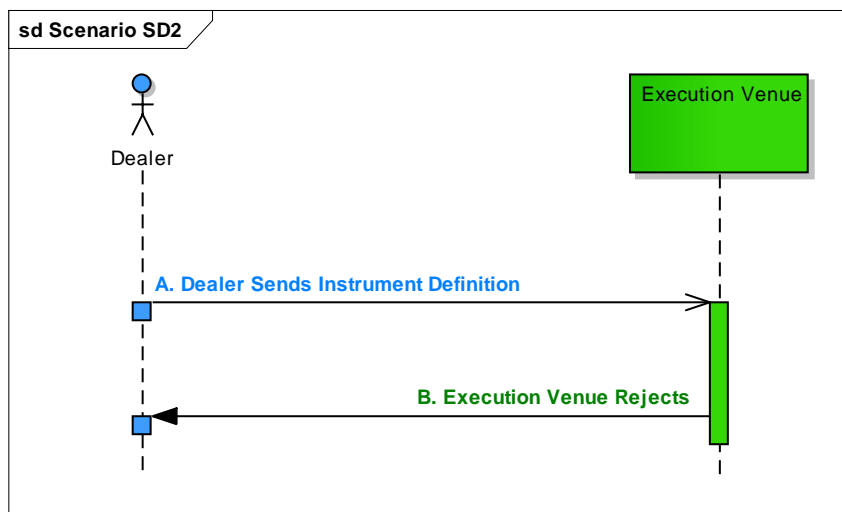


Figure 15: Scenario SD2 – Dealer Sends Instrument Definition Request, Execution Venue Rejects

Model Flow

The following table illustrates the flows expected when communicating with an Execution Venue implementing FIX 5.0 SP2.

Model FIX 5.0				
(A) Dealer Sends Instrument Definition	Dealer	→	c – SecurityDefinitionRequest SecurityReqID(320)= ❶ SecurityType(167)=MLEG SecuritySubType(762)=<sub type> NoLegs(555)=<count> > LegSecurityID(602) >> LegSecurityIDSource(603)=8/M (Exchange Symbol/Marketplace assigned identifier) > LegRatioQty(623) > LegSide(624)=< enum value >	Execution Venue
(B) Execution Venue Rejects		←	d – SecurityDefinition SecurityReqID(320)= ❶ SecurityResponseType(323)=Reject security proposal(5)	

Table 13: Scenario SD2 – Dealer Sends Instrument Definition Request, Execution Venue Rejects

6.14 Scenario SD3 – Execution Venue Publishes Instrument Definition

This scenario is where the Execution Venue publishes an Instrument definition to the Dealer.

Typical examples for using this scenario are:

- When an instrument definition request is sent to the Execution Venue and the request is accepted, all Dealers expect to receive a new security definition
- When a new instrument is added during the FIX session
- When one or more of the Instrument's attributes is updated/modified during the FIX session
- Some Execution Venues publish Instrument definitions after a successful login, where the subscription can be an 'offline' request (e.g. via profile setup)

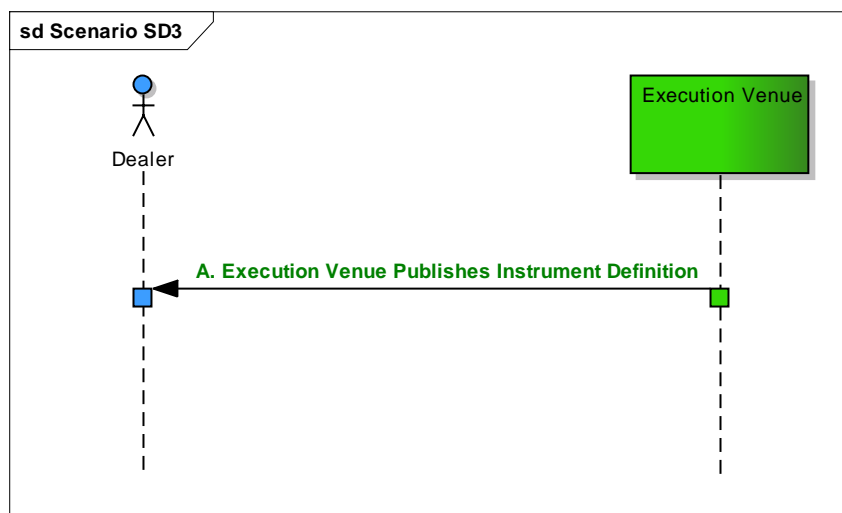


Figure 16: Scenario SD3 – Execution Venue Publishes Instrument Definition

Model Flow

The following table illustrates the flows expected when communicating with an Execution Venue implementing FIX 5.0 SP2.

Model FIX 5.0			
(A) Execution Published by the Execution Venue	Dealer	←	d – SecurityDefinition SecurityResponseType(323)=Accept security proposal as-is(1) Symbol(55)=<human readable name of the instrument> SecurityID(48)=<ID > SecurityType(167)=MLEG SecuritySubType(762)=<sub type> >NoLegs(555)=<count> > LegSecurityID(602) > LegSecurityIDSource(603)=8/M (Exchange Symbol/Marketplace assigned identifier) > LegRatioQty(623) > LegSide(624)=< enum value >
			Execution Venue

Table 14: Scenario SD3 – Execution Venue Publishes Instrument Definition

7 Pre-Trade – Price Contribution

This section describes messages relevant for contributing prices to an Execution Venue.

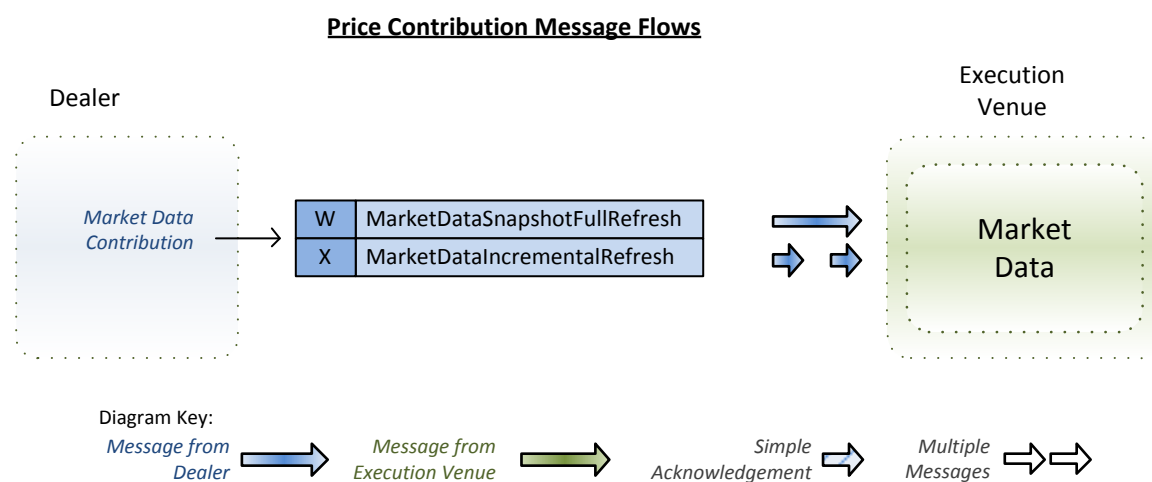
The scenarios in this section are common in market making. Prices may be contributed to either:

- Quote Negotiation trading as quotes (see volume 3)
- Central Limit Order Book as orders (see volume 4)

Dealers use these workflows to contribute tradable and/or indicative prices for different market streams (i.e. tiers).

7.1 Overview Diagram

The following diagram illustrates the FIX messages and the Workflows described in this chapter.



7.2 Message Flows Summary

The following scenarios illustrate the use of these messages.

Scenario	Description
PP1	Dealer Sends Different Prices for the Same Instrument to Multiple Customers – Dealer Amends Trade Information (Price & type)

7.1 Scenario PP1 – Dealer Sends Different Prices for the Same Instrument to Multiple Customers – Dealer Amends Trade Information (Price & type)

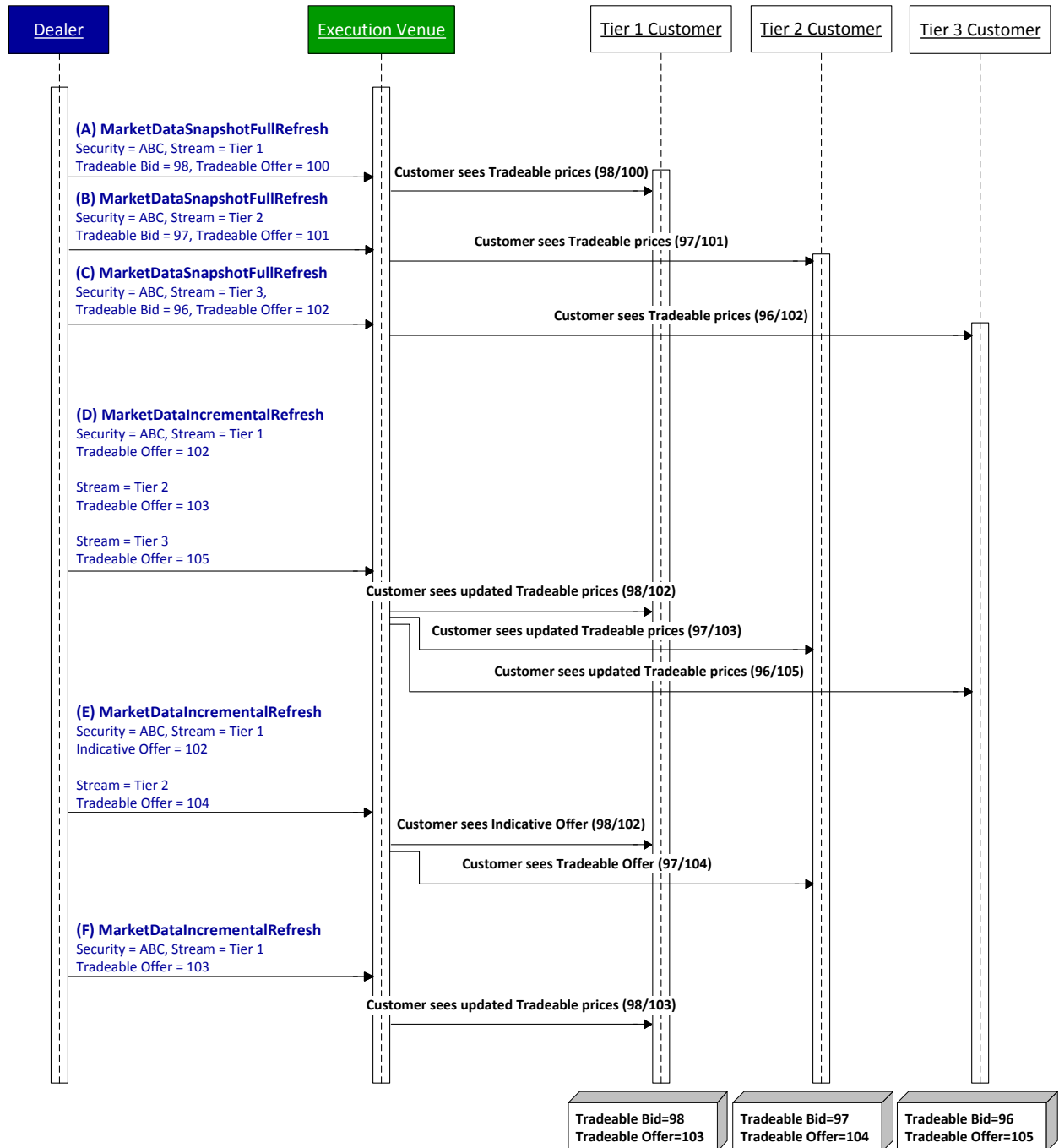


Figure 17: Scenario PP1 – Dealer Sends Different Prices for the Same Instrument to Multiple Customers – Dealer Amends Trade Information (Price & type)

Model Flow

The following table illustrates the flows expected when communicating with an Execution Venue implementing FIX 5.0 SP2

Model FIX 5.0				
(A) Dealer submits Tradeable prices	Dealer	→	W – MarketDataSnapshotFullRefresh MDStreamID(1500) = Tier 1 NoMDEntries(268)=2 MDEntryRefID(280)=❶ MDEntryType(269) = Bid MDEntryPx(270) = 98 MDQuoteType(1070) = Tradeable(1) MDEntryRefID(280)=❷ MDEntryType(269) = Offer MDEntryPx(270) = 100 MDQuoteType(1070) = Tradeable(1)	Execution Venue
(B) Dealer submits Tradeable prices		→	W - MarketDataSnapshotFullRefresh MDStreamID(1500) = Tier 2 NoMDEntries(268)=2 MDEntryRefID(280)=❸ MDEntryType(269) = Bid MDEntryPx(270) = 97 MDQuoteType(1070) = Tradeable(1) MDEntryRefID(280)=❹ MDEntryType(269) = Offer MDEntryPx(270) = 101 MDQuoteType(1070) = Tradeable(1)	
(C) Dealer submits Tradeable prices		→	W - MarketDataSnapshotFullRefresh MDStreamID(1500) = Tier3 NoMDEntries(268)=2 MDEntryRefID(280)=❺ MDEntryType(269) = Bid MDEntryPx(270) = 96 MDQuoteType(1070) = Tradeable(1) MDEntryRefID(280)=❻ MDEntryType(269) = Offer MDEntryPx(270) = 102 MDQuoteType(1070) = Tradeable(1)	
(D) Dealer updates Tradeable prices		→	X - MarketDataIncrementalRefresh NoMDEntries(268)=3 MDEntryRefID(280)=❷ MSEntryAction(279)=Change(1) MDEntryType(269) = Offer MDEntryPx(270) = 102 MDQuoteType(1070) = Tradeable(1) MDEntryRefID(280)=❹ MSEntryAction(279)=Change(1) MDEntryType(269) = Offer MDEntryPx(270) = 103 MDQuoteType(1070) = Tradeable(1) MDEntryRefID(280)=❺ MSEntryAction(279)=Change(1) MDEntryType(269) = Offer MDEntryPx(270) = 105 MDQuoteType(1070) = Tradeable(1)	

Model FIX 5.0			
(E) Dealer updates price/QuoteType	→	X - MarketDataIncrementalRefresh NoMDEntries(268)=2 MDEntryRefID(280)=❷ MSEntryAction(279)=Change(1) MDEntryType(269) = Offer MDEntryPx(270) = 102 MDQuoteType(1070) = Indicative(0) MDEntryRefID(280)=❹ MSEntryAction(279)=Change(1) MDEntryType(269) = Offer MDEntryPx(270) = 104 MDQuoteType(1070) = Tradeable(1)	
(F) Dealer updates price/QuoteType	→	X - MarketDataIncrementalRefresh NoMDEntries(268)=1 MDEntryRefID(280)=❷ MSEntryAction(279)=Change(1) MDEntryType(269) = Offer MDEntryPx(270) = 103 MDQuoteType(1070) = Tradeable(1)	

Table 15: Scenario PP1 – Dealer Sends Different Prices for the Same Instrument to Multiple Customers – Dealer Amends Trade Information (Price & type)

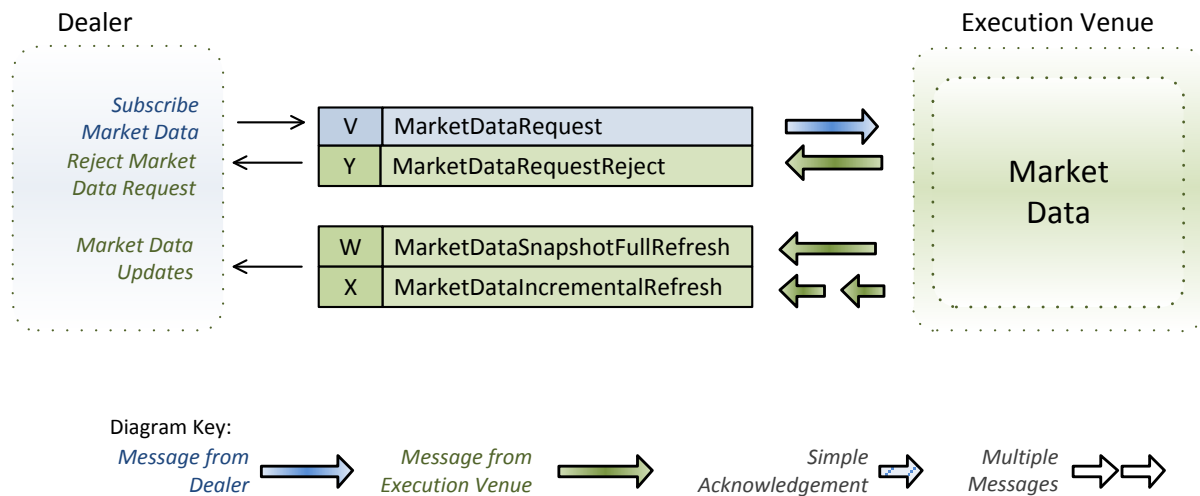
8 Pre-Trade – Price Subscription

This section describes messages relevant for subscribing to Market Data provided by the Execution Venue. The FIX messages are flexible in the data that is communicated – the data can be a simple streaming of prices (e.g. composite prices) or a representation of a central order book showing full market depth.

8.1 Overview diagram

The following diagram illustrates the FIX messages and the Workflows described in this chapter.

Price Subscription Message Flows



8.2 Message Flows Summary

The following scenarios illustrate the use of these messages.

Scenario	Description
MDS1	Dealer Requests/Receives Market Data
MDS2	Dealer Unsubscribes Market Data
MDS3	Dealer Requests Market Data, Execution Venue Rejects

8.3 Scenario MDS1 – Dealer Requests/Receives Market Data

The Dealer subscribes to retrieve Market prices from the Execution Venue (e.g. Composite Prices).

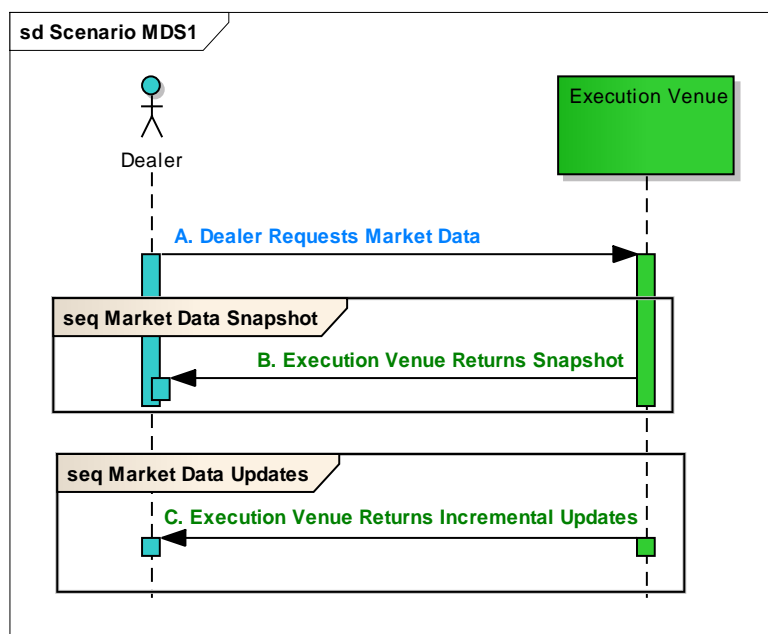


Figure 18: Scenario MDS1 – Dealer Requests/Receives Market Data

Model Flow

The following table illustrates the flows expected when communicating with an Execution Venue implementing FIX 5.0 SP2.

Model FIX 5.0				
(A) Dealer Requests Market Data	Dealer	→	V – MarketDataRequest MDReqID(262) = ❶ SubscriptionRequestType(263)= required e.g. SnapshotAndUpdates(1) MarketDepth(264)= required e.g. full book depth(0) MDUpdateType(265)= required e.g. Incremental refresh(1) AggregatedBook(266)=required e.g. BookEntriesShouldNotBeAggregated(N)	Execution Venue
(B) Execution Venue Returns Snapshot		←	W – MarketDataSnapshotFullRefresh MDReqID(262) = ❶ NoMDEntires(268)=<number of entries> MDBookType(1022)=requirede.g OrderDepth(3)	
(C) Execution Venue Returns Incremental Updates		←	X – MarketDataIncrementalRefresh MDReqID(262) = ❶ NoMDEntires(268)=<number of entries> MDBookType(1021)=requirede.g OrderDepth(3)	

Table 16: Scenario MDS1 – Dealer Requests/Receives Market Data

Note: Each Market Data Request will trigger an update. If there are multiple Market Data Requests (with different MDReqID) for the same instrument then the Dealer should expect duplicate updates (one per request). Further detail of the Market Data is available in “Book Management Recommended Practises” available at <http://www.fixprotocol.org/specifications/TechDoc-MktData>.

8.4 Scenario MDS2 – Dealer Unsubscribes Market Data

The Dealer unsubscribes from Market Data.

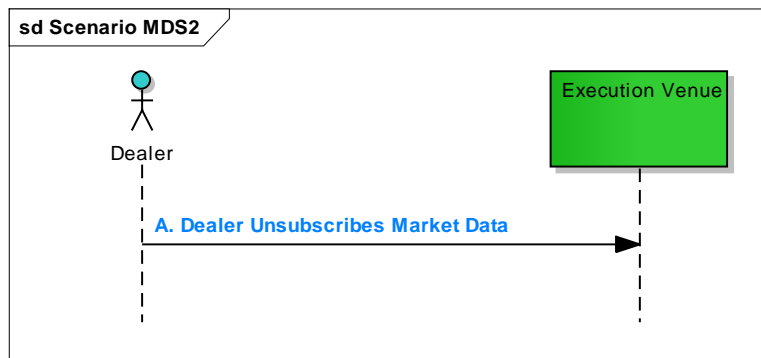


Figure 19: Scenario MDS2 – Dealer Unsubscribes Market Data

Model Flow

The following table illustrates the flows expected when communicating with an Execution Venue implementing FIX 5.0 SP2.

Model FIX 5.0				
(A) Dealer Unsubscribes Market Data	Dealer	→	V – MarketDataRequest MDReqID(262) = ❶ SubscriptionRequestType(263)=DisablePreviousSnapshot(2)	Execution Venue

Table 17: Scenario MDS2 – Dealer Unsubscribes Market Data

8.5 Scenario MDS3 – Dealer Requests Market Data, Execution Venue Rejects

The Dealer requests Market Data (e.g. for an unknown instrument). Execution Venue rejects.

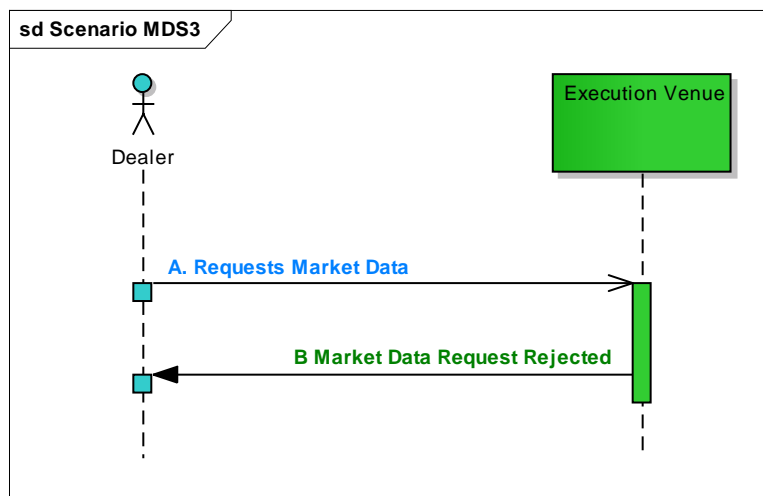


Figure 20: Scenario MDS3 – Dealer Requests Market Data, Execution Venue Rejects

Model Flow

The following table illustrates the flows expected when communicating with an Execution Venue implementing FIX 5.0 SP2.

Model FIX 5.0				
(A) Dealer Requests market Data	Dealer	→	V – MarketDataRequest MDReqID(262) = ❶ SubscriptionRequestType(263)=SnapshotAndUpdates(1) MarketDepth(264)= full book depth(0) MDUpdateType(265)=Incremental refresh(1)	Execution Venue
(B) Market Data Request Reject		←	Y – MarketDataRequestReject MDReqID(262) = ❶ MDReqRejReason(281)=<Reason for rejection>	

Table 18: Scenario MDS3 – Dealer Requests Market Data, Execution Venue Rejects

The same message flow is used for different reject reasons that are specified in field [MDReqRejReason](#).

9 Pre-Trade – Quote Contribution

This section describes messages relevant for contributing quotes to an Execution Venue.

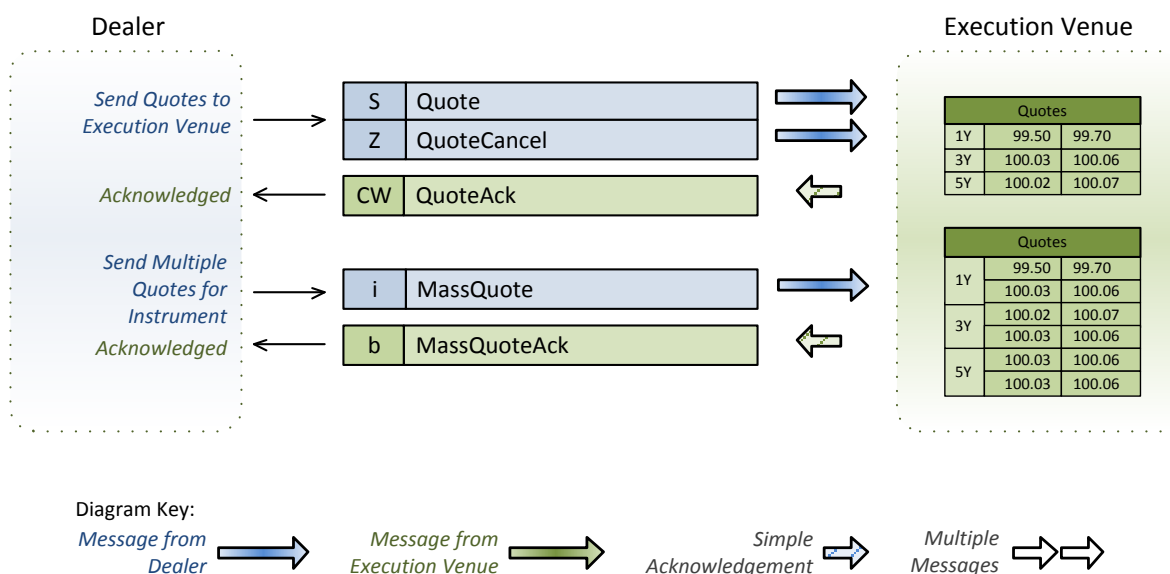
The scenarios in this section are common in ‘click to trade’ markets and may be followed by scenarios described in *Volume 3- Quote-Driven Workflows*.

See also *Volume 4 - Central Limit Order Book Workflows*, section *Quote Contribution to Central Limit Order Book* which is largely used by market makers.

9.1 Overview diagram

The following diagram illustrates the FIX messages and the Workflows described in this chapter.

Quote Contribution Message Flows



9.2 Message Flows Summary

The following scenarios illustrate the use of these messages.

Scenario	Description
PC1	Dealer Sends Quote to Market
PC2	Dealer Cancels Quote on Market
PC3	Dealer Cancels All Quotes
PC4	Dealer Sends Quote to Market, Execution Venue Rejects
PC5	Dealer Cancels a Quote, Execution Venue Rejects
MQ1	Dealer Sends Mass Quote to Execution Venue
MQ2	Dealer Cancels a Single Quote Sent Within a Mass Quote
MQ3	Dealer Cancels All Quotes Sent within a Mass Quote

9.3 Scenario PC1 – Dealer Sends Quote to Market

This scenario shows a Quote being sent to the market and being updated.

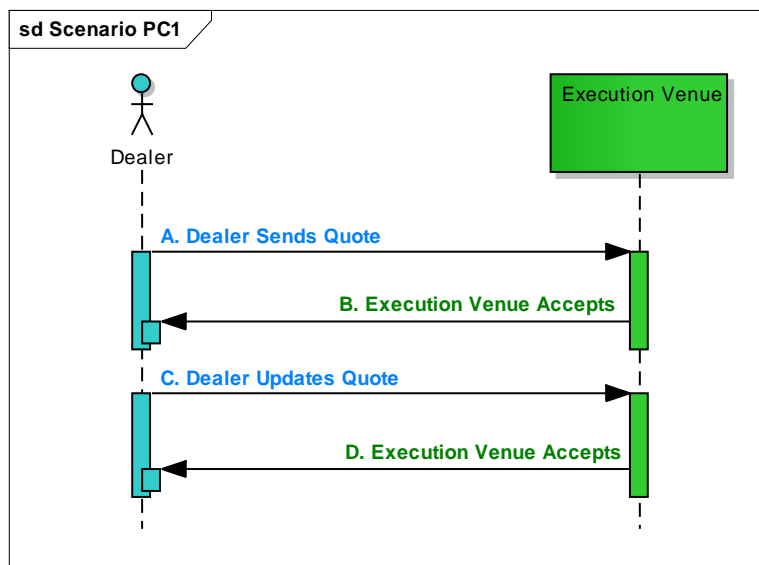


Figure 21: Scenario PC1 – Dealer Sends Quote to Market

Model Flow

The following table illustrates the flows expected when communicating with an Execution Venue implementing FIX 5.0 SP2.

Model FIX 5.0				
(A) Dealer Sends Quote	Dealer	→	S – Quote QuoteID(117)= ❶ QuoteMsgID(1166)=❷ QuoteResponseLevel(301)= required e.g. Acknowledge each quote message(2)	Execution Venue
(B) Execution Venue Accepts		←	CW - QuoteAck QuoteID(117)= ❶ QuoteMsgID(1166)= ❷ QuoteAckStatus (1865)=Accepted(1)	
(C) Dealer Updates Quote		→	S – Quote QuoteID(117)= ❶ QuoteMsgID(1166)=❸ QuoteResponseLevel(301)= required e.g Acknowledge each quote message(2)	
(D) Execution Venue Accepts		←	CW - QuoteAck QuoteID(117)= ❶ QuoteMsgID(1166)= ❸ QuoteAckStatus (1865)=Accepted(1)	
The message flow in this scenario maybe followed by the message flow for the scenarios in:Trading - Quotes/Orders Based Trading (Volume 3)				

Table 19: Scenario PC1 – Dealer Sends Quote to Market

9.4 Scenario PC2 – Dealer Cancels Quote on Market

This scenario shows a Quote being sent to the market and later cancelled.

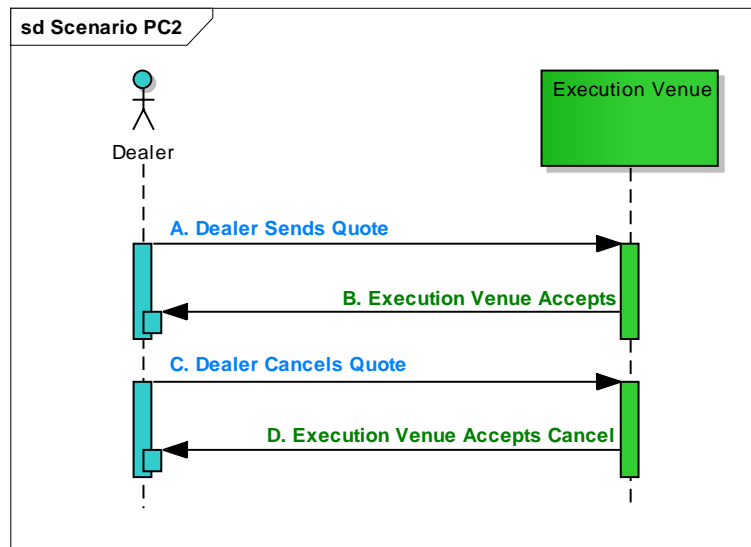


Figure 22: Scenario PC2 – Dealer Cancels Quote on Market

Model Flow

The following table illustrates the flows expected when communicating with an Execution Venue implementing FIX 5.0 SP2.

Model FIX 5.0				
(A) Dealer Sends Quote	Dealer	→	S – Quote QuoteID(117)= ❶ QuoteMsgID(1166)=❷ QuoteResponseLevel(301)= required e.g. Acknowledge each quote message(2)	Execution Venue
(B) Execution Venue Accepts		←	CW - QuoteAck QuoteID(117)= ❶ QuoteMsgID(1166)=❷ QuoteAckStatus (1865)=Accepted(1)	
(C) Dealer Cancels Quote		→	Z – QuoteCancel QuoteID(117)= ❶ QuoteMsgID(1166)=❸ QuoteCancelType(298)=required e.g. Cancel quote specified in QuoteID(5) QuoteResponseLevel(301)= required e.g.Acknowledge each quote message(2)	
(D) Execution Venue Accepts Cancel		←	CW - QuoteAck QuoteID(117)= ❶ QuoteMsgID(1166)=❸ QuoteAckStatus (1865)= Accepted(1) <Accepted the cancellation>	

Table 20: Scenario PC2 – Dealer Cancels Quote on Market

9.5 Scenario PC3 – Dealer Cancels All Quotes

This scenario shows a Dealer cancelling all his quotes on the market that are associated with this FIX session.

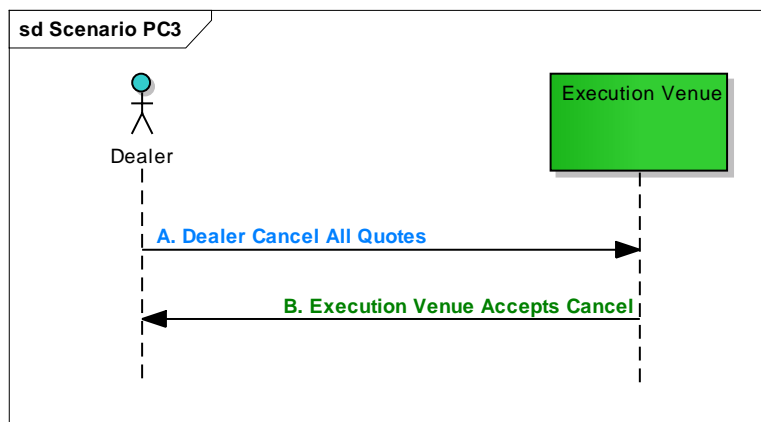


Figure 23: Scenario PC3 – Dealer Cancels All Quotes

Model Flow

The following table illustrates the flows expected when communicating with an Execution Venue implementing FIX 5.0 SP2.

Model FIX 5.0				
(A) Dealer Cancels all quotes	Dealer	→	Z – QuoteCancel QuoteID(117)= ❶ QuoteMsgID(1166)=❷ QuoteCancelType(298)= CancelAllQuotes(4) QuoteResponseLevel(301)= required e.g. Acknowledge each quote message(2)	Execution Venue
(B) Execution Venue Accepts cancel		←	b – MassQuoteAck QuoteID(117)= ❶ QuoteStatus(297)= Accepted(0) <MassQuote Cancellation accepted> NoQuoteSets(296)=required e.g. (1) >TotNoCxlQuotes(1168) >TotNoAccQuote(1169) >TotNoRejQuotes(1170) >NoQuoteEntries(295)=required >>QuoteEntryStatus(1167) <The current status held by the Execution Venue for that quote> >>QuoteEntryRejectReason(368)	

Table 21: Scenario PC3 – Dealer Cancels All Quotes

Note: Extensions to this scenario are described in *FIX Specifications Version 5.0 Service Pack 2 – Volume 3 Quote Cancel Page 55*.

9.6 Scenario PC4 – Dealer Sends Quote to Market, Execution Venue Rejects

This scenario shows a Quote being sent to the market and the Execution Venue rejects.

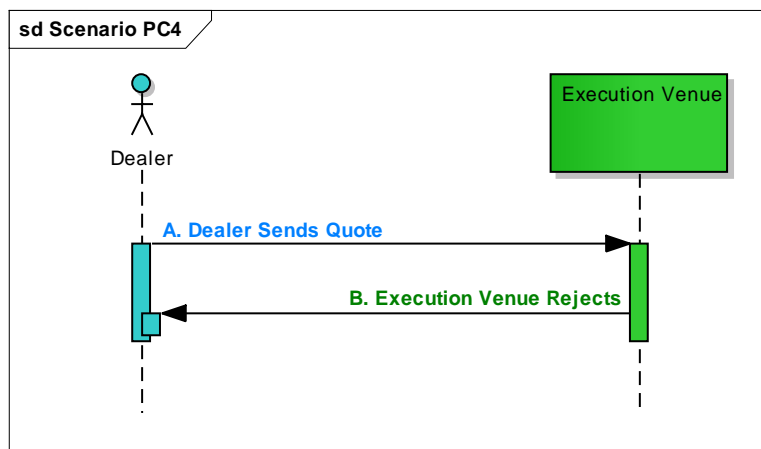


Figure 24: Scenario PC4 – Dealer Sends Quote to Market, Execution Venue Rejects

Model Flow

The following table illustrates the flows expected when communicating with an Execution Venue implementing FIX 5.0 SP2.

Model FIX 5.0				
(A) Dealer Sends Quote	Dealer	→	S – Quote QuoteID(117)= ❶ QuoteMsgID(1166)= ❷ QuoteResponseLevel(301)= required e.g. Acknowledge each quote message(2)	Execution Venue
(B) Execution Venue Rejects		←	CW - QuoteAck QuoteID(117)= ❶ QuoteMsgID(1166)= ❷ QuoteAckStatus (1865)= Rejected(2) QuoteRejectReason(300)= required e.g. Unknown Symbol - security(1)	

Table 22: Scenario PC4 – Dealer Sends Quote to Market, Execution Venue Rejects

9.7 Scenario PC5 – Dealer Cancels a Quote, Execution Venue Rejects

This scenario represents the case where a Dealer sends a Quote cancel to the market and Execution Venue rejects.

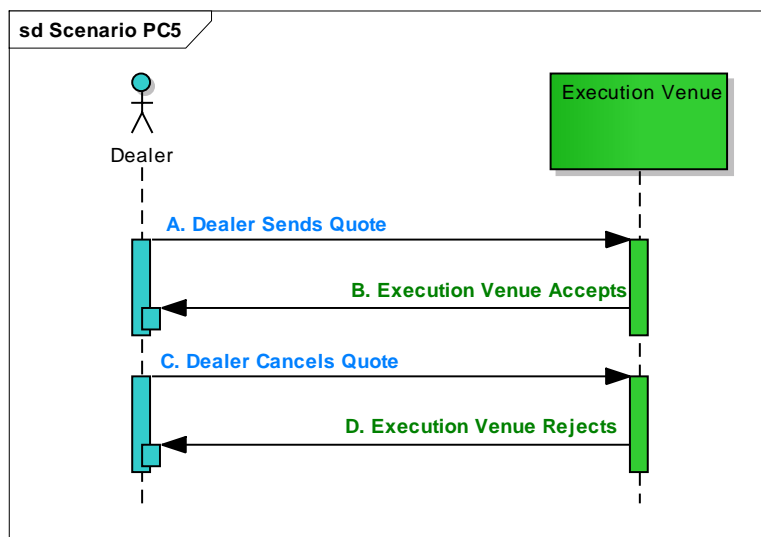


Figure 25: Scenario PC5 – Dealer Cancels Quote, Execution Venue Rejects

Model Flow

The following table illustrates the flows expected when communicating with an Execution Venue implementing FIX 5.0 SP2.

Model FIX 5.0				
(A) Dealer Sends Quote	Dealer	→	S – Quote QuoteID(117)= ❶ QuoteMsgID(1166)= ❷ QuoteResponseLevel(301)= required e.g. Acknowledge each quote message(2)	Execution Venue
(B) Execution Venue Accepts		←	CW - QuoteAck QuoteID(117)= ❶ QuoteMsgID(1166)= ❷ QuoteAckStatus (1865)= Accepted(1)	
(C) Dealer Cancels Quote		→	Z – QuoteCancel QuoteID(117)= ❶ QuoteMsgID(1166)= ❸ QuoteCancelType(298)=CancelQuoteSpecifiedInQuoteID(5) QuoteResponseLevel(301)= required e.g. AcknowledgeEachQuoteMessage(2)	
(D) Execution Venue Rejects		←	CW - QuoteAck QuoteID(117)= ❶ QuoteMsgID(1166)= ❸ QuoteAckStatus (1865)= Rejected(2) <Cancellation rejected> QuoteRejectReason(300)=required e.g. Unknown Quote(5)	

Table 23: Scenario PC5 – Dealer Cancels Quote, Execution Venue Rejects

9.8 Scenario MQ1 – Dealer Sends Mass Quote to Execution Venue

This scenario represents the case where the Dealer sends a Mass Quote to the market and later updates.

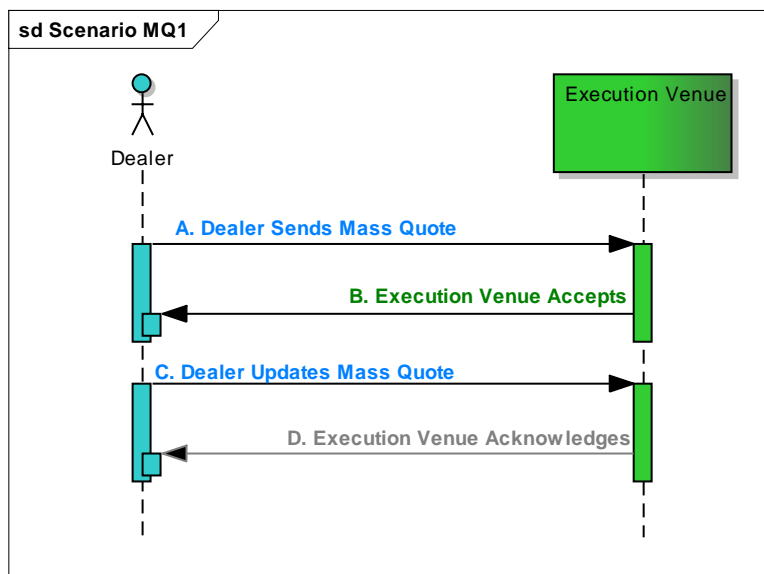


Figure 26: Scenario MQ1 – Dealer Sends Mass Quote to Execution Venue

Model Flow

The following table illustrates the flows expected when communicating with an Execution Venue implementing FIX 5.0 SP2.

Model FIX 5.0			
(A) Dealer Sends Mass Quote	Dealer	i - MassQuote QuoteID(117)= ❶ NoQuoteSets(296)= required e.g.(1) > NoQuoteEntries(295)= required e.g.(3) >> QuoteEntryID(299)= ❶ >> + quote attributes >> QuoteEntryID(299)= ❷ >> + quote attributes >> QuoteEntryID(299)= ❸ >> + quote attributes QuoteResponseLevel(301)= Acknowledge only negative or erroneous quotes(1)	Execution Venue
(B) Execution Venue Accepts		b- MassQuoteAck QuoteID(117)= ❶ QuoteStatus(297)=Accepted(0) <MassQuote Accepted> NoQuoteSets(296)=<count> (1) > NoQuoteEntries(295)=<count> (1) <i>Response only for bad quotes e.g. if ❷ is bad:</i> >> QuoteEntryRejectReason(368)=<enum :required> >> QuoteEntryID(299)= ❷ >>QuoteEntryStatus(1167)= Rejected(5)	

Model FIX 5.0			
(C) Dealer Updates Mass Quote	→	i - MassQuote QuoteID(117)= ⑤ NoQuoteSets(296)= required e.g. (1) >NoQuoteEntries(295)= required e.g.(2) >> QuoteEntryID(299)= ① >> + quote attributes >> QuoteMsgID(1166)=④ >> + quote attributes QuoteResponseLevel(301)= AcknowledgeOnlyNegativeOrErroneousQuotes(1)	
(D) Execution Venue Acknowledges (Optional)	←	b- MassQuoteAck QuoteID(117)= ⑤ QuoteStatus(297)=Accepted(0)	
The message flow in this scenario maybe followed by the message flow for the scenarios in:Trading - Quotes/Orders Based Trading (Volume 3)			

Table 24: Scenario MQ1 – Dealer Sends Mass Quote to Execution Venue

9.9 Scenario MQ2 – Dealer Cancels a Single Quote Sent Within a Mass Quote

This scenario shows the case where the Dealer sends a Mass Quote to the Execution Venue and later cancels one of these quotes.

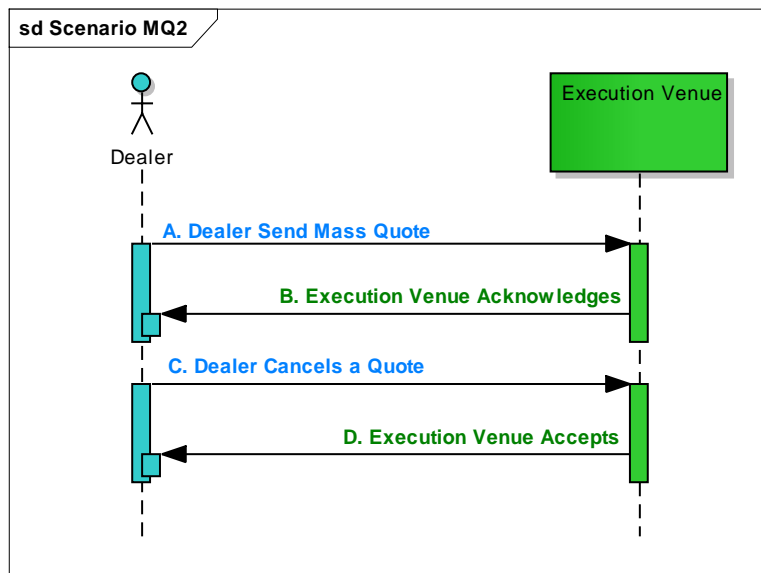


Figure 27: Scenario MQ2 – Dealer Cancels a Single Quote Sent Within a Mass Quote

Model Flow

The following table illustrates the flows expected when communicating with an Execution Venue implementing FIX 5.0 SP2.

Model FIX 5.0			
(A) Dealer Sends Mass Quote	Dealer	→	i - MassQuote QuoteID(117)= ❶ NoQuoteSets(296)= required e.g.(1) > NoQuoteEntries(295)= required e.g.(3) >> QuoteEntryID(299)= ❶ >> + quote attributes >> QuoteEntryID(299)= ❷ >> + quote attributes >> QuoteEntryID(299)= ❸ >> + quote attributes QuoteResponseLevel(301)= Acknowledge only negative or erroneous quotes(1)
(B) Execution Venue Acknowledges		←	b- MassQuoteAck QuoteID(117)= ❶ QuoteStatus(297)=Accepted(0) NoQuoteSets(296)= required > NoQuoteEntries(295)= required <i>Response only for bad quotes only, e.g.:</i> >> QuoteEntryRejectionReason(368)= required >> QuoteEntryID(299)= ❸ >>QuoteEntryStatus(1167)= Rejected(5)

Model FIX 5.0			
(C) Dealer Cancels a Quote		→	Z – QuoteCancel QuoteID(117)= ❶ QuoteCancelType(298)=required e.g. Cancel quote specified in QuoteID(5) QuoteResponseLevel(301)= required e.g. Acknowledge each quote message(2)
(D) Execution Venue Accepts		←	b– MassQuoteAck QuoteID(117)= ❶ QuoteStatus(297)=Accepted(0) NoQuoteSets(296)= 1 > NoQuoteEntries(295)= 1 >>QuoteEntryStatus (1167)= Accepted(0) >> QuoteEntryID(299)=❶

Table 25: Scenario MQ2 – Dealer Cancels a Single Quote Sent Within a Mass Quote

9.10 Scenario MQ3 – Dealer Cancels All Quotes Sent within a Mass Quote

This scenario is for the Dealer to cancel all quotes sent within Mass Quote.

Refer to scenario [PC3](#) for the message flow.

10 Message Detail

This section describes in detail all FIX application messages used in this volume. A summary of all the messages described in this volume is provided below.

Component name / Tag name	Description
Instrument	Instrument Component
InstrumentExtention	Extended Instrument Component
InstrAttribType(871)	FIX Tag InstrAttribType(871) values
UndInstrmtGrp	Undelying Instrument Component
Stipulations	Stipulations Component
StipulationType(233)	FIX Tag StipulationType(233) values
SpreadOrBenchmarkCurveData	Spread or Benchmark Curve Data Component
YieldData	Yield Data Component
YieldType(235)	FIX Tag YieldType(235) values
InstrmntLrgGrp	Multi-leg Instrument Component
MsgType	Description
x	Security List Request
y	Security List
BK	Security List Update Report
c	Security Definition Request
d	Security Definition
e	Security Status Request
f	Security Status
V	Market Data Request
W	Market Data Snapshot Full Refresh
X	Market Data Incremental Refresh
Y	Market Data Request Reject
S	Quote
Z	Quote Cancel
AI	Quote Status Report
i	Mass Quote
b	Mass Quote Ack
CW	Quote Ack

In the Message Detail tables below:

- Text appearing in **blue font** in **Req'd column** and/or **Descriptions column** indicates that the standard FIX description or Req'd field has been modified
- The Comment column contains the Best Practices comments

10.1 Tag PartyIDSource(447) values

Value	Description
For PartyRole = "InvestorID" and for CIV	
6	UK National Insurance or Pension Number
7	US Social Security Number
8	US Employer or Tax ID Number
9	Australian Business Number
A	Australian Tax File Number
For PartyRole = "InvestorID" and for Equities	
1	Korean Investor ID
2	Taiwanese Qualified Foreign Investor ID QFII/FID
3	Taiwanese Trading Acct
4	Malaysian Central Depository (MCD) number
5	Chinese Investor ID
For PartyRole="Broker of Credit"	
I	Directed broker three character acronym as defined in ISITC "ETC Best Practice" guidelines document
For all PartyRoles	
B	BIC (Bank Identification Code - SWIFT managed) code (ISO9362 - See "Appendix 6-B")
C	Generally accepted market participant identifier (e.g. NASD mnemonic)
D	Proprietary / Custom code
E	ISO Country Code
F	Settlement Entity Location (note if Local Market Settlement use "E=ISO Country Code") (see "Appendix 6-G" for valid values)
G	MIC (ISO 10383 - Market Identifier Code) (See "Appendix 6-C")
H	CSD participant/member code (e.g.. Euroclear, DTC, CREST or Kassenverein number)
K	Australian Company Number
L	Australian Registered Body Number
M	CFTC reporting firm identifier
N	Legal Entity Identifier (ISO 17442) LEI

10.2 Tag PartyRole(452) values

Value	Description
1	Executing Firm (formerly FIX 4.2 ExecBroker)
2	Broker of Credit (formerly FIX 4.2 BrokerOfCredit)
3	Client ID (formerly FIX 4.2 ClientID)
4	Clearing Firm (formerly FIX 4.2 ClearingFirm)
5	Investor ID
6	Introducing Firm
7	Entering Firm
8	Locate / Lending Firm (for short-sales)
9	Fund Manager Client ID (for CIV)
10	Settlement Location (formerly FIX 4.2 SettlLocation)
11	Order Origination Trader (associated with Order Origination Firm - i.e. trader who initiates/submits the order)
12	Executing Trader (associated with Executing Firm - actually executes)
13	Order Origination Firm (e.g. buy-side firm)
14	Giveup Clearing Firm (firm to which trade is given up)
15	Correspondant Clearing Firm
16	Executing System
17	Contra Firm
18	Contra Clearing Firm
19	Sponsoring Firm
20	Underlying Contra Firm
21	Clearing Organization
22	Exchange
24	Customer Account
25	Correspondent Clearing Organization
26	Correspondent Broker
27	Buyer/Seller (Receiver/Deliverer)
28	Custodian
29	Intermediary
30	Agent
31	Sub-custodian
32	Beneficiary
33	Interested party
34	Regulatory body
35	Liquidity provider
36	Entering trader
37	Contra trader
38	Position account
39	Contra Investor ID
40	Transfer to Firm
41	Contra Position Account
42	Contra Exchange
43	Internal Carry Account
44	Order Entry Operator ID
45	Secondary Account Number
46	Foreign Firm
47	Third Party Allocation Firm
48	Claiming Account
49	Asset Manager
50	Pledgor Account
51	Pledgee Account

52	Large Trader Reportable Account
53	Trader mnemonic
54	Sender Location
55	Session ID
56	Acceptable Counterparty
57	Unacceptable Counterparty
58	Entering Unit
59	Executing Unit
60	Introducing Broker
61	Quote originator
62	Report originator
63	Systematic internaliser (SI)
64	Multilateral Trading Facility (MTF)
65	Regulated Market (RM)
66	Market Maker
67	Investment Firm
68	Host Competent Authority (Host CA)
69	Home Competent Authority (Home CA)
70	Competent Authority of the most relevant market in terms of liquidity (CAL)
71	Competent Authority of the Transaction (Execution) Venue (CATV)
72	Reporting intermediary (medium/vendor via which report has been published)
73	Execution Venue
74	Market data entry originator
75	Location ID
76	Desk ID
77	Market data market
78	Allocation Entity
79	Prime Broker providing General Trade Services
80	Step-Out Firm (Prime Broker)
81	BrokerClearingID
82	Central Registration Depository (CRD)
83	Clearing Account
84	Acceptable Settling Counterparty
85	Unacceptable Settling Counterparty

10.3 Instrument Component

CDS & IRS Instrument Component

Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages" of the requested Security

Tag	FieldName	Req'd	Description	Comment
55	Symbol	Y	Common, "human understood" representation of the security. SecurityID value can be specified if no symbol exists (e.g. non-exchange traded Collective Investment Vehicles) Use "[N/A]" for products which do not have a symbol.	
65	SymbolSfx	N	Used in Fixed Income with a value of "WI" to indicate "When Issued" for a security to be reissued under an old CUSIP or ISIN or with a value of "CD" to indicate a EUCP with lump-sum interest rather than discount price.	
48	SecurityID	Y	Takes precedence in identifying security to counterparty over SecurityAltID block. Requires SecurityIDSource if specified.	
22	SecurityIDSource	Y	Required if SecurityID is specified.	
460	Product	N	Indicates the type of product the security is associated with (high-level category)	
167	SecurityType	Y	It is recommended that CFICode be used instead of SecurityType for non-Fixed Income instruments. Required for Fixed Income. Refer to Volume 7 - Fixed Income 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.)	
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. When using MaturityMonthYear, it is recommended that markets and sell sides report the MaturityDate on all outbound messages as a means of data enrichment. For NDFs this represents the fixing date of the contract.	
224	CouponPaymentDate	N	Date interest is to be paid. Used in identifying Corporate Bond issues.	
1449	RestructuringType	N	A category of CDS credit even in which the underlying bond experiences a restructuring. Used to define a CDS instrument.	
1450	Seniority	N	Specifies which issue (underlying bond) will receive payment priority in the event of a default. Used to define a CDS instrument.	
1451	NotionalPercentageOutstanding	N	Indicates the notional percentage of the deal that is still outstanding based on the remaining components of the index. Used to calculate the true value of a CDS trade or position.	
1452	OriginalNotionalPercentageOutstanding	N	Used to reflect the Original value prior to the application of a credit event. See NotionalPercentageOutstanding(1451).	
1457	AttachmentPoint	N	Lower bound percentage of the loss that the tranche can endure.	

1458	DetachmentPoint	N	Upper bound percentage of the loss the tranche can endure.	
1739	ObligationType	N	Type of reference obligation for credit derivatives contracts.	
225	IssueDate	N	Date instrument was issued. For Fixed Income IOIs for new issues, specifies the issue date.	
228	Factor	N	For Fixed Income: Amortization Factor for deriving Current face from Original face for ABS or MBS securities, note the fraction may be greater than, equal to or less than 1. In TIPS securities this is the Inflation index. Qty * Factor * Price = Gross Trade Amount For Derivatives: Contract Value Factor by which price must be adjusted to determine the true nominal value of one futures/options contract. (Qty * Price) * Factor = Nominal Value	
255	CreditRating	N	An evaluation of a company's ability to repay obligations or its likelihood of not defaulting. These evaluation are provided by Credit Rating Agencies, i.e. S&P, Moody's. (Note tag # was reserved in FIX 4.1, added in FIX 4.3)	
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).	
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.	
1191	RestructuringType	N	A category of CDS credit even in which the underlying bond experiences a restructuring. Used to define a CDS instrument.	
1197	ValuationMethod	N	Specifies the type of valuation method applied.	
223	CouponRate	N	For Fixed Income.	
106	Issuer	N	Name of security issuer (e.g. International Business Machines, GNMA). see also Volume 7: "PRODUCT: FIXED INCOME - Euro Issuer Values"	
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.	
691	Pool	N	Identifies MBS / ABS pool	
667	ContractSettlMonth	N	Must be present for MBS/TBA	

875	CPPProgram	N	The program under which a commercial paper is issued	
873	DatedDate	N	If different from IssueDate	
874	InterestAccrualDate	N	If different from IssueDate and DatedDate	

Table 26: Instrument Attributes

10.4 Security List Request (MsgType=x)

SecurityListRequest (x)				Dealer -> Execution Venue
The Security List Request message is used to return a list of securities from the counterparty that match criteria provided on the request				
Tag	FieldName	Req'd	Description	Comment
	StandardHeader	Y	MsgType = x (lowercase X)	
320	SecurityReqID	Y	Unique ID of a Security Definition Request.	
559	SecurityListRequestType	Y	Type of Security List Request being made	
1465	SecurityListID	N	Identifies a specific list	
1470	SecurityListType	N	Specifies a type of Security List.	Used as filter [Used when SecurityListRequestType(559)=SecurityType(1)]
1301	MarketID	N	Identifies the market which lists and trades the instrument.	Used as filter [Used when SecurityListRequestType(559)=MarketID(5)]
1300	MarketSegmentID	N	Identifies the segment of the market to which the specify trading rules and listing rules apply. The segment may indicate the venue, whether retail or wholesale, or even segregation by nationality.	Used as filter [Used when SecurityListRequestType(559)=MarketID + MarketSegmentID(5)]
	Instrument	N	Use of instrument identifiers is described in a separate section	Used to filter a single instrument request [Used when SecurityListRequestType(559)=Symbol(0)]
	InstrmtLegGrp	N	Use of instrument Leg identifiers is described in	

			a separate section	
555	NoLegs	N	Number of legs	
263	SubscriptionRequestType	N	Subscribe or unsubscribe for security status to security specified in request.	Mandatory for Execution Venues that supports reference data updates
	StandardTrailer	Y	The standard FIX message trailer	

Table 27: Security List Request (MsgType=x)

10.5 Security List (MsgType=y)

SecurityList (y)			Execution Venue -> Dealer	
The Security List message is used to return a list of securities that matches the criteria specified in a Security List Request.				
Tag	FieldName	Req'd	Description	Comment
	StandardHeader	Y	MsgType = y (lowercase Y)	
964	SecurityReportID	N	Identifies a Security List message.	
1465	SecurityListID	N	Identifies a specific Security List Entry	
1466	SecurityListRefID	N	Provides a reference to another Security List	
1467	SecurityListDesc	N	Specifies a description or name of a Security List.	
1470	SecurityListType	N	Identifies a list type	
320	SecurityReqID	Y	Unique ID of a Security Definition Request.	
322	SecurityResponseID	Y	Identifier for the Security List message	
560	SecurityRequestResult	Y	Result of the Security Request identified by the SecurityReqID	
60	TransactTime	N	Timestamp when the business transaction represented by the message occurred.	
393	TotNoRelatedSym	Y	Used to indicate the total number of securities being returned for this request. Used in the event that message fragmentation is required.	
1301	MarketID	N	Identifies the market which lists and trades the instrument.	
1300	MarketSegmentID	N	Identifies the segment of the market to which the specify trading rules and listing rules apply. The segment may indicate the venue, whether retail or wholesale, or even segregation by nationality.	
893	LastFragment	N	Indicates whether this is the last fragment in a sequence of message fragments. Only required where message has been fragmented.	
	SecListGrp	N	Specifies the number of repeating symbols (instruments) specified	
146	NoRelatedSym	N	Specifies the number of repeating symbols (instruments) specified	
->	Instrument	N	Use of instrument identifiers is described in a separate section	
->	InstrmtLegSecListGrp	N	Use of instrument Leg identifiers is described in a separate section	
	StandardTrailer	Y	The standard FIX message trailer	

Table 28: Security List (MsgType=y)

10.6 Security List Update Report (MsgType=BK)

SecurityListUpdateReport (BK)			Execution Venue -> Dealer	
The Security List Update Report is used for reporting updates to a Contract Security Masterfile. Updates could be due to Corporate Actions or other business events. Update may include additions, modifications and deletions.				
Tag	FieldName	Req'd	Description	Comment
	StandardHeader	Y	MsgType = BK	
964	SecurityReportID	N	Identifier for the Security List Update message in a bulk transfer environment (No Request/Response)	
1465	SecurityListID	N	Identifies a specific Security List entity	
1466	SecurityListRefID	N	Provides a reference to another Security List	
1467	SecurityListDesc	N	Specifies a description or name of a Security List.	
1470	SecurityListType	N	Identifies a list type	
320	SecurityReqID	Y	Unique ID of a Security Definition Request.	
322	SecurityResponseID	Y	Identifier for the Security List message.	
560	SecurityRequestResult	Y	Result of the Security Request identified by the SecurityReqID.	
393	TotNoRelatedSym	Y	Used to indicate the total number of securities being returned for this request. Used in the event that message fragmentation is required.	
980	SecurityUpdateAction	N		
1301	MarketID	N	Identifies the market which lists and trades the instrument.	
1300	MarketSegmentID	N	Identifies the segment of the market specified in MarketID(96)	
60	TransactTime	N	Timestamp when the business transaction represented by the message occurred.	
893	LastFragment	N	Indicates whether this is the last fragment in a sequence of message fragments. Only required where message has been fragmented.	
	SecLstUpdRelSymGrp	N	Specifies the number of repeating symbols (instruments) specified	
146	NoRelatedSym	N	Specifies the number of repeating symbols (instruments) specified	
->	Instrument	N	Use of instrument identifiers is described in a separate section	
->	SecLstUpdRelSymsLegGrp	N	Use of instrument Leg identifiers is described in a separate section	
	StandardTrailer	Y	The standard FIX message trailer	

Table 29: Security List Update Report (MsgType=BK)

10.7 Security Definition Request (MsgType=c)

SecurityDefinitionRequest (c)			Dealer -> Execution Venue	
<i>The Security Definition Request message is used for the following: 1. Request a specific Security to be traded with the second party. The request security can be defined as a multileg security made up of one or more instrument legs. 2. Request a set of individual securities for a single market segment. 3. Request all securities, independent of market segment.</i>				
Tag	FieldName	Req'd	Description	Comment
	StandardHeader	Y	MsgType = c (lowercase)	
320	SecurityReqID	Y	Unique ID of a Security Definition Request.	
	InstrmtLegGrp	Y	Use of instrument Leg identifiers is described in a separate section	
555	NoLegs	N	Number of legs	
	StandardTrailer	Y	The standard FIX message trailer	

Table 30: Security Definition Request (MsgType=c)

10.8 Security Definition (MsgType=d)

SecurityDefinition (d)			Execution Venue -> Dealer	
<i>The Security Definition message is used for the following: 1. Accept the security defined in a Security Definition message. 2. Accept the security defined in a Security Definition message with changes to the definition and/or identity of the security. 3. Reject the security requested in a Security Definition message. 4. Respond to a request for securities within a specified market segment. 5. Convey comprehensive security definition for all market segments that the security participates in. 6. Convey the security's trading rules that differ from default rules for the market segment.</i>				
Tag	FieldName	Req'd	Description	Comment
	StandardHeader	Y	MsgType = d (lowercase)	
964	SecurityReportID	N	Identifier for Security Definition message	
320	SecurityReqID	Y	Unique ID of a Security Definition Request.	
322	SecurityResponseID	N	Identifier for the Security Definition message	
323	SecurityResponseType	Y	Response to the Security Definition Request	
	Instrument	N	Use of instrument identifiers is described in a separate section	
	InstrmtLegGrp	N	Use of instrument Leg identifiers is described in a separate section	
555	NoLegs	N	Number of legs	
60	TransactTime	N	Timestamp when the business transaction represented by the message occurred.	

Table 31: Security Definition (MsgType=d)

10.9 Security Status Request (MsgType=e)

SecurityStatusRequest (e)			Dealer -> Execution Venue	
The Security Status Request message provides for the ability to request the status of a security. One or more Security Status messages are returned as a result of a Security Status Request message.				
Tag	FieldName	Req'd	Description	Comment
	StandardHeader	Y	MsgType = e (lowercase)	
324	SecurityStatusReqID	Y	Must be unique, or the ID of previous Security Status Request to disable if SubscriptionRequestType = Disable previous Snapshot + Updates Request (2).	
	Instrument	Y	Use of instrument identifiers is described in a separate section	Used as filter
	InstrmtLegGrp	N	Number of legs that make up the Security	
555	NoLegs	N	Number of legs	
263	SubscriptionRequestType	Y	SubscriptionRequestType indicates to the other party what type of response is expected. A snapshot request only asks for current information. A subscribe request asks for updates as the status changes. Unsubscribe will cancel any future update messages from the counter party.	
1301	MarketID	N	Identifies the Market	Used as filter
1300	MarketSegmentID	N	Identifies the market segment	Used as filter
336	TradingSessionID	N	Identifier for Trading Session A trading session spans an extended period of time that can also be expressed informally in terms of the trading day. Usage is determined by market or counterparties. To specify good for session where session spans more than one calendar day, use TimeInForce = Day in conjunction with TradingSessionID. Bilaterally agreed values of data type "String" that start with a character can be used for backward compatibility.	Used as filter
625	TradingSessionSubID	N	Optional market assigned sub identifier for a trading phase within a trading session. Usage is determined by market or counterparties. Used by US based futures markets to identify exchange specific execution time bracket codes as required by US market regulations. Bilaterally agreed values of data type "String" that start with a character can be used for backward compatibility	Used as filter
	StandardTrailer	Y	The standard FIX message trailer	

Table 32: Security Status Request (MsgType=e)

10.10 Security Status (MsgType=f)

SecurityStatus (f)			Execution Venue -> Dealer	
The Security Status message provides for the ability to report changes in status to a security. The Security Status message contains fields to indicate trading status, corporate actions, financial status of the company. The Security Status message is used by one trading entity (for instance an exchange) to report changes in the state of a security.				
Tag	FieldName	Req'd	Description	Comment
	StandardHeader	Y	MsgType = f (lowercase)	
324	SecurityStatusReqID	Y	Unique ID of a Security Status Request message.	
	Instrument	N	Use of instrument identifiers is described in a separate section	Mandatory unless the Security Status message is an error reply to an invalid Security Status Request [in such case, SecurityTradingStatus(326)=Unknown Or Invalid(20)]
	InstrmtLegGrp	N	Use of instrument Leg identifiers is described in a separate section	
555	NoLegs	N	Number of legs	
1301	MarketID	N	Identifies the Market	
1300	MarketSegmentID	N	Identifies the market segment	
336	TradingSessionID	N	Identifier for Trading Session A trading session spans an extended period of time that can also be expressed informally in terms of the trading day. Usage is determined by market or counterparties. To specify good for session where session spans more than one calendar day, use TimeInForce = Day in conjunction with TradingSessionID. Bilaterally agreed values of data type "String" that start with a character can be used for backward compatibility.	
625	TradingSessionSubID	N	Optional market assigned sub identifier for a trading phase within a trading session. Usage is determined by market or counterparties. Used by US based futures markets to identify exchange specific	

			execution time bracket codes as required by US market regulations. Bilaterally agreed values of data type "String" that start with a character can be used for backward compatibility	
325	UnsolicitedIndicator	N	Set to 'Y' if message is sent as a result of a subscription request not a snapshot request	
326	SecurityTradingStatus	Y	Identifies the trading status applicable to the transaction.	
1174	SecurityTradingEvent	N	Identifies an event related to the trading status	
291	FinancialStatus	N	Identifies a firm's or a security's financial status	
327	HaltReason	N	Denotes the reason for the Opening Delay or Trading Halt.	
328	InViewOfCommon	N	Indicates whether or not the halt was due to Common Stock trading being halted.	
329	DueToRelated	N	Indicates whether or not the halt was due to the Related Security being halted.	
1021	MDBBookType	N	Used to relay changes in the book type	
264	MarketDepth	N	Used to relay changes in Market Depth.	
330	BuyVolume	N	Quantity bought.	
331	SellVolume	N	Quantity sold.	
332	HighPx	N	Represents an indication of the high end of the price range for a security prior to the open or reopen	
333	LowPx	N	Represents an indication of the low end of the price range for a security prior to the open or reopen	
31	LastPx	N	Represents the last price for that security either on a Consolidated or an individual participant basis at the time it is disseminated.	
60	TransactTime	N	Trade Dissemination Time	

1025	FirstPx	N	Represents the price of the first fill of the trading session.	
58	Text	N	Comment, instructions, or other identifying information.	
	StandardTrailer	Y	The standard FIX message trailer	

Table 33: Security Status (MsgType=f)

10.11 Market Data Request (MsgType=V)

MarketDataRequest (V)			Dealer -> Execution Venue	
Some systems allow the transmission of real-time quote, order, trade, trade volume, open interest, and/or other price information on a subscription basis. A Market Data Request is a general request for market data on specific securities or forex quotes.				
Tag	FieldName	Req'd	Description	Comment
	StandardHeader	Y	MsgType = V	
262	MDReqID	Y	Must be unique, or the ID of previous Market Data Request to disable if SubscriptionRequestType = Disable previous Snapshot + Updates Request (2).	
263	SubscriptionRequestType	Y	SubscriptionRequestType indicates to the other party what type of response is expected. A snapshot request only asks for current information. A subscribe request asks for updates as the status changes. Unsubscribe will cancel any future update messages from the counter party.	
264	MarketDepth	Y	Depth of market for Book Snapshot / Incremental updates 0 - full book depth 1 - top of book 2 and above - book depth (number of levels)	
265	MDUpdateType	N	Required if SubscriptionRequestType = Snapshot + Updates (1).	
266	AggregatedBook	N	Specifies whether or not book entries should be aggregated. (Not specified) = broker option	
	MDReqGrp	Y	Number of MDEntryType fields requested.	
267	NoMDEntryTypes	Y	Number of MDEntryType fields requested.	
-> 269	MDEntryType	Y	Must be the first field in this repeating group. This is a list of all the types of Market Data Entries that the firm requesting the Market Data is interested in receiving.	
	InstrmtMDReqGrp	Y	Number of symbols (instruments) requested.	
146	NoRelatedSym	Y	Number of symbols (instruments) requested.	
->	Instrument	Y	Use of instrument identifiers is described in a separate section	
1070	MDQuoteType	N	Identifies market data quote type.	
	StandardTrailer	Y	The standard FIX message trailer	

Table 34: Market Data Request (MsgType=V)

10.12 Market Data Snapshot Full Refresh (MsgType=W)

MarketDataSnapshotFullRefresh (W)			Execution Venue -> Dealer	
The Market Data messages are used as the response to a Market Data Request message. In all cases, one Market Data message refers only to one Market Data Request. It can be used to transmit a 2-sided book of orders or list of quotes, a list of trades, index values, opening, closing, settlement, high, low, or VWAP prices, the trade volume or open interest for a security, or any combination of these.				
Tag	FieldName	Req'd	Description	Comment
	StandardHeader	Y	MsgType = W	
911	TotNumReports	N	Total number or reports returned in response to a request.	
963	MDReportID	Y	Unique indentifier for Market Data Report	
264	MarketDepth	N	Depth of market for Book Snapshot / Incremental updates 0 - full book depth 1 - top of book 2 and above - book depth (number of levels)	
1187	RefreshIndicator	N	Set by the sender to tell the receiver to perform an immediate refresh of the book due to disruptions in the accompanying real-time feed 'Y' - Mandatory refresh by all participants 'N' - Process as required	
262	MDReqID	N	Conditionally required if this message is in response to a Market Data Request.	
	Instrument	Y	Use of instrument identifiers is described in a separate section	
	MDFullGrp	Y	Number of entries following.	This component should include all the pricing fields. All Pricing FIX fields in this component are optional.
268	NoMDEntries	Y	Number of entries following.	
->269	MDEntryType	Y	Must be the first field in this repeating group.	
->270	MDEntryPx	N	Conditionally required if MDEntryType is not Imbalance(A)), Trade Volume (B), or Open Interest(C); Conditionally required when MDEntryType = "auction clearing price"	
->271	MDEntrySize	N	Conditionally required if MDEntryType = Bid(0), Offer(1), Trade(2)), Trade Volume (B), or Open Interest(C) conditionally required when MDEntryType = "auction clearing price"	
	StandardTrailer	Y	The standard FIX message trailer	

Table 35: Market Data Snapshot Full Refresh (MsgType=W)

10.13 Market Data Incremental Refresh (MsgType=X)

MarketDataIncrementalRefresh (X)			Execution Venue -> Dealer	
The Market Data message for incremental updates may contain any combination of new, changed, or deleted Market Data Entries, for any combination of instruments, with any combination of trades, imbalances, quotes, index values, open, close, settlement, high, low, and VWAP prices, trade volume and open interest so long as the maximum FIX message size is not exceeded. All of these types of Market Data Entries can be changed and deleted.				
Tag	FieldName	Req'd	Description	Comment
	StandardHeader	Y	MsgType = X	
262	MDReqID	N	Conditionally required if this message is in response to a Market Data Request.	
	MDIncGrp	Y	Number of entries following.	This component should include all the pricing fields. All Pricing FIX fields in this component are optional.
268	NoMDEntries	Y	Number of entries following.	
-> 279	MDUpdateAction	Y	Must be first field in this repeating group.	
-> 264	MarketDepth	N	Depth of market for Book Snapshot / Incremental updates 0 - full book depth 1 - top of book 2 and above - book depth (number of levels)	
-> 269	MDEntryType	N	Conditionally required if MDUpdateAction = New(0). Cannot be changed.	
-> 278	MDEntryID	N	If specified, must be unique among currently active entries if MDUpdateAction = New (0), must be the same as a previous MDEntryID if MDUpdateAction = Delete (2), and must be the same as a previous MDEntryID if MDUpdateAction = Change (1) and MDEntryRefID is not specified, or must be unique among currently active entries if MDUpdateAction = Change(1) and MDEntryRefID is specified..	
-> 280	MDEntryRefID	N	If MDUpdateAction = New(0), for the first Market Data Entry in a message, either this field or a Symbol must be specified. If MDUpdateAction = Change(1), this must refer to a previous MDEntryID.	
->	Instrument	N	Use of instrument identifiers is described in a separate section	
-> 270	MDEntryPx	N	Conditionally required when MDUpdateAction = New(0) and MDEntryType is not Imbalance(A)), Trade Volume (B), or Open Interest (C). Conditionally required when MDEntryType = "auction clearing	

			price"	
-> 271	MDEntrySize	N	Conditionally required when MDUpdateAction = New(0) and MDEntryType = Bid(0), Offer(1), Trade(2)), Trade Volume(B), or Open Interest(C). Conditionally required when MDEntryType = "auction clearing price"	
-> 60	TransactTime	N	For optional use in reporting Trades. Used to specify the time of matching.	
	StandardTrailer	Y	The standard FIX message trailer	

Table 36: Market Data Incremental Refresh (MsgType=X)

10.14 Market Data Request Reject (MsgType=Y)

MarketDataRequestReject (Y)			Execution Venue -> Dealer	
<i>The Market Data Request Reject is used when the broker cannot honor the Market Data Request, due to business or technical reasons. Brokers may choose to limit various parameters, such as the size of requests, whether just the top of book or the entire book may be displayed, and whether Full or Incremental updates must be used.</i>				
Tag	FieldName	Req'd	Description	Comment
	StandardHeader	Y	MsgType = Y	
262	MDReqID	Y	Must refer to the MDReqID of the request.	
281	MDReqRejReason	N	Reason for the rejection of a Market Data request.	
	StandardTrailer	Y	The standard FIX message trailer	

Table 37: Market Data Request Reject (MsgType=Y)

10.15 Quote (MsgType=S)

Quote (S)			Dealer -> Execution Venue	
The Quote message is used as the response to a Quote Request or a Quote Response message in both indicative, tradeable, and restricted tradeable quoting markets.				
Tag	FieldName	Req'd	Description	Comment
	StandardHeader	Y	MsgType = S	
131	QuoteReqID	N	Required when quote is in response to a Quote Request message	
117	QuoteID	Y	Unique identifier for quote	
1166	QuoteMsgID	N	Optionally used to supply a message identifier for a quote.	
693	QuoteRespID	N	Required when responding to the Quote Response message. The counterparty specified ID of the Quote Response message.	
537	QuoteType	N	Quote Type If not specified, the default is an indicative quote	
301	QuoteResponseLevel	N	Level of Response requested from receiver of quote messages.	
	Parties	N	For use by the Dealer to specify the trader/desk who issued the quote	
	Instrument	Y	Use of instrument identifiers is described in a separate section	
54	Side	N	Required for Tradeable or Counter quotes of single instruments	
	OrderQtyData	N	Quote Quantity	
38	OrderQty	N	The notional amount for outright IRS and CDS quotes.	
	LegQuotGrp	N	Required for multileg quotes	
555	NoLegs	N	Required for multileg quotes	
->	InstrumentLeg	N	Required for multileg quotes Use of instrument identifiers is described in a separate section	
-> 686	LegPriceType	N	Code to represent type of price presented in LegBidPx and LegOfferPx. Required if LegBidPx or PegOfferPx is present.	
-> 681	LegBidPx	N	Bid price of this leg. See BidPx (32) for description and valid values.	
-> 684	LegOfferPx	N	Offer price of this leg. See OfferPx (133) for description and valid values	
132	BidPx	N	BidPx, OfferPx or both must be specified.	
133	OfferPx	N	BidPx, OfferPx or both must be specified.	
134	BidSize	N	Specifies the bid size. If MinBidSize is specified, BidSize is interpreted to contain the maximum bid size.	
135	OfferSize	N	Specified the offer size. If MinOfferSize is specified, OfferSize is interpreted to contain the maximum offer size.	
62	ValidUntilTime	N	The time when the quote will expire	
423	PriceType	N	Code to represent the price type. (For	

			Financing transactions PriceType implies the "repo type" - Fixed or Floating - 9 (Yield) or 6 (Spread) respectively - and Price (44) gives the corresponding "repo rate". See Volume : "Glossary" for further value definitions)	
	StandardTrailer	Y	The standard FIX message trailer	

Table 38: Quote (MsgType=S)

10.16 Quote Cancel (MsgType=Z)

QuoteCancel (Z)		Dealer -> Execution Venue		
The Quote Cancel message is used by an originator of quotes to cancel quotes. The Quote Cancel message supports cancellation of: • All quotes • Quotes for a specific symbol or security ID • All quotes for a security type • All quotes for an underlying				
Tag	FieldName	Req'd	Description	Comment
	StandardHeader	Y	MsgType = Z	
131	QuoteReqID	N	Required when quote is in response to a Quote Request message	
117	QuoteID	N	Conditionally required when QuoteCancelType(298) = 5 (cancel quote specified in QuoteID). Maps to QuoteID(117) of a single Quote(MsgType=S) or QuoteEntryID(299) of a MassQuote(MsgType=i).	
1166	QuoteMsgID	N	Optionally used to supply a message identifier for a quote cancel.	
298	QuoteCancelType	Y	Identifies the type of Quote Cancel request.	
537	QuoteType	N	Conditional Required when QuoteCancelType(298)=6[Cancel by QuoteType]	
301	QuoteResponseLevel	N	Level of Response requested from receiver of quote messages.	
	QuotCxlEntriesGrp	N	The number of securities (instruments) whose quotes are to be canceled Not required when cancelling all quotes.	
295	NoQuoteEntries	N	The number of securities (instruments) whose quotes are to be canceled Not required when cancelling all quotes.	
->	Instrument	N	Use of instrument identifiers is described in a separate section	
	StandardTrailer	Y	The standard FIX message trailer	

Table 39: Quote Cancel (MsgType=Z)

10.17 Quote Status Report (MsgType=AI)

QuoteStatusReport (AI)			Execution Venue -> Dealer	
The quote status report message is used: • as the response to a Quote Status Request message • as a response to a Quote Cancel message • as a response to a Quote Response message in a negotiation dialog (see Volume 7 – PRODUCT: FIXED INCOME and USER GROUP: EXCHANGES AND MARKETS)				
Tag	FieldName	Req'd	Description	Comment
	StandardHeader	Y	MsgType = AI	
649	QuoteStatusReqID	N	Unique identifier for Quote Status Request.	
131	QuoteReqID	N	Required when quote is in response to a Quote Request message	
117	QuoteID	Y	Maps to QuoteID(117) of a single Quote(MsgType=S) or QuoteEntryID(299) of a MassQuote(MsgType=i).	
1166	QuoteMsgID	Y	Maps to QuoteComponentID(1166) of a single Quote(MsgType=S) or QuoteID(117) of a MassQuote(MsgType=i).	
693	QuoteRespID	N	Required when responding to a Quote Response message.	
537	QuoteType	N	Quote Type If not specified, the default is an indicative quote	
298	QuoteCancelType	N	Identifies the type of quote cancel.	
	Instrument	N	Conditionally required when reporting status of a single security quote. Use of instrument identifiers is described in a separate section	
54	Side	N	Side of order (see Volume : "Glossary" for value definitions)	
126	ExpireTime	N	Time/Date of order expiration (always expressed in UTC (Universal Time Coordinated, also known as "GMT")) The meaning of expiration is specific to the context where the field is used. For orders, this is the expiration time of a Good Til Date TimeInForce. For Quotes - this is the expiration of the quote. Expiration time is provided across the quote message dialog to control the length of time of the overall quoting process. For collateral requests, this is the time by which collateral must be assigned. For collateral assignments, this is the time by which a response to the assignment is expected.	
44	Price	N	Price per unit of quantity (e.g. per share)	
423	PriceType	N	Code to represent the price type. (For Financing transactions PriceType implies the "repo type" - Fixed or Floating - 9 (Yield) or 6 (Spread) respectively - and Price (44) gives the corresponding "repo rate". See Volume : "Glossary" for further value	

			definitions)	
132	BidPx	N	Bid Price	
133	OfferPx	N	Offer Price	
134	BidSize	N	Specifies the bid size. If MinBidSize is specified, BidSize is interpreted to contain the maximum bid size.	
135	OfferSize	N	Specified the offer size. If MinOfferSize is specified, OfferSize is interpreted to contain the maximum offer size.	
62	ValidUntilTime	N	Indicates expiration time of indication message (always expressed in UTC (Universal Time Coordinated, also known as "GMT"))	
60	TransactTime	N	Timestamp when the business transaction represented by the message occurred.	
297	QuoteStatus	Y	Quote Status	
300	QuoteRejectReason	N	Reason Quote was rejected	
	StandardTrailer	Y	The standard FIX message trailer	

Table 40: Quote Status Report (MsgType=AI)

10.18 Mass Quote (MsgType=i)

MassQuote (i)		Dealer -> Execution Venue		
The Mass Quote message can contain quotes for multiple securities to support applications that allow for the mass quoting of an option series. Two levels of repeating groups have been provided to minimize the amount of data required to submit a set of quotes for a class of options (e.g. all option series for IBM).				
Tag	FieldName	Req'd	Description	Comment
	StandardHeader	Y	MsgType = i (lowercase)	
131	QuoteReqID	N	Required when quote is in response to a Quote Request message	
117	QuoteID	Y	Unique identifier for quote	
537	QuoteType	N	Type of Quote Default is Indicative if not specified	
301	QuoteResponseLevel	N	Level of Response requested from receiver of quote messages.	
	Parties	N	For use by the Dealer to specify the trader/desk who issued the quote	
293	DefBidSize	N	Default Bid Size for quote contained within this quote message - if not explicitly provided.	
294	DefOfferSize	N	Default Offer Size for quotes contained within this quote message - if not explicitly provided.	
	QuotSetGrp	Y	The number of sets of quotes in the message	
296	NoQuoteSets	Y	The number of sets of quotes in the message	
->302	QuoteSetID	Y	Sequential number for the Quote Set. For a given QuoteID - assumed to start at 1. Must be the first field in the repeating group.	
->367	QuoteSetValidUntilTime	N	Indicates expiration time of this particular QuoteSet (always expressed in UTC (Universal Time Coordinated, also known as "GMT"))	
->304	TotNoQuoteEntries	Y	Total number of quotes for the quote set across all messages. Should be the sum of all NoQuoteEntries in each message that has repeating quotes that are part of the same quote set.	
->893	LastFragment	N	Indicates whether this is the last fragment in a sequence of message fragments. Only required where message has been fragmented.	
->	QuotEntryGrp	Y		This component should include all the pricing fields. All Pricing FIX fields in this component are

				optional.
->->> 295	NoQuoteEntries	Y	The number of quotes for this Symbol (instrument) (QuoteSet) that follow in this message.	
->->>> 299	QuoteEntryID	Y	Uniquely identifies the quote across the complete set of all quotes for a given quote provider.	
->->>	Instrument	N	Use of instrument identifiers is described in a separate section	
->->>> 62	ValidUntilTime	N	Indicates expiration time of indication message (always expressed in UTC (Universal Time Coordinated, also known as "GMT"))	
->->>> 60	TransactTime	N	Timestamp when the business transaction represented by the message occurred.	
	StandardTrailer	Y	The standard FIX message trailer	

Table 41: Mass Quote (MsgType=i)

10.19 Mass Quote Ack (MsgType=b)

MassQuoteAcknowledgement (b)			Execution Venue -> Dealer	
Mass Quote Acknowledgement is used as the application level response to a Mass Quote message.				
Tag	FieldName	Req'd	Description	Comment
	StandardHeader	Y	MsgType = b (lowercase)	
131	QuoteReqID	N	Required when acknowledgment is in response to a Quote Request message	
117	QuoteID	N	Required when acknowledgment is in response to a Mass Quote, mass Quote Cancel or mass Quote Status Request message. Maps to: - QuoteID(117) of a Mass Quote - QuoteMsgID(1166) of Quote Cancel - QuoteStatusReqID(649) of Quote Status Request	
297	QuoteStatus	Y	MassQuote or QuoteCancel acknowledgement status	Accepted - MassQuote or QuoteCancel accepted; Rejected - MassQuote or QuoteCancel rejected;
300	QuoteRejectReason	N	Reason Quote was rejected.	
301	QuoteResponseLevel	N	Level of Response requested from receiver of quote messages. Is echoed back to the counterparty.	
537	QuoteType	N	Type of Quote	
	QuotSetAckGrp	N	The number of sets of quotes in the message	
296	NoQuoteSets	N	The number of sets of quotes in the message	
-> 302	QuoteSetID	N	First field in repeating group. Required if NoQuoteSets > 0	
-> 304	TotNoQuoteEntries	N	Total number of quotes for the quote set across all messages. Should be the sum of all NoQuoteEntries in each message that has repeating quotes that are part of the same quote set. Required if NoQuoteEntries > 0	
-> 1168	TotNoCxlQuotes	N	Total number of quotes canceled for the quote set across all messages.	
-> 1169	TotNoAccQuotes	N	Total number of quotes accepted for the quote set across all messages.	
-> 1170	TotNoRejQuotes	N	Total number of quotes rejected for the quote set across all messages.	
-> 893	LastFragment	N	Indicates whether this is the last	

			fragment in a sequence of message fragments. Only required where message has been fragmented.	
->	QuotEntryAckGrp	N		
->-> 295	NoQuoteEntries	N	The number of quotes for this Symbol (QuoteSet) that follow in this message.	
->->- 299	QuoteEntryID	N	Uniquely identifies the quote across the complete set of all quotes for a given quote provider. First field in repeating group. Required if NoQuoteEntries > 0.	
->->- 1167	QuoteEntryStatus	N	MassQuote or QuoteCancel entry acknowledgement status	Accepted - Entry in MassQuote or QuoteCancel accepted; Rejected - Entry in MassQuote or QuoteCancel rejected
->->- 368	QuoteEntryRejectReason	N	Reason Quote Entry was rejected.	
	StandardTrailer	Y	The standard FIX message trailer	

Table 42: Mass Quote Ack (MsgType=b)

10.20 Quote Ack (MsgType=CW)

QuoteAck (CW)		Execution Venue -> Dealer		
The QuoteAck(35=CW) message is used to acknowledge a Quote(35=S) submittal or request to cancel an individual quote using the QuoteCancel(35=Z) message during a Quote/Negotiation dialog.				
The QuoteAck(35=CW) is available for optional use to acknowledge the request to cancel an individual quote (QuoteCancel(35=Z) with QuoteCancelType(298) =5(Cancel specified single quote))				
Tag	FieldName	Req'd	Description	Comment
	StandardHeader	Y	MsgType = CW	
117	QuoteID	N	Contains the QuoteID(117) of a single Quote(35=S).	
1166	QuoteMsgID	N	Contains the QuoteMsgID(1166) of a single Quote(35=S) or QuoteCancel(35=Z)	
131	QuoteReqID	N	Required when acknowledgment is in response to a Quote Request message	
537	QuoteType	N	Type of Quote	
298	QuoteCancelType	N		
1865	QuoteAckStatus	Y	Acknowledgement status of a Quote(35=S) or QuoteCancel(35=Z) message submission. Valid values: 0 = Received, not yet processed 1 = Accepted 2 = Rejected	
300	QuoteRejectReason	N	Reason Quote was rejected.	
1328	RejectText			
1664	EncodedRejectTextLen			
1665	EncodedRejectText			
	StandardTrailer	Y	The standard FIX message trailer	

Table 43: Quote Ack (MsgType=CW)

Appendix I - Document Figures

Figure 1: Scenario MC1- Example of IRS Gap Spread Definition and Usage	26
Figure 2: Scenario MC2 - Example of Duration Based IRS Butterfly	29

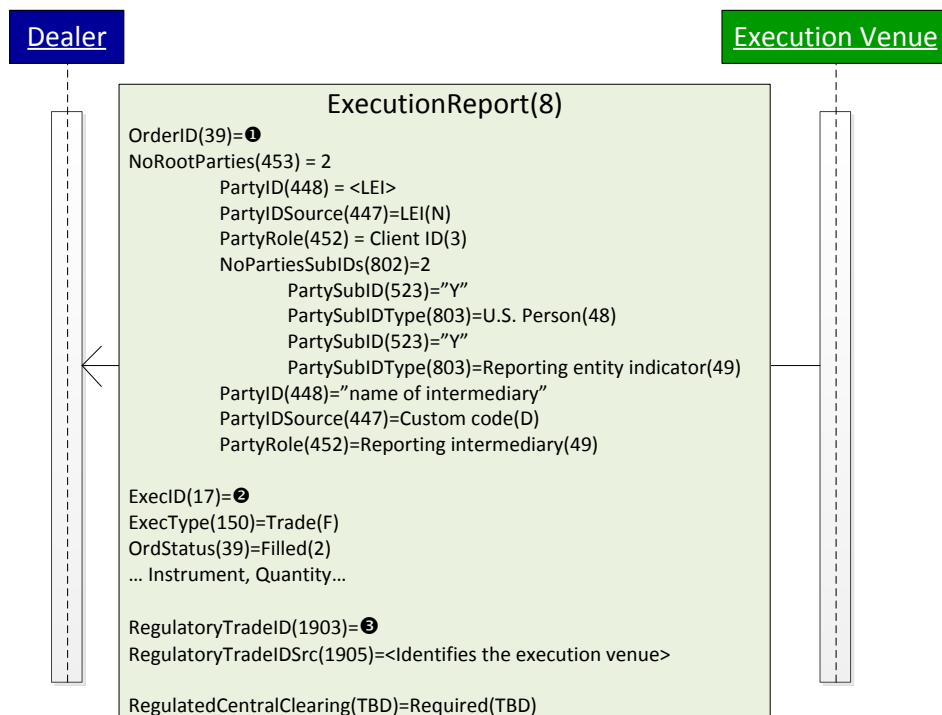


Figure 3: Flow of

'Required Transaction'	36
Figure 4: Flow of 'Permitted Transaction'	36
Figure 6: Scenario SL2 – Dealer Requests Snapshot of Instruments, Execution Venue Returns Snapshot (fragmented)	41
Figure 7: Scenario SL3 – Dealer Subscribes to Instruments Reference Data, Execution Venue Returns Snapshot and Updates	43
Figure 8: Scenario SL4 – Dealer Cancels Instrument Reference Data Subscription	45
Figure 9: Scenario SL5 – Dealer Requests Snapshot or Subscription of Instruments Reference Data, Execution Venue Returns Error	46
Figure 10: Scenario SL6 – Dealer Cancels Instrument Reference Data Subscription, Execution Venue Rejects	47
Figure 11: Scenario SS1 – Dealer Requests Instrument Status, Execution Venue Returns Status	48
Figure 12: Scenario SS2 – Dealer Subscribes to Instrument Status, Execution Venue Returns Snapshot and Updates	49
Figure 13: Scenario SS3 – Dealer Requests Instrument Status, Execution Venue Returns Error	50
Figure 14: Scenario SD1 – Dealer Sends Instrument Definition Request, Execution Venue Confirms	51
Figure 15: Scenario SD2 – Dealer Sends Instrument Definition Request, Execution Venue Rejects	52
Figure 16: Scenario SD3 – Execution Venue Publishes Instrument Definition	53
Figure 17: Scenario PP1 – Dealer Sends Different Prices for the Same Instrument to Multiple Customers – Dealer Amends Trade Information (Price & type)	55
Figure 18: Scenario MDS1 – Dealer Requests/Receives Market Data	59
Figure 19: Scenario MDS2 – Dealer Unsubscribes Market Data	60
Figure 20: Scenario MDS3 – Dealer Requests Market Data, Execution Venue Rejects	61
Figure 21: Scenario PC1 – Dealer Sends Quote to Market	63
Figure 22: Scenario PC2 – Dealer Cancels Quote on Market	64
Figure 23: Scenario PC3 – Dealer Cancels All Quotes	65
Figure 24: Scenario PC4 – Dealer Sends Quote to Market, Execution Venue Rejects	66
Figure 25: Scenario PC5 – Dealer Cancels Quote, Execution Venue Rejects	67
Figure 26: Scenario MQ1 – Dealer Sends Mass Quote to Execution Venue	68
Figure 27: Scenario MQ2 – Dealer Cancels a Single Quote Sent Within a Mass Quote	70

Appendix II - Document Tables

Table 1: Scenario MC1- Example of IRS Gap Spread Definition and Usage.....	28
Table 2: Scenario MC2 - Example of Duration Based IRS Butterfly	31
Table 3: Scenario SL1 – Dealer Requests Snapshot of Instruments, Execution Venue Returns Snapshot	39
Table 4: Scenario SL2 – Dealer Requests Snapshot of Instruments, Execution Venue Returns Snapshot (fragmented)	42
Table 5: Scenario SL3 – Dealer Subscribes to Instruments Reference Data, Execution Venue Returns Snapshot and Updates...	44
Table 6: Scenario SL4 – Dealer Cancels Instrument Reference Data Subscription.....	45
Table 7: Scenario SL5 – Dealer Requests Snapshot or Subscription of Instruments Reference Data, Execution Venue Returns Error	46
Table 8: Scenario SL6 – Dealer Cancels Instrument Reference Data Subscription, Execution Venue Rejects	47
Table 9: Scenario SS1 – Dealer Requests Instrument Status, Execution Venue Returns Status.....	48
Table 10: Scenario SS2 – Dealer Subscribes to Instrument Status, Execution Venue Returns Snapshot and Updates.....	49
Table 11: Scenario SS3 – Dealer Requests Instrument Status, Execution Venue Returns Error	50
Table 12: Scenario SD1 – Dealer Sends Instrument Definition Request, Execution Venue Confirms	51
Table 13: Scenario SD2 – Dealer Sends Instrument Definition Request, Execution Venue Rejects	52
Table 14: Scenario SD3 – Execution Venue Publishes Instrument Definition	53
Table 15: Scenario PP1 – Dealer Sends Different Prices for the Same Instrument to Multiple Customers – Dealer Amends Trade Information (Price & type)	57
Table 16: Scenario MDS1 – Dealer Requests/Receives Market Data	59
Table 17: Scenario MDS2 – Dealer Unsubscribes Market Data	60
Table 18: Scenario MDS3 – Dealer Requests Market Data, Execution Venue Rejects	61
Table 19: Scenario PC1 – Dealer Sends Quote to Market	63
Table 20: Scenario PC2 – Dealer Cancels Quote on Market.....	64
Table 21: Scenario PC3 – Dealer Cancels All Quotes.....	65
Table 22: Scenario PC4 – Dealer Sends Quote to Market, Execution Venue Rejects.....	66
Table 23: Scenario PC5 – Dealer Cancels Quote, Execution Venue Rejects	67
Table 24: Scenario MQ1 – Dealer Sends Mass Quote to Execution Venue.....	69
Table 25: Scenario MQ2 – Dealer Cancels a Single Quote Sent Within a Mass Quote	71
Table 26: Instrument Attributes	79
Table 27: Security List Request (MsgType=x).....	81
Table 28: Security List (MsgType=y).....	82
Table 29: Security List Update Report (MsgType=BK).....	83
Table 30: Security Definition Request (MsgType=c)	84
Table 31: Security Definition (MsgType=d)	85
Table 32: Security Status Request (MsgType=e)	86
Table 34: Market Data Request (MsgType=V)	90
Table 35: Market Data Snapshot Full Refresh (MsgType=W)	91
Table 36: Market Data Incremental Refresh (MsgType=X)	93
Table 37: Market Data Request Reject (MsgType=Y).....	94
Table 38: Quote (MsgType=S).....	96
Table 39: Quote Cancel (MsgType=Z)	97
Table 40: Quote Status Report (MsgType=AI)	99
Table 41: Mass Quote (MsgType=i)	101
Table 42: Mass Quote Ack (MsgType=b).....	103
Table 43: Quote Ack (MsgType=CW)	104