Global Fixed Income Committee

Best Practices for Trading Fixed Income Instruments

Cash Bonds

VOLUME 2 – Common Workflows

Editor(s): Produced by: FIX Trading Community Global Fixed Income Committee
Date: Tuesday, 16 December 2014
Version: 2.4
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 ORCONSEQUENTIAL (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 FIX TRADING COMMUNITY 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).

Table of Contents

Document History

Date	Change	Version
20 June 2012	Initial Version	0.1
26 June 2012	Modified description of scenarios: SL1 and SD3	0.2
25 Sept 2012	Added section for post-trade allocations	0.3
	Public release phase 1	1.0
12 October 2012	Split post-trade allocation to volume 5	1.1
02 April 2012	Public release phase 2	1.2
30 April 2012	Added new section for Pre-Trade Price Contribution	1.3
10 May 2013	Minor corrections in Pre-Trade Price Contribution	1.4
07 April 2014	Document ready for FIX Trading Community member firm review followed by public release (Phase 3)	2.0
17 October 2014	Added new section for Pre-Trade Indications	2.1
23 October 2014	Updated the Pre-Trade Indications section	2.2
17 November 2014	Updated the Pre-Trade Indications section	2.3
16 December 2014	Updated with FIX specification changes (EP194)	2.4

VOLUME INDEX

1 Introduction

This document represents volume 2 of the *Best Practices for Trading Fixed Income Instruments - Cash Bonds* 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
- Pre-Trade – Indications

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 'ResentRequest' 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), ResentRequest(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 Infrastructre 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.

3 Instrument Identifiers

3.1 Scope

3.1.1 Instrument Types

The following table summarizes the Instrument types that are in scope.

Cash Bonds	
Rates	Credit
• U.S. Treasuries • European Government Bonds • Japanese Government Bonds • Australian Government Bonds • U.S. Agencies • Gilts	• European Credit • Covered Bonds • Supranationals • Agencies and Sovereigns • Mortgage-Backed Securities • U.S. Corporate Bonds

3.1.2 Trading Types

- Outright Instruments (Cash Bonds)
- 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 Workflows

- Reference Data
- Market Data
- Quote Contribution
- Quotation Negotiation
- Order Quote Based
- Central Order Book
- Post Trade – to be completed

3.2 Instrument Identifier Recommendations

1. The definition of an instrument should be supplied within the FIX Instrument component
2. Each intrument should have a unique identifier which is provided in tag SecurityID(48)
3. For Cash Bonds, many Execution Venues use ISIN or CUSIP as the SecurityIDSource, although other Execution Venues may use different values
4. Symbol tag is requires and should contain human readable name of the Instrument or "[N/A]" if not available

3.3 Instrument Identifier Attributes

3.3.1 Required Instrument Identifier Attributes

- The following minimum set of attributes are used in all FIX messages to refer to an Instrument (some FIX messages will require additional 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> SecurityType(167)=<required>

3.3.2 Cash Bond Instrument Identifier Attributes

The following table summarizes the the relevant FIX component that conatains the instrument identifier attributes:

FIX component	Component Description
Instrument Component	Instrument attributes that are relevant to Cash Bonds
Extended Instrument Component	Extended Instrument attributes that are relevant to Cash Bonds
Undelying Instrument Component	Attributes that are related to the underlying Instruments
Stipulations Component	Stipulations that are related to the Instrument
Spread or Benchmark Curve Data Component	Attributes that are related to the curve or benchmark of an Instrument
Yield Data Component	Attributes that are related to the yield of an Instrument
Multi-leg Instrument Component	Attributes that are related to multi-leg Instrument

4 Market Conventions

4.1 Multi-leg Strategies – Bond vs. Future

- Bond v/s Future trades (i.e. basis trades) are a common strategy supported today by the major execution venues. There are 2 legs to this strategy:
  - The first leg is the bond leg which is identified by an ISIN or CUSIP for example
  - The second leg is the future leg where the product is identified using the LegSecurityType(609) field

- The market convention is that the buyer of the strategy buys the bond and sells the future and the seller of the strategy sells the bond and buys the future.

Strategy Side	Bond Side	Future Side
Buy	Buy	Sell
Sell	Sell	Buy

- The multi-leg strategy definition is specified in the instrument component block with the details of each individual product provided in the Instrument Leg component. The SecurityID(48) field contains the identifier of the strategy assigned by the Execution Venue. The LegSide(624) field is used to specify the buyer/seller of each instrument and the LegMaturityDate(611) specifies the maturity of the future.

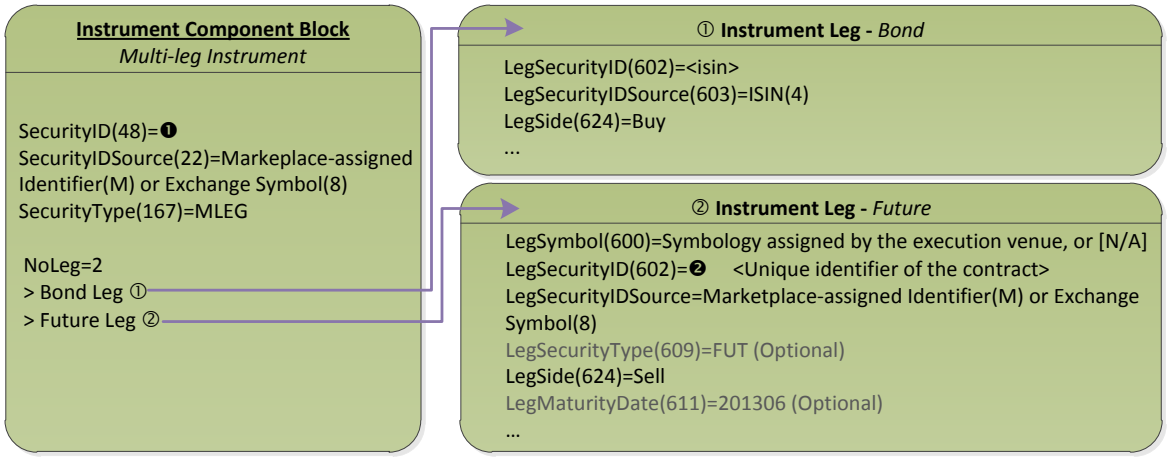


Figure 1: Example of Multi-leg Strategy - Bonds vs. Future

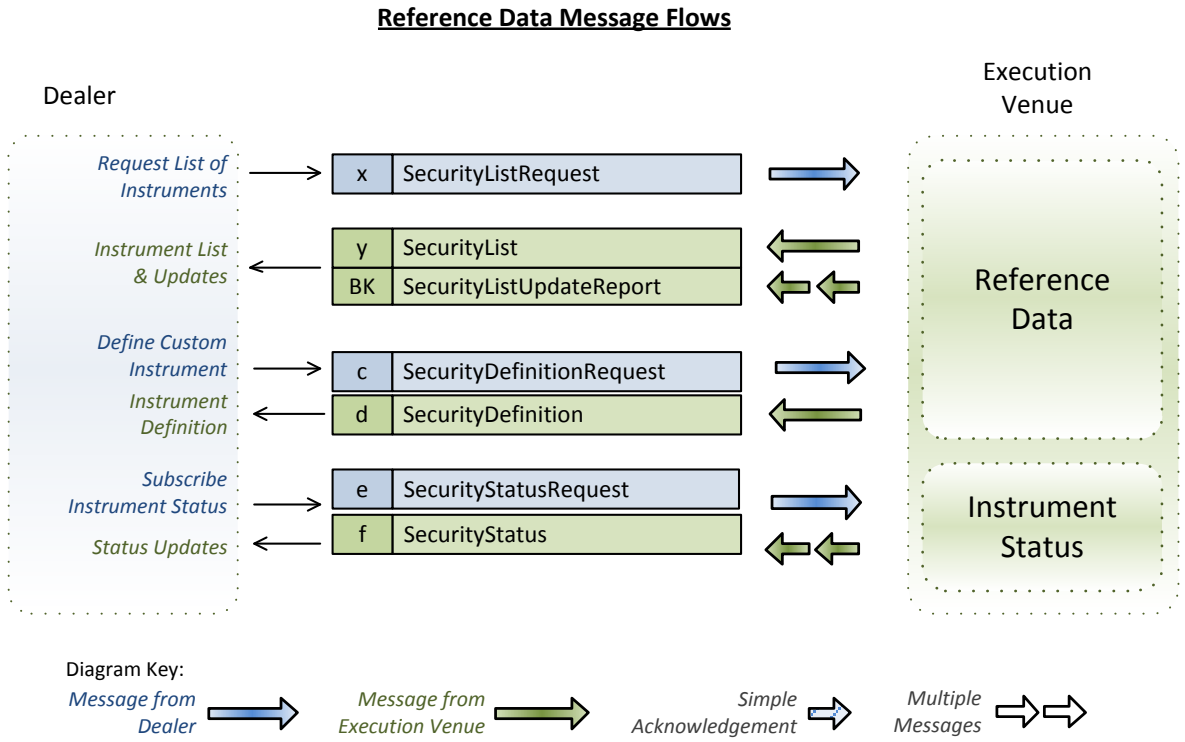
5 Pre-Trade - Reference Data

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)

5.1 Overview diagram

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



5.2 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

5.3 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) .

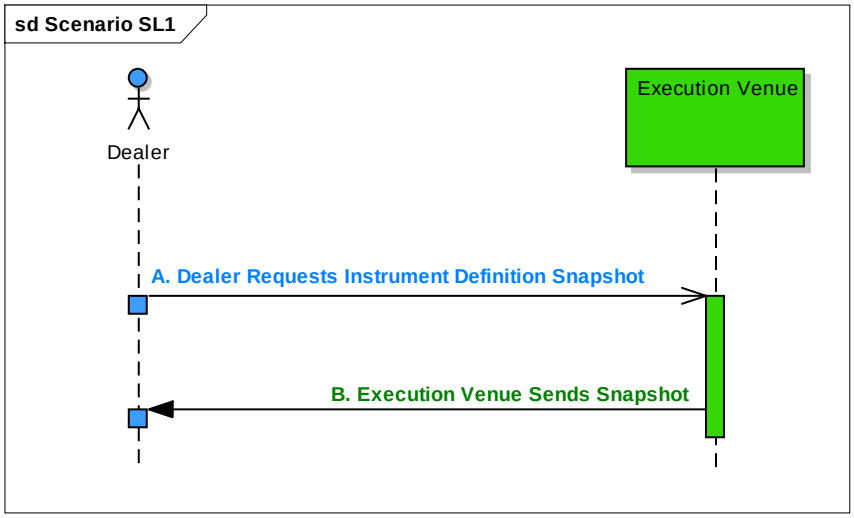


Figure 2: Scenario SL1 – Dealer Requests Snapshot of Instruments, Execution Venue Returns Snapshot

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> *Additional instrument attributes*	

Table 1: 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.

5.4 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.

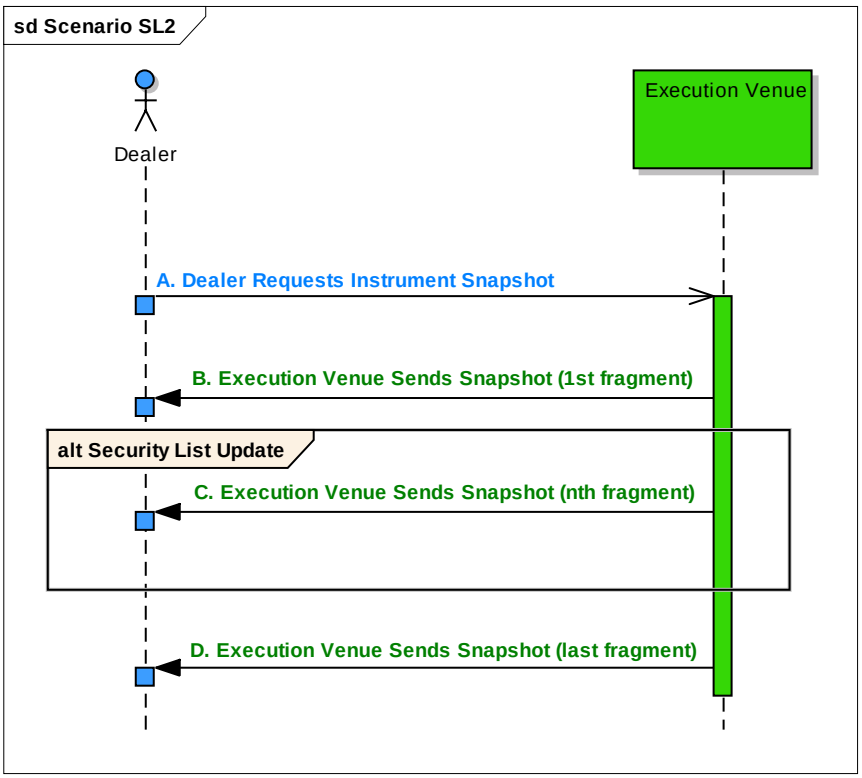


Figure 3: Scenario SL2 – Dealer Requests Snapshot of Instruments, Execution Venue Returns Snapshot (fragmented)

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 fragemented 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> *Additional instrument attributes*	
(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> *Additional instrument attributes*	
(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> *Additional instrument attributes*	

Table 2: Scenario SL2 – Dealer Requests Snapshot of Instruments, Execution Venue Returns Snapshot (fragmented)

5.5 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.

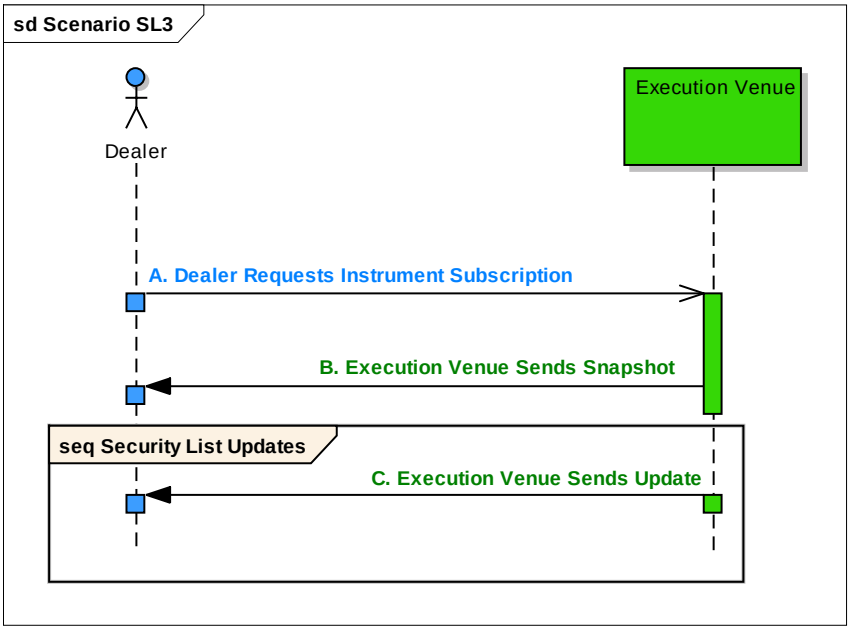


Figure 4: Scenario SL3 – Dealer Subscribes to Instruments Reference Data, Execution Venue Returns Snapshot and Updates

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> *Additional instrument attributes*	
(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> *Additional instrument attributes*	

Table 3: Scenario SL3 – Dealer Subscribes to Instruments Reference Data, Execution Venue Returns Snapshot and Updates

5.6 Scenario SL4 – Dealer Cancels Instrument Reference Data Subscription

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

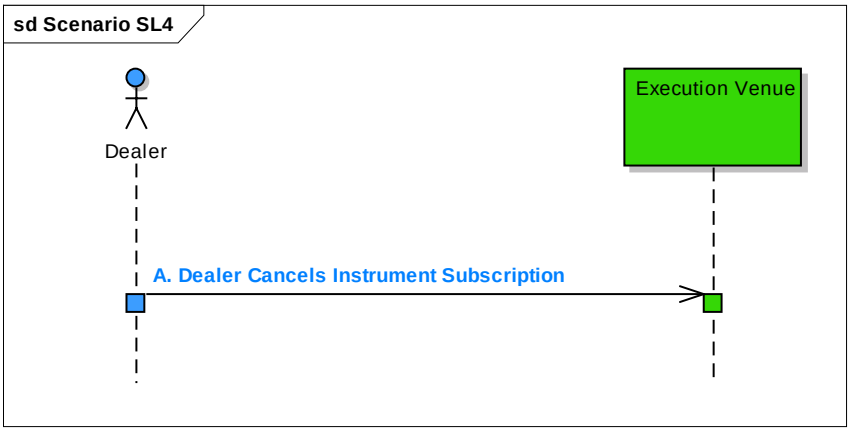


Figure 5: Scenario SL4 – Dealer Cancels Instrument Reference Data Subscription

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 4: Scenario SL4 – Dealer Cancels Instrument Reference Data Subscription

5.7 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.

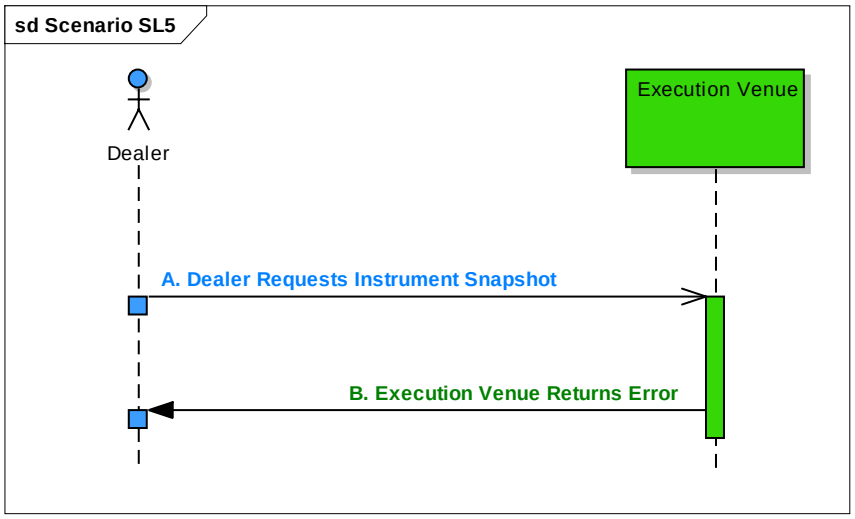


Figure 6: Scenario SL5 – Dealer Requests Snapshot or Subscription of Instruments Reference Data, Execution Venue Returns Error

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 5: 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.

5.8 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.

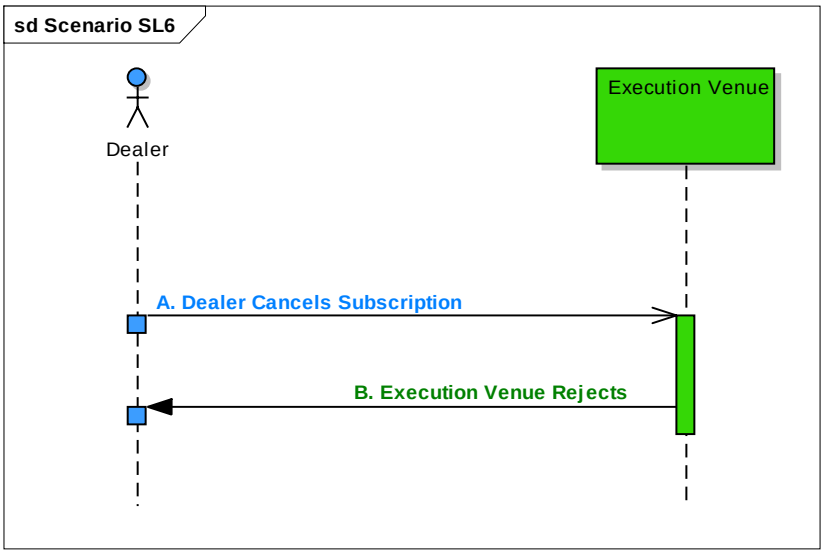


Figure 7: Scenario SL6 – Dealer Cancels Instrument Reference Data Subscription, Execution Venue Rejects

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

5.9 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.

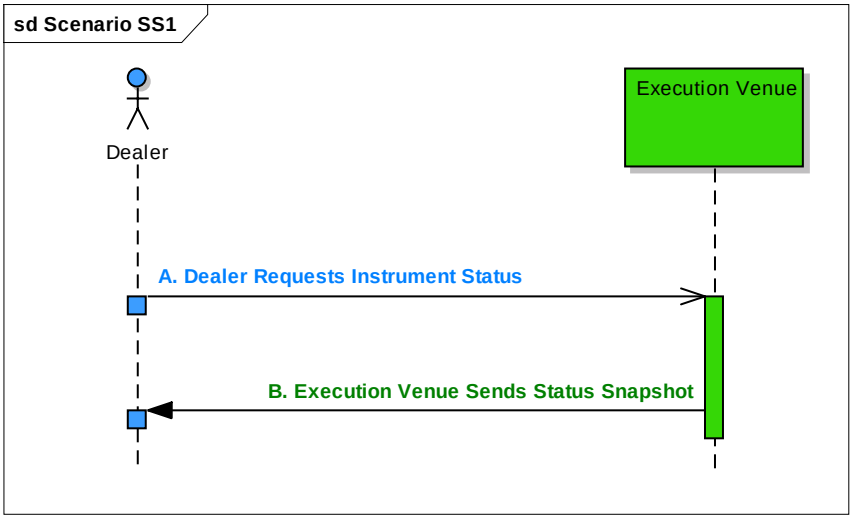


Figure 8: Scenario SS1 – Dealer Requests Instrument Status, Execution Venue Returns Status

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

5.10 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

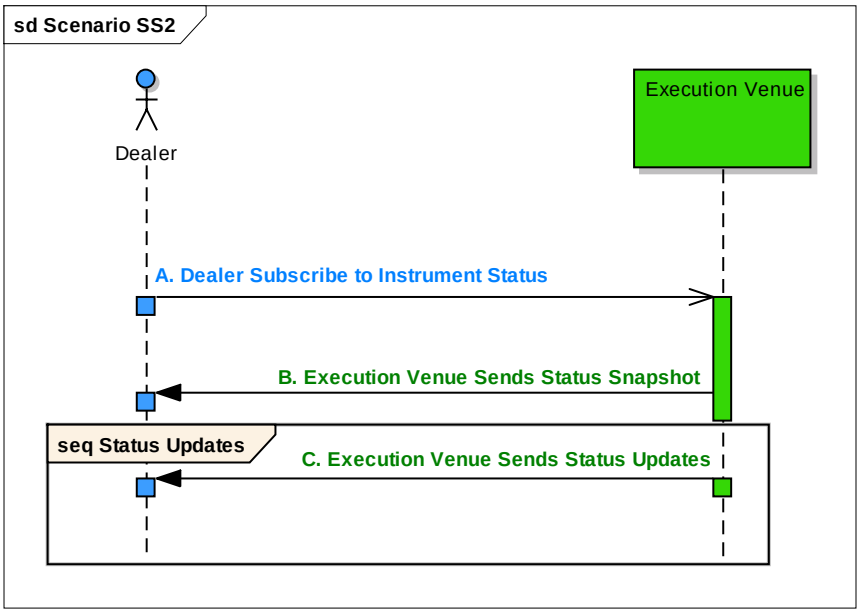


Figure 9: Scenario SS2 – Dealer Subscribes to Instrument Status, Execution Venue Returns Snapshot and Updates

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

5.11 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

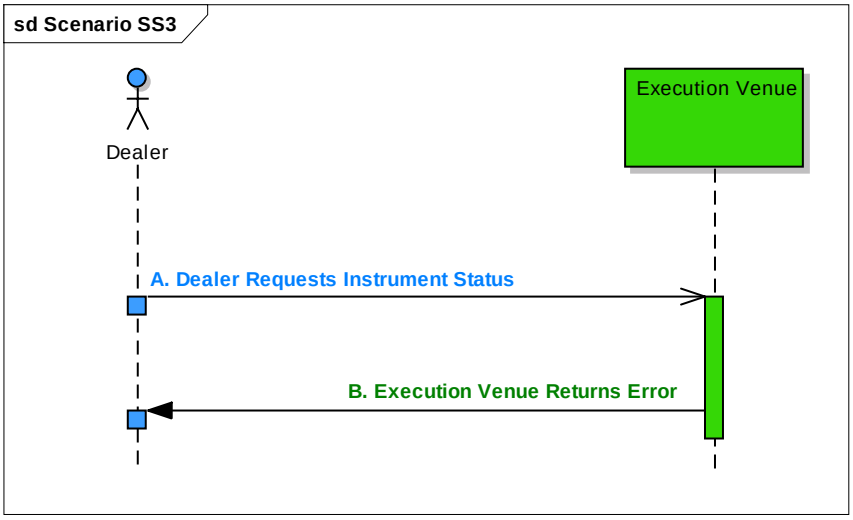


Figure 10: Scenario SS3 – Dealer Requests Instrument Status, Execution Venue Returns Error

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

5.12 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.

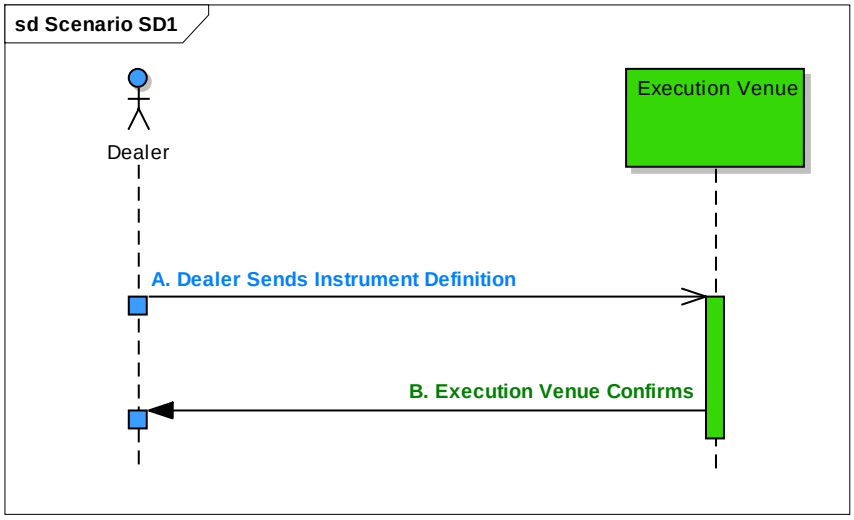


Figure 11: Scenario SD1 – Dealer Sends Instrument Definition Request, Execution Venue Confirms

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

5.13 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.

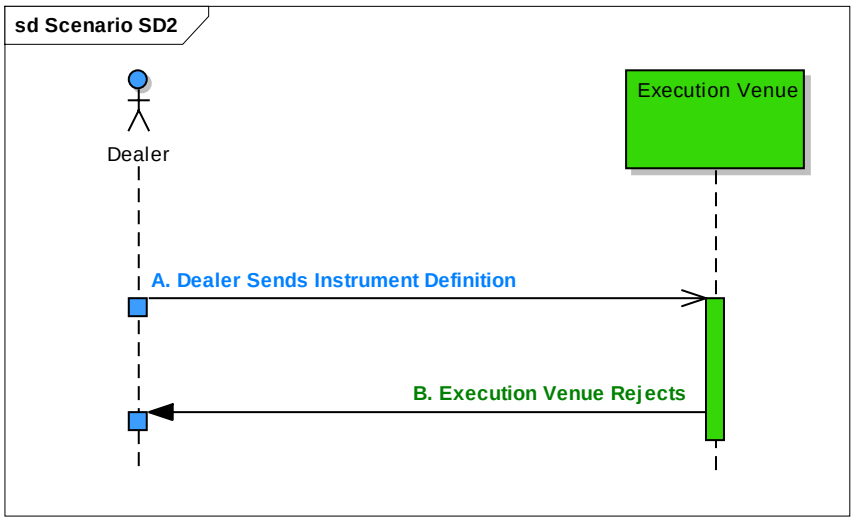


Figure 12: Scenario SD2 – Dealer Sends Instrument Definition Request, Execution Venue Rejects

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

5.14 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 sents 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 logon, where the subscription can be an 'offline' request (e.g. via profile setup)

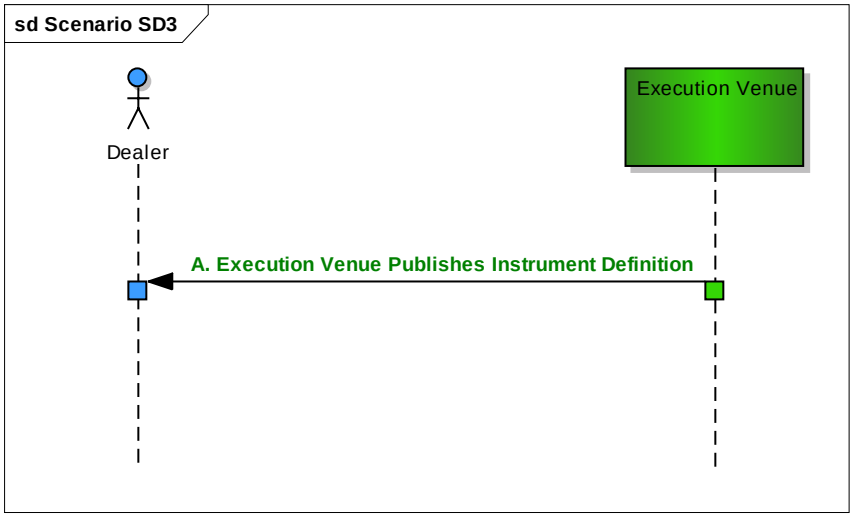


Figure 13: Scenario SD3 – Execution Venue Publishes Instrument Definition

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 12: Scenario SD3 – Execution Venue Publishes Instrument Definition

6 Pre-Trade – Price Contribution

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

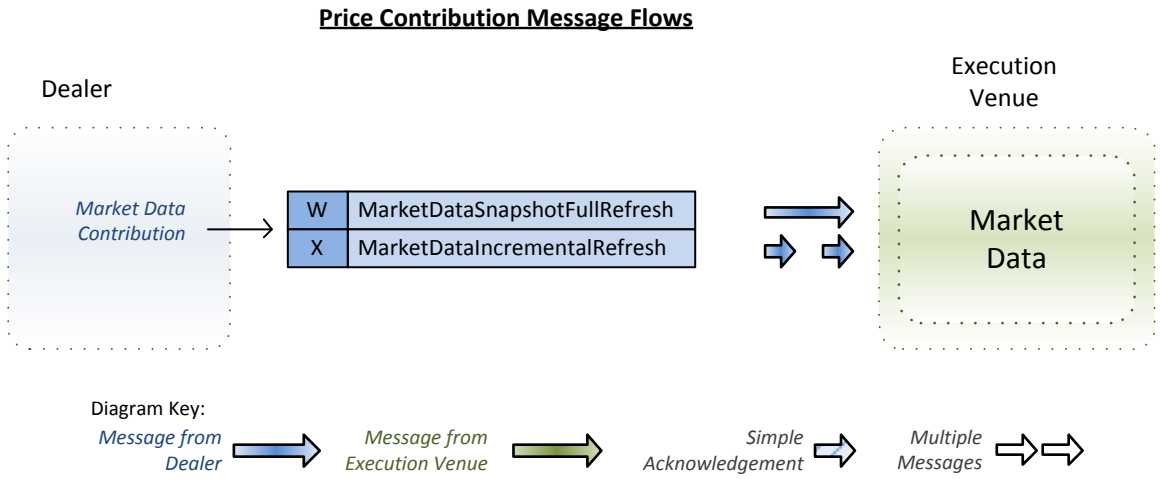
The scenarios in this section are common in market making. Prices may be contributed to either:

- Quote Negitiation 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).

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.

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

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

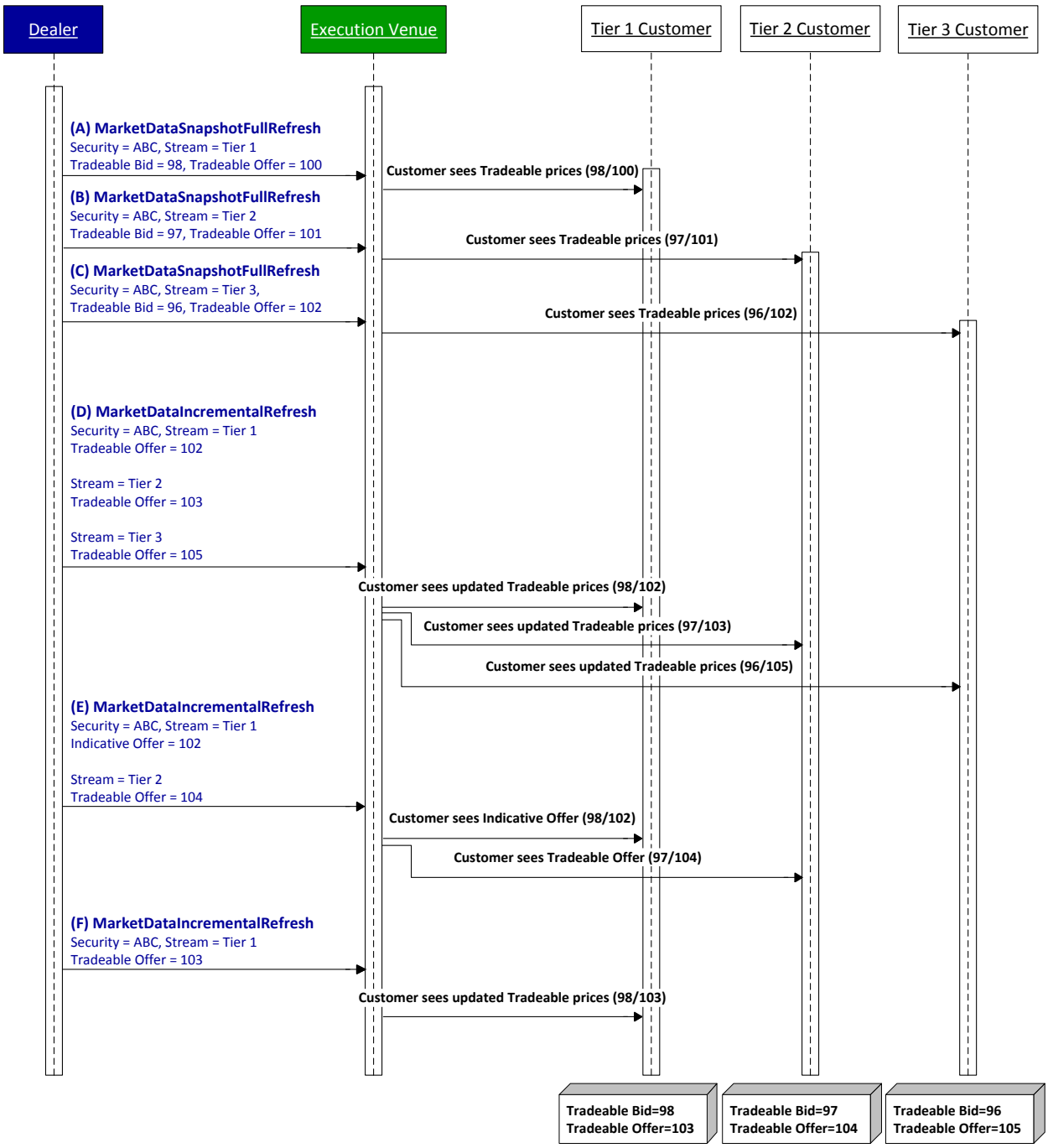


Figure 14: Scenario PP1 – Dealer Sends Different Prices for the Same Instrument to Multiple Customers – Dealer Amends Trade Information (Price & type)

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)=❻ 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)

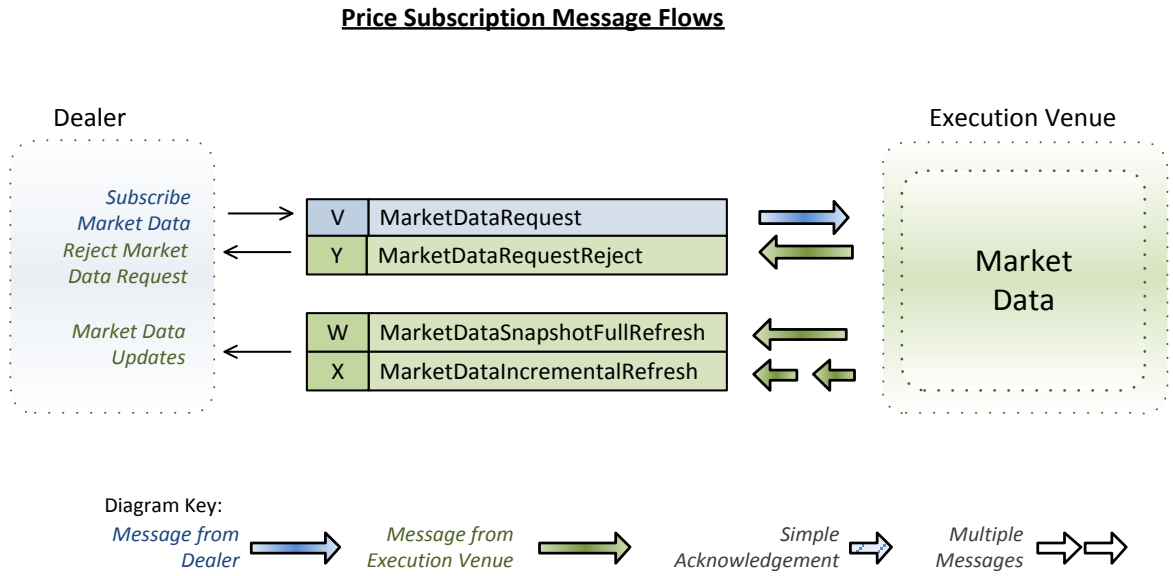
Table 13: Scenario PP1 – Dealer Sends Different Prices for the Same Instrument to Multiple Customers – Dealer Amends Trade Information (Price & type)

7 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.

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

7.3 Scenario MDS1 – Dealer Requests/Receives Market Data

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

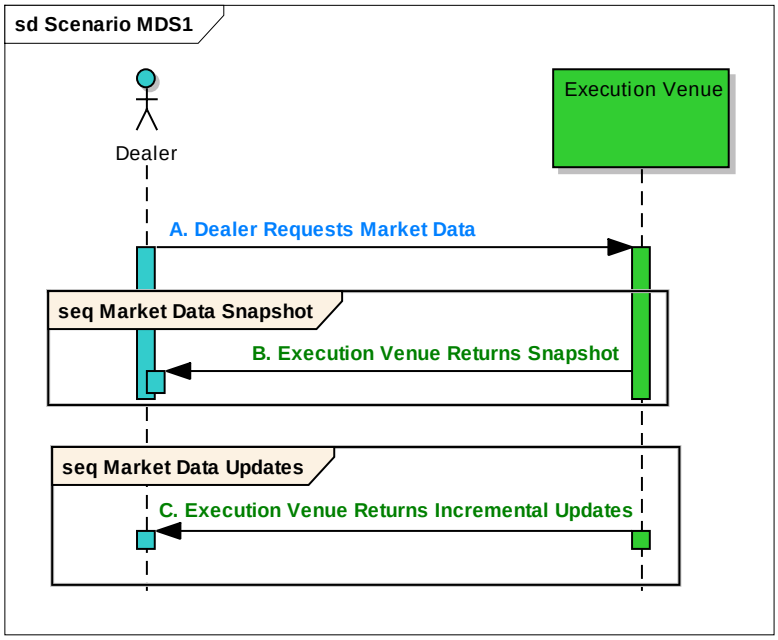


Figure 15: Scenario MDS1 – Dealer Requests/Receives Market Data

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 14: 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.

7.4 Scenario MDS2 – Dealer Unsubscribes Market Data

The Dealer unsubscribes from Market Data.

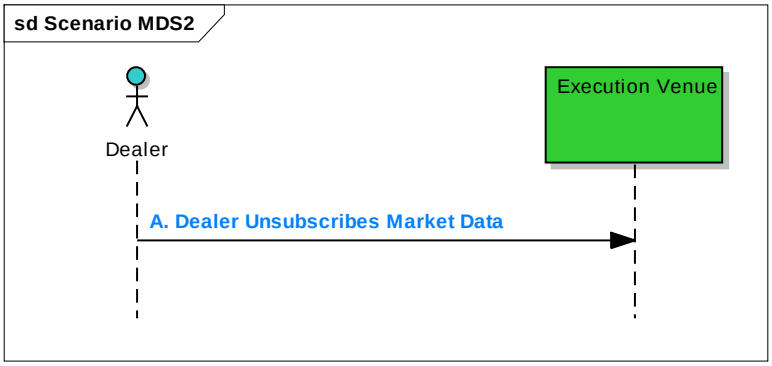


Figure 16: Scenario MDS2 – Dealer Unsubscribes Market Data

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 15: Scenario MDS2 – Dealer Unsubscribes Market Data

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

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

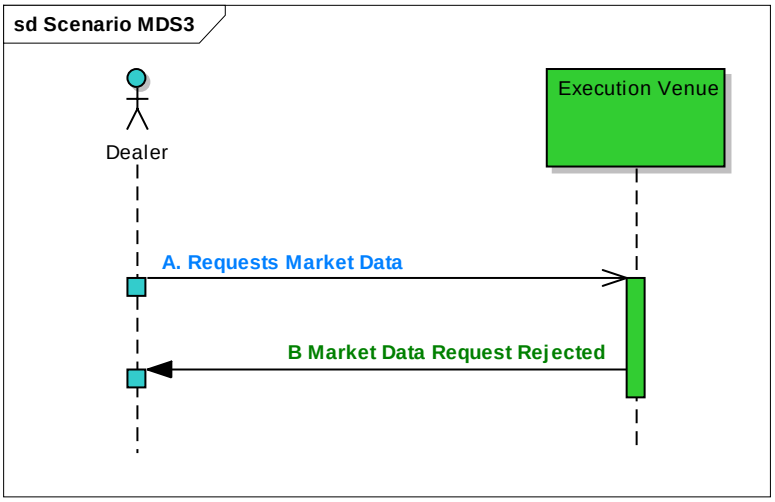


Figure 17: Scenario MDS3 – Dealer Requests Market Data, Execution Venue Rejects

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 16: 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.

8 Pre-Trade – Quote Contribution

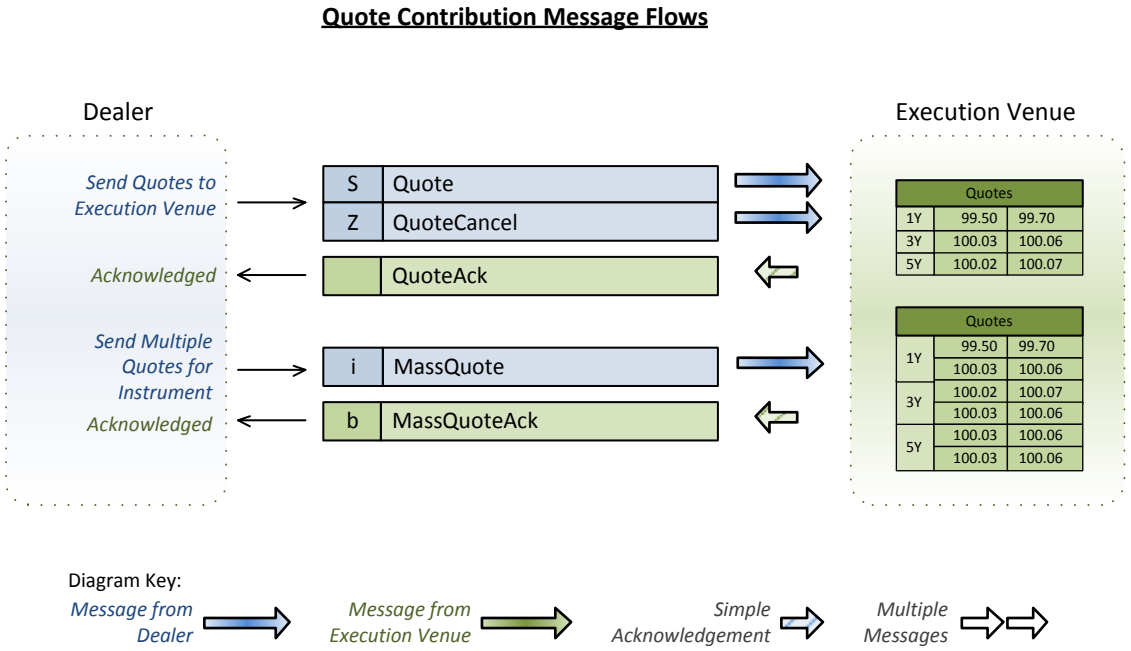
This section describes messages relevant for contributing quotes to an Execution Venue.

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.

8.1 Overview diagram

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



8.2 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

8.3 Scenario PC1 – Dealer Sends Quote to Market

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

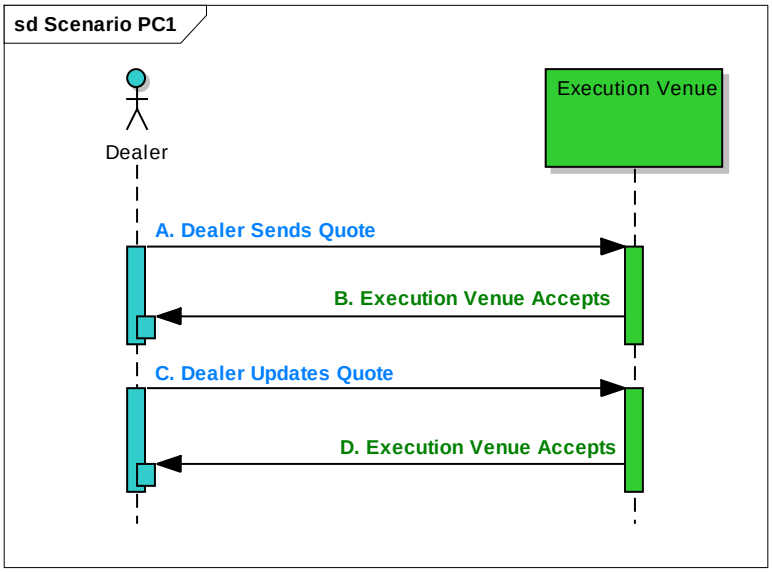


Figure 18: Scenario PC1 – Dealer Sends Quote to Market

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)	Execution Venue
(B) Execution Venue Accepts (optional : According to the value of QuoteResponseLevel (301))		←	**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)	
(D) Execution Venue Accepts (optional : According to the value of QuoteResponseLevel (301))		←	**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 17: Scenario PC1 – Dealer Sends Quote to Market

8.4 Scenario PC2 – Dealer Cancels Quote on Market

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

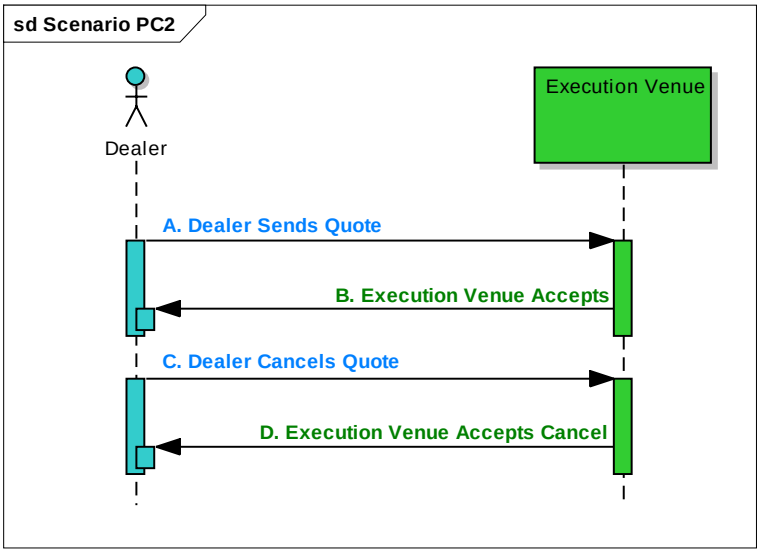


Figure 19: Scenario PC2 – Dealer Cancels Quote on Market

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)	Execution Venue
(B) Execution Venue Accepts (optional : According to the value of QuoteResponseLevel (301))		←	**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)	
(D) Execution Venue Accepts Cancel (optional : According to the value of QuoteResponseLevel (301))		←	**CW - QuoteAck** QuoteID(117)= ❶ QuoteMsgID(1166)=❸ **QuoteAckStatus (1865)**= Accepted(1) <Accepted the cancellation>	

Table 18: Scenario PC2 – Dealer Cancels Quote on Market

8.5. 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.

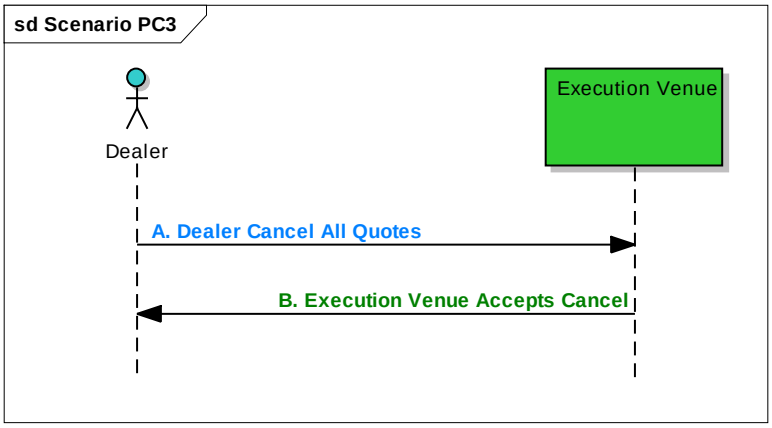


Figure 20: Scenario PC3 – Dealer Cancels All Quotes

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)	Execution Venue
(B) Execution Venue Accepts cancel (optional : According to the value of QuoteResponseLevel (301))		←	**b – MassQuoteAck** QuoteID(117)= ❶ QuoteStatus(297)= Accepted(0) <MassQuote Cancellation accepted> NoQuoteSets(296)=required e.g. (1) >TotNoCxldQuotes(1168) >TotNoAccQuote(1169) >TotNoRejQuotes(1170) >NoQuoteEntries(295)=required >>QuoteEntryStatus(1167) <The current status held by the Execution Venue for that quote> >>QuoteEntryRejectReason(368)	

Table 19: 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*.

8.6 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.

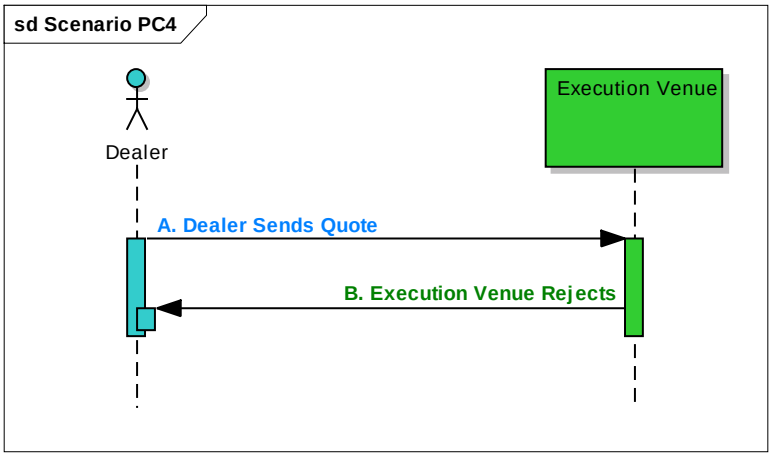


Figure 21: Scenario PC4 – Dealer Sends Quote to Market, Execution Venue Rejects

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)	Execution Venue
(B) Execution Venue Rejects (optional : According to the value of QuoteResponseLevel (301))		←	**CW - QuoteAck** QuoteID(117)= ❶ QuoteMsgID(1166)= ❷ **QuoteAckStatus (1865)**= Rejected(2) QuoteRejectReason(300)= required e.g. Unknown Symbol - security(1)	

Table 20: Scenario PC4 – Dealer Sends Quote to Market, Execution Venue Rejects

8.7 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.

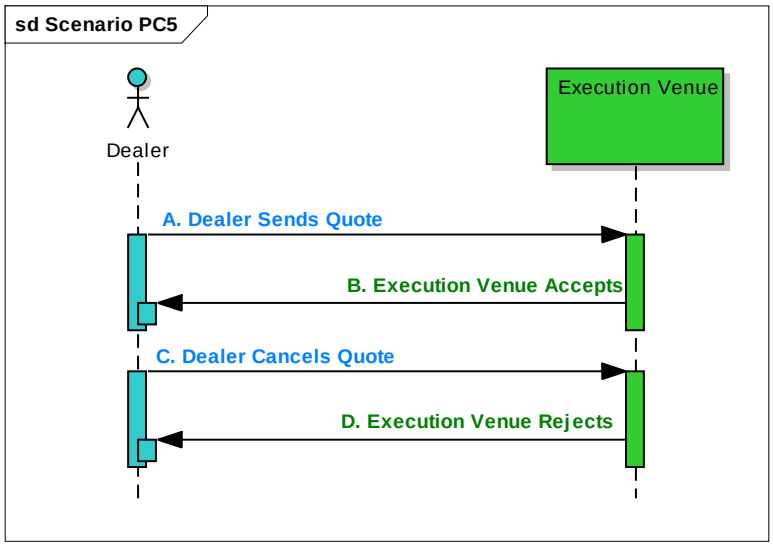


Figure 22: Scenario PC5 – Dealer Cancels Quote, Execution Venue Rejects

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)	Execution Venue
(B) Execution Venue Accepts (optional : According to the value of QuoteResponseLevel (301))		←	**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)	
(D) Execution Venue Rejects (optional : According to the value of QuoteResponseLevel (301))		←	**CW - QuoteAck** QuoteID(117)= ❶ QuoteMsgID(1166)= ❸ **QuoteAckStatus (1865)**= Rejected(2) <Cancellation rejected> QuoteRejectReason(300)=required e.g. Unknown Quote(5)	

Table 21: Scenario PC5 – Dealer Cancels Quote, Execution Venue Rejects

8.8 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.

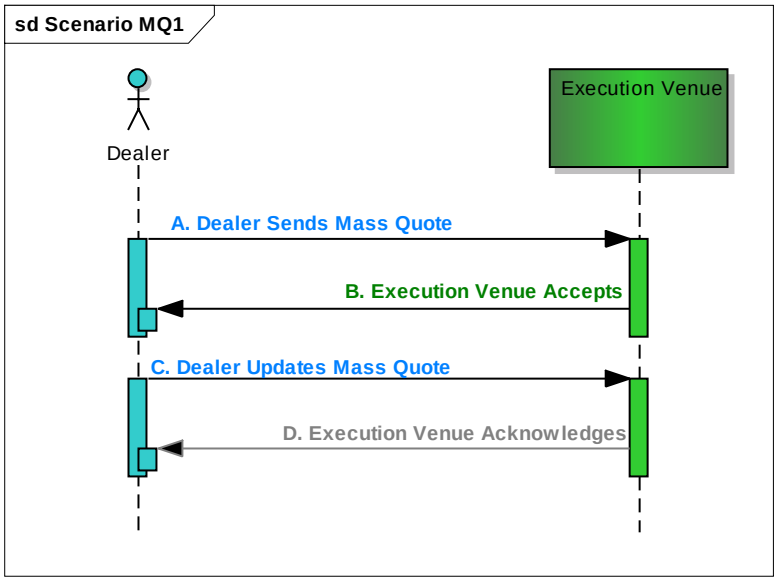


Figure 23: Scenario MQ1 – Dealer Sends Mass Quote to Execution Venue

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)	Execution Venue
(B) Execution Venue Accepts (optional : According to the value of QuoteResponseLevel (301))		←	**b– MassQuoteAck** QuoteID(117)= ❶ QuoteStatus(297)=Accepted(0) <MassQuote Accepted> NoQuoteSets(296)=<count> (1) > NoQuoteEntries(295)=<count> (1) *Response only for bad quotes e.g. if ❷ is bad:* >> QuoteEntryRejectReason(368)=<enum :required> >> QuoteEntryID(299)=❷ >>QuoteEntryStatus(1167)= Rejected(5)	

Model FIX 5.0			
(C) Dealer Updates Mass Quote		→	**i - MassQuote** QuoteID(117)= ❺ NoQuoteSets(296)= required e.g. (1) >NoQuoteEntries(295)= required e.g.(2) >> QuoteEntryID(299)= ❶ >> + quote attributes >> QuoteMsgID(1166)=❹ >> + quote attributes QuoteResponseLevel(301)
(D) Execution Venue Acknowledges (optional : According to the value of QuoteResponseLevel (301))		←	**b– MassQuoteAck** QuoteID(117)= ❺ 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 22: Scenario MQ1 – Dealer Sends Mass Quote to Execution Venue

8.9 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.

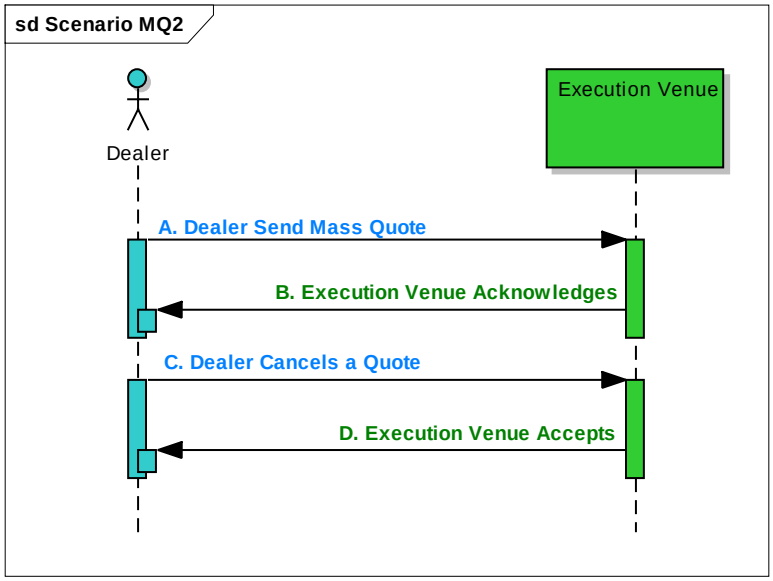


Figure 24: Scenario MQ2 – Dealer Cancels a Single Quote Sent Within a Mass Quote

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)	Execution Venue
(B) Execution Venue Acknowledges (optional : According to the value of QuoteResponseLevel (301))		←	**b– MassQuoteAck** QuoteID(117)= ❶ QuoteStatus(297)=Accepted(0) NoQuoteSets(296)= required > NoQuoteEntries(295)= required *Response only for bad quotes only, e.g.:* >> 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)
(D) Execution Venue Accepts (optional : According to the value of QuoteResponseLevel (301))		←	**b– MassQuoteAck** QuoteID(117)= ❶ QuoteStatus(297)=Accepted(0) NoQuoteSets(296)= 1 > NoQuoteEntries(295)= 1 >>QuoteEntryStatus (1167)= Accepted(0) >> QuoteEntryID(299)= ❶

Table 23: Scenario MQ2 – Dealer Cancels a Single Quote Sent Within a Mass Quote

8.10 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.

9 Pre-Trade - Indications

An Indication allows the Dealer to communicate to the Customer an interest to buy and / or sell a particular instrument. Historically this has been communicated via email or a venue's messaging system. The message is constructed from minimal details and is used by the Customer in support of their pre-trade liquidity discovery process.

Indications are recommended to be sent as either an IOI (35=6) message or a Quote (35=S) message with QuoteType (537) = Indicative (0). In a Quote (35=S) message, the use of QuoteType (537) = Indicative (0) shows that the Indications are not firm and that a "last look" would always be expected regardless of the setting of QuoteQualifier (695).

The Dealer may include one or more IOIQualifier (104) to an IOI (35=6) or QuoteQualifier (695) to a Quote message (35=S) in order to qualify the Indication. It is typically the case that the IOI (35=6) message is used for one-sided Axes and Inventories, whilst the Quote (35=S) message is used for two-sided Runs (where each side can be "axed") with the following qualifiers being used:

CommonIndication Type	IOIQualifier (104)	QuoteQualifier (695)
Axe	E = Axe F = Axe on Bid G = Axe on Offer	F = Axe on Bid G = Axe on Offer
Inventory	S = Inventory or Portfolio Shown	
Run		R = Ready To Trade

Note: Each IOI (35=6) message can support a one-sided Indication, whilst a Quote (35=S) message may support a two-sided Indication.

The Indication message can contain a Price (percent of par), a Yield and/or a Benchmark Spread value (and one, two or all of these values *could* be included on the message) – but if Benchmark Spread is included, then the message must also include a Benchmark Security ID (699).

The message may also contain one or more additional Spreads (eg: ASW Spread, Z-Spread, Discount Margin etc.). A Run has two sides and so the Indication may include additional Spread values for both sides (bid and offer).

Although the FIX messages are directed to a particular recipient institution - defined in the standard header - the Sell-Side has the ability to indicate to the Buy-Side which user(s) or desk(s) they think would have a particular interest in seeing the indication. This is defined in the RoutingGrp component (a repeating group) by placing a bilaterally agreed identifier in RoutingID (217) with the type defined in the RoutingType (216).

Instrument reference data (such as Sector, Rating etc) should be handled through the mechanism documented in Section 5: Pre-Trade Reference Data.

Recommendations regarding the identification of Instruments on the Indication message can be found in Section 3: Instrument Identifiers.

The following key tags are recommended for use with Indication messages using either the IOI (35=6) message or the Quote (35=S) message.

Business Attribute	IOI (35=6) tags	Quote (35=S) tags
Message ID Used in subsequent messages for traceability	IOIID(23)	QuoteID(117)
Instrument	SecurityID (48) Further details can be found in Section 3: Instrument Identifiers	
Side	Side(54)	*Not Applicable*
Quantity	IOIQty(27)	BidSize(134) / OfferSize(135)
Extended Quantity	IOIQualifier(104)=M(More Behind)	QuoteQualifier(695)=M(More Behind)
Currency	Currency(15)	
Commentary	Text(58)	
Indication is based on an Order	IOINaturalFlag(130): Y=Yes; N=No	*Not Applicable*
Buy-Side Interested Desk	RoutingID(217) RoutingType(216): 2=TargetList	
Buy-Side Interested User	RoutingID(217) RoutingType(216): **5=TargetPerson**	
Time of Indication	TransactTime(60)	
Levels		
Price Level	Price(44)	BidPx(132) / OfferPx(133)
Price Type	PriceType(423) For example: 1=Percentage (use for Price, par=100)	
Yield	Yield(236) Optional YieldType (235)	BidYield(632)/OfferYield(634) Optional YieldType (235)
Benchmark Spread	Spread (218)	**BidSpread(2533) / OfferSpread(2534)**
Benchmark Security	BenchmarkSecurityID (699) BenchmarkSecurityIDSource (761)	
Additional Spread Values		
ASW-Spread	**RelativeValue(2531)** **RelativeValueType(2530): 1=ASW Spread** **RelativeValueSide(2532): 1=Bid; 2=Mid; 3=Offer**	
OIS-Spread	**RelativeValue(2531)** **RelativeValueType(2530): 2=OIS-Spread** **RelativeValueSide(2532): 1=Bid; 2=Mid; 3=Offer**	
Z-Spread	**RelativeValue(2531)** **RelativeValueType(2530): 3=Z-Spread** **RelativeValueSide(2532): 1=Bid; 2=Mid; 3=Offer**	
Discount Margin	**RelativeValue(2531)** **RelativeValueType(2530): 4=Discount Margin** **RelativeValueSide(2532): 1=Bid; 2=Mid; 3=Offer**	
I-Spread	**RelativeValue(2531)** **RelativeValueType(2530): 5=I-Spread** **RelativeValueSide(2532): 1=Bid; 2=Mid; 3=Offer**	
OA-Spread	**RelativeValue(2531)**	

	RelativeValueType(2530): 6=OA-Spread **RelativeValueSide(2532): 1=Bid; 2=Mid; 3=Offer**
G-Spread	**RelativeValue(2531)** **RelativeValueType(2530): 7=G-Spread** **RelativeValueSide(2532): 1=Bid; 2=Mid; 3=Offer**
CDS Basis	**RelativeValue(2531)** **RelativeValueType(2530): 8=CDS Basis** **RelativeValueSide(2532): 1=Bid; 2=Mid; 3=Offer**
CDS Interpolated Basis	**RelativeValue(2531)** **RelativeValueType(2530): 9=CDS Interpolated Basis** **RelativeValueSide(2532): 1=Bid; 2=Mid; 3=Offer**

9.1 Message Flows Summary

The following scenarios in this section describe the communication between a Dealers a Customer. Whilst these workflows illustrate the typical direction of the messages, the definition of the messages could equally apply if they were sent from the Customer to the Dealer.

Scenario	Description
IND1	Dealer Sends an Axe or Inventory to a Customer
IND2	Dealer Sends a Run to a Customer
IND3	Dealer Replaces an Axe or Inventory
IND4	Dealer Cancels an Axe or Inventory
IND5	Dealer Replaces a Run
IND6	Dealer Cancels a Run
IND7	Dealer Sends an Axe or Inventory to a Customer, Axe or Inventory Rejected
IND8	Dealer Sends a Run to a Customer, Run Rejected

9.2 Scenario IND1: Dealer Sends an Axe or Inventory to a Customer

This scenario illustrates the workflow in which the Dealer sends an Axe or Inventory Indication message to a Customer. The Indication of Interest message (35=6) is to be used for one-sided Indications.
The type of Indication is denoted by the IOIQualifier(104):

Common Indication Type	IOIQualifier (104)
Axe	E = Axe F = Axe on Bid G = Axe on Offer
Inventory	S = Inventory or Portfolio Shown

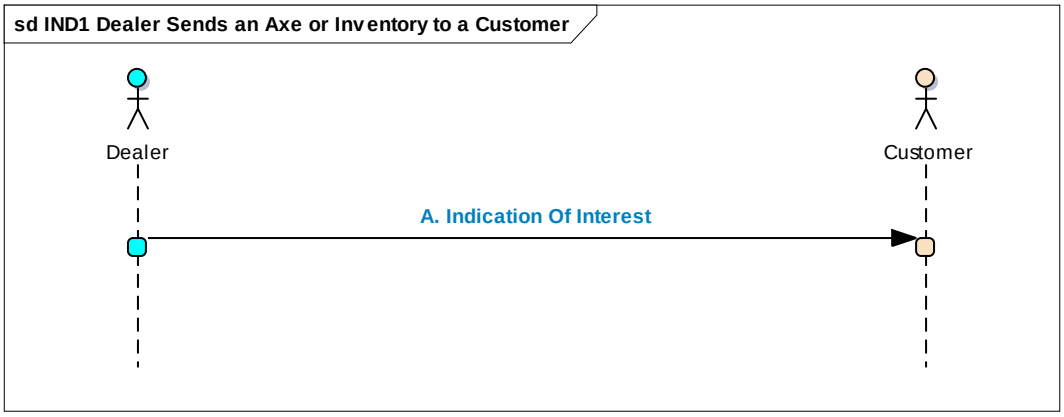


Figure 25: Scenario IND1: Dealer Sends an Axe or Inventory to a Customer

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 IOI	Dealer	→	**6 – IOI** IOIID(23)= ❶ IOITransType(28)=New(N) Instrument <required> Side(54) <required> Price(44) <Axe Indication price> IOIQty(27) <Axe Indication size> NoIOIQualifiers(199)=1 -> IOIQualifier(104)= *Axe(E) / Axe on Bid(F) / Axe on Offer (G) / Inventory or Portfolio Shown (S)*	Customer

Table 24: Scenario IND1: Dealer Sends an Axe or Inventory to a Customer

9.3 Scenario IND2: Dealer Sends a Run to a Customer

This scenario illustrates the workflow in which the Dealer sends a Run Indication message to a Customer. The Quote message (35=S) is to be used for two-sided Indications.
The type of Indication is denoted by the QuoteQualifier(695) with an "axed" side denoted by the appropriate QuoteQualifier(695).

Common Indication Type	QuoteQualifier (695)
Run	R = Ready To Trade
Axed Side	F = Axe on Bid G = Axe on Offer

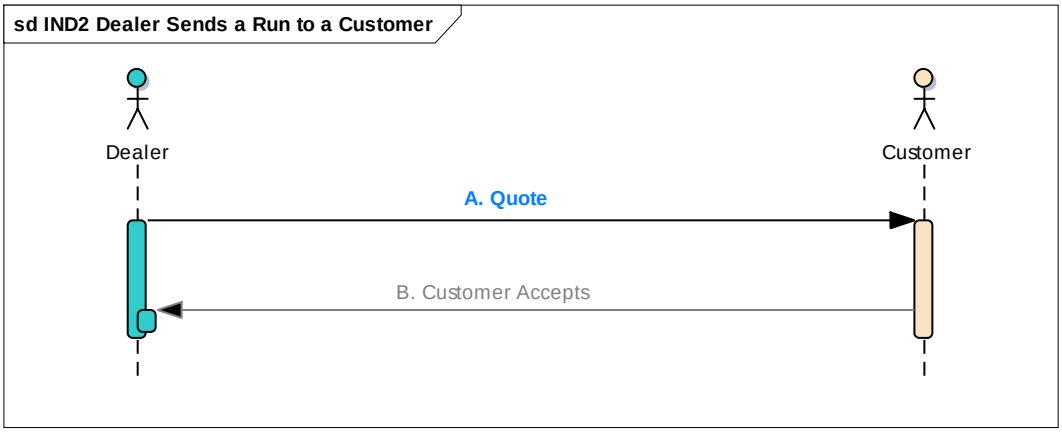


Figure 26: Scenario IND2: Dealer Sends a Run to a Customer

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 send Quote	Dealer	→	**S – Quote** QuoteID(117)= ❶ QuoteType(537)=Indicative(0) Instrument <required> QuoteRespType(301) BidPx(132) <Run bid price> OfferPx(133) <Run offer price> BidSize(134) <Run bid size> OfferSize(135) <Run offer size> NoQuoteQualifiers(735)=<number> -> QuoteQualifier(695)= *Ready To Trade (R)* -> QuoteQualifier(695)= *<optional> Axe on Bid(F) / Axe on Offer (G)*	Customer
(B) Execution Venue Accepts (optional : According to the value of QuoteResponseLevel (301))		←	**CW - QuoteAck** QuoteID(117)= ❶ QuoteMsgID(1166)=❷ **QuoteAckStatus (1865)**=Accepted(1)	

Table 25: Scenario IND2: Dealer Sends a Run to a Customer

9.4 Scenario IND3: Dealer Replaces an Axe or Inventory

This scenario illustrates the workflow in which the Dealer sends an Axe or Inventory Indication message to a Customer which is then replaced.

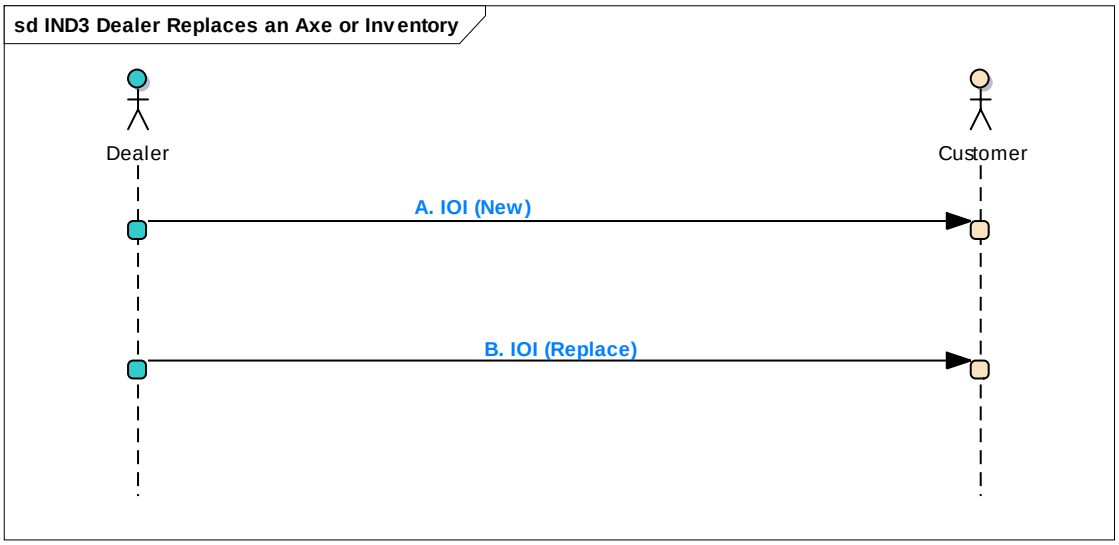


Figure 27: Scenario IND3: Dealer Replaces an Axe or Inventory

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 send IOI	Dealer	→	**6 – IOI** IOIID(23)= ❶ IOITransType(28)=New(N) Instrument <required> Side(54) <required> Price(44) <Axe Indication price> IOIQty(27) <Axe Indication size> NoIOIQualifiers(199)=1 -> IOIQualifier(104)= *Axe(E) / Axe on Bid(F) / Axe on Offer (G) / Inventory or Portfolio Shown (S)*	Customer
(B) Dealer Replaces IOI		→	**6 – IOI** IOIID(23)= ❷ IOIRefID(26)= ❶ IOITransType(28)=Replace(R) Instrument <required> Side(54) <required> Price(44) <Replaced Axe Indication price> IOIQty(27) <Replaced Axe Indication size> NoIOIQualifiers(199)=1 -> IOIQualifier(104)= *Axe(E) / Axe on Bid(F) / Axe on Offer (G) / Inventory or Portfolio Shown (S)*	

Table 26: Scenario IND3: Dealer Replaces an Axe or Inventory

9.5 Scenario IND4: Dealer Cancels an Axe or Inventory

This scenario illustrates the workflow in which the Dealer sends an Axe Indication message to a Customer which is then cancelled.

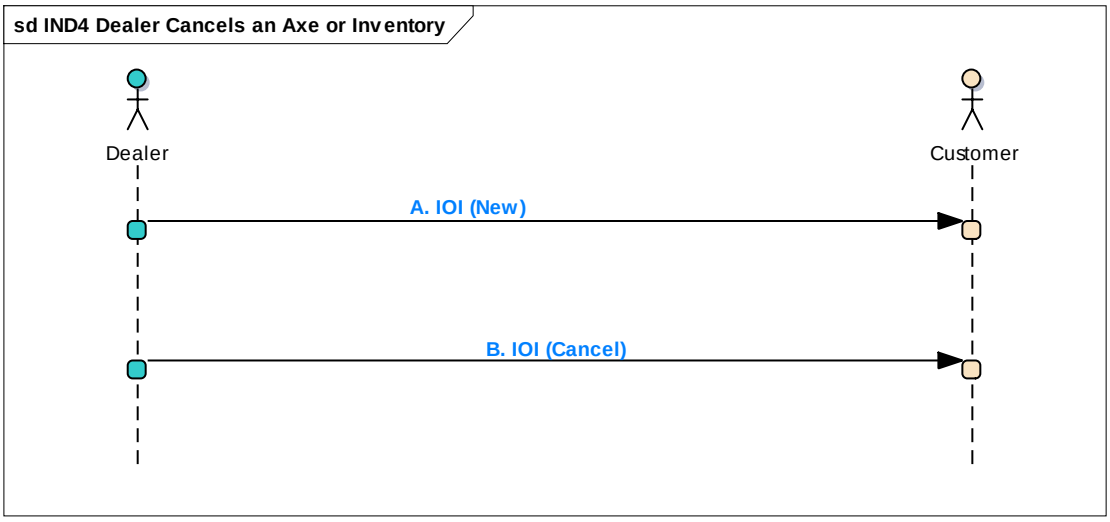


Figure 28: Scenario IND4: Dealer Cancels an Axe or Inventory

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 send IOI	Dealer	→	**6 – IOI** IOIID(23)= ❶ IOITransType(28)=New(N) Instrument <required> Side(54) <required> Price(44) <Axe Indication price> IOIQty(27) <Axe Indication size> NoIOIQualifiers(199)=1 -> IOIQualifier(104)= *Axe(E) / Axe on Bid(F) / Axe on Offer (G) / Inventory or Portfolio Shown (S)*	Customer
(B) Dealer Cancels IOI		→	**6 – IOI** IOIID(23)= ❷ IOIRefID(26)= ❶ IOITransType(28)=Cancel(C) Side(54) <required> IOIQty(27) <Cancelled Axe Indication size>	

Table 27: Scenario IND4: Dealer Cancels an Axe or Inventory

9.6 Scenario IND5: Dealer Replaces a Run

This scenario is for the Dealer to replace a Run sent with the Quote message (35=S).

Refer to scenario PC1 for the message flow.

9.7 Scenario IND6: Dealer Cancels a Run

This scenario is for the Dealer to cancel a Run sent with the Quote message (35=S).

Refer to scenario PC2 for the message flow.

9.8 Scenario IND7: Dealer Sends an Axe or Inventory to a Customer, Axe or Inventory Rejected

This scenario occurs when the Dealer sends the Customer an Axe or Inventory which is to be rejected with an explicit rejection.

The BusinessMessageReject (MsgTyp=j) message is used to reject an IOI (35=6) message at an application-level which fulfils session-level rules and cannot be rejected via any other means. Examples of reasons for rejecting an Indication in this way may include: an unknown Instrument, the receiving application not be available or that a conditionally required field is unavailable. For more details on this workflow, please refer to Section 2.2 of this document.

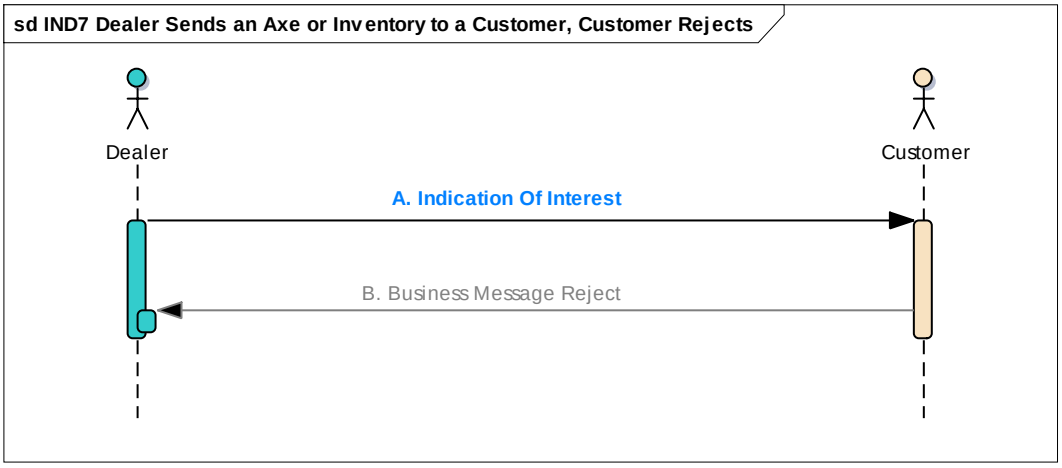


Figure 29: Scenario IND7: Dealer Sends an Axe or Inventory to a Customer, Indication Rejected

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 IOI	Dealer	→	**6 – IOI** IOIID(23)= ❶ MsgSeqNum(34)= ❷ IOITransType(28)=New(N) Instrument <required> Side(54) <required> Price(44) <Axe Indication price> IOIQty(27) <Axe Indication size> NoIOIQualifiers(199)=1 -> IOIQualifier(104)= *Axe(E) / Axe on Bid(F) / Axe on Offer (G) / Inventory or Portfolio Shown (S)*	Customer
(B) Customer Rejects IOI (optional)		←	**j – BusinessMessageReject** RefSeqNum(45)= ❷ RefMsgType(372)= 6 BusinessRejectRefID=❶ BusinessRejectReason=<required>	

Table 28: Scenario IND7: Dealer Sends an Axe or Inventory to a Customer, Indication Rejected

9.9 Scenario IND8: Dealer Sends a Run to a Customer, Run Rejected

This scenario is for when the Dealer sends the Customer a Run using the Quote (35=S) message which is then rejected.

Refer to scenario PC4 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
6	Indication of Interest

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 Instrument Component

Cash Bonds 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.	
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.	
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.	
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	**CPProgram**	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 29: Instrument Attributes

10.2 Extended Instrument Component

Extended Instrument Component				
Insert here the set of "InstrumentExtension" fields defined in "Common Components of Application Messages"				
Tag	**FieldName**	**Req'd**	**Description**	**Comment**
668	**DeliveryForm**	N	Identifies the form of delivery.	
	AttrbGrp	N	Number of repeating InstrAttrib group entries.	
->870	**NoInstrAttrib**	N	Number of repeating InstrAttribType entries.	
->->871	**InstrAttribType**	N	Code to represent the type of instrument attribute	See below
->->872	**InstrAttribValue**	N	Attribute value appropriate to the InstrAttribType (87) field.	

Table 30: Extended Instrument Attributes

10.2.1 FIX Tag InstrAttribType(871) values

Value	Description
1	Flat (securities pay interest on a current basis but are traded without interest)
2	Zero coupon
3	Interest bearing (for Euro commercial paper when not issued at discount)
4	No periodic payments
5	Variable rate
6	Less fee for put
7	Stepped coupon
8	Coupon period (if not semi-annual). Supply redemption date in the InstrAttribValue (872) field.
9	When [and if] issued
10	Original issue discount
11	Callable, puttable
12	Escrowed to Maturity
13	Escrowed to redemption date - callable. Supply redemption date in the InstrAttribValue (872) field
14	Pre-refunded
15	In default
16	Unrated
17	Taxable
18	Indexed
19	Subject To Alternative Minimum Tax
20	Original issue discount price. Supply price in the InstrAttribValue (872) field
21	Callable below maturity value
22	Callable without notice by mail to holder unless registered
23	Price tick rules for security.
24	Trade type eligibility details for security.
25	Instrument Denominator
26	Instrument Numerator
27	Instrument Price Precision
28	Instrument Strike Price
29	Tradeable Indicator
99	Text. Supply the text of the attribute or disclaimer in the InstrAttribValue (872) field.

Table 31: FIX Tag InstrAttribType(871) values

10.3 Undelying Instrument Component

Underlying Instrument Component				
Tag	**FieldName**	**Req'd**	**Description**	**Comment**
711	**NoUnderlyings**	N	Number of underlyings	
->	**UnderlyingInstrument**	N	Must be provided if Number of underlyings > 0	
->->311	**UnderlyingSymbol**	N	Underlying security's Symbol. See Symbol (55) field for description	
->->312	**UnderlyingSymbolSfx**	N	Underlying security's SymbolSfx. See SymbolSfx (65) field for description	
->->309	**UnderlyingSecurityID**	N	Underlying security's SecurityID. See SecurityID (48) field for description	
->->305	**UnderlyingSecurityIDSource**	N	Underlying security's SecurityIDSource. Valid values: see SecurityIDSource (22) field	
->->	**UndSecAltIDGrp**	N		
->->462	**UnderlyingProduct**	N	Underlying security's Product. Valid values: see Product(460) field	
->->310	**UnderlyingSecurityType**	N	Underlying security's SecurityType. Valid values: see SecurityType (167) field (see below for details concerning this fields use in conjunction with SecurityType=REPO) The following applies when used in conjunction with SecurityType=REPO Represents the general or specific type of security that underlies a financing agreement Valid values for SecurityType=REPO: If bonds of a particular issuer or country are wanted in an Order or are in the basket of an Execution and the SecurityType is not granular enough, include the UnderlyingIssuer (306), UnderlyingCountryOfIssue (592), UnderlyingProgram, UnderlyingRegType and/or < UnderlyingStipulations > block e.g.:	
->->763	**UnderlyingSecuritySubType**	N	Underlying security's SecuritySubType. See SecuritySubType (762) field for description	
->->313	**UnderlyingMaturityMonthYear**	N	Underlying security's MaturityMonthYear. Can be used with standardized	

			derivatives vs. the UnderlyingMaturityDate (542) field. See MaturityMonthYear (200) field for description	
->->542	**UnderlyingMaturityDate**	N	Underlying security's maturity date. See MaturityDate (541) field for description	
->->1213	**UnderlyingMaturityTime**	N	Time of security's maturity expressed in local time with offset to UTC specified	
->->241	**UnderlyingCouponPaymentDate**	N	Underlying security's CouponPaymentDate. See CouponPaymentDate (224) field for description (Note tag # was reserved in FIX 4.1, added in FIX 4.3) (prior to FIX 4.4 field was of type UTCDate)	
->->1453	**UnderlyingRestructuringType**	N	See RestructuringType(1449)	
->->1454	**UnderlyingSeniority**	N	See Seniority(1450)	
->->1455	**UnderlyingNotionalPercentageOutstanding**	N	See NotionalPercentageOutstanding(1451)	
->->1456	**UnderlyingOriginalNotionalPercentageOutstanding**	N	See OriginalNotionalPercentageOutstanding(1452)	
->->1459	**UnderlyingAttachmentPoint**	N	See AttachmentPoint(1457).	
->->1460	**UnderlyingDetachmentPoint**	N	See DetachmentPoint(1458).	
->->242	**UnderlyingIssueDate**	N	Underlying security's IssueDate. See IssueDate (225) field for description (Note tag # was reserved in FIX 4.1, added in FIX 4.3) (prior to FIX 4.4 field was of type UTCDate)	
->->246	**UnderlyingFactor**	N	Underlying security's Factor. See Factor (228) field for description (Note tag # was reserved in FIX 4.1, added in FIX 4.3)	
->->256	**UnderlyingCreditRating**	N	Underlying security's CreditRating. See CreditRating (255) field for description (Note tag # was reserved in FIX 4.1, added in FIX 4.3)	
->->595	**UnderlyingInstrRegistry**	N	Underlying security's InstrRegistry. See InstrRegistry (543) field for description	
->->592	**UnderlyingCountryOfIssue**	N	Underlying security's CountryOfIssue. See CountryOfIssue (470) field for description	
->->593	**UnderlyingStateOrProvinceOfIssue**	N	Underlying security's StateOrProvinceOfIssue. See StateOrProvinceOfIssue (471) field for description	
->->594	**UnderlyingLocaleOfIssue**	N	Underlying security's LocaleOfIssue. See LocaleOfIssue (472) field for	

			description	
->->435	**UnderlyingCouponRate**	N	Underlying security's CouponRate. See CouponRate (223) field for description	
->->306	**UnderlyingIssuer**	N	Underlying security's Issuer. See Issuer (06) field for description	
->->362	**EncodedUnderlyingIssuerLen**	N	Byte length of encoded (non-ASCII characters) EncodedUnderlyingIssuer (363) field.	
->->363	**EncodedUnderlyingIssuer**	N	Encoded (non-ASCII characters) representation of the UnderlyingIssuer (306) field in the encoded format specified via the MessageEncoding (347) field. If used, the ASCII (English) representation should also be specified in the UnderlyingIssuer field.	
->->307	**UnderlyingSecurityDesc**	N	Description of the Underlying security. See SecurityDesc(107).	
->->364	**EncodedUnderlyingSecurityDescLen**	N	Byte length of encoded (non-ASCII characters) EncodedUnderlyingSecurityDesc (365) field.	
->->365	**EncodedUnderlyingSecurityDesc**	N	Encoded (non-ASCII characters) representation of the UnderlyingSecurityDesc (307) field in the encoded format specified via the MessageEncoding (347) field. If used, the ASCII (English) representation should also be specified in the UnderlyingSecurityeDesc field.	
->->877	**UnderlyingCPProgram**	N	The program under which the underlying commercial paper is issued	
->->878	**UnderlyingCPRegType**	N	The registration type of the underlying commercial paper issuance	
->->318	**UnderlyingCurrency**	N	Specific to the (not in)	
->->810	**UnderlyingPx**	N	Specific to the (not in) In a financing deal clean price (percent-of-par or per unit) of the underlying security or basket.	
->->882	**UnderlyingDirtyPrice**	N	Specific to the (not in) In a financing deal price (percent-of-par or per unit) of the underlying security or basket. "Dirty" means it includes accrued interest	
->->883	**UnderlyingEndPrice**	N	Specific to the (not in) In a financing deal price (percent-of-par or per unit) of the underlying security or basket at the end of	

			the agreement.	
->->884	**UnderlyingStartValue**	N	Specific to the (not in) Currency value attributed to this collateral at the start of the agreement	
->->885	**UnderlyingCurrentValue**	N	Specific to the (not in) Currency value currently attributed to this collateral	
->->886	**UnderlyingEndValue**	N	Specific to the (not in) Currency value attributed to this collateral at the end of the agreement	
->->	**UnderlyingStipulations**	N	Specific to the (not in) Insert here the contents of the Component Block	
->->1044	**UnderlyingAdjustedQuantity**	N	Specific to the (not in). For listed derivatives margin management, this is the number of shares adjusted for upcoming corporate action. Used only for securities which are optionable and are between ex-date and settlement date (4 days).	
->->1045	**UnderlyingFXRate**	N	Specific to the (not in). Foreign exchange rate used to compute UnderlyingCurrentValue (885) (or market value) from UnderlyingCurrency (318) to Currency (15).	
->->1046	**UnderlyingFXRateCalc**	N	Specific to the (not in). Specified whether UnderlyingFxRate (1045) should be multiplied or divided to derive UnderlyingCurrentValue (885).	
->->1038	**UnderlyingCapValue**	N	Maximum notional value for a capped financial instrument	
->->	**UndlyInstrumentParties**	N	The use of this component block is restricted to instrument definition only and is not permitted to contain transactional information. Only a specified subset of party roles will be supported within the InstrumentParty block.	

Table 32: Underlying Instrument Attributes

10.4 Stipulations Component

Stipulation Component				
The Stipulations component block is used in Fixed Income to provide additional information on a given security. This additional information is usually not considered static data information.				
Tag	**FieldName**	**Req'd**	**Description**	**Comment**
232	**NoStipulations**	N	Number of stipulation entries (Note tag # was reserved in FIX 4.1, added in FIX 4.3).	
->233	**StipulationType**	N	Required if NoStipulations >0	
->234	**StipulationValue**	N	For Fixed Income. Value of stipulation. The expression can be an absolute single value or a combination of values and logical operators: < value > value <= value >= value value value - value2 value OR value2 value AND value2 YES NO Bargain conditions recognized by the London Stock Exchange - to be used when StipulationType is "BGNCON". CD = Special cum Dividend XD = Special ex Dividend CC = Special cum Coupon XC = Special ex Coupon CB = Special cum Bonus XB = Special ex Bonus CR = Special cum Rights XR = Special ex Rights CP = Special cum Capital Repayments XP = Special ex Capital Repayments CS = Cash Settlement SP = Special Price TR = Report for European Equity Market Securities in accordance with Chapter 8 of the Rules. GD = Guaranteed Delivery Values for StipulationType = "PXSOURCE": BB GENERIC BB FAIRVALUE BROKERTEC ESPEED GOVPX HILLIARD FARBER ICAP TRADEWEB TULLETT LIBERTY If a particular side of the market is wanted append /BID /OFFER or /MID. plus appropriate combinations of the above and other expressions by mutual agreement of the counterparties. Examples: ">=60", ".25", "ORANGE OR CONTRACOSTA", etc. (Note tag # was reserved in FIX 4.1, added in FIX 4.3)	

Table 33: Underlying Instrument Attributes

10.4.1 FIX Tag StipulationType(233) values

Value	Description
AMT	Alternative Minimum Tax (Y/N)
AUTOREINV	Auto Reinvestment at <rate> or better
BANKQUAL	Bank qualified (Y/N)
BGNCON	Bargain conditions (see StipulationValue (234) for values)
COUPON	Coupon range
CURRENCY	ISO Currency Code
CUSTOMDATE	Custom start/end date
GEOG	Geographics and % range (ex. 234=CA 0-80 [minimum of 80% California assets])
HAIRCUT	Valuation Discount
INSURED	Insured (Y/N)
ISSUE	Year Or Year/Month of Issue (ex. 234=2002/09)
ISSUER	Issuer's ticker
ISSUESIZE	issue size range
LOOKBACK	Lookback Days
LOT	Explicit lot identifier
LOTVAR	Lot Variance (value in percent maximum over- or under-allocation allowed)
MAT	Maturity Year And Month
MATURITY	Maturity range
MAXSUBS	Maximum substitutions (Repo)
MINDNOM	Minimum denomination
MININCR	Minimum increment
MINQTY	Minimum quantity
PAYFREQ	Payment frequency, calendar
PIECES	Number Of Pieces
PMAX	Pools Maximum
PPL	Pools per Lot
PPM	Pools per Million
PPT	Pools per Trade
PRICE	Price Range
PRICEFREQ	Pricing frequency
PROD	Production Year
PROTECT	Call protection
PURPOSE	Purpose
PXSOURCE	Benchmark price source
RATING	Rating source and range
REDEMPTION	Type Of Redemption - values are: NonCallable, Prefunded, EscrowedToMaturity, Putable, Convertible
RESTRICTED	Restricted (Y/N)
SECTOR	Market Sector
SECTYPE	Security Type included or excluded
STRUCT	Structure
SUBSFREQ	Substitutions frequency (Repo)
SUBSLEFT	Substitutions left (Repo)
TEXT	Freeform Text
TRDVAR	Trade Variance (value in percent maximum over- or under-allocation allowed)
WAC	Weighted Average Coupon - value in percent (exact or range) plus "Gross" or "Net" of servicing spread (the default) (ex. 234=6.5-Net [minimum of 6.5% net of servicing fee])
WAL	Weighted Average Life Coupon - value in percent (exact or range)
WALA	Weighted Average Loan Age - value in months (exact or range)
WAM	Weighted Average Maturity - value in months (exact or range)

WHOLE	Whole Pool (Y/N)
YIELD	Yield Range
-- Other --	
AVFICO	Average FICO Score
AVSIZE	Average Loan Size
MAXBAL	Maximum Loan Balance
POOL	Pool Identifier
ROLLTYPE	Type of Roll trade
REFTRADE	reference to rolling or closing trade
REFPRIN	principal of rolling or closing trade
REFINT	interest of rolling or closing trade
AVAILQTY	Available offer quantity to be shown to the street
BROKERCREDIT	Broker's sales credit
INTERNALPX	Offer price to be shown to internal brokers
INTERNALQTY	Offer quantity to be shown to internal brokers
LEAVEQTY	The minimum residual offer quantity
MAXORDQTY	Maximum order size
ORDRINCR	Order quantity increment
PRIMARY	Primary or Secondary market indicator
SALESCREDITOVR	Broker sales credit override
TRADERCREDIT	Trader's credit
DISCOUNT	Discount Rate (when price is denominated in percent of par)
YTM	Yield to Maturity (when YieldType(235) and Yield(236) show a different yield)
-- Prepayment Speeds --	
ABS	Absolute Prepayment Speed
CPP	Constant Prepayment Penalty
CPR	Constant Prepayment Rate
CPY	Constant Prepayment Yield
HEP	final CPR of Home Equity Prepayment Curve
MHP	Percent of Manufactured Housing Prepayment Curve
MPR	Monthly Prepayment Rate
PPC	Percent of Prospectus Prepayment Curve
PSA	Percent of BMA Prepayment Curve
SMM	Single Monthly Mortality

Table 34: FIX Tag StipulationType(233) values

10.5 Spread or Benchmark Curve Data Component

Spread or Benchmark Curve Component				
The SpreadOrBenchmarkCurveData component block is primarily used for Fixed Income to convey spread to a benchmark security or curve.				
Tag	**FieldName**	**Req'd**	**Description**	**Comment**
218	**Spread**	N	For Fixed Income	
220	**BenchmarkCurveCurrency**	N	Identifies currency used for benchmark curve. See "Appendix 6-A: Valid Currency Codes" for information on obtaining valid values. (Note tag # was reserved in FIX 4.1, added in FIX 4.3)	
221	**BenchmarkCurveName**	N	Name of benchmark curve. (Note tag # was reserved in FIX 4.1, added in FIX 4.3)	
222	**BenchmarkCurvePoint**	N	Point on benchmark curve. Free form values: e.g. "Y", "7Y", "INTERPOLATED". Sample values: M = combination of a number between 1-12 and a "M" for month Y = combination of number between 1-100 and a "Y" for year} 10Y-OLD = see above, then add "-OLD" when appropriate INTERPOLATED = the point is mathematically derived 2/2031 5 3/8 = the point is stated via a combination of maturity month / year and coupon See Fixed Income-specific documentation at http://www.fixprotocol.org for additional values. (Note tag # was reserved in FIX 4.1, added in FIX 4.3)	
662	**BenchmarkPrice**	N	Specifies the price of the benchmark.	
663	**BenchmarkPriceType**	N	Must be present if BenchmarkPrice is used.	
699	**BenchmarkSecurityID**	N	The identifier of the benchmark security, e.g. Treasury against Corporate bond.	
761	**BenchmarkSecurityIDSource**	N	Source of BenchmarkSecurityID. If not specified, then ID Source is understood to be the same as that in the Instrument block.	

Table 35: Spread Or Benchmark Curve Data Attributes

10.6 Yield Data Component

Yield Data Component				
The YieldData component block conveys yield information for a given Fixed Income security.				
Tag	**FieldName**	**Req'd**	**Description**	**Comment**
235	**YieldType**	N	Type of yield. (Note tag # was reserved in FIX 4.1, added in FIX 4.3)	
236	**Yield**	N	Yield percentage. (Note tag # was reserved in FIX 4.1, added in FIX 4.3)	
701	**YieldCalcDate**	N	Include as needed to clarify yield irregularities associated with date, e.g. when it falls on a non-business day.	
696	**YieldRedemptionDate**	N	Date to which the yield has been calculated (i.e. maturity, par call or current call, pre-refunded date).	
697	**YieldRedemptionPrice**	N	Price to which the yield has been calculated.	
698	**YieldRedemptionPriceType**	N	The price type of the YieldRedemptionPrice (697) See PriceType (423) for description and valid values.	

Table 36: Yield Data Attributes

10.6.1 FIX Tag YieldType(235) values

Value	Description
AFTERTAX	After Tax Yield (Municipals)
ANNUAL	Annual Yield
ATISSUE	Yield At Issue (Municipals)
AVGMATURITY	Yield To Avg Maturity
BOOK	Book Yield
CALL	Yield to Next Call
CHANGE	Yield Change Since Close
CLOSE	Closing Yield
COMPOUND	Compound Yield
CURRENT	Current Yield
GOVTEQUIV	Gvnt Equivalent Yield
GROSS	True Gross Yield
INFLATION	Yield with Inflation Assumption
INVERSEFLOATER	Inverse Floater Bond Yield
LASTCLOSE	Most Recent Closing Yield
LASTMONTH	Closing Yield Most Recent Month
LASTQUARTER	Closing Yield Most Recent Quarter
LASTYEAR	Closing Yield Most Recent Year
LONGAVGLIFE	Yield to Longest Average Life
MARK	Mark to Market Yield
MATURITY	Yield to Maturity
NEXTREFUND	Yield to Next Refund (Sinking Fund Bonds)
OPENAVG	Open Average Yield
PREVCLOSE	Previous Close Yield
PROCEEDS	Proceeds Yield
PUT	Yield to Next Put
SEMIANNUAL	Semi-annual Yield
SHORTAVGLIFE	Yield to Shortest Average Life
SIMPLE	Simple Yield
TAXEQUIV	Tax Equivalent Yield
TENDER	Yield to Tender Date
TRUE	True Yield
VALUE1_32	Yield Value Of 1/32
WORST	Yield To Worst

Table 37: FIX Tag YieldType(233) values

10.7 Multi-leg Instrument Component

Multi-leg Instrument Component				
Number of legs that make up the Security				
Tag	**FieldName**	**Req'd**	**Description**	**Comment**
555	**NoLegs**	N	Number of legs	
->	**InstrumentLeg**	N	Must be provided if Number of legs > 0	
->->600	**LegSymbol**	N	Multileg instrument's individual security's Symbol. See Symbol (55) field for description	
->->601	**LegSymbolSfx**	N	Multileg instrument's individual security's SymbolSfx. See SymbolSfx (65) field for description	
->->602	**LegSecurityID**	N	Multileg instrument's individual security's SecurityID. See SecurityID (48) field for description	
->->603	**LegSecurityIDSource**	N	Multileg instrument's individual security's SecurityIDSource. See SecurityIDSource (22) field for description	
->->	**LegSecAltIDGrp**	N		
->->607	**LegProduct**	N	Multileg instrument's individual security's Product. See Product (460) field for description	
->->608	**LegCFICode**	N	Multileg instrument's individual security's CFICode. See CFICode (461) field for description	
->->609	**LegSecurityType**	N	Refer to definition of SecurityType(167)	
->->764	**LegSecuritySubType**	N	SecuritySubType of the leg instrument. See SecuritySubType (762) field for description	
->->610	**LegMaturityMonthYear**	N	Multileg instrument's individual security's MaturityMonthYear. See MaturityMonthYear (200) field for description	
->->611	**LegMaturityDate**	N	Multileg instrument's individual security's MaturityDate. See MaturityDate (54) field for description	
->->1212	**LegMaturityTime**	N	Time of security's maturity expressed in local time with offset to UTC specified	
->->248	**LegCouponPaymentDate**	N	Multileg instrument's individual leg security's CouponPaymentDate. See CouponPaymentDate (224) field for description (Note tag # was reserved in FIX 4.1, added in FIX 4.3) (prior to FIX 4.4 field was of type UTCDate)	

->->249	**LegIssueDate**	N	Multileg instrument's individual leg security's IssueDate. See IssueDate (225) field for description (Note tag # was reserved in FIX 4.1, added in FIX 4.3) (prior to FIX 4.4 field was of type UTCDate)	
->->253	**LegFactor**	N	Multileg instrument's individual leg security's Factor. See Factor (228) field for description (Note tag # was reserved in FIX 4.1, added in FIX 4.3)	
->->257	**LegCreditRating**	N	Multileg instrument's individual leg security's CreditRating. See CreditRating (255) field for description (Note tag # was reserved in FIX 4.1, added in FIX 4.3)	
->->599	**LegInstrRegistry**	N	Multileg instrument's individual leg security's InstrRegistry. See InstrRegistry (543) field for description	
->->596	**LegCountryOfIssue**	N	Multileg instrument's individual leg security's CountryOfIssue. See CountryOfIssue (470) field for description	
->->597	**LegStateOrProvinceOfIssue**	N	Multileg instrument's individual leg security's StateOrProvinceOfIssue. See StateOrProvinceOfIssue (471) field for description	
->->598	**LegLocaleOfIssue**	N	Multileg instrument's individual leg security's LocaleOfIssue. See LocaleOfIssue (472) field for description	
->->612	**LegStrikePrice**	N	Multileg instrument's individual security's StrikePrice. See StrikePrice (202) field for description	
->->942	**LegStrikeCurrency**	N	Currency in which the strike price of a instrument leg of a multileg instrument is denominated	
->->613	**LegOptAttribute**	N	Multileg instrument's individual security's OptAttribute. See OptAttribute (206) field for description	
->->614	**LegContractMultiplier**	N	Multileg instrument's individual security's ContractMultiplier. See ContractMultiplier (23) field for description	
->->1436	**LegContractMultiplierUnit**	N	"Indicates the type of multiplier being applied to the contract. Can be optionally used to further define what unit LegContractMultiplier(tag 614) is	

			expressed in.	
->->1440	**LegFlowScheduleType**	N	The industry standard flow schedule by which electricity or natural gas is traded. Schedules exist by regions and on-peak and off-peak status, such as "Western Peak".	
->->615	**LegCouponRate**	N	Multileg instrument's individual security's CouponRate. See CouponRate (223) field for description	
->->616	**LegSecurityExchange**	N	Multileg instrument's individual security's SecurityExchange. See SecurityExchange (207) field for description	
->->617	**LegIssuer**	N	Multileg instrument's individual security's Issuer. See Issuer (106) field for description	
->->618	**EncodedLegIssuerLen**	N	Multileg instrument's individual security's EncodedIssuerLen. See EncodedIssuerLen (348) field for description	
->->619	**EncodedLegIssuer**	N	Multileg instrument's individual security's EncodedIssuer. See EncodedIssuer (349) field for description	
->->620	**LegSecurityDesc**	N	Description of a leg of a multileg instrument. See SecurityDesc(107).	
->->621	**EncodedLegSecurityDescLen**	N	Multileg instrument's individual security's EncodedSecurityDescLen. See EncodedSecurityDescLen (350) field for description	
->->622	**EncodedLegSecurityDesc**	N	Multileg instrument's individual security's EncodedSecurityDesc. See EncodedSecurityDesc (35) field for description	
->->623	**LegRatioQty**	N	Specific to the (not in)	
->->624	**LegSide**	N	Specific to the (not in)	
->->556	**LegCurrency**	N	Specific to the (not in)	
->->740	**LegPool**	N	Identifies MBS / ABS pool	
->->739	**LegDatedDate**	N	The effective date of a new securities issue determined by its underwriters. Often but not always the same as the Issue Date and the Interest Accrual Date	
->->955	**LegContractSettlMonth**	N	Specifies when the contract (i.e. MBS/TBA) will settle.	
->->956	**LegInterestAccrualDate**	N	The start date used for calculating accrued interest on debt instruments which are being sold between interest payment dates. Often but not always the same as the Issue Date and the Dated Date	
->->566	**LegPrice**	N	Used to specify an anchor price	

			for a leg as part of the definition or creation of the strategy - not used for execution price.	

Table 38: Multi-leg Instrument Attributes

10.8 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 specific 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 39: Security List Request (MsgType=x)

10.9 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 specific 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 40: Security List (MsgType=y)

10.10 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 41: Security List Update Report (MsgType=BK)

10.11 Security Definition Request (MsgType=c)

SecurityDefinitionRequest (c)			Dealer -> Execution Venue	
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.				
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 42: Security Definition Request (MsgType=c)

10.12 Security Definition (MsgType=d)

SecurityDefinition (d)			Execution Venue -> Dealer	
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.				
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 43: Security Definition (MsgType=d)

10.13 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 44: Security Status Request (MsgType=e)

10.14 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	**MDBookType**	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 45: Security Status (MsgType=f)

10.15 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 46: Market Data Request (MsgType=V)

10.16 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 47: Market Data Snapshot Full Refresh (MsgType=W)

10.17 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) andMDEntryType = 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 48: Market Data Incremental Refresh (MsgType=X)

10.18 Market Data Request Reject (MsgType=Y)

MarketDataRequestReject (Y)			Execution Venue -> Dealer	
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.				
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 49: Market Data Request Reject (MsgType=Y)

10.19 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	**Can be used in subsequent messages (eg: RFQ, New Order Single) for traceability.**
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 Not required for Indications	
	OrderQtyData	N	Quote Quantity	
38	**OrderQty**	N	The notional amount for outight 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.	
632	**BidYield**	N	Specified Bid Yield when used for an Indication (Run)	
634	**OfferYield**	N	Specified Offer Yield when used for an Indication (Run)	
60	**TransactTime**	N	Recommended value to show the Date/time of when an Indication was	

			sent.	
62	**ValidUntilTime**	N	The time when the quote will expire.	Not required for Indications (Runs)
423	**PriceType**	N	Defines the contents of the BidPx and OfferPx fields. For example: 1 = Percentage (use for Price, par=100) See Volume : "Glossary" for further value definitions)	
	RoutingGrp	N	To be used to indicate to the Buy-Side which users or desks may have a particular interest in seeing the Indication.	
->215	**NoRoutingIDs**	N		
->216	**RoutingType**	N	The type of recipient of the Indication: 1=Target List (Desk) **5=Target Person (User)**	
->217	**RoutingID**	N	The ID of the desk or user	
2533	**BidSpread**	**N**	**Basis points relative to a benchmark curve on the bid side, such as LIBOR, or a known security, such as 10Y US Treasury bond.**	**SpreadOfBenchmarkCurveData component may be used to specify the benchmark**
2534	**OfferSpread**	**N**	**Basis points relative to a benchmark curve on the offer side, such as LIBOR, or a known security, such as 10Y US Treasury bond.**	**SpreadOfBenchmarkCurveData component may be used to specify the benchmark**
	SpreadOrBenchmarkCurveData	N		
->218	**Spread**	N		**May be used for a mid-spread value.**
->220	**BenchmarkCurveCurrency**	N		
->221	**BenchmarkCurveName**	N		
->222	**BenchmarkCurvePoint**	N		
->662	**BenchmarkPrice**	N		
->663	**BenchmarkPriceType**	N		

->699	**BenchmarkSecurityID**	N	**Contains the ID of the security that is to be used as the Benchmark (contained in BidSpread(2533) /OfferSpread(2534)**	
->761	**BenchmarkSecurityIDSource**	N		
	RelativeValueGrp	**N**	**A repeating group that supports the definition of additional Spread values.**	
->2529	**NoRelativeValues**	**N**		
->2530	**RelativeValueType**	**N**	**Defines the type of Spread contained in RelativeValue (2531) 1 = ASW-Spread 2 = OIS-Spread 3 = Z-Spread 4 = Discount Margin 5 = I-Spread 6 = OA-Spread 7 = G-Spread 8 = CDS Basis 9 = CDS Interpolated Basis**	
->2531	**RelativeValue**	**N**		
->2532	**RelativeValueSide**		**Defines the Side to which the RelativeValue (2531) applies: 1 = Bid 2 = Mid 3 = Offer**	
	YieldData	N		
	StandardTrailer	Y	The standard FIX message trailer	

Table 50: Quote (MsgType=S)

10.20 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 51: Quote Cancel (MsgType=Z)

10.21 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 52: Quote Status Report (MsgType=AI)

10.22 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 53: Mass Quote (MsgType=i)

10.23 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	**TotNoCxldQuotes**	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 54: Mass Quote Ack (MsgType=b)

10.24 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 55: Quote Ack (MsgType=CW)

10.25 IOI (MsgType=6)

Indication Of Interest (6)			Dealer -> Execution Venue	
Indication of interest messages are used to market merchandise which the broker is buying or selling in either a proprietary or agency capacity. The indications can be time bound with a specific expiration value. Indications are distributed with the understanding that other firms may react to the message first and that the merchandise may no longer be available due to prior trade. Indication messages can be transmitted in various transaction types; NEW, CANCEL, and REPLACE. All message types other than NEW modify the state of the message identified in IOIRefID.				
Tag	**FieldName**	**Req'd**	**Description**	**Comment**
	StandardHeader	Y	MsgType = 6	
	ApplicationSequenceControl	N		
23	**IOIID**	Y	Unique identifier of the IOI	**Can be used in subsequent messages (eg: RFQ, New Order Single) for traceability.**
28	**IOITransType**	Y	N=New C=Cancel R=Replace	
26	**IOIRefID**	N	Required for Cancel and Replace IOITransType messages	
	Instrument	Y	Use of instrument identifiers is described in a separate section	
54	**Side**	Y	Side of Indication	**1 = Buy** **2 = Sell**
27	**IOIQty**	Y	A numeric value that represents the quantity of the Indication. Text values should not be used in this field.	**IOIQualifier(104) = M (More Behind) to denote additional Quantity.**
15	**Currency**	N		
423	**PriceType**	N	Defines the contents of the Price field. For example: 1 = Percentage (use for Price, par=100)	
44	**Price**	N	Price of the IOI : denoted by the value of Price Type (423)	
62	**ValidUntilTime**	N	Not Required	
25	**IOIQltyInd**	N	Not required	
130	**IOINaturalFlag**	N	Optional field that can be used to indicate that there is an Order behind the IOI.	
	IOIQualGrp	N	Required if any IOIQualifiers are specified.	
->199	**NoIOIQualifiers**	N	Indicates the number of repeating IOIQualifiers.	
->104	**IOIQualifier**	N	Required if NoIOIQualifiers > 0	**Common Indication Types:** **E = Axe** **F = Axe on Bid** **G = Axe on Offer** **S = Inventory or Portfolio Shown**

				(Indicates Inventory) **R = Ready To Trade (Indicates Run)** **M = More Behind**
58	**Text**	N		
60	**TransactTime**	N	Recommended value to show the Date/time of when the IOI was sent.	
	RoutingGrp	N	To be used to indicate to the Buy-Side which users or desks may have a particular interest in seeing the IOI.	
->215	**NoRoutingIDs**	N		
->216	**RoutingType**	N	The type of recipient of the Indication: 1=Target List (Desk) **5=Target Person (User)**	
->217	**RoutingID**	N	The ID of the desk or user	
	SpreadOrBenchmarkCurveData	N		
->218	**Spread**	N	May contain the Benchmark Spread for the IOI	
->220	**BenchmarkCurveCurrency**	N		
->221	**BenchmarkCurveName**	N		
->222	**BenchmarkCurvePoint**	N		
->662	**BenchmarkPrice**	N		
->663	**BenchmarkPriceType**	N		
->699	**BenchmarkSecurityID**	N	Contains the ID of the security that is to be used as the Benchmark	
->761	**BenchmarkSecurityIDSource**	N		
	RelativeValueGrp	**N**	**A repeating group that supports the definition of additional Spread values.**	
->2529	**NoRelativeValues**	**N**		
->2530	**RelativeValueType**	**N**	**Defines the type of Spread contained in RelativeValue (2531)** **1 = ASW-Spread** **2 = OIS-Spread** **3 = Z-Spread** **4 = Discount Margin** **5 = I-Spread** **6 = OA-Spread** **7 = G-Spread** **8 = CDS Basis** **9 = CDS Interpolated Basis**	
->2531	**RelativeValue**	**N**		
->2532	**RelativeValueSide**		**Defines the Side to which the RelativeValue (2531) applies:** **1 = Bid** **2 = Mid** **3 = Offer**	
	YieldData	N		

->235	**YieldType**	N		
->236	**Yield**	N	May contain the Yield of the IOI	
->701	**YieldCalcDate**	N		
->696	**YieldRedemptionDate**	N		
->697	**YieldRedemptionPrice**	N		
->698	**YieldRedemptionPriceType**	N		
	StandardTrailer	Y		

Table 56: IOI (MsgType=6)

Appendix I - Document Figures

Appendix II - Document Tables