FIX PROTOCOL
INDUSTRY-DRIVEN MESSAGING STANDARD℠

FINANCIAL INFORMATION EXCHANGE PROTOCOL (FIX)

Version 5.0

VOLUME 3 – FIX APPLICATION MESSAGES: PRE-TRADE

December 2006

DISCLAIMER

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

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

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

REPRODUCTION

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

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

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

Contents – Volume 3

FIX APPLICATION MESSAGES: PRE-TRADE

Pre-trade messaging is characterized as messages which are typically communicated prior to the placement of an order.

The specific FIX pre-trade messaging categories are:

1. INDICATION
2. EVENT COMMUNICATIONS
3. QUOTATION / NEGOTIATION
4. MARKET DATA
5. SECURITY AND TRADING SESSION DEFINITION/STATUS

Descriptions and formats of the specific FIX pre-trade application messages follow.

CATEGORY: INDICATION

Advertisements

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

The advertisement message format is as follows:

Advertisement

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = 7
2	AdvId	Y	
5	AdvTransType	Y	
3	AdvRefID	N	Required for Cancel and Replace AdvTransType messages
component block <Instrument>		Y	Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages"
555	NoLegs	N	Number of legs Identifies a Multi-leg Execution if present and non-zero.
→	**component block <InstrumentLeg>**	N	Must be provided if Number of legs > 0
711	NoUnderlyings	N	Number of underlyings
→	**component block <UnderlyingInstrument>**	N	Must be provided if Number of underlyings > 0
4	AdvSide	Y	
53	Quantity	Y	
854	QtyType	N	
44	Price	N	
15	Currency	N	
75	TradeDate	N	
60	TransactTime	N	
58	Text	N	
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.

149	URLLink	N	A URL (Uniform Resource Locator) link to additional information (i.e. http://www.XYZ.com/research.html)
30	LastMkt	N	
336	TradingSessionID	N	
625	TradingSessionSubID	N	
	StandardTrailer	Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details
Refer to FIXML element Adv

Indications of Interest

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.

The indication of interest message format is as follows:

Indication of Interest

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = 6
23	IOIID	Y	
28	IOITransType	Y	
26	IOIRefID	N	Required for Cancel and Replace IOITransType messages
component block <Instrument>		Y	Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages"
component block <Parties>		N	Insert here the set of "Parties" (firm identification) fields defined in "Common Components of Application Messages".
component block <FinancingDetails>		N	Insert here the set of "FinancingDetails" (symbology) fields defined in "Common Components of Application Messages"
711	NoUnderlyings	N	Number of underlyings
→	**component block <UnderlyingInstrument>**	N	Must be provided if Number of underlyings > 0
54	Side	Y	Side of Indication Valid values: 1 = Buy 2 = Sell 7 = Undisclosed (for IOIs) B = As Defined (for multilegs) C = Opposite (for multilegs)
854	QtyType	N	
component block <OrderQtyData>		N	Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages" The value zero is used if NoLegs repeating group is

				used Applicable if needed to express CashOrder Qty (tag 152)
27	IOIQty		Y	The value zero is used if NoLegs repeating group is used
15	Currency		N	
component block <Stipulations>			N	Insert here the set of "Stipulations" (symbology) fields defined in "Common Components of Application Messages"
555	NoLegs		N	Required for multileg IOIs
→	**component block <InstrumentLeg>**		N	Required for multileg IOIs For Swaps one leg is Buy and other leg is Sell
→	**682**	**LegIOIQty**	N	Required for multileg IOIs and for each leg.
→	**component block <LegStipulations>**		N	
423	PriceType		N	
44	Price		N	
62	ValidUntilTime		N	
25	IOIQltyInd		N	
130	IOINaturalFlag		N	
199	NoIOIQualifiers		N	Required if any IOIQualifiers are specified. Indicates the number of repeating IOIQualifiers.
→	**104**	**IOIQualifier**	N	Required if NoIOIQualifiers > 0
58	Text		N	
354	EncodedTextLen		N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText		N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
60	TransactTime		N	
149	URLLink		N	A URL (Uniform Resource Locator) link to additional information (i.e. http://www.XYZ.com/research.html)
215	NoRoutingIDs		N	Required if any RoutingType and RoutingIDs are specified. Indicates the number within repeating group.
→	**216**	**RoutingType**	N	Indicates type of RoutingID. Required if NoRoutingIDs is > 0.
→	**217**	**RoutingID**	N	Identifies routing destination. Required if NoRoutingIDs is > 0.
component block <SpreadOrBenchmarkCurveData>			N	Insert here the set of "SpreadOrBenchmarkCurveData" (Fixed Income spread or benchmark curve) fields defined in "Common Components of Application Messages"

component block <YieldData>		N	
	StandardTrailer	Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details
Refer to FIXML element IOI

CATEGORY: EVENT COMMUNICATION

News

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

The news message format is as follows:

News

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = B
42	OrigTime	N	
61	Urgency	N	
148	Headline	Y	Specifies the headline text
358	EncodedHeadlineLen	N	Must be set if EncodedHeadline field is specified and must immediately precede it.
359	EncodedHeadline	N	Encoded (non-ASCII characters) representation of the Headline field in the encoded format specified via the MessageEncoding field.
215	NoRoutingIDs	N	Required if any RoutingType and RoutingIDs are specified. Indicates the number within repeating group.
→	**216** **RoutingType**	N	Indicates type of RoutingID. Required if NoRoutingIDs is > 0.
→	**217** **RoutingID**	N	Identifies routing destination. Required if NoRoutingIDs is > 0.
146	NoRelatedSym	N	Specifies the number of repeating symbols (instruments) specified
→	**component block <Instrument>**	N	Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages"
555	NoLegs	N	Number of legs Identifies a Multi-leg Execution if present and non-zero.
→	**component block <InstrumentLeg>**	N	Must be provided if Number of legs > 0
711	NoUnderlyings	N	Number of underlyings
→	**component block <UnderlyingInstrument>**	N	Must be provided if Number of underlyings > 0
33	NoLinesOfText	Y	Specifies the number of repeating lines of text specified
→	**58** **Text**	Y	Repeating field, number of instances defined in

				LinesOfText
→	**354**	**EncodedTextLen**	N	Must be set if EncodedText field is specified and must immediately precede it.
→	**355**	**EncodedText**	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
149	URLLink		N	A URL (Uniform Resource Locator) link to additional information (i.e. http://www.XYZ.com/research.html)
95	RawDataLength		N	
96	RawData		N	
	StandardTrailer		Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details
Refer to FIXML element News

Email

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

The email message format is as follows:

Email

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = C
164	EmailThreadID	Y	Unique identifier for the email message thread
94	EmailType	Y	
42	OrigTime	N	
147	Subject	Y	Specifies the Subject text
356	EncodedSubjectLen	N	Must be set if EncodedSubject field is specified and must immediately precede it.
357	EncodedSubject	N	Encoded (non-ASCII characters) representation of the Subject field in the encoded format specified via the MessageEncoding field.
215	NoRoutingIDs	N	Required if any RoutingType and RoutingIDs are specified. Indicates the number within repeating group.
→	**216** **RoutingType**	N	Indicates type of RoutingID. Required if NoRoutingIDs is > 0.
→	**217** **RoutingID**	N	Identifies routing destination. Required if NoRoutingIDs is > 0.
146	NoRelatedSym	N	Specifies the number of repeating symbols (instruments) specified
→	**component block <Instrument>**	N	Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages"
711	NoUnderlyings	N	Number of underlyings
→	**component block <UnderlyingInstrument>**	N	Must be provided if Number of underlyings > 0
555	NoLegs	N	Number of legs Identifies a Multi-leg Execution if present and non-zero.
→	**component block <InstrumentLeg>**	N	Must be provided if Number of legs > 0
37	OrderID	N	
11	ClOrdID	N	
33	NoLinesOfText	Y	Specifies the number of repeating lines of text specified
→	**58** **Text**	Y	Repeating field, number of instances defined in

				LinesOfText
→	**354**	**EncodedTextLen**	N	Must be set if EncodedText field is specified and must immediately precede it.
→	**355**	**EncodedText**	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
95	RawDataLength		N	
96	RawData		N	
	StandardTrailer		Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details

Refer to FIXML element Email

CATEGORY: QUOTATION / NEGOTIATION

The quotation messages fall into two main sub-categories – those used for quoting in single instruments ‘Single product quoting’ and those used to quote on multiple instruments such as option series - ‘Mass quoting’

Within the ‘single product quoting’ suite of messages three business models have been identified

- Indicative quoting. – the predominant business model for retail quoting, where the expected response to a quote is a ‘previously quoted’ order which may be accepted or rejected. In the retail model the quote may be preceded by a Quote Request
- Tradeable quoting – a model where the response to a quote may be an execution (rather than an order). A common model where participants are posting quotes to an exchange. Quote may be issued in response to a Quote Request in a ‘quote on demand’ market
- Restricted Tradeable quoting – as per Tradeable quoting but the response to a quote may be either an execution or an order depending on various parameters.

The Negotiation (a.k.a. counter quoting) dialog is also supported. The Negotiation dialog may begin with either an indicative quote or a tradeable quote. **For specific usage guidance for Fixed Income negotiation and counter quotes using the quotation messages, see *Volume 7 – PRODUCT: FIXED INCOME***

The common thread linking the models is the use of the Quote message.

Quote Request

In some markets it is the practice to request quotes from brokers prior to placement of an order. The quote request message is used for this purpose. This message is commonly referred to as an Request For Quote (RFQ)

Quotes can be requested on specific securities, on specified stipulations when specific security is not known or forex rates. The quote request message can be used to request quotes on single products or multiple products.

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

In the tradeable and restricted tradeable quote models the Quote Request may be preceded by the RFQ Request message described further below.

For tradeable quote requests it is possible to specify the time period in which the request is valid for and the time period which the resulting quote must be valid for.

See VOLUME 7 - PRODUCT: FOREIGN EXCHANGE section for detailed usage notes specific to Foreign Exchange.

The quote request message format is as follows:

Quote Request

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = R
131	QuoteReqID	Y	
644	RFQReqID	N	For tradeable quote model - used to indicate to which RFQ Request this Quote Request is in response.
11	ClOrdID	N	Required when QuoteType is Tradeable and the OrdType is Limit.

528	OrderCapacity		N	
146	NoRelatedSym		Y	Number of related symbols (instruments) in Request
→	**component block <Instrument>**		Y	Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages"
→	**component block <FinancingDetails>**		N	Insert here the set of "FinancingDetails" (symbology) fields defined in "Common Components of Application Messages"
→	**711**	**NoUnderlyings**	N	Number of underlyings
→	→	**component block <UnderlyingInstrument>**	N	Must be provided if Number of underlyings > 0
→	**140**	**PrevClosePx**	N	Useful for verifying security identification
→	**303**	**QuoteRequestType**	N	Indicates the type of Quote Request (e.g. Manual vs. Automatic) being generated.
→	**537**	**QuoteType**	N	Type of quote being requested from counterparty or market (e.g. Indicative, Firm, or Restricted Tradeable) Valid values used by FX in the request: 0 = Indicative, 1 = Tradeable; Absence implies a request for an indicative quote.
→	**336**	**TradingSessionID**	N	
→	**625**	**TradingSessionSubID**	N	
→	**229**	**TradeOriginationDate**	N	
→	**54**	**Side**	N	If OrdType = "Forex - Swap", should be the side of the future portion of a F/X swap. The absence of a side implies that a two-sided quote is being requested. For single instrument use. FX values, 1 = Buy, 2 = Sell; This is from the perspective of the Initiator. If absent then a two-sided quote is being requested for spot or forward.
→	**854**	**QtyType**	N	Type of quantity specified in a quantity field. For FX, if used, should be "0".
→	**component block <OrderQtyData>**		N	Required for single instrument quoting. Required for Fixed Income if QuoteType is Tradeable.
→	**63**	**SettlType**	N	
→	**64**	**SettlDate**	N	Can be used (e.g. with forex quotes) to specify the desired "value date"
→	**193**	**SettlDate2**	N	Can be used with OrdType = "Forex - Swap" to specify the "value date" for the future portion of a F/X swap.
→	**192**	**OrderQty2**	N	Can be used with OrdType = "Forex - Swap" to specify

					the order quantity for the future portion of a F/X swap.
→	15	**Currency**		N	Can be used to specify the desired currency of the quoted price. May differ from the ‘normal’ trading currency of the instrument being quote requested.
→	**component block <Stipulations>**			N	Insert here the set of "Stipulations" (repeating group of Fixed Income stipulations) fields defined in "Common Components of Application Messages"
→	**1**	**Account**		N	
→	**660**	**AcctIDSource**		N	
→	**581**	**AccountType**		N	
→	**555**	**NoLegs**		N	Required for multileg quotes.
→	→	**component block <InstrumentLeg>**		N	Required for multileg quotes For Swaps one leg is Buy and other leg is Sell
→	→	**1017**	**LegOptionRatio**	N	LegOptionRatio is provided on Covering leg to create a delta neutral spread. In Listed Derivatives, LegDelta LegOptionRatio is multiplied by LegOptionRatio and OrderQty to determine the covering quantity
→	→	**566**	**LegPrice**	N	Specifies the price of a Leg
→	→	**687**	**LegQty**	N	
→	→	**685**	**LegOrderQty**	N	When reporting an Execution, LegOrderQty may be used on Execution Report to echo back original LegOrderQty submission. This field should be used to specify OrderQty at the leg level rather than LegQty (deprecated).
→	→	**690**	**LegSwapType**	N	
→	→	**587**	**LegSettlType**	N	
→	→	**588**	**LegSettlDate**	N	
→	→	**component block <LegStipulations>**		N	
→	→	**component block <NestedParties>**		N	
→	→	**component block <LegBenchmarkCurveData>**		N	
→	→	**654**	**LegRefID**	N	Initiator can optionally provide a unique identifier for the specific leg.
→	**735**	**NoQuoteQualifiers**		N	
→	→	**695**	**QuoteQualifier**	N	Required if NoQuoteQualifiers > 1
→	**692**	**QuotePriceType**		N	Initiator can specify the price type the quote needs to be quoted at. If not specified, the Respondent has option to

				specify how quote is quoted.
→	**40**	**OrdType**	N	Can be used to specify the type of order the quote request is for
→	**62**	**ValidUntilTime**	N	Used by the quote initiator to indicate the period of time the resulting Quote must be valid until
→	**126**	**ExpireTime**	N	The time when Quote Request will expire.
→	**60**	**TransactTime**	N	Time transaction was entered
→	**component block <SpreadOrBenchmarkCurveData>**		N	Insert here the set of "SpreadOrBenchmarkCurveData" (Fixed Income spread or benchmark curve) fields defined in "Common Components of Application Messages"
→	**423**	**PriceType**	N	
→	**44**	**Price**	N	Quoted or target price
→	**640**	**Price2**	N	Can be used with OrdType = "Forex - Swap" to specify the Quoted or target price for the future portion of a F/X swap.
→	**component block <YieldData>**		N	Insert here the set of "YieldData" (yield-related) fields defined in "Common Components of Application Messages"
→	**component block <Parties>**		N	
58	Text		N	
354	EncodedTextLen		N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText		N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
	StandardTrailer		Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details

Refer to FIXML element QuotReq

Quote Response

The Quote Response message is used to respond to a IOI message or Quote message. It is also used to counter a Quote or end a negotiation dialog.

For usage of this message in a negotiation or counter quote dialog in the fixed income space see Volume 7, Fixed Income.

The Quote Response message format is as follows:

Quote Response

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = AJ
693	QuoteRespID	Y	Unique ID as assigned by the Initiator
117	QuoteID	N	Required only when responding to a Quote.
694	QuoteRespType	Y	Type of response this Quote Response is.
11	ClOrdID	N	Unique ID as assigned by the Initiator. Required only when QuoteRespType is 1 (Hit/Lift) or 2 (Counter quote).
528	OrderCapacity	N	
23	IOIID	N	Required only when responding to an IOI.
537	QuoteType	N	(Deprecated in FIX.5.0)Default is Indicative.
735	NoQuoteQualifiers	N	
→	**695** **QuoteQualifier**	N	Required if NoQuoteQualifiers > 1
component block <Parties>		N	Insert here the set of "Parties" (firm identification) fields defined in "Common Components of Application Messages"
336	TradingSessionID	N	
625	TradingSessionSubID	N	
component block <Instrument>		Y	Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages" For multilegs supply minimally a value for Symbol (55).
component block <FinancingDetails>		N	Insert here the set of "FinancingDetails" (symbology) fields defined in "Common Components of Application Messages" For multilegs supply minimally a value for Symbol (55).
711	NoUnderlyings	N	Number of underlyings
→	**component block <UnderlyingInstrument>**	N	Must be provided if Number of underlyings > 0

54	Side		N	Required when countering a single instrument quote or "hit/lift" an IOI or Quote.
component block <OrderQtyData>			N	Insert here the set of "OrderQtyData" fields defined in "Common Components of Application Messages" Required when countering a single instrument quote or "hit/lift" an IOI or Quote.
63	SettlType		N	
64	SettlDate		N	Can be used with forex quotes to specify a specific "value date"
193	SettlDate2		N	(Deprecated in FIX.5.0)Can be used with OrdType = "Forex - Swap" to specify the "value date" for the future portion of a F/X swap.
192	OrderQty2		N	(Deprecated in FIX.5.0)Can be used with OrdType = "Forex - Swap" to specify the order quantity for the future portion of a F/X swap.
15	Currency		N	Can be used to specify the currency of the quoted prices. May differ from the 'normal' trading currency of the instrument being quoted
component block <Stipulations>			N	Optional
1	Account		N	
660	AcctIDSource		N	Used to identify the source of the Account code.
581	AccountType		N	Type of account associated with the order (Origin)
555	NoLegs		N	Required for multileg quotes
→	**component block <InstrumentLeg>**		N	Required for multileg quotes For Swaps one leg is Buy and other leg is Sell
→	**687**	**LegQty**	N	
→	**685**	**LegOrderQty**	N	When reporting an Execution, LegOrderQty may be used on Execution Report to echo back original LegOrderQty submission. This field should be used to specify OrderQty at the leg level rather than LegQty (deprecated).
→	**690**	**LegSwapType**	N	
→	**587**	**LegSettlType**	N	
→	**588**	**LegSettlDate**	N	
→	**component block <LegStipulations>**		N	
→	**component block <NestedParties>**		N	
→	**686**	**LegPriceType**	N	Code to represent type of price presented in LegBidPx and LegOfferPx. Required if LegBidPx or PegOfferPx is present.

→	**681**	**LegBidPx**	N	
→	**684**	**LegOfferPx**	N	
→	**component block <LegBenchmarkCurveData>**		N	
→	**654**	**LegRefID**	N	Initiator can optionally provide a unique identifier for the specific leg. Required for FX Swaps
→	**1067**	**LegBidForwardPoints**	N	
→	**1068**	**LegOfferForwardPoints**	N	
132	BidPx		N	If F/X quote, should be the "all-in" rate (spot rate adjusted for forward points). Note that either BidPx, OfferPx or both must be specified.
133	OfferPx		N	If F/X quote, should be the "all-in" rate (spot rate adjusted for forward points). Note that either BidPx, OfferPx or both must be specified.
645	MktBidPx		N	Can be used by markets that require showing the current best bid and offer
646	MktOfferPx		N	Can be used by markets that require showing the current best bid and offer
647	MinBidSize		N	Specifies the minimum bid size. Used for markets that use a minimum and maximum bid size.
134	BidSize		N	Specifies the bid size. If MinBidSize is specified, BidSize is interpreted to contain the maximum bid size.
648	MinOfferSize		N	Specifies the minimum offer size. If MinOfferSize is specified, OfferSize is interpreted to contain the maximum offer size.
135	OfferSize		N	Specified the offer size. If MinOfferSize is specified, OfferSize is interpreted to contain the maximum offer size.
62	ValidUntilTime		N	The time when the quote will expire. Required for FI when the QuoteRespType is 2 (Counter quote) to indicate to the Respondent when the counter offer is valid until.
188	BidSpotRate		N	May be applicable for F/X quotes
190	OfferSpotRate		N	May be applicable for F/X quotes
189	BidForwardPoints		N	May be applicable for F/X quotes
191	OfferForwardPoints		N	May be applicable for F/X quotes
631	MidPx		N	
632	BidYield		N	
633	MidYield		N	

634	OfferYield	N	
60	TransactTime	N	
40	OrdType	N	Can be used to specify the type of order the quote is for.
642	BidForwardPoints2	N	(Deprecated in FIX.5.0)Bid F/X forward points of the future portion of a F/X swap quote added to spot rate. May be a negative value
643	OfferForwardPoints2	N	(Deprecated in FIX.5.0)Offer F/X forward points of the future portion of a F/X swap quote added to spot rate. May be a negative value
656	SettlCurrBidFxRate	N	Can be used when the quote is provided in a currency other than the instrument's 'normal' trading currency. Applies to all bid prices contained in this quote message
657	SettlCurrOfferFxRate	N	Can be used when the quote is provided in a currency other than the instrument's 'normal' trading currency. Applies to all offer prices contained in this quote message
156	SettlCurrFxRateCalc	N	Can be used when the quote is provided in a currency other than the instruments trading currency.
12	Commission	N	Can be used to show the counterparty the commission associated with the transaction.
13	CommType	N	Can be used to show the counterparty the commission associated with the transaction.
582	CustOrderCapacity	N	For Futures Exchanges
100	ExDestination	N	Used when routing quotes to multiple markets
1133	ExDestinationIDSource	N	
58	Text	N	
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
44	Price	N	
423	PriceType	N	
component block <SpreadOrBenchmarkCurveData>		N	Insert here the set of "SpreadOrBenchmarkCurveData" fields defined in "Common Components of Application Messages"
component block <YieldData>		N	Insert here the set of "YieldData" fields defined in "Common Components of Application Messages"
	StandardTrailer	Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details

Refer to FIXML element QuotRsp

Quote Request Reject

The Quote Request Reject message is used to reject Quote Request messages for all quoting models.

The quote request reject message format is as follows:

Quote Request Reject

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = AG
131	QuoteReqID	Y	
644	RFQReqID	N	For tradeable quote model - used to indicate to which RFQ Request this Quote Request is in response.
658	QuoteRequestRejectReason	Y	Reason Quote was rejected
146	NoRelatedSym	Y	Number of related symbols (instruments) in Request
→	**component block <Instrument>**	Y	Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages"
→	**component block <FinancingDetails>**	N	Insert here the set of "FinancingDetails" (symbology) fields defined in "Common Components of Application Messages"
→	**711** **NoUnderlyings**	N	Number of underlyings
→	→ **component block <UnderlyingInstrument>**	N	Must be provided if Number of underlyings > 0
→	**140** **PrevClosePx**	N	Useful for verifying security identification
→	**303** **QuoteRequestType**	N	Indicates the type of Quote Request (e.g. Manual vs. Automatic) being generated.
→	**537** **QuoteType**	N	Type of quote being requested from counterparty or market (e.g. Indicative, Firm, or Restricted Tradeable)
→	**336** **TradingSessionID**	N	
→	**625** **TradingSessionSubID**	N	
→	**229** **TradeOriginationDate**	N	
→	**54** **Side**	N	If OrdType = "Forex - Swap", should be the side of the future portion of a F/X swap. The absence of a side implies that a two-sided quote is being requested. Required if specified in Quote Request message.
→	**854** **QtyType**	N	
→	**component block <OrderQtyData>**	N	Insert here the set of "OrderQytData" fields defined in "Common Components of Application Messages" Required if component is specified in Quote Request

					message.
→	**63**	**SettlType**		N	
→	**64**	**SettlDate**		N	Can be used (e.g. with forex quotes) to specify the desired "value date"
→	**193**	**SettlDate2**		N	Can be used with OrdType = "Forex - Swap" to specify the "value date" for the future portion of a F/X swap.
→	**192**	**OrderQty2**		N	Can be used with OrdType = "Forex - Swap" to specify the order quantity for the future portion of a F/X swap.
→	**15**	**Currency**		N	Can be used to specify the desired currency of the quoted price. May differ from the 'normal' trading currency of the instrument being quote requested.
→	**component block <Stipulations>**			N	Insert here the set of "Stipulations" (repeating group of Fixed Income stipulations) fields defined in "Common Components of Application Messages"
→	**1**	**Account**		N	
→	**660**	**AcctIDSource**		N	
→	**581**	**AccountType**		N	
→	**555**	**NoLegs**		N	Required for multileg quotes.
→	→	**component block <InstrumentLeg>**		N	Required for multileg quotes For Swaps one leg is Buy and other leg is Sell
→	→	**1017**	**LegOptionRatio**	N	LegOptionRatio is provided on Covering leg to create a delta neutral spread. In Listed Derivatives, LegDelta LegOptionRatio is multiplied by LegOptionRatio and OrderQty to determine the covering quantity
→	→	**566**	**LegPrice**	N	Specifies the price of a Leg
→	→	**687**	**LegQty**	N	
→	→	**685**	**LegOrderQty**	N	When reporting an Execution, LegOrderQty may be used on Execution Report to echo back original LegOrderQty submission. This field should be used to specify OrderQty at the leg level rather than LegQty (deprecated).
→	→	**690**	**LegSwapType**	N	
→	→	**587**	**LegSettlType**	N	
→	→	**588**	**LegSettlDate**	N	
→	→	**component block <LegStipulations>**		N	
→	→	**component block <NestedParties>**		N	
→	→	**component block <LegBenchmarkCurveData>**		N	

→	→	**654**	**LegRefID**	N	Initiator can optionally provide a unique identifier for the specific leg.
→	**735**	**NoQuoteQualifiers**		N	
→	→	**695**	**QuoteQualifier**	N	Required if NoQuoteQualifiers > 1
→	**692**	**QuotePriceType**		N	Initiator can specify the price type the quote needs to be quoted at. If not specified, the Respondent has option to specify how quote is quoted.
→	**40**	**OrdType**		N	Can be used to specify the type of order the quote request is for
→	**126**	**ExpireTime**		N	The time when Quote Request will expire.
→	**60**	**TransactTime**		N	Time transaction was entered
→	**component block <SpreadOrBenchmarkCurveData>**			N	Insert here the set of "SpreadOrBenchmarkCurveData" (Fixed Income spread or benchmark curve) fields defined in "Common Components of Application Messages"
→	**423**	**PriceType**		N	
→	**44**	**Price**		N	Quoted or target price
→	**640**	**Price2**		N	Can be used with OrdType = "Forex - Swap" to specify the Quoted or target price for the future portion of a F/X swap.
→	**component block <YieldData>**			N	Insert here the set of "YieldData" (yield-related) fields defined in "Common Components of Application Messages"
→	**component block <Parties>**			N	Insert here the set of "Parties" (firm identification) fields defined in "Common Components of Application Messages"
58	Text			N	
354	EncodedTextLen			N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText			N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
	StandardTrailer			Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details

Refer to FIXML element QuotReqRej

RFQ Request

In tradeable and restricted tradeable quoting markets – Quote Requests are issued by counterparties interested in ascertaining the market for an instrument. Quote Requests are then distributed by the market to liquidity providers who make markets in the instrument. The RFQ Request is used by liquidity providers to indicate to the market for which instruments they are interested in receiving Quote Requests. It can be used to register interest in receiving quote requests for a single instrument or for multiple instruments

The RFQ Request message format is as follows:

RFQ Request

Tag	*FieldName*		*Req'd*	*Comments*
	StandardHeader		Y	MsgType = AH
644	RFQReqID		Y	
146	NoRelatedSym		Y	Number of related symbols (instruments) in Request
→	**component block <Instrument>**		Y	Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages"
→	**711**	**NoUnderlyings**	N	Number of underlyings
→	→	**component block <UnderlyingInstrument>**	N	Must be provided if Number of underlyings > 0
→	**555**	**NoLegs**	N	Number of legs Identifies a Multi-leg Execution if present and non-zero.
→	→	**component block <InstrumentLeg>**	N	Must be provided if Number of legs > 0
→	**140**	**PrevClosePx**	N	Useful for verifying security identification
→	**303**	**QuoteRequestType**	N	Indicates the type of Quote Request (e.g. Manual vs. Automatic) being generated.
→	**537**	**QuoteType**	N	Type of quote being requested from counterparty or market (e.g. Indicative, Firm, or Restricted Tradeable)
→	**336**	**TradingSessionID**	N	
→	**625**	**TradingSessionSubID**	N	
263	SubscriptionRequestType		N	Used to subscribe for Quote Requests that are sent into a market
	StandardTrailer		Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details

Refer to FIXML element RFQReq

Tradeable Quote Model - Using the RFQ Request

In the quote on demand model – markets are not necessarily available until someone interested in the market generates a request.

First Party		**Market**		**Second Party (usually market maker or specialist)**
			←	RFQ Request Subscribes for Quote Requests for instruments in which party is interested in making markets
Quote Request Submits Quote Requests for instruments	→			
		Quote Requests are distributed to subscribers	→	Receives Quote Request
			←	Quote Sends Quote in response to Quote Request
	←	Market Data Quote results in change to market – causing Market Data to be distributed		

Quote

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.

In tradeable and restricted tradeable quoting models, the market maker sends quotes into a market as opposed to sending quotes directly to a counterparty.

For Fixed Income in the indicative and tradeable quoting models, the quotes are typically sent directly to an interested counterparty as opposed to a market place. ***See Volume 7 – PRODUCT: FIXED INCOME for specific descriptions and usage details.***

The quote message can be used to send unsolicited quotes in both indicative, tradeable, and restricted tradeable quoting markets.

The quote message contains a quote for a single product.

If the issuer of the quote requires a response (i.e. notification that the quote message has been accepted) then the QuoteResponseLevel field should be populated on the quote message – the response would be made using the Quote Status Report message

The quote should not be used in tradeable and restricted tradeable quoting markets, such as electronic trading systems, to broadcast quotes to market participants. The recommended approach to reporting market state changes that result from quotes received by a market is to use the market data messages.

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

See VOLUME 7 - PRODUCT: FOREIGN EXCHANGE section for more detailed usage notes specific to Foreign Exchange.

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

The time in force for a quote is determined by agreement between counterparties.

A quote can be canceled either using the Quote Cancel message or by sending a quote message with bid and offer prices and sizes all set to zero (BidPx, OfferPx, BidSize, OfferSize)

The quote message format is as follows:

Quote

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = S
131	QuoteReqID	N	Required when quote is in response to a Quote Request message
117	QuoteID	Y	
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
735	NoQuoteQualifiers	N	
→	**695** **QuoteQualifier**	N	Required if NoQuoteQualifiers > 1

301	QuoteResponseLevel	N	Level of Response requested from receiver of quote messages.
component block <Parties>		N	Insert here the set of "Parties" (firm identification) fields defined in "Common Components of Application Messages"
336	TradingSessionID	N	
625	TradingSessionSubID	N	
component block <Instrument>		Y	Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages"
component block <FinancingDetails>		N	Insert here the set of "FinancingDetails" (symbology) fields defined in "Common Components of Application Messages"
711	NoUnderlyings	N	Number of underlyings
→	**component block <UnderlyingInstrument>**	N	Must be provided if Number of underlyings > 0
54	Side	N	Required for Tradeable or Counter quotes of single instruments
component block <OrderQtyData>		N	Required for Tradeable quotes or Counter quotes of single instruments
63	SettlType	N	
64	SettlDate	N	Can be used with forex quotes to specify a specific "value date"
193	SettlDate2	N	(Deprecated in FIX.5.0)Can be used with OrdType = "Forex - Swap" to specify the "value date" for the future portion of a F/X swap.
192	OrderQty2	N	(Deprecated in FIX.5.0)Can be used with OrdType = "Forex - Swap" to specify the order quantity for the future portion of a F/X swap.
15	Currency	N	Can be used to specify the currency of the quoted prices. May differ from the 'normal' trading currency of the instrument being quoted
component block <Stipulations>		N	Insert here the set of "Stipulations" (repeating group of Fixed Income stipulations) fields defined in "Common Components of Application Messages"
1	Account	N	
660	AcctIDSource	N	
581	AccountType	N	Type of account associated with the order (Origin)
555	NoLegs	N	Required for multileg quotes
→	**component block <InstrumentLeg>**	N	Required for multileg quotes For Swaps one leg is Buy and other leg is Sell

→	**687**	**LegQty**	N	
→	**685**	**LegOrderQty**	N	When reporting an Execution, LegOrderQty may be used on Execution Report to echo back original LegOrderQty submission. This field should be used to specify OrderQty at the leg level rather than LegQty (deprecated).
→	**690**	**LegSwapType**	N	
→	**587**	**LegSettlType**	N	
→	**588**	**LegSettlDate**	N	
→	**component block <LegStipulations>**		N	
→	**component block <NestedParties>**		N	
→	**686**	**LegPriceType**	N	Code to represent type of price presented in LegBidPx and LegOfferPx. Required if LegBidPx or PegOfferPx is present.
→	**681**	**LegBidPx**	N	
→	**684**	**LegOfferPx**	N	
→	**component block <LegBenchmarkCurveData>**		N	
→	**654**	**LegRefID**	N	Initiator can optionally provide a unique identifier for the specific leg. Required for FX Swaps
→	**1067**	**LegBidForwardPoints**	N	
→	**1068**	**LegOfferForwardPoints**	N	
132	BidPx		N	If F/X quote, should be the "all-in" rate (spot rate adjusted for forward points). Note that either BidPx, OfferPx or both must be specified.
133	OfferPx		N	If F/X quote, should be the "all-in" rate (spot rate adjusted for forward points). Note that either BidPx, OfferPx or both must be specified.
645	MktBidPx		N	Can be used by markets that require showing the current best bid and offer
646	MktOfferPx		N	Can be used by markets that require showing the current best bid and offer
647	MinBidSize		N	Specifies the minimum bid size. Used for markets that use a minimum and maximum bid size.
134	BidSize		N	Specifies the bid size. If MinBidSize is specified, BidSize is interpreted to contain the maximum bid size.
648	MinOfferSize		N	Specifies the minimum offer size. If MinOfferSize is specified, OfferSize is interpreted to contain the

			maximum offer size.
135	OfferSize	N	Specified the offer size. If MinOfferSize is specified, OfferSize is interpreted to contain the maximum offer size.
62	ValidUntilTime	N	The time when the quote will expire
188	BidSpotRate	N	May be applicable for F/X quotes
190	OfferSpotRate	N	May be applicable for F/X quotes
189	BidForwardPoints	N	May be applicable for F/X quotes
191	OfferForwardPoints	N	May be applicable for F/X quotes
1065	BidSwapPoints	N	Bid swap points of an FX Swap quote.
1066	OfferSwapPoints	N	
631	MidPx	N	
632	BidYield	N	
633	MidYield	N	
634	OfferYield	N	
60	TransactTime	N	
40	OrdType	N	Can be used to specify the type of order the quote is for
642	BidForwardPoints2	N	(Deprecated in FIX.5.0)Bid F/X forward points of the future portion of a F/X swap quote added to spot rate. May be a negative value
643	OfferForwardPoints2	N	(Deprecated in FIX.5.0)Offer F/X forward points of the future portion of a F/X swap quote added to spot rate. May be a negative value
656	SettlCurrBidFxRate	N	Can be used when the quote is provided in a currency other than the instrument's 'normal' trading currency. Applies to all bid prices contained in this quote message
657	SettlCurrOfferFxRate	N	Can be used when the quote is provided in a currency other than the instrument's 'normal' trading currency. Applies to all offer prices contained in this quote message
156	SettlCurrFxRateCalc	N	Can be used when the quote is provided in a currency other than the instruments trading currency.
13	CommType	N	Can be used to show the counterparty the commission associated with the transaction.
12	Commission	N	Can be used to show the counterparty the commission associated with the transaction.
582	CustOrderCapacity	N	For Futures Exchanges
100	ExDestination	N	Used when routing quotes to multiple markets
1133	ExDestinationIDSource	N	
528	OrderCapacity	N	

423	PriceType	N	
component block <SpreadOrBenchmarkCurveData>		N	
component block <YieldData>		N	
58	Text	N	
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
	StandardTrailer	Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details

Refer to FIXML element Quot

Example: Quote for Single Security

QuoteID=XXX
QuoteReqID=YYY
Symbol=AA
MaturyMonthYear=199901
StrikePrice=25.00
CFICode=”OCXXXS”
BixPx=5.00
OfferPx=5.25
BidSize=10
OfferSize=10

Quote Cancel

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

The Quote Cancel message supports cancelation of:

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

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

It is recommended that all Cancel messages be acknowledged using the Quote Status Report message.

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

The Quote Cancel message format is as follows:

Quote Cancel

Tag	*FieldName*	*Req'd*	*Comments*
	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 = 5 (cancel quote specified in QuoteID)
298	QuoteCancelType	Y	Identifies the type of Quote Cancel request.
301	QuoteResponseLevel	N	Level of Response requested from receiver of quote messages.
component block <Parties>		N	Insert here the set of "Parties" (firm identification) fields defined in "Common Components of Application Messages"
1	Account	N	
660	AcctIDSource	N	
581	AccountType	N	Type of account associated with the order (Origin)
336	TradingSessionID	N	
625	TradingSessionSubID	N	
295	NoQuoteEntries	N	The number of securities (instruments) whose quotes are to be canceled Not required when cancelling all quotes.
→	**component block <Instrument>**	N	Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages"
→	**component block <FinancingDetails>**	N	Insert here the set of "FinancingDetails" (symbology) fields defined in "Common Components of Application

				Messages"
→	711	**NoUnderlyings**	N	Number of underlyings
→	→	**component block <UnderlyingInstrument>**	N	Must be provided if Number of underlyings > 0
→	555	**NoLegs**	N	Number of legs Identifies a Multi-leg Execution if present and non-zero.
→	→	**component block <InstrumentLeg>**	N	Must be provided if Number of legs > 0
	StandardTrailer		Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details

Refer to FIXML element QuotCxl

Options usage notes:

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

Examples of the types of Quote Cancel operations:

Cancel for Symbol(s)

Cancel all option quotes for symbol: IBM

QuoteID=*user defined identifier for this cancel request*

QuoteCancelType=1

NoQuoteEntries=1

Symbol=IBM

CFICode=O

Cancel for Security Type(s)

Cancel all futures quotes for symbol: T (notice that CFICode is specified not SecurityType).

QuoteID=*user defined identifier for this cancel request*

QuoteCancelType=2

NoQuoteEntries=1

Symbol=N/A

CFICode=F

Cancel Quotes for underlying symbols

Cancel all quotes for options with an underlying symbol of IBM

QuoteID= *user defined identifier for this cancel request*

QuoteCancelType=3

NoQuoteEntries=1

Symbol=IBM

CFICode=O

Cancel All Quotes

Cancel all quotes associated with this FIX Session

QuoteID= *user defined identifier for this cancel request*

QuoteCancelType=4

Cancel all quotes for a specific trading session

QuoteID= *user defined identifier for this cancel request*

QuoteCancelType=4

TradingSessionID=*a trading session identifier in a market*

Cancel All Quotes for specific parties

QuoteID= *user defined identifier for this cancel request*

QuoteCancelType=4

PartyID=party identifier

NoPartyIDs=1

- PartyID=*party identifier*
- PartyIDSource=*source*
- PartyRole=*role*

Quote Status Request

The quote status request message is used for the following purposes in markets that employ tradeable or restricted tradeable quotes:

- For the issuer of a quote in a market to query the status of that quote (using the QuoteID to specify the target quote).
- To subscribe and unsubscribe for Quote Status Report messages for one or more securities.

The format of the quote status request message is:

Quote Status Request

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = a (lowercase)
649	QuoteStatusReqID	N	
117	QuoteID	N	
component block <Instrument>		Y	Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages"
component block <FinancingDetails>		N	Insert here the set of "FinancingDetails" (symbology) fields defined in "Common Components of Application Messages"
711	NoUnderlyings	N	Number of underlyings
→	**component block <UnderlyingInstrument>**	N	Must be provided if Number of underlyings > 0
555	NoLegs	N	Number of legs Identifies a Multi-leg Execution if present and non-zero.
→	**component block <InstrumentLeg>**	N	Must be provided if Number of legs > 0
component block <Parties>		N	Insert here the set of "Parties" (firm identification) fields defined in "Common Components of Application Messages"
1	Account	N	
660	AcctIDSource	N	
581	AccountType	N	Type of account associated with the order (Origin)
336	TradingSessionID	N	
625	TradingSessionSubID	N	
263	SubscriptionRequestType	N	Used to subscribe for Quote Status Report messages
	StandardTrailer	Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details
Refer to FIXML element QuotStatReq

Application of Quote Status Request to Options Markets using tradeable or restricted tradeable quoting models:

To retrieve status of all quotes for a given underlying symbol for options enter the Symbol[55] and optionally the SecurityID[167] along with a CFICode[537]=”OXXXXX”.

Quote Status Report

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)***

Quote Status Report

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = AI
649	QuoteStatusReqID	N	
131	QuoteReqID	N	Required when quote is in response to a Quote Request message
117	QuoteID	Y	
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
component block <Parties>		N	Insert here the set of "Parties" (firm identification) fields defined in "Common Components of Application Messages"
336	TradingSessionID	N	
625	TradingSessionSubID	N	
component block <Instrument>		Y	Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages"
component block <FinancingDetails>		N	Insert here the set of "FinancingDetails" (symbology) fields defined in "Common Components of Application Messages"
711	NoUnderlyings	N	Number of underlyings
→	**component block <UnderlyingInstrument>**	N	Must be provided if Number of underlyings > 0
54	Side	N	
component block <OrderQtyData>		N	Required for Tradeable quotes of single instruments
63	SettlType	N	
64	SettlDate	N	Can be used with forex quotes to specify a specific "value date"
193	SettlDate2	N	(Deprecated in FIX.5.0)Can be used with OrdType =

				"Forex - Swap" to specify the "value date" for the future portion of a F/X swap.
192	OrderQty2		N	(Deprecated in FIX.5.0)Can be used with OrdType = "Forex - Swap" to specify the order quantity for the future portion of a F/X swap.
15	Currency		N	Can be used to specify the currency of the quoted prices. May differ from the 'normal' trading currency of the instrument being quoted
component block <Stipulations>			N	
1	Account		N	
660	AcctIDSource		N	
581	AccountType		N	Type of account associated with the order (Origin)
555	NoLegs		N	Required for multileg quote status reports
→	**component block <InstrumentLeg>**		N	Required for multileg quote status reports For Swaps one leg is Buy and other leg is Sell
→	**687**	**LegQty**	N	
→	**685**	**LegOrderQty**	N	When reporting an Execution, LegOrderQty may be used on Execution Report to echo back original LegOrderQty submission. This field should be used to specify OrderQty at the leg level rather than LegQty (deprecated).
→	**690**	**LegSwapType**	N	
→	**587**	**LegSettlType**	N	
→	**588**	**LegSettlDate**	N	
→	**component block <LegStipulations>**		N	
→	**component block <NestedParties>**		N	
735	NoQuoteQualifiers		N	
→	**695**	**QuoteQualifier**	N	Required if NoQuoteQualifiers > 1
126	ExpireTime		N	
44	Price		N	
423	PriceType		N	
component block <SpreadOrBenchmarkCurveData>			N	
component block <YieldData>			N	
132	BidPx		N	If F/X quote, should be the "all-in" rate (spot rate adjusted for forward points). Note that either BidPx, OfferPx or both must be specified.

133	OfferPx	N	If F/X quote, should be the "all-in" rate (spot rate adjusted for forward points). Note that either BidPx, OfferPx or both must be specified.
645	MktBidPx	N	Can be used by markets that require showing the current best bid and offer
646	MktOfferPx	N	Can be used by markets that require showing the current best bid and offer
647	MinBidSize	N	Specifies the minimum bid size. Used for markets that use a minimum and maximum bid size.
134	BidSize	N	Specifies the bid size. If MinBidSize is specified, BidSize is interpreted to contain the maximum bid size.
648	MinOfferSize	N	Specifies the minimum offer size. If MinOfferSize is specified, OfferSize is interpreted to contain the maximum offer size.
135	OfferSize	N	Specified the offer size. If MinOfferSize is specified, OfferSize is interpreted to contain the maximum offer size.
62	ValidUntilTime	N	
188	BidSpotRate	N	May be applicable for F/X quotes
190	OfferSpotRate	N	May be applicable for F/X quotes
189	BidForwardPoints	N	May be applicable for F/X quotes
191	OfferForwardPoints	N	May be applicable for F/X quotes
631	MidPx	N	
632	BidYield	N	
633	MidYield	N	
634	OfferYield	N	
60	TransactTime	N	
40	OrdType	N	Can be used to specify the type of order the quote is for
642	BidForwardPoints2	N	(Deprecated in FIX.5.0)Bid F/X forward points of the future portion of a F/X swap quote added to spot rate. May be a negative value
643	OfferForwardPoints2	N	(Deprecated in FIX.5.0)Offer F/X forward points of the future portion of a F/X swap quote added to spot rate. May be a negative value
656	SettlCurrBidFxRate	N	Can be used when the quote is provided in a currency other than the instrument's 'normal' trading currency. Applies to all bid prices contained in this message
657	SettlCurrOfferFxRate	N	Can be used when the quote is provided in a currency other than the instrument's 'normal' trading currency. Applies to all offer prices contained in this message
156	SettlCurrFxRateCalc	N	Can be used when the quote is provided in a currency

			other than the instruments trading currency.
13	CommType	N	Can be used to show the counterparty the commission associated with the transaction.
12	Commission	N	Can be used to show the counterparty the commission associated with the transaction.
582	CustOrderCapacity	N	For Futures Exchanges
100	ExDestination	N	Used when routing quotes to multiple markets
1133	ExDestinationIDSource	N	
297	QuoteStatus	N	Quote Status
58	Text	N	
354	EncodedTextLen	N	
355	EncodedText	N	
	StandardTrailer	Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details

Refer to FIXML element QuotStatRpt

Indicative Quoting Model

FIX supports an Indicative Quoting Model that is frequently used between two counterparties. In the Indicative Quoting Model a party interested in a particular security issues a Quote Request to a counterparty. The counterparty responds with an indicative quote. The first party – assuming the quote meets their requirements – can send back a New Order – Single (order type = Previously Quoted). The New Order – Single message should contain the QuoteID of the Quote. The issuer of the quote does not necessarily have to execute the order – based upon market conditions or characteristics contained on the New Order Message.

Indicative Quoting Model Message Scenario

First Party		Second Party
QuoteRequest This is an optional first step. Counterparties may agree to provide indicative quotes in a continuous manner.	→	Accepts Quote Request Creates a Quote for the product specified in the Quote Request
Accepts Quote – after examining market indicated in quote decides whether to place a New Order	←	Send Quote message (can be a one or two sided market). The QuoteReqID should be set to the QuoteReqID from the Quote Request to which this Quote is a response.
New Order –Single – should reference the QuoteID for which the New Order message in which the New Order is a response. The OrdType should be set to previously quoted.	→	Accepts the New Order message. Should be acknowledged as New.
	←	Sends Execution Report for NEW (Optional)
	←	Sends Execution Report OrdStatus=FILL if the order is acceptable or Or Send Execution Report OrdStatus=PARTIALLY FILLED Or Send Execution Report OrdStatus=REJECTED

Indicative quotes can also be sent out on an unsolicited basis. The correct response is the New Order (previously quoted) as above

Tradeable Quote Model

Beginning with FIX 4.2 support was provided for markets requiring tradeable quotes. A tradeable quote market has market makers or specialist issue quotes that are kept as part of a market. A tradeable quote can be directly traded against orders or other quotes (depending on market rules). The market created by these quotes should be distributed using the Market Data messages. When orders are entered in response to the markets created by the tradeable quotes – trades may result. Trades are reported with an Execution Report.

Tradeable Quote model markets can be continuously quoted or quoted on demand or a combination of the two. In continuously quoted markets – market makers or specialists are required to maintain two sided markets which comply with market requirements for bid-ask spread and minimum quantity. In the quote on demand market – market makers and specialists are usually required to respond to Quote Requests (RFQs) within a market prescribed time limit with a quote which complies with exchange prescribed bid-ask spread and minimum quantity.

Tradeable Quote Model - Reporting Quote Status back to Issuer

The market should provide unsolicited quote status back to the quote issuer if the state of a quote changes with the exception of trades (fills) that occur against a quote. Trades (fills) are reported using the Execution Report.

NOTE: The Quote Message should not be used to report trades. Only the Execution Report should be used to report fills against a tradeable or restricted tradeable quote.

Market maker or specialist		Market
Quote Valid tradeable or restricted tradeable quote sent into market – either unsolicited or in reply to a Quote Request from the market.	→	Accepts Quote and applies to the market
Accepts Quote and updates trading system based upon status reported by market	←	Based upon market rules or the QuoteResponseLevel requested by Quote Issuer the market will send Quote Status Report messages back to the quote issuer to report quote status (using the QuoteStatus field).
	←	If a trade (fill) occurs against a tradeable quote an Execution Report (ExecType=Fill or Partial Fill) is sent to the quote issuer.

Using the Execution Report to report a trade on a Tradeable Quote

The Execution Report should be used to report trades involving a tradeable quote. Because quotes are usually replaced or replenished – often times with the same QuoteID – it is not always possible, nor does it necessarily make sense for markets to keep track and transmit the detailed quantity information required on the quote. Execution Reports for trades against a tradeable quote can use the quantity fields in the following manner.

Tag#	Field Name	Reqd	**Usage in reporting trades on** tradeable or restricted tradeable **quotes**
38	OrderQty	N	Quote quantity when the fill occurred.
32	LastQty	N	Same as for a fill against an order
31	LastPx	N	Same as for a fill against an order
151	LeavesQty	Y	Quantity remaining open in the market
14	CumQty	Y	Use 0.0 if market is unable to provide a cumulative total.
6	AvgPx	Y	Use 0.0 if market is unable to provide an average price

Tradeable Quote Model - Quote on Demand Message Scenario

In the quote on demand model – markets are not necessarily available until someone interested in the market generates a request.

First Party		Market		Second Party (usually market maker or specialist)
			←	Optional Quote Status Request to subscribe for Quote Status for one or more instruments (some markets may chose to configure this out of band).
		Tracks Subscription Requests for each party connected to market NOTE: Some markets may choose to configure subscription and dissemination of Quote Request out-of-band – instead of in-band.	←	RFQ Request Subscribe for Quote Requests
Quote Request (Optional request for quote if no quote exists in the market)	→	Market checks validity of Quote Request and then sends it to subscribed participants	→	Accepts Quote Request Generates a quote based upon request
		Interprets quotes and applies them to a market Interprets QuoteResponse Level to determine if quote status should be sent back to the quote issuer using a Quote Status Report message with the QuoteStatus field set appropriately Valid quote that changes market should be disseminated using Market Data messages Optional Quote Status Report	← →	Quote Quote is sent that complies with market requirements.
Receives Market Data Will use Market Data to make market participation and pricing decision	←	If the Quote is valid and has an impact on the market Market Data is published (NOTE: The process of subscribing for market data is omitted from this example)	→	Receives Market Data Useful in creating subsequent quotes

First Party		Market		Second Party (usually market maker or specialist)
Sends New Order – Single Receives Execution Report – Pending New (optional) Received Execution Report – NEW	→ ← ←	Order is matched against other orders and quotes according to market rules. (NOTE: This can be either open-outcry based markets with or without limit book or a fully electronic market)		
Receipt of Execution Report – Reporting Fill or Partial Fill	←	If the order is matched against the tradeable or restricted tradeable quote resulting in a trade – Execution Reports are sent to the counterparties of the trade	→	Receipt of Market Maker side Execution Report reporting Fill against the previously submitted tradeable or restricted tradeable Quote (Optionally can choose to replenish market or wait for next Quote Request)

Tradeable Quote Model Message Scenario - Continuous markets

The Tradeable Quote Model in a continuously quoted marketed – requires market makers or specialist to maintain market compliant two sided markets.

First Party		Market		Second Party (usually market maker or specialist)
Uses market data to determine market participation and pricing on orders	←	Market Data is disseminated (NOTE: This may include the need to transmit expected opening prices based upon current state of the book at the opening)	→	Uses market data to create subsequent quotes
		Interprets quotes and applies them to a market Interprets QuoteResponse Level to determine if quote status should be sent back to the quote issuer using a Quote Status Report message with the QuoteStatus field set appropriately Market Data will be generated to report state of the book is changed by the quote Optional Quote Status Report	← →	Quote Market Makers / Specialist are expected to maintain two sided quotes that comply with market required bid-ask spread and minimum quantities

First Party		Market		Second Party (usually market maker or specialist)
Receives Market Data Will use Market Data to make market participation and pricing decision	←	If the Quote is valid and has an impact on the market Market Data is published (NOTE: The process of subscribing for market data is omitted from this example)	→	Receives Market Data Used to create subsequent quotes
Sends New Order – Single Receives Execution Report – Pending New (optional) Received Execution Report – NEW	→ ← ←	Order is matched against other orders and quotes according to market rules. (NOTE: This can be either open-outcry based markets with or without limit book or a fully electronic market)		
Receipt of Execution Report – Reporting Fill or Partial Fill	←	If the order is matched against the tradeable or restricted tradeable quote resulting in a trade – Execution Reports are sent to the counterparties of the trade	→	Receipt of Market Maker side Execution Report reporting Fill against the previously submitted tradeable or restricted tradeable Quote (Optionally can choose to replenish market or wait for next Quote Request)
		Quote is processed as above – market data is generated – an optional Quote Status Report message is generated	←	Replenishes Quote – possibly changing prices and quantities

Tradeable Quote Model - Querying for Quote Status

Market participants may need to query the status of their current quotes. Normally a market will provide status in an unsolicited manner back to the quote issuer. However, to support system or session recovery – the Quote Status Request can be used to query the current state of quotes within a market.

Market maker or specialist		Market
Quote Status Request Contains information on the securities for which the quote status request is being issued or the QuoteID of a previously submitted quote.	→	Accepts Quote Status Request

Accepts Quote and updates trading system.	←	Sends Quote Status Report messages with the QuoteStatus field set, bid and ask prices, and quantities for each quote belonging to the request issuer that meet the criteria in the request. If there is a current quote in the market – the Quote Status Report in response to a Quote Status Request should be sent with a QuoteStatus of “Query”. The Quote Status Report message can also contain a QuoteStatus of “Quote Not Found” if no quote currently exists.

Restricted Tradeable Quote Model

The Restricted Tradeable Quote Model extends the behavior of the Tradeable Quote Model to place limits on quantity or price. Orders received against the Restricted Tradeable Quote that are within limits set by the market – will execute against the quote automatically – just as in the case of the Tradeable Quote Model.

If the order is outside the limits specified by the market – the order is forwarded to the quote issuer(s) to be filled, partially filled with remaining quantity cancelled, or canceled.

Restricted Tradeable Quote Model Message Scenario

The Restricted Tradeable Quote Model will automatically trade against orders within restrictions specified by the market in terms of quantity or price.

First Party		Market		Second Party (usually market maker or specialist)
Uses market data to determine market participation and pricing on orders	←	Market Data is disseminated (NOTE: This may include the need to transmit expected opening prices based upon current state of the book at the opening)	→	Uses market data to create subsequent quotes
		Interprets quotes and applies them to a market Interprets QuoteResponse Level to determine if quote status should be sent back to the quote issuer using a Quote Status Report message with the QuoteStatus field set appropriately Market Data will be generated to report state of the book is changed by the quote	←	Quote Market Makers / Specialist are expected to maintain two sided quotes that comply with market required bid-ask spread and minimum quantities
Receives Market Data Will use Market Data to make market participation and pricing decision	←	If the Quote is valid and has an impact on the market Market Data is published (NOTE: The process of subscribing for market data is omitted from this example)	→	Receives Market Data Used to create subsequent quotes
Sends New Order – Single Receives Execution Report – Pending New (optional) Received Execution Report – NEW	→ ← ←	Order is matched against other orders and quotes according to market rules. (NOTE: This can be either open-outcry based markets with or without limit book or a fully electronic market)		

First Party		**Market**		**Second Party (usually market maker or specialist)**
Receipt of Execution Report – Reporting Fill or Partial Fill	←	If the order is matched against the tradeable or restricted tradeable quote resulting in a trade – Execution Reports are sent to the counterparties of the trade	→	Receipt of Market Maker side Execution Report reporting Fill against the previously submitted tradeable or restricted tradeable Quote (Optionally can choose to replenish market or wait for next Quote Request)
		Quote is processed as above – market data is generated – an optional quote status message is generated	←	Replenishes Quote – possibly changing prices and quantities
Sends New Order – Single that is outside the restrictions specified by the market	→	Order is identified as being outside automatic execution parameters. The order is sent to the quote issuer(s)	→	Receives order and decides if the order is acceptable
				Sends back an execution for partial quantity, full quantity, or cancels the order

Mass Quote

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

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

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

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

The grouping of quotes is as follows:

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

- QuoteSetID – Is a unique ID given to the quote set
- Information regarding the security to which all of the quotes belong
- TotQuoteEntries – defines the number of quotes for the quote set across all messages
- NoQuoteEntries – defines the number of quotes contained within this message for this quote set
    - QuoteEntryID – Is a unique ID given to a specific quote entry
    - Information regarding the specific quote (bid/ask size and price)

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

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

Requesting Acknowledgement for Mass Quotes

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

See "***Mass Quote Message Scenarios***"

The Mass Quote message format is as follows:

Mass Quote

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = i (lowercase)
131	QuoteReqID	N	Required when quote is in response to a Quote Request message
117	QuoteID	Y	
537	QuoteType	N	Type of Quote Default is Indicative if not specified
301	QuoteResponseLevel	N	Level of Response requested from receiver of quote messages.
component block <Parties>		N	Insert here the set of "Parties" (firm identification) fields defined in "Common Components of Application Messages"
1	Account	N	
660	AcctIDSource	N	
581	AccountType	N	Type of account associated with the order (Origin)
293	DefBidSize	N	Default Bid Size for quote contained within this quote message - if not explicitly provided.
294	DefOfferSize	N	Default Offer Size for quotes contained within this quote message - if not explicitly provided.
296	NoQuoteSets	Y	The number of sets of quotes in the message
→ **302**	**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.
→	**component block <UnderlyingInstrument>**	N	Insert here the set of "UnderlyingInstrument" (underlying symbology) fields defined in "Common Components of Application Messages"
→ **367**	**QuoteSetValidUntilTime**	N	
→ **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.
→	**295**	**NoQuoteEntries**		Y	The number of quotes for this Symbol (instrument) (QuoteSet) that follow in this message. ** Nested Repeating Group follows **
→	→	**299**	**QuoteEntryID**	Y	Uniquely identifies the quote as part of a QuoteSet. Must be used if NoQuoteEntries is used
→	→	**component block <Instrument>**		N	Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages"
→	**555**	**NoLegs**		N	Number of legs Identifies a Multi-leg Execution if present and non-zero.
→	→	**component block <InstrumentLeg>**		N	Must be provided if Number of legs > 0
→	→	**132**	**BidPx**	N	If F/X quote, should be the "all-in" rate (spot rate adjusted for forward points). Note that either BidPx, OfferPx or both must be specified.
→	→	**133**	**OfferPx**	N	If F/X quote, should be the "all-in" rate (spot rate adjusted for forward points). Note that either BidPx, OfferPx or both must be specified.
→	→	**134**	**BidSize**	N	
→	→	**135**	**OfferSize**	N	
→	→	**62**	**ValidUntilTime**	N	
→	→	**188**	**BidSpotRate**	N	May be applicable for F/X quotes
→	→	**190**	**OfferSpotRate**	N	May be applicable for F/X quotes
→	→	**189**	**BidForwardPoints**	N	May be applicable for F/X quotes
→	→	**191**	**OfferForwardPoints**	N	May be applicable for F/X quotes
→	→	**631**	**MidPx**	N	
→	→	**632**	**BidYield**	N	
→	→	**633**	**MidYield**	N	
→	→	**634**	**OfferYield**	N	
→	→	**60**	**TransactTime**	N	
→	→	**336**	**TradingSessionID**	N	
→	→	**625**	**TradingSessio**	N	

			nSubID		
→	→	**64**	**SettlDate**	N	Can be used with forex quotes to specify a specific "value date"
→	→	**40**	**OrdType**	N	Can be used to specify the type of order the quote is for
→	→	**193**	**SettlDate2**	N	Can be used with OrdType = "Forex - Swap" to specify the "value date" for the future portion of a F/X swap.
→	→	**192**	**OrderQty2**	N	Can be used with OrdType = "Forex - Swap" to specify the order quantity for the future portion of a F/X swap.
→	→	**642**	**BidForwardPoints2**	N	Bid F/X forward points of the future portion of a F/X swap quote added to spot rate. May be a negative value
→	→	**643**	**OfferForwardPoints2**	N	Offer F/X forward points of the future portion of a F/X swap quote added to spot rate. May be a negative value
→	→	**15**	**Currency**	N	Can be used to specify the currency of the quoted price.
	StandardTrailer			Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details

Refer to FIXML element MassQuot

Notes on usage:

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

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

```
QuoteID=XXX
QuoteReqID=YYY
NoQuoteSets=1
   QuoteSetID=1
   Symbol=AA
   TotQuoteEntries=2
   NoQuoteEntries=2
   Other quote set fields
   QuoteEntryID=1
      MaturyMonthYear=199901
      StrikePrice=25.00
      CFICode="OCXXXS"
      BixPx=5.00
      OfferPx=5.25
      BidSize=10
      OfferSize=10
   QuoteEntryID=2
      MaturyMonthYear=199901
      StrikePrice=30.00
      CFICode="OCXXXS"
      BixPx=3.00
```

```
        OfferPx=3.25
        BidSize=10
        OfferSize=10
```

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

First Message:

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

Second Message:

```
    QuoteID=XXX
    QuoteReqID=YYY
    NoQuoteSets=1
      QuoteSetID=1
      Symbol=AA
      Other quote set fields
      TotQuoteEntries=3
      NoQuoteEntries=1
      QuoteEntryID=3
        MaturyMonthYear=199901
        StrikePrice=35.00
        CFICode=”OCXXXS”

        BixPx=2.00
        OfferPx=2.25
        BidSize=10
        OfferSize=10
```

Example: Multiple Quotes for Fixed Income publishing

```
    QuoteID=XXX
    NoQuoteSets=1
```

```
QuoteSetID=1
TotQuoteEntries=3
NoQuoteEntries=3
Other quote set fields
QuoteEntryID=1
   Symbol=DE10003453
   SecurityID=DE10003453
   SecurityIDSource=4
   BixPx=105
   BidYield=.043
   OfferPx=102.3
   OfferYield=.0525
   BidSize=10
   OfferSize=10
QuoteEntryID=2
   Symbol=NL0000102606
   SecurityID=NL0000102606
   SecurityIDSource=4
   MidPx=105
   MidYield=4.3
QuoteEntryID=3
   Symbol=FR0100059601
   SecurityID=FR0100059601
   SecurityIDSource=4
   BidYield=.048
   OfferYield=.057
   BidSize=5
   OfferSize=5
```

Mass Quote Acknowledgement

Mass Quote Acknowledgement is used as the application level response to a Mass Quote message. The Mass Quote Acknowledgement contains a field for reporting the reason in the event that the entire quote is rejected (QuoteRejectReason[300]). The Mass Quote Acknowledgement also contains a field for each quote that is used in the event that the quote entry is rejected (QuoteEntryRejectReason[368]). The ability to reject an individual quote entry is important so that the majority of quotes can be successfully applied to the market instead of having to reject the entire Mass Quote for a minority of rejected quotes.

Derivative markets are characterized by high bandwidth consumption – due to a change in an underlying security price causing multiple (often in the hundreds) of quotes to be recalculated and retransmitted to the market. For that reason the ability for market participants (and the market) to be able to set the level of response requested to a Mass Quote message is specified using the QuoteResponseLevel[301] field.

The Mass Quote Acknowledgement message format is as follows:

Mass Quote Acknowledgement

Tag	*FieldName*	*Req'd*	*Comments*
	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 Quote message
297	QuoteStatus	Y	Status of the mass quote acknowledgement.
300	QuoteRejectReason	N	Reason Quote was rejected.
301	QuoteResponseLevel	N	Level of Response requested from receiver of quote messages. Is echoed back to the counterparty.
537	QuoteType	N	Type of Quote
component block <Parties>		N	Insert here the set of "Parties" (firm identification) fields defined in "Common Components of Application Messages"
1	Account	N	
660	AcctIDSource	N	
581	AccountType	N	Type of account associated with the order (Origin)
58	Text	N	
354	EncodedTextLen	N	
355	EncodedText	N	
296	NoQuoteSets	N	The number of sets of quotes in the message
→	**302** **QuoteSetID**	N	First field in repeating group. Required if NoQuoteSets > 0
→	**component block**	N	Insert here the set of "UnderlyingInstrument"

	<UnderlyingInstrument>				(underlying symbology) fields defined in "Common Components of Application Messages" 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
→	**893**	**LastFragment**		N	Indicates whether this is the last fragment in a sequence of message fragments. Only required where message has been fragmented.
→	**295**	**NoQuoteEntries**		N	The number of quotes for this Symbol (QuoteSet) that follow in this message.
→	→	**299**	**QuoteEntryID**	N	Uniquely identifies the quote as part of a QuoteSet. First field in repeating group. Required if NoQuoteEntries > 0.
→	→	**component block <Instrument>**		N	Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages"
→	**555**	**NoLegs**		N	Number of legs Identifies a Multi-leg Execution if present and non-zero.
→	→	**component block <InstrumentLeg>**		N	Must be provided if Number of legs > 0
→	→	**132**	**BidPx**	N	If F/X quote, should be the "all-in" rate (spot rate adjusted for forward points). Note that either BidPx, OfferPx or both must be specified.
→	→	**133**	**OfferPx**	N	If F/X quote, should be the "all-in" rate (spot rate adjusted for forward points). Note that either BidPx, OfferPx or both must be specified.
→	→	**134**	**BidSize**	N	
→	→	**135**	**OfferSize**	N	
→	→	**62**	**ValidUntilTime**	N	
→	→	**188**	**BidSpotRate**	N	May be applicable for F/X quotes
→	→	**190**	**OfferSpotRate**	N	May be applicable for F/X quotes
→	→	**189**	**BidForwardPoints**	N	May be applicable for F/X quotes
→	→	**191**	**OfferForwardPoints**	N	May be applicable for F/X quotes
→	→	**631**	**MidPx**	N	

→	→	**632**	**BidYield**	N	
→	→	**633**	**MidYield**	N	
→	→	**634**	**OfferYield**	N	
→	→	**60**	**TransactTime**	N	
→	→	**336**	**TradingSessionID**	N	
→	→	**625**	**TradingSessionSubID**	N	
→	→	**64**	**SettlDate**	N	Can be used with forex quotes to specify a specific "value date"
→	→	**40**	**OrdType**	N	Can be used to specify the type of order the quote is for
→	→	**193**	**SettlDate2**	N	Can be used with OrdType = "Forex - Swap" to specify the "value date" for the future portion of a F/X swap.
→	→	**192**	**OrderQty2**	N	Can be used with OrdType = "Forex - Swap" to specify the order quantity for the future portion of a F/X swap.
→	→	**642**	**BidForwardPoints2**	N	Bid F/X forward points of the future portion of a F/X swap quote added to spot rate. May be a negative value
→	→	**643**	**OfferForwardPoints2**	N	Offer F/X forward points of the future portion of a F/X swap quote added to spot rate. May be a negative value
→	→	**15**	**Currency**	N	Can be used to specify the currency of the quoted price.
→	→	**368**	**QuoteEntryRejectReason**	N	Reason Quote Entry was rejected.
	StandardTrailer			Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details
Refer to FIXML element MassQuotAck

Mass Quote Message Scenarios

Unsolicited quote(s) no response requested

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

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

Unsolicited quote(s) negative response only requested

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

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

Unsolicited quote(s) full response requested

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

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

Cancel All Quotes

The First Party asks the second party to cancel all quotes. Quotes with a quote status are sent in response to the Cancel All Quotes message.

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

Use of other Quote Messages in Mass Quoting

Once the Mass Quote message is submitted to a market and after the initial Mass Quote Acknowledgement - the Quote Entries are treated as separate quotes. Report of Quote Status should be done using the Quote Status Request and Quote messages. Fills are reported for each QuoteEntry using the Execution Report.

Reporting Quote Status back to Mass Quote Issuer

Markets should report the status of quotes back to the quote issuer when the state of one of the quotes in a Mass Quote changes. Quote Status Report messages should be issued for each change in state of a quote entry. The QuoteID of the original Mass Quote message should be used as the QuoteID on the Quote Status Report. It is acceptable to append the QuoteSetID and QuoteEntryID to indicate the specific quote in the Mass Quote message referred to in the Quote Status Report if this information is maintained by the market. NOTE: The Quote Message should not be used to report trades. Only the Execution Report should be used to report fills against a tradeable or restricted tradeable quote.

Market maker or specialist		Market

Mass Quote Valid tradeable or restricted tradeable quote sent into market – either unsolicited or in reply to a Quote Request from the market.	→	Accepts Mass Quote and applies to the market
Accepts Quote and updates trading system based upon status reported by market	←	Based upon market rules or based upon the QuoteResponseLevel requested by Quote Issuer the market will send Mass Quote Acknowledgement message back to the quote issuer to report quote status in the QuoteStatus field.
Updates trading system with quote status	←	Quote messages are sent back unsolicited as the quote state changes. The QuoteEntryID should be used as the QuoteID.
Updates trading system with execution report	←	If a trade (fill) occurs against a tradeable or restricted tradeable quote an Execution Report (ExecType=Trade) is sent to the quote issuer.

Querying for Mass Quote Status

If the issuer of a Mass Quote queries the current status of the quote the market should reply with a sequence of individual quote messages with status. This is recommended to eliminate the need for markets to store QuoteSetIds and QuoteEntryIds that were provided as part of the Mass Quote message. Also, as quote status is very dynamic data – sending quote status on securities as soon as it is available – instead of combining it into a single message – will provide more timely information to the quote issuer. The use of a Quote Status Request for a Mass Quote is provided as a method of recovery for market maker trading systems – due to the volume of information that can be generated and the short lived nature of quote status – this usage is not recommended for normal processing.

Market maker or specialist		**Market**
Quote Status Request Contains the QuoteId of a previously submitted Mass Quote.	→	Accepts Quote Status Request
Accepts Quote and updates trading system.	←	Sends Quote messages with the QuoteStatus field, bid and ask prices and quantities for each quote belonging to the request issuer that meet the criteria in the request. If there is a current quote in the market – the Quote in response to a Quote Status Request should be sent with a QuoteStatus of "Query". The Quote message can also contain a QuoteStatus of "Quote Not Found" if no quote currently exists.

CATEGORY: MARKET DATA

Market Data Request

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.

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

If the message is used for disseminating imbalance information, conventions are as follows:

- MDEntrySize represents the size of the imbalance and is always a positive integer.
- A TradeCondition of either P or Q is required to indicate the side of the imbalance.
- Markets may wish to indicate the presence of an imbalance but not the actual size. In this case, MDEntrySize need not be specified.

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

Types of Market Data Requests

1. A market data feed may consist of both Market Data Snapshot Full Refresh messages and Market Data Incremental Refresh messages.
2. The Market Data Request message is used to request a static book snapshot or subscribe to a stream of snapshots and updates.
3. Market Data Snapshot Full Refresh should be used to provide a snapshot of the market when Snapshot is requested using SubscriptionRequestType (263). Use of Market Data Incremental Refresh is being discouraged for this purpose.
4. Market Data Snapshot Full Refresh will be used to provide initial snapshot when Snapshot + Updates are requested using SubscriptionRequestType (263)
5. The Market Data Request scenarios that will be supported are as follows:

Customer Requests	Subscription RequestType (263)	MDUpdateType(265)	Response Messages
Requests state of the book and	0=Snapshot	Not Provided	Market Data

receives one and only one snapshot for each request (i.e. customer only wants single snapshot of prices)		(customer is not requesting a subscription)	Snapshot/Full Refresh message (only one message is sent)
Requests state of the book + updates and specifies that only Full Refresh Message is used (i.e. full refresh update of data is to be sent)	1 = Snapshot + Updates	0 = Full Refresh	Market Data Snapshot/Full Refresh messages only
Requests state of the book + updates and specifies that updates are to be sent using Incremental Refresh Message (i.e. incremental updates on data is to be sent)	1 = Snapshot + Updates	1 = Incremental Refresh	Market Data Snapshot/Full Refresh message with updates provided using Market Data Incremental Refresh messages

Indicating an Empty Book

1. An empty book contains no bids or asks and indicates that the market has no open orders in a given instrument. This can also be referred to as a “null” book.
2. When this occurs in a scenario in which the Snapshot Full Refresh Message is being used to provide a static snapshot or snapshot + updates then a special MDEntryType (tag 269) of “J” (Null Market) should be used.
3. The Snapshot Full Refresh Message should contain a single MDEntry with MDEntryType (269) = J specified. MDEntryPrice (270) = 0 and MDEntrySize (271) = 0 may also be provided but are not required. Other tags may be specified as well in order to convey the time and conditions under which the market generated a null book.

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

See VOLUME 7 - PRODUCT: FOREIGN EXCHANGE section for more detailed usage notes specific to Foreign Exchange.

The Market Data Request message format is as follows:

Market Data Request

Tag	*FieldName*	*Req'd*	*Comments*
	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	SubcriptionRequestType indicates to the other party what type of response is expected. A snapshot request only asks for current information. A subscribe request asks for updates as the status changes. Unsubscribe will cancel

				any future update messages from the counter party.
264	MarketDepth		Y	
265	MDUpdateType		N	Required if SubscriptionRequestType = Snapshot + Updates (1).
266	AggregatedBook		N	
286	OpenCloseSettlFlag		N	Can be used to clarify a request if MDEntryType = Opening Price(4), Closing Price(5), or Settlement Price(6).
546	Scope		N	Defines the scope(s) of the request
547	MDImplicitDelete		N	Can be used when MarketDepth >= 2 and MDUpdateType = Incremental Refresh(1).
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.
146	NoRelatedSym		Y	Number of symbols (instruments) requested.
→	**component block <Instrument>**		Y	Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages"
→	**711**	**NoUnderlyings**	N	Number of underlyings
→	→	**component block <UnderlyingInstrument>**	N	Must be provided if Number of underlyings > 0
→	**555**	**NoLegs**	N	Number of legs Identifies a Multi-leg Execution if present and non-zero.
→	→	**component block <InstrumentLeg>**	N	Must be provided if Number of legs > 0
→	**15**	**Currency**	N	
→	**537**	**QuoteType**	N	
→	**63**	**SettlType**	N	
→	**64**	**SettlDate**	N	
→	**271**	**MDEntrySize**	N	Quantity or volume represented by the Market Data Entry. In the context of the Market Data Request this allows the Initiator to indicate the quantity of the market data request. Specific to FX this field indicates the ceiling amount the customer is seeking prices for.
386	NoTradingSessions		N	Specifies the number of repeating TradingSessionIDs
→	**336**	**TradingSessionID**	N	Required if NoTradingSessions is > 0.
→	**625**	**TradingSessionSubID**	N	

815	ApplQueueAction	N	Action to take if application level queuing exists
812	ApplQueueMax	N	Maximum application queue depth that must be exceeded before queuing action is taken.
1070	MDQuoteType	N	
	StandardTrailer	Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details
MktDataReq

Market Data - Snapshot / Full Refresh

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

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

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

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

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

See VOLUME 7 - PRODUCT: FOREIGN EXCHANGE section for more detailed usage notes specific to Foreign Exchange.

Market Data - Snapshot / Full Refresh

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = W
963	MDReportID	N	Unique indentifier for Market Data Report
715	ClearingBusinessDate	N	
1021	MDBookType	N	Describes the type of book for which the feed is intended. Can be used when multiple feeds are provided over the same connection
1022	MDFeedType	N	Describes a class of service for a given data feed, ie Regular and Market Maker
75	TradeDate	N	Used to specify the trading date for which a set of market data applies
262	MDReqID	N	Conditionally required if this message is in response to a Market Data Request.
component block <Instrument>		Y	Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages"

711	NoUnderlyings		N	Number of underlyings
→	**component block <UnderlyingInstrument>**		N	Must be provided if Number of underlyings > 0
555	NoLegs		N	Number of legs Identifies a Multi-leg Execution if present and non-zero.
→	**component block <InstrumentLeg>**		N	Must be provided if Number of legs > 0
291	FinancialStatus		N	
292	CorporateAction		N	
451	NetChgPrevDay		N	
268	NoMDEntries		Y	Number of entries following.
→	**269**	**MDEntryType**	Y	Must be the first field in this repeating group.
→	**278**	**MDEntryID**	N	Conditionally required when maintaining an order-depth book, that is, when AggregatedBook (266) is "N". allows subsequent Incremental changes to be applied using MDEntryID.
→	**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"
→	**40**	**OrdType**	N	Used to support market mechanism type; limit order, market order, committed principal order
→	**15**	**Currency**	N	Can be used to specify the currency of the quoted 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"
→	**272**	**MDEntryDate**	N	
→	**273**	**MDEntryTime**	N	
→	**274**	**TickDirection**	N	
→	**275**	**MDMkt**	N	Market posting quote / trade. Valid values: See Volume 6: Appendix 6-C
→	**336**	**TradingSessionID**	N	
→	**625**	**TradingSessionSubID**	N	
→	**276**	**QuoteCondition**	N	Space-delimited list of conditions describing a quote.
→	**277**	**TradeCondition**	N	Space-delimited list of conditions describing a trade
→	**282**	**MDEntryOriginator**	N	

→	**283**	**LocationID**	N	
→	**284**	**DeskID**	N	
→	**286**	**OpenCloseSettlFlag**	N	Used if MDEntryType = Opening Price(4), Closing Price(5), or Settlement Price(6).
→	**59**	**TimeInForce**	N	For optional use when this Bid or Offer represents an order
→	**432**	**ExpireDate**	N	For optional use when this Bid or Offer represents an order. ExpireDate and ExpireTime cannot both be specified in one Market Data Entry.
→	**126**	**ExpireTime**	N	For optional use when this Bid or Offer represents an order. ExpireDate and ExpireTime cannot both be specified in one Market Data Entry.
→	**110**	**MinQty**	N	For optional use when this Bid or Offer represents an order
→	**18**	**ExecInst**	N	Can contain multiple instructions, space delimited.
→	**287**	**SellerDays**	N	
→	**37**	**OrderID**	N	For optional use when this Bid, Offer, or Trade represents an order
→	**198**	**SecondaryOrderID**	N	For optional use to support Hit/Take (selecting a specific order from the feed) without disclosing a private order id.
→	**299**	**QuoteEntryID**	N	For optional use when this Bid, Offer, or Trade represents a quote
→	**288**	**MDEntryBuyer**	N	For optional use in reporting Trades
→	**289**	**MDEntrySeller**	N	For optional use in reporting Trades
→	**346**	**NumberOfOrders**	N	In an Aggregated Book, used to show how many individual orders make up an MDEntry
→	**290**	**MDEntryPositionNo**	N	Display position of a bid or offer, numbered from most competitive to least competitive, per market side, beginning with 1
→	**546**	**Scope**	N	
→	**811**	**PriceDelta**	N	
→	**58**	**Text**	N	Text to describe the Market Data Entry. Part of repeating group.
→	**354**	**EncodedTextLen**	N	Must be set if EncodedText field is specified and must immediately precede it.
→	**355**	**EncodedText**	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
→	**1023**	**MDPriceLevel**	N	Display position of a bid or offer, numbered from most competitive to least competitive, per market side, beginning with 1

→	**528**	**OrderCapacity**	N	Designates the capacity of the firm placing the order
→	**1024**	**MDOriginType**	N	
→	**332**	**HighPx**	N	Used to report high price in association with trade, bid or ask rather than a separate entity
→	**333**	**LowPx**	N	Used to report low price in association with trade, bid or ask rather than a separate entitty
→	**1020**	**TradeVolume**	N	Used to report trade volume in association with trade, bid or ask rather than a separate entity
→	**63**	**SettlType**	N	
→	**64**	**SettlDate**	N	
→	**1070**	**MDQuoteType**	N	
→	**83**	**RptSeq**	N	Used to identify the sequence number within a feed type
→	**1048**	**DealingCapacity**	N	Identifies role of dealer; Agent, Principal, RisklessPrincipal
→	**1026**	**MDEntrySpotRate**	N	
→	**1027**	**MDEntryForwardPoints**	N	
→	**component block <Parties>**		N	
813	ApplQueueDepth		N	Depth of application messages queued for transmission as of delivery of this message
814	ApplQueueResolution		N	Action taken to resolve application queuing
215	NoRoutingIDs		N	Required if any RoutingType and RoutingIDs are specified. Indicates the number within repeating group.
→	**216**	**RoutingType**	N	Indicates type of RoutingID. Required if NoRoutingIDs is > 0.
→	**217**	**RoutingID**	N	Identifies routing destination. Required if NoRoutingIDs is > 0.
	StandardTrailer		Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details

Refer to the FIXML element MktDataSnpFullRefresh

Market Data - Incremental Refresh

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

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

The Market Data message for incremental updates may contain any combination of new, changed, or deleted Market Data Entries, for any combination of instruments, with any combination of trades, 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.

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

Several techniques are employed to conserve bandwidth:

- An instrument only needs to be identified when a Market Data Entry is first created.
- In cases where the identification of an instrument is long, the sender has the option of referring to a previous active Market Data Entry of the same instrument instead of duplicating the information.
- A new Market Data Entry will default to the same instrument of the previous Market Data Entry in the same Market Data message if neither Symbol nor MDEntryRefID are specified.
- In the case of a change in a Market Data Entry, only the fields changing need to be sent as part of the change to the Market Data Entry; for example, a change of the MDEntrySize but not the MDEntryPx or other attributes of the Market Data Entry only requires listing the MDEntrySize field, in addition to MDUpdateAction and MDEntryID if used in the original Market Data Entry
- When creating a new Market Data Entry with a future or option instrument similar to the instrument in the previous Market Data Entry in the same FIX message, one may send just symbol identification fields that have changed, such as MaturityMonthYear, MaturityDay, StrikePrice, OptAttribute, and SecurityExchange.
- MDEntryID can be reused within the same day after it is deleted. This is helpful for distributing order books because an order that is suspended and then reinstated can have its MDEntryID deleted upon suspension and later reused, with MDUpdateAction = New(0) upon reinstatement, thus avoiding having to re-map the MDEntryID.

Market Data - Incremental Refresh

Tag		*FieldName*	*Req'd*	*Comments*
		StandardHeader	Y	MsgType = X
1021		MDBookType	N	Describes the type of book for which the feed is intended. Can be used when multiple feeds are provided over the same connection
1022		MDFeedType	N	Describes a class of service for a given data feed, ie Regular and Market Maker
75		TradeDate	N	Used to specify the trading date for which a set of market data applies
262		MDReqID	N	Conditionally required if this message is in response to a Market Data Request.
268		NoMDEntries	Y	Number of entries following.
→	**279**	**MDUpdateAction**	Y	Must be first field in this repeating group.
→	**285**	**DeleteReason**	N	If MDUpdateAction = Delete(2), can be used to specify a reason for the deletion.
→	**269**	**MDEntryType**	N	Conditionally required if MDUpdateAction = New(0). Cannot be changed.
→	**278**	**MDEntryID**	N	If specified, must be unique among currently active entries if MDUpdateAction = New (0), must be the same as a previous MDEntryID if MDUpdateAction = Delete (2), and must be the same as a previous MDEntryID if MDUpdateAction = Change (1) and MDEntryRefID is not specified, or must be unique among currently active entries if MDUpdateAction = Change(1) and MDEntryRefID is specified..
→	**280**	**MDEntryRefID**	N	If MDUpdateAction = New(0), for the first Market Data Entry in a message, either this field or a Symbol must be specified. If MDUpdateAction = Change(1), this must refer to a previous MDEntryID.
→		**component block <Instrument>**	N	Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages" Either Symbol (the instrument component block) or MDEntryRefID must be specified if MDUpdateAction = New(0) for the first Market Data Entry in a message. For subsequent Market Data Entries where MDUpdateAction = New(0), the default is the instrument used in the previous Market Data Entry if neither Symbol nor MDEntryRefID are specified, or in the case of options and futures, the previous instrument with changes specified in MaturityMonthYear, MaturityDay, StrikePrice, OptAttribute, and SecurityExchange. May not be changed.

→	**711**	**NoUnderlyings**	N	Number of underlyings
→	→	**component block <UnderlyingInstrument>**	N	Must be provided if Number of underlyings > 0
→	**555**	**NoLegs**	N	Number of legs Identifies a Multi-leg Execution if present and non-zero.
→	→	**component block <InstrumentLeg>**	N	Must be provided if Number of legs > 0
→	**291**	**FinancialStatus**	N	
→	**292**	**CorporateAction**	N	
→	**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"
→	**40**	**OrdType**	N	Used to support market mechanism type; limit order, market order, committed principal order
→	**15**	**Currency**	N	Can be used to specify the currency of the quoted 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"
→	**272**	**MDEntryDate**	N	
→	**273**	**MDEntryTime**	N	
→	**274**	**TickDirection**	N	
→	**275**	**MDMkt**	N	Market posting quote / trade. Valid values: See Volume 6: Appendix 6-C
→	**336**	**TradingSessionID**	N	
→	**625**	**TradingSessionSubID**	N	
→	**276**	**QuoteCondition**	N	Space-delimited list of conditions describing a quote.
→	**277**	**TradeCondition**	N	Space-delimited list of conditions describing a trade
→	**282**	**MDEntryOriginator**	N	
→	**283**	**LocationID**	N	
→	**284**	**DeskID**	N	
→	**286**	**OpenCloseSettlFlag**	N	Used if MDEntryType = Opening Price(4), Closing Price(5), or Settlement Price(6).
→	**59**	**TimeInForce**	N	For optional use when this Bid or Offer represents an

				order
→	**432**	**ExpireDate**	N	For optional use when this Bid or Offer represents an order. ExpireDate and ExpireTime cannot both be specified in one Market Data Entry.
→	**126**	**ExpireTime**	N	For optional use when this Bid or Offer represents an order. ExpireDate and ExpireTime cannot both be specified in one Market Data Entry.
→	**110**	**MinQty**	N	For optional use when this Bid or Offer represents an order
→	**18**	**ExecInst**	N	Can contain multiple instructions, space delimited.
→	**287**	**SellerDays**	N	
→	**37**	**OrderID**	N	For optional use when this Bid, Offer, or Trade represents an order
→	**198**	**SecondaryOrderID**	N	For optional use to support Hit/Take (selecting a specific order from the feed) without disclosing a private order id.
→	**299**	**QuoteEntryID**	N	For optional use when this Bid, Offer, or Trade represents a quote
→	**288**	**MDEntryBuyer**	N	For optional use in reporting Trades
→	**289**	**MDEntrySeller**	N	For optional use in reporting Trades
→	**346**	**NumberOfOrders**	N	In an Aggregated Book, used to show how many individual orders make up an MDEntry
→	**290**	**MDEntryPositionNo**	N	Display position of a bid or offer, numbered from most competitive to least competitive, per market side, beginning with 1
→	**546**	**Scope**	N	
→	**811**	**PriceDelta**	N	
→	**451**	**NetChgPrevDay**	N	
→	**58**	**Text**	N	Text to describe the Market Data Entry. Part of repeating group.
→	**354**	**EncodedTextLen**	N	Must be set if EncodedText field is specified and must immediately precede it.
→	**355**	**EncodedText**	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
→	**1023**	**MDPriceLevel**	N	
→	**528**	**OrderCapacity**	N	
→	**1024**	**MDOriginType**	N	
→	**332**	**HighPx**	N	
→	**333**	**LowPx**	N	
→	**1020**	**TradeVolume**	N	

→	**63**	**SettlType**	N	
→	**64**	**SettlDate**	N	Indicates date on which instrument will settle
→	**1070**	**MDQuoteType**	N	
→	**83**	**RptSeq**	N	Allows sequence number to be specified within a feed type
→	**1048**	**DealingCapacity**	N	Identifies role of dealer; Agent, Principal, RisklessPrincipal
→	**1026**	**MDEntrySpotRate**	N	
→	**1027**	**MDEntryForwardPoints**	N	
→	**component block <Parties>**		N	
813	ApplQueueDepth		N	Depth of application messages queued for transmission as of delivery of this message
814	ApplQueueResolution		N	Action taken to resolve application queuing
215	NoRoutingIDs		N	Required if any RoutingType and RoutingIDs are specified. Indicates the number within repeating group.
→	**216**	**RoutingType**	N	Indicates type of RoutingID. Required if NoRoutingIDs is > 0.
→	**217**	**RoutingID**	N	Identifies routing destination. Required if NoRoutingIDs is > 0.
	StandardTrailer		Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details

Refer to the FIXML element MktDataIncRefresh

Market Data Request Reject

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

The market data request reject message format is as follows:

Market Data Request Reject

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = Y
262	MDReqID	Y	Must refer to the MDReqID of the request.
281	MDReqRejReason	N	
816	NoAltMDSource	N	
→	**817** **AltMDSourceID**	N	Alternative Market Data Source
58	Text	N	
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
	StandardTrailer	Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details
Refer to FIXML element MktDataReqRej

CATEGORY: SECURITY AND TRADING SESSION DEFINITION/STATUS

Security Definition Request

The Security Definition Request message is used for the following:

1. Request a specific Security to be traded with the second party. The request security can be defined as a multileg security made up of one or more instrument legs.

Subscription for security status can be optionally specified by including the SubscriptionRequestType[263] field on the message.

See "Security Definition, Security Status, and Trading Session Message Scenarios"

Security Definition Request

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = c (lowercase)
320	SecurityReqID	Y	
321	SecurityRequestType	Y	
component block <Instrument>		N	Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages" of the requested Security
component block <InstrumentExtension>		N	Insert here the set of "InstrumentExtension" fields defined in "Common Components of Application Messages"
711	NoUnderlyings	N	Number of underlyings
→	**component block <UnderlyingInstrument>**	N	Must be provided if Number of underlyings > 0
15	Currency	N	
58	Text	N	Comment, instructions, or other identifying information.
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
336	TradingSessionID	N	Optional Trading Session Identifier to specify a particular trading session for which you want to obtain a list of securities that are tradeable.
625	TradingSessionSubID	N	
555	NoLegs	N	Number of legs

			Identifies a Multi-leg Execution if present and non-zero.
→	**component block <InstrumentLeg>**	N	Must be provided if Number of legs > 0
→	**1017 LegOptionRatio**	N	
→	**566 LegPrice**	N	
827	ExpirationCycle	N	
263	SubscriptionRequestType	N	Subscribe or unsubscribe for security status to security specified in request.
	StandardTrailer	Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details

Refer to the FIXML element SecDefReq

Security Definition

The Security Definition message is used for the following:

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

Security Definition

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = d (lowercase)
964	SecurityReportID	N	Identifier for Security Definition message
715	ClearingBusinessDate	N	
320	SecurityReqID	N	
322	SecurityResponseID	N	Identifier for the Security Definition message
323	SecurityResponseType	N	Response to the Security Definition Request
component block <Instrument>		N	Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages" of the requested Security
component block <InstrumentExtension>		N	Insert here the set of "InstrumentExtension" fields defined in "Common Components of Application Messages"
711	NoUnderlyings	N	Number of underlyings
→	**component block <UnderlyingInstrument>**	N	Must be provided if Number of underlyings > 0
15	Currency	N	
336	TradingSessionID	N	
625	TradingSessionSubID	N	
58	Text	N	Comment, instructions, or other identifying information.
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
555	NoLegs	N	Number of legs Identifies a Multi-leg Execution if present and non-zero.
→	**component block <InstrumentLeg>**	N	Must be provided if Number of legs > 0

→	**1017**	**LegOptionRatio**	N	LegOptionRatio is provided on Covering leg to create a delta neutral spread. In Listed Derivatives, LegDelta LegOptionRatio is multiplied by LegOptionRatio and OrderQty to determine the covering quantity
→	**566**	**LegPrice**	N	Specifies the price of a Leg
827	ExpirationCycle		N	
561	RoundLot		N	
562	MinTradeVol		N	
	StandardTrailer		Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details
Refer to FIXML element SecDef

Security Definition Update Report

This message is used for reporting updates to a Product Security Masterfile. Updates could be the result of corporate actions or other business events. Updates may include additions, modifications or deletions.

Security Defintion Update Report

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = BP
964	SecurityReportID	N	Identifier for the Security Definition Update message in a bulk transfer environment (No Request/Response)
320	SecurityReqID	N	
322	SecurityResponseID	N	Identifier for the Security Definition message.
323	SecurityResponseType	N	Response to the Security Definition Request.
715	ClearingBusinessDate	N	
980	SecurityUpdateAction	N	Specifies New (0), Cancel (1) or Replace (2)
292	CorporateAction	N	Identifies the type of Corporate Action
component block <Instrument>		N	
component block <UnderlyingInstrument>		N	
15	Currency	N	
336	TradingSessionID	N	
625	TradingSessionSubID	N	
58	Text	N	Comment, instructions, or other identifying information.
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
555	NoLegs	N	Number of legs Identifies a Multi-leg Execution if present and non-zero.
→	**component block <InstrumentLeg>**	N	Must be provided if Number of legs > 0
827	ExpirationCycle	N	
561	RoundLot	N	
562	MinTradeVol	N	
	StandardTrailer	Y	

Security Type Request

The Security Type Request message is used to return a list of security types available from a counterparty or market,

The request can include a specific TradingSessionID for which Security Types should be returned.

Security Type Request

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = v (lowercase V)
320	SecurityReqID	Y	
58	Text	N	Comment, instructions, or other identifying information.
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
336	TradingSessionID	N	Optional Trading Session Identifier to specify a particular trading session for which you want to obtain a list of securities that are tradeable.
625	TradingSessionSubID	N	
460	Product	N	Used to qualify which security types are returned
167	SecurityType	N	Used to qualify which security type is returned
762	SecuritySubType	N	Used to qualify which security types are returned
	StandardTrailer	Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details

Refer to FIXML element SecTypReq

Security Types

The Security Type message is used to return a list of security types available from a counterparty or market.

Security Types

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = w (lowercase W)
320	SecurityReqID	Y	
322	SecurityResponseID	Y	Identifier for the security response message
323	SecurityResponseType	Y	The result of the security request identified by SecurityReqID
557	TotNoSecurityTypes	N	Indicates total number of security types in the event that multiple Security Type messages are used to return results
893	LastFragment	N	Indicates whether this is the last fragment in a sequence of message fragments. Only required where message has been fragmented.
558	NoSecurityTypes	N	
→	**167 SecurityType**	N	Required if NoSecurityTypes > 0
→	**762 SecuritySubType**	N	
→	**460 Product**	N	
→	**461 CFICode**	N	
58	Text	N	Comment, instructions, or other identifying information.
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
336	TradingSessionID	N	Optional Trading Session Identifier to specify a particular trading session for which you want to obtain a list of securities that are tradeable.
625	TradingSessionSubID	N	
263	SubscriptionRequestType	N	Subscribe or unsubscribe for security status to security specified in request.
	StandardTrailer	Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details

Refer to FIXML element SecTyps

Security List Request

The Security List Request message is used to return a list of securities from the counterparty that match criteria provided on the request

Subscription for security status can be optionally specified by including the SubscriptionRequestType[263] field on the message.

SecurityListRequestType[559] specifies the criteria of the request:
0 - Symbol
1 - SecurityType and/or CFICode
2 - Product
3 - TradingSessionID
4 - All Securities

Security List Request

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = x (lowercase X)
320	SecurityReqID	Y	
559	SecurityListRequestType	Y	Type of Security List Request being made
component block <Instrument>		N	Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages" of the requested Security
component block <InstrumentExtension>		N	Insert here the set of "InstrumentExtension" fields defined in "Common Components of Application Messages"
component block <FinancingDetails>		N	Insert here the set of "FinancingDetails" fields defined in "Common Components of Application Messages"
711	NoUnderlyings	N	Number of underlyings
→	**component block <UnderlyingInstrument>**	N	Must be provided if Number of underlyings > 0
555	NoLegs	N	Number of legs Identifies a Multi-leg Execution if present and non-zero.
→	**component block <InstrumentLeg>**	N	Must be provided if Number of legs > 0
15	Currency	N	
58	Text	N	Comment, instructions, or other identifying information.
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
336	TradingSessionID	N	Optional Trading Session Identifier to specify a particular

			trading session for which you want to obtain a list of securities that are tradeable.
625	TradingSessionSubID	N	
263	SubscriptionRequestType	N	Subscribe or unsubscribe for security status to security specified in request.
	StandardTrailer	Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details
Refer to FIXML element SecListReq

Security List

The Security List message is used to return a list of securities that matches the criteria specified in a Security List Request.

Security List

Tag	*FieldName*		*Req'd*	*Comments*
	StandardHeader		Y	MsgType = y (lowercase Y)
964	SecurityReportID		N	
715	ClearingBusinessDate		N	
320	SecurityReqID		N	
322	SecurityResponseID		N	Identifier for the Security List message
560	SecurityRequestResult		N	Result of the Security Request identified by the SecurityReqID
393	TotNoRelatedSym		N	Used to indicate the total number of securities being returned for this request. Used in the event that message fragmentation is required.
893	LastFragment		N	Indicates whether this is the last fragment in a sequence of message fragments. Only required where message has been fragmented.
146	NoRelatedSym		N	Specifies the number of repeating symbols (instruments) specified
→	**component block <Instrument>**		N	Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages" of the requested Security
→	**component block <InstrumentExtension>**		N	Insert here the set of "InstrumentExtension" fields defined in "Common Components of Application Messages"
→	**component block <FinancingDetails>**		N	Insert here the set of "FinancingDetails" fields defined in "Common Components of Application Messages"
→	**711**	**NoUnderlyings**	N	Number of underlyings
→	→	**component block <UnderlyingInstrument>**	N	Must be provided if Number of underlyings > 0
→	**15**	**Currency**	N	
→	**component block <Stipulations>**		N	Insert here the set of "Stipulations" fields defined in "Common Components of Application Messages"
→	**555**	**NoLegs**	N	Number of legs that make up the Security
→	→	**component block <InstrumentLeg>**	N	Insert here the set of "Instrument Legs" (leg symbology) fields defined in "Common Components of Application Messages"

					Required if NoLegs > 0
→	→	**690**	**LegSwapType**	N	
→	→	**587**	**LegSettlType**	N	
→	→	**component block <LegStipulations>**		N	Insert here the set of "LegStipulations" (leg symbology) fields defined in "Common Components of Application Messages" Required if NoLegs > 0
→	→	**component block <LegBenchmarkCurveData>**		N	Insert here the set of "LegBenchmarkCurveData" (leg symbology) fields defined in "Common Components of Application Messages" Required if NoLegs > 0
→	**component block <SpreadOrBenchmarkCurveData>**			N	Insert here the set of "SpreadOrBenchmarkCurveData" fields defined in "Common Components of Application Messages"
→	**component block <YieldData>**			N	Insert here the set of "YieldData" fields defined in "Common Components of Application Messages"
→	**561**	**RoundLot**		N	
→	**562**	**MinTradeVol**		N	
→	**336**	**TradingSessionID**		N	
→	**625**	**TradingSessionSubID**		N	
→	**827**	**ExpirationCycle**		N	
→	**58**	**Text**		N	Comment, instructions, or other identifying information.
→	**354**	**EncodedTextLen**		N	Must be set if EncodedText field is specified and must immediately precede it.
→	**355**	**EncodedText**		N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
	StandardTrailer			Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details

Refer to FIXML element SecList

Security List Update Report

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.

Security List Update Report

Tag	*FieldName*		*Req'd*	*Comments*
	StandardHeader		Y	MsgType = BK
964	SecurityReportID		N	Identifier for the Security List Update message in a bulk transfer environment (No Request/Response)
320	SecurityReqID		N	
322	SecurityResponseID		N	Identifier for the Security List message.
560	SecurityRequestResult		N	Result of the Security Request identified by the SecurityReqID.
393	TotNoRelatedSym		N	Used to indicate the total number of securities being returned for this request. Used in the event that message fragmentation is required.
715	ClearingBusinessDate		N	
980	SecurityUpdateAction		N	Specifies New (0), Cancel (1) or Replace (2)
292	CorporateAction		N	Identifies the type of Corporate Action that triggered the update
893	LastFragment		N	Indicates whether this is the last fragment in a sequence of message fragments. Only required where message has been fragmented.
146	NoRelatedSym		N	Specifies the number of repeating symbols (instruments) specified
→	**component block <Instrument>**		N	Insert here the set of "Instrument" (symbology) fields defined in "common components of application messages" of the requested Security
→	**component block <InstrumentExtension>**		N	Insert here the set of " InstrumentExtension " fields defined in " COMMON COMPONENTS OF APPLICATION MESSAGES "
→	**component block <FinancingDetails>**		N	Insert here the set of " FinancingDetails " fields defined in " COMMON COMPONENTS OF APPLICATION MESSAGES "
→	**component block <UnderlyingInstrument>**		N	
→	**15**	**Currency**	N	
→	**component block <Stipulations>**		N	
→	**555**	**NoLegs**	N	Number of legs that make up the Security
→	→	**component block**	N	Insert here the set of "Instrument Legs" (leg symbology)

		<InstrumentLeg>			fields defined in "common components of application messages" Required if NoLegs > 0
→	→	**690**	**LegSwapType**	N	
→	→	**587**	**LegSettlType**	N	
→	→	**component block <LegStipulations>**		N	Insert here the set of "LegStipulations" (leg symbology) fields defined in "common components of application messages" Required if NoLegs > 0
→	→	**component block <LegBenchmarkCurveData>**		N	Insert here the set of "LegBenchmarkCurveData" (leg symbology) fields defined in "common components of application messages" Required if NoLegs > 0
→	**component block <SpreadOrBenchmarkCurveData>**			N	Insert here the set of " SpreadOrBenchmarkCurveData " fields defined in " COMMON COMPONENTS OF APPLICATION MESSAGES "
→	**component block <YieldData>**			N	Insert here the set of " YieldData " fields defined in " COMMON COMPONENTS OF APPLICATION MESSAGES "
→	**561**	**RoundLot**		N	
→	**562**	**MinTradeVol**		N	
→	**336**	**TradingSessionID**		N	
→	**625**	**TradingSessionSubID**		N	
→	**827**	**ExpirationCycle**		N	
→	**58**	**Text**		N	Comment, instructions, or other identifying information.
→	**354**	**EncodedTextLen**		N	Must be set if EncodedText field is specified and must immediately precede it.
→	**355**	**EncodedText**		N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
	StandardTrailer			Y	

Derivative Security List Request

The Derivative Security List Request message is used to return a list of securities from the counterparty that match criteria provided on the request

Subscription for security status can be optionally specified by including the SubscriptionRequestType[263] field on the message.

SecurityListRequestType[559] specifies the criteria of the request:
0 - Symbol
1 - SecurityType and/or CFICode
2 - Product
3 - TradingSessionID
4 - All Securities

Derivative Security List Request

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = z (lowercase Z)
320	SecurityReqID	Y	
559	SecurityListRequestType	Y	
component block <UnderlyingInstrument>		N	Specifies the underlying instrument
762	SecuritySubType	N	
15	Currency	N	
58	Text	N	Comment, instructions, or other identifying information.
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
336	TradingSessionID	N	Optional Trading Session Identifier to specify a particular trading session for which you want to obtain a list of securities that are tradeable.
625	TradingSessionSubID	N	
263	SubscriptionRequestType	N	Subscribe or unsubscribe for security status to security specified in request.
	StandardTrailer	Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details
Refer to FIXML element DerivSecListReq

Derivative Security List

The Derivative Security List message is used to return a list of securities that matches the criteria specified in a Derivative Security List Request.

Derivative Security List

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = AA (2 A's)
320	SecurityReqID	Y	
322	SecurityResponseID	Y	Identifier for the Derivative Security List message
560	SecurityRequestResult	Y	Result of the Security Request identified by SecurityReqID
component block <UnderlyingInstrument>		N	Underlying security for which derivatives are being returned
393	TotNoRelatedSym	N	Used to indicate the total number of securities being returned for this request. Used in the event that message fragmentation is required.
893	LastFragment	N	Indicates whether this is the last fragment in a sequence of message fragments. Only required where message has been fragmented.
146	NoRelatedSym	N	Specifies the number of repeating symbols (instruments) specified
→	**component block <Instrument>**	N	Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages" of the requested Security
→	**15** **Currency**	N	
→	**827** **ExpirationCycle**	N	
→	**component block <InstrumentExtension>**	N	Insert here the set of "InstrumentExtension" fields defined in "Common Components of Application Messages"
→	**555** **NoLegs**	N	Number of legs Identifies a Multi-leg Execution if present and non-zero.
→	→ **component block <InstrumentLeg>**	N	Must be provided if Number of legs > 0
→	**336** **TradingSessionID**	N	
→	**625** **TradingSessionSubID**	N	
→	**58** **Text**	N	Comment, instructions, or other identifying information.
→	**354** **EncodedTextLen**	N	Must be set if EncodedText field is specified and must immediately precede it.

→	**355**	**EncodedText**	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
	StandardTrailer		Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details
Refer to FIXML element DerivSecList

Security Status Request

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

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

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

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

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

Security Status Request

Tag	*FieldName*	*Req'd*	*Comments*
	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).
component block <Instrument>		Y	Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages"
component block <InstrumentExtension>		N	Insert here the set of "InstrumentExtension" fields defined in "Common Components of Application Messages"
711	NoUnderlyings	N	Number of underlyings
→	**component block <UnderlyingInstrument>**	N	Must be provided if Number of underlyings > 0
555	NoLegs	N	Number of legs Identifies a Multi-leg Execution if present and non-zero.
→	**component block <InstrumentLeg>**	N	Must be provided if Number of legs > 0
15	Currency	N	
263	SubscriptionRequestType	Y	SubcriptionRequestType indicates to the other party what type of response is expected. A snapshot request only asks for current information. A subscribe request asks for updates as the status changes. Unsubscribe will cancel any future update messages from the counter party.)
336	TradingSessionID	N	
625	TradingSessionSubID	N	
	StandardTrailer	Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details
Refer to FIXML element SecStatReq

Security Status

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

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

Security Status

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = f (lowercase)
324	SecurityStatusReqID	N	
component block <Instrument>		Y	Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages"
component block <InstrumentExtension>		N	Insert here the set of "InstrumentExtension" fields defined in "Common Components of Application Messages"
711	NoUnderlyings	N	Number of underlyings
→	**component block <UnderlyingInstrument>**	N	Must be provided if Number of underlyings > 0
555	NoLegs	N	Number of legs Identifies a Multi-leg Execution if present and non-zero.
→	**component block <InstrumentLeg>**	N	Must be provided if Number of legs > 0
15	Currency	N	
336	TradingSessionID	N	
625	TradingSessionSubID	N	
325	UnsolicitedIndicator	N	Set to 'Y' if message is sent as a result of a subscription request not a snapshot request
326	SecurityTradingStatus	N	Identifies the trading status applicable to the transaction.
291	FinancialStatus	N	
292	CorporateAction	N	
327	HaltReason	N	Denotes the reason for the Opening Delay or Trading Halt.
328	InViewOfCommon	N	
329	DueToRelated	N	
330	BuyVolume	N	

331	SellVolume	N	
332	HighPx	N	
333	LowPx	N	
31	LastPx	N	Represents the last price for that security either on a Consolidated or an individual participant basis at the time it is disseminated.
60	TransactTime	N	Trade Dissemination Time
334	Adjustment	N	
1025	FirstPx	N	Represents the price of the first fill of the trading session.
58	Text	N	Comment, instructions, or other identifying information.
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
	StandardTrailer	Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details
Refer to FIXML element SecStat

Trading Session Status Request

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

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

To list the securities available during a particular trading session, ***see the SecurityDefinitionRequest message***.

Trading Session Status Request

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = g (lowercase)
335	TradSesReqID	Y	Must be unique, or the ID of previous Trading Session Status Request to disable if SubscriptionRequestType = Disable previous Snapshot + Updates Request (2).
336	TradingSessionID	N	Trading Session for which status is being requested
625	TradingSessionSubID	N	
338	TradSesMethod	N	Method of trading
339	TradSesMode	N	Trading Session Mode
263	SubscriptionRequestType	Y	
207	SecurityExchange	N	
	StandardTrailer	Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details
Refer to FIXML element TrdgSesStatReq

Trading Session Status

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

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

Trading Session Status

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

FIXML Definition for this message – see http://www.fixprotocol.org for details
Refer to FIXML element TrdgSesStat

Trading Session List Request

The Trading Session List Request is used to request a list of trading sessions available in a market place and the state of those trading sessions. The request can be modified to request status on a particular trading session (by specifying the TradingSessionID (tag 336) and TradingSessionSubID (tag 625) (if used by the market place). The request can be used to request a list of trading sessions that use a particular trading method or mode (such as electronic) by specifying the TradSesMethod (tag 338) and/or TradSesMode(tag 339).

A successful request will result in a response from the counterparty of a Trading Session List (MsgType=BJ) message that contains a list of zero or more trading sessions.

It is recommended that the TradSesReqID be used to provide a unique identifier for the request. This value should be returned by the counterparty in the Trading Session List messages sent in response to the request.

The Trading Session List Request follows the standard request model in providing the SubscriptionRequestType (tag 263) field which can be used to obtain a snapshot of trading session information, subscribe for a snapshot with subsequent updates, or to unsubscribe from a previous subscription request.

Trading Session List Request

Tag	*FieldName*	*Req'd*	*Comments*
	StandardHeader	Y	MsgType = BI
335	TradSesReqID	Y	Must be unique, or the ID of previous Trading Session Status Request to disable if SubscriptionRequestType = Disable previous Snapshot + Update Request (2).
336	TradingSessionID	N	Trading Session for which status is being requested
625	TradingSessionSubID	N	
207	SecurityExchange	N	
338	TradSesMethod	N	Method of Trading
339	TradSesMode	N	Trading Session Mode
263	SubscriptionRequestType	Y	
	StandardTrailer	Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details
Refer to FIXML element TrdSessListReq

Trading Session List

The Trading Session List message is sent as a response to a Trading Session List Request. The Trading Session List should contain the characteristics of the trading session and the current state of the trading session.

The Trading Session List should return the TradSesReqID(tag 335) value from the Trading Session List Request originally sent by a counterparty.

Trading Session List

Tag	*FieldName*		*Req'd*	*Comments*
	StandardHeader		Y	MsgType = BJ
335	TradSesReqID		N	Provided for a response to a specific Trading Session List Request message (snapshot).
386	NoTradingSessions		Y	
→	**336**	**TradingSessionID**	Y	Identifier for Trading Session
→	**625**	**TradingSessionSubID**	N	
→	**207**	**SecurityExchange**	N	
→	**338**	**TradSesMethod**	N	Method of Trading
→	**339**	**TradSesMode**	N	Trading Session Mode
→	**325**	**UnsolicitedIndicator**	N	"Y" if message is sent unsolicited as a result of a previous subscription request.
→	**340**	**TradSesStatus**	Y	State of trading session.
→	**567**	**TradSesStatusRejReason**	N	Used with TradSesStatus = "Request Rejected"
→	**341**	**TradSesStartTime**	N	Starting time of trading session
→	**342**	**TradSesOpenTime**	N	Time of the opening of the trading session
→	**343**	**TradSesPreCloseTime**	N	Time of pre-close of trading session
→	**344**	**TradSesCloseTime**	N	Closing time of trading session
→	**345**	**TradSesEndTime**	N	End time of trading session
→	**387**	**TotalVolumeTraded**	N	
→	**58**	**Text**	N	
→	**354**	**EncodedTextLen**	N	Must be set if EncodedText field is specified and must immediately precede it.
→	**355**	**EncodedText**	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.
	StandardTrailer		Y	

FIXML Definition for this message – see http://www.fixprotocol.org for details
Refer to FIXML element TrdSessList

Security Definition, Security Status, and Trading Session Message Scenarios

Overview

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

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

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

The *Security Status* message provides solicited or unsolicited status information on securities. An exchange can use this message to transmit change in trading state of a product. The *Security Status Request* message can be used to query the state of a product or to subscribe for security state changes.

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

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

Background

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

Definitions

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

  Examples:

  - Vertical Spread
  - Butterfly Spread
  - Calendar Spread
  - Covered Write
- Strategy Leg - One Security within a strategy

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

Approach

A *Security Definition Request* message can be used to define and/or request a specific Security to be traded with a counterparty.

The *Security Definition message* is used to:

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

Extensions to other messages

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

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

For an example:

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

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

Rules

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

Specifying Derivative Trading Strategies using the Security Definition message

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

Two additional pieces are needed specify the strategy.

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

Example using *RatioQty* and *Side*:

A Butterfly strategy consists of simultaneously:

Buying 1 Call at Strike Price #1

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

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

The Legs that would describe this strategy are as follows:

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

Scenarios

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

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

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

Scenario 2 - Inquire Securities Types Available

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

First Party		Second Party
Security Definition Request message SecurityRequest = 2	→	Processes Security Definition message

First party can use this to select a list of messages	←	Security Definition message
		In this scenario, the trading party only trades three types of securities
		SecurityResponseType= 2
		NoRelatedSym=3
		UnderlyingSecuritySymbol=SecurityType#1
		UnderlyingSecuritySymbol=SecurityType#2
		UnderlyingSecuritySymbol=SecurityType#3

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

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

First Party		Second Party

Security Definition Request message
In this scenario the initiator wants to obtain a list of common stock available for trading with the counterparty.
SecurityRequest=3
SecurityType=”CS”
	→	Processes Security request
		Create a list of common stocks that are available for trading.

First party can use this to select a list of messages	←	Security Definition message
		Contains list of common stocks available for trading with the second party
		SecurityResponse=3
		NoRelatedSym=25
		UnderlyingSecuritySymbol=”AOL”
		….Other fields for this security
		UnderlyingSecuritySymbol=”GM”
		….Other fields for this security
		UnderlyingSecuritySymbol=”IBM”
		….Other fields for this security

Scenario 4 - Inquire all securities traded by a trading party

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

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

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

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

First Party		Second Party

Security Definition Request message
In this scenario the initiator wants to see a list of option series for IBM that are traded by the counterparty (that may be an exchange)
SecurityRequest=3
SecurityType=”OPT”
	→	Processes Security request
		Create a list of common stocks that are available for trading.

First party can use this to select a list of messages	←	Security Definition message
		Contains list of common stocks available for trading with the second party
		SecurityResponse=3
		NoRelatedSym=25
		UnderlyingSecuritySymbol=”AOL”
		UnderlyingSecuritySymbol=”GM”
		UnderlyingSecuritySymbol=”IBM”

Scenario 6 - Inquire list of option series for a class

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

First Party		Second Party

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

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

User Defined Spreads using Security Definition Messages

User Defined Spreads (UDS) allow users to construct strategies that support their unique trading and risk needs. In an exchange-centric model, a user may request a custom-designed strategy when the pre-listed instruments offered by an exchange or counterparty are insufficient to meet these needs. If accepted by the exchange or counterparty, it will become a listed instrument.

FIX currently provides support for User Defined Spreads through the Security Definition Request and Security Definition messages. These messages allow single-leg or multi-leg requests to be submitted for instrument creation, and provide confirmation of the fully elaborated instrument. Once the UDS has been established, the requestor will generally submit a subsequent Order or Request for Quote on the newly defined instrument.

Creating a User Defined Spread - Business Flow

The Business Process for User Defined Spreads is expressed by the workflow shown below. One-step and two-step processes are illustrated as they represent the recommended flow in FIX 4.4 (and above). The requesting party makes known its desire to define an instrument which has not been pre-listed by the Respondent - usually an exchange entity - by sending in a Request for a New Strategy. The Strategy will generally be a complex, multi-legged strategy or an options strategy which will provide neutral risk.

Upon receiving the request, the Respondent will perform validation and either accept or reject the request. If accepted, the Respondent will create a new instrument which is now considered to be “listed” [on the exchange], and send back confirmation that a new instrument has been created. Generally, the Respondent will not revise the requested instrument definition but will simply reject the request. The confirmation will carry all the details of the new instrument. The Requestor will then submit orders and/or request for quotes on the newly established instrument which will then follow the normal flow for these processes.

Exhibit 1: One-step process

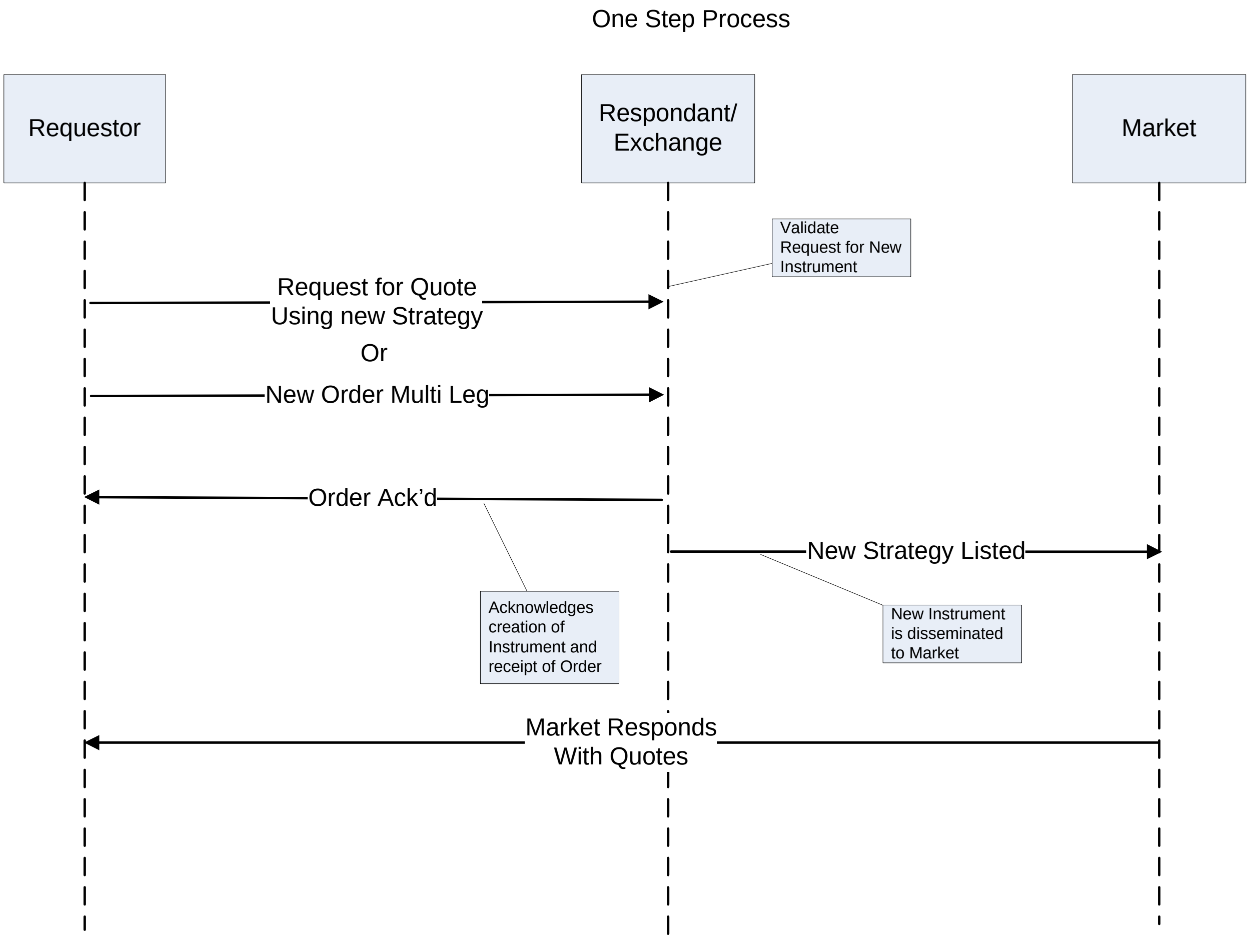
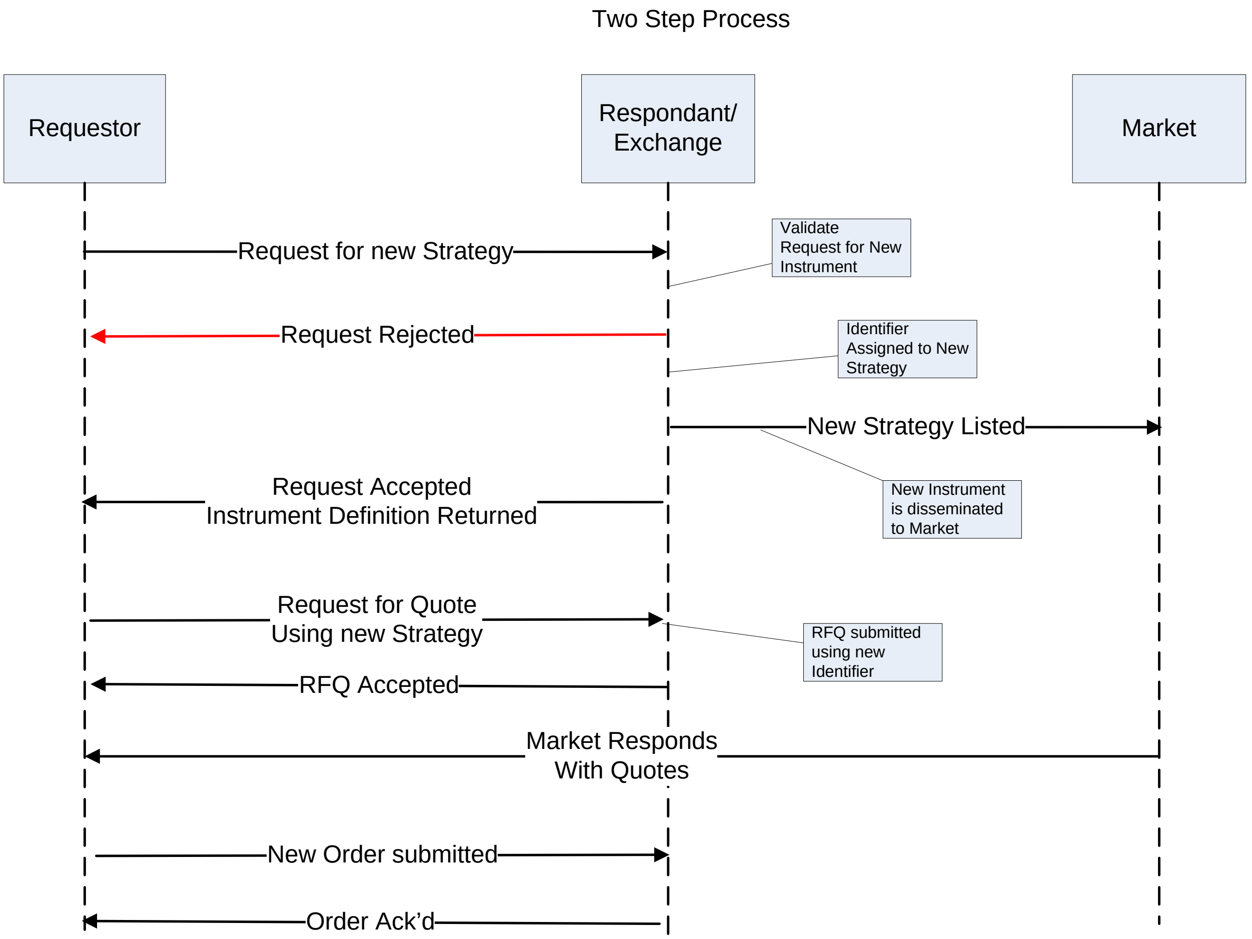

One Step Process
Requestor
Respondant/ Exchange
Market
Validate Request for New Instrument
Request for Quote Using new Strategy
Or
New Order Multi Leg
Order Ack'd
New Strategy Listed
Acknowledges creation of Instrument and receipt of Order
New Instrument is disseminated to Market
Market Responds With Quotes

Exhibit 2: Two-step process

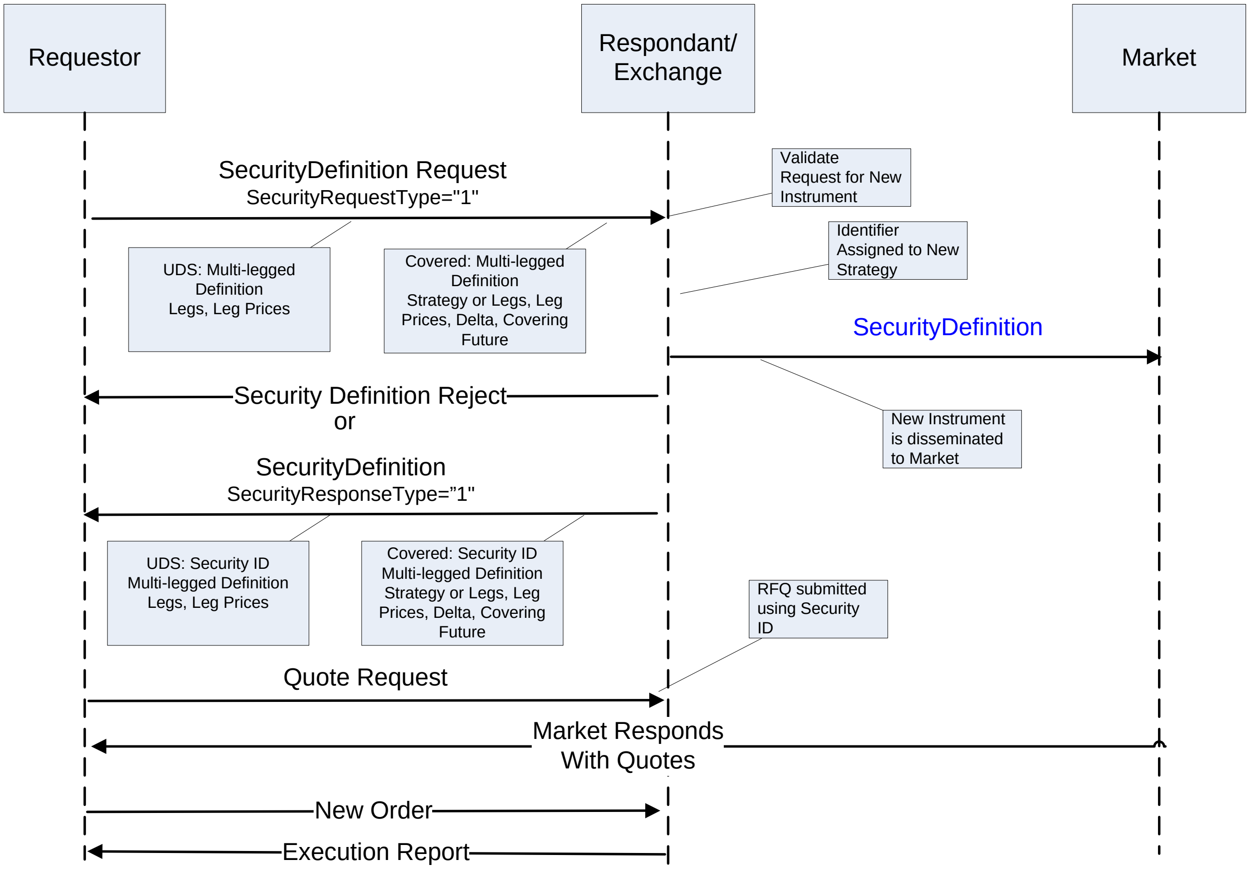

Two Step Process
Requestor
Respondant/ Exchange
Market
Request for new Strategy
Validate Request for New Instrument
Request Rejected
Identifier Assigned to New Strategy
New Strategy Listed
New Instrument is disseminated to Market
Request Accepted Instrument Definition Returned
Request for Quote Using new Strategy
RFQ submitted using new Identifier
RFQ Accepted
Market Responds With Quotes
New Order submitted
Order Ack'd

Creating a User Defined Spread - FIX Message Flow

The message flow for creation of a User Defined Spread is shown in the Exhibit below. The requesting party submits a Security Definition Request Message with the objective of defining a new instrument. The requestor will submit the specifications for the new instrument as part of the Security Definition Request. The Security ID of the strategy will not be provided as it is not yet know. It is important to note that an Instrument Block need not be included on the message. The InstrumentLeg block will be used to convey the legs of the strategy.

The respondent will validate the Security Definition Request, create the instrument, and respond with a Security Definition Message which will carry a Security ID or Symbol for the new instrument. The Security Definition Message will carry all the details of the new instrument. Upon receiving the Security Definition Message, the requestor will then submit an order and/or request for quote on the newly established instrument which will then follow the normal flow for these processes.

Exhibit 3: FIX Message flow for User Defined Spread


Requestor
Respondant/ Exchange
Market
SecurityDefinition Request
SecurityRequestType="1"
Validate Request for New Instrument
Identifier Assigned to New Strategy
UDS: Multi-legged Definition Legs, Leg Prices
Covered: Multi-legged Definition Strategy or Legs, Leg Prices, Delta, Covering Future
SecurityDefinition
Security Definition Reject
or
New Instrument is disseminated to Market
SecurityDefinition
SecurityResponseType="1"
UDS: Security ID Multi-legged Definition Legs, Leg Prices
Covered: Security ID Multi-legged Definition Strategy or Legs, Leg Prices, Delta, Covering Future
RFQ submitted using Security ID
Quote Request
Market Responds With Quotes
New Order
Execution Report

Appendix 3-A: Pre-Trade Message Targeting/Routing

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

Targeting

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

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

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

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

Note: The ^ character represents the SOH delimiter.

Tag Explanation

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

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

Blocking

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

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

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

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

Note: The ^ character represents the SOH delimiter.

Tag Explanation

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

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

Other Issues

It is expected that every indication of interest message will have a unique IOIid for the FIX session for the trading day.

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

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