FIX Global Exchanges and Markets Committee

ISE Market Data Extensions

October 5, 2009

v0.6

Proposal Status: ~~Final~~Approved

DISCLAIMER

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

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

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

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

Copyright 2003-2009 FIX Protocol Limited, all rights reserved

Table of Contents

Table of Figures

Document History

Revision	Date	Author	Revision Comments
0.1	Aug 4, 2009	Hanno Klein, Deutsche Börse Group	Initial draft
0.2	Aug 10, 2009	Hanno Klein, Deutsche Börse Group	Updated version after discussion in GexMC on Aug 6, 2009
0.3	Aug 24, 2009	Hanno Klein, Deutsche Börse Group	Updated version after discussion in GexMC on Aug 20, 2009 • Renamed SecurityListStatus message to MassSecurityStatus • Added MassSecurityStatusRequest • Re-used <InstrumentScope> block from Parties Reference Data proposal to specifiy multiple instruments
0.4	Sep 10, 2009	Hanno Klein, Deutsche Börse Group	Updated version after discussion in GTC on Sep 3 and Sep 10, 2009 • Add use case description for security trading status (section 2.2) • Change proposal to convey market order quantity (section 2.3) • Remove proposal to convey market order presence (section 2.3) • Change proposal to convey market segment information for incremental message on the root level (section 2.5)
0.5	Sep 17, 2009	Hanno Klein, Deutsche Börse Group	Final version for public comment, no changes to previous version (changes accepted).
0.6	Oct 5, 2009	Hanno Klein, Deutsche Börse Group	Final version for GTC Governance vote • Adjusted the introduction to complete the summary of changes • Changed name of messages from MassSecurityXXX to SecurityMassXXX • Changed name of MassSecurityTradingStatus to SecurityMassTradingStatus • Changed name of MassSecurityTradingEvent to SecurityMassTradingEvent
	Dec. 9, 2013	Lisa T.	Edits made based on FIXML Schema error reported and logged in SPEC-1065. Changed component category from PartiesReferenceData to

			Common for InstrumentScope and InstrumentScopeSecurityAltIDGrp.

1 Introduction

Market data comprises various pieces of information including book data as well as status information. This gap analysis suggests a number of extensions in this areas to cover the requirements of the Internal Securities Exchange (ISE), one of the major options exchanges in the US.

The proposal is looking for the following extensions:

- New message SecurityMassStatus to convey the trading status of an entire list (group) of securities together with a list of exceptions. Currently, only the SecurityStatus message is available which allows to convey the status of a single security only.

- New message SecurityMassStatusRequest to enable the request of the security trading status for an entire list (group) of securities.

- New fields SecurityMassTradingStatus, SecurityMassTradingEvent, MassHaltReason to convey a trading status, a trading event and a halt reason for multiple securities.

- New fields MDSecurityTradingStatus and MDHaltReason on the root level of MDSnapshotFullRefresh to convey a single trading status and halt reason for the instrument specified in this message.

- New enumerations Market Bid and Market Offer for MDEntryType to convey market order quantities

- New field MDSubFeedType in MDSnapshotFullRefresh, MDIncrementalRefresh and the component <MarketDataFeedTypes> to increase the granularity currently provided by the field MDFeedType.

- Add fields MarketID and MarketSegmentID to the root level of MDSnapshotFullRefresh and MDIncrementalRefresh to convey information about the market (segment).

2 Business Workflow

2.1 Security Mass Status

State changes on an instrument level can be conveyed by means of the SecurityStatus message. A new message, SecurityMassStatus, is being proposed that follows the same business workflow but can convey state changes of multiple instruments that belong to a well defined group.

The SecurityMassStatus carries a single security trading status value, event value and halt reason that applies to all securities of the list. Additionally a repeating group of exceptions can optionally be provided which contains securities of the list together with a security trading status different from the one at the list level.

The use case for such a message is an electronic marketplace for options with market makers that have to initiate continuous trading in a large number of series. This is often done by a single request for an entire product or market segment. The marketplace needs to communicate the state change and would require a large number of SecurityStatus messages if it were to do so by tradable entity (individual option). Large marketplaces might a few hundred thousand individual options grouped into products or market segments. The reference data describes which options belong to which product or market segment. The SecurityMassStatus message then only has to refer to the product or market segment and the receiver of the message can propagate the security trading status to all options belonging to the given product or market segment.

It is also proposed to add a corresponding SecurityMassStatusRequest message to define the group of instruments for which the trading status is to be returned. The request can then trigger multiple SecurityStatus messages or a single SecurityMassStatus message. This is left to the implementation and should be defined in the rules of engagement.

2.2 Security Trading Status

The market data messages MDSnapshotFullRefresh and MDIncrementalRefresh allow to convey the current security trading status at the entry level, i.e. within the component block <MDFullGrp> and <MDIncGrp>. Whilst this is sufficient for the MDIncrementalRefresh message which also defined the instrument at this level, it can be inefficient for the MDSnapshotFullRefresh message which only convey information about a single instrument. The security trading status might be the same for all entries in the message. The proposal is to support such semantic by adding a second security trading status field (MDSecurityTradingStatus) at the root level of the message where also the instrument is being defined. It should contain the same valid values as the existing field SecurityTradingStatus. The trading status is typically accompanied by another (optional) field to describe a reason for a halted trading status. This require another field MDHaltReason which should contain the same valid values as the existing field HaltReason.

An important use case requiring both the root level and the entry level security trading status field is when the actual status (root level) and the historic status values for each entry need to be conveyed in a single snapshot message.

2.3 Market Orders

Market orders can be part of the market data if they are allowed to rest on the order book alongside with other order types such as limit orders. This is especially the case prior to continuous trading where there might be an imbalance caused by market orders.

The proposal is to allow market order quantities to be conveyed explicitly as part of the standard book data messages MDSnapshotFullRefresh and MDIncrementalRefresh. The existing values for MDEntryType only cover bid and offer in a general sense and it is proposed to add two specific values "Market Bid" and "Market Offer" for which the entry only contains a size (MDEntrySize). Information about market order quantities allows a market maker to resolve a potential imbalance before the trading state can go into continuous trading. Currently, the only imbalance information that can be conveyed are the netted amount and the side of an imbalance.

2.4 Market Data Feed Types

Market data feeds can be identified by means of the MDFeedType field. It permits arbitrary values and allows to, for example, distinguish a regular feed from a market maker feed which might be entitled to additonal information. The proposal suggest to increase the level of granularity by adding sub-feed types that further qualify the given value for MDFeedType.

This affects the actual market data flows via MDSnapshotFullRefresh and MDIncrementalRefresh as well the area of reference data where market data feeds are defined by means of the component <MDFeedTypes>.

2.5 Market Segment Information

The market data messages MDSnapshotFullRefresh and MDIncrementalRefresh provide market data on an instrument level. Electronic marketplaces often do not allow the recipient to customize the feed but pre-define a number of feeds over a number of lines. This requires the recipient to filter out those messages that are of interest to him. The proposal is to provide a higher level of granularity for this purpose by adding market (segment) information to the feed at the same level as the instrument information.

3 Issues and Discussion Points

3.1 *Design of Status Request Message*

There are basically two options to design the message used to request the status of a list of securities:

- Re-use of existing SecurityStatusRequest message by changing its <Instrument> component block from mandatory to optional
- Introduction of a new SecurityMassStatusRequest message

The re-use avoids a new message being very similar to an existing one. On the other hand it adds complexity to the response processing as a single request can lead to two different responses depending on the fields filled on the request. It was decided to go with a new dedicated request message as this is also more in line with existing ISO concepts.

3.2 *Usage of Security Status Message*

CME is using the SecurityStatus message to communicate the trading status of more than a single security by using the SecurityGroup field of the <Instrument> component block and not the specific field Symbol or SecurityID + SecurityIDSource. An option could thus be to re-use the SecurityStatus message instead of introducing a new SecurityMassStatus message.

However, the exception list of securities with a differing status is not supported by the SecurityStatus message and could be confusing when added to a message perceived to be about a single security. The alternative to send exceptions as separate SecurityStatus messages is not ideal as there is a requirement of atomicity to ensure that the receipient has a consistent view of all status values. The delay or loss of a message with a deviating status would lead to such an inconsistency. Therefore the better design is thought to be the introduction of a new message with a single default trading status and a repeating group of securities with a deviating trading status.

3.3 *Scope of Security Mass Status Message*

The proposed new SecurityMassStatus message allows to use SecurityListID, MarketID and MarketSegmentID to identify the group of securities for which a siingle trading status is conveyed. This could be insufficient as there are more such fields in the <Instrument> component block that could be useful, i.e.

- Product (460)
- ProductComplex (1227)
- SecurityGroup (1151)
- CFICode (461)
- SecurityType (167)
- SecuritySubType (762)

However, putting any of these fields at the main level of the SecurityMassStatus message would create a conflict in the tag=value syntax with the <Instrument> component block which is being used within the repeating group used for the exception list. A solution would be to introduce new fields with the same name but with a prefix of "SecurityList", e.g. SecurityListProduct. The easier alternative is to use the generic SecurityListID field to contain whatever grouping identifier is needed. This corresponds to the mechanism available for the existing SecurityList message that can be used to define such lists.

There is, however, another gap analysis in the area of parties reference data that will require a new component block called <InstrumentScope> that will cover the requirements described above. It has the ability to specify one or more instruments.

3.4 Name of Security Mass Status Message

The name of the new status message was "SecurityListStatus" initially. However, there were concerns with the name as there are existing messages such as SecurityList and SecurityListRequest as well as ListStatus and ListStatusRequest. "SecurityListStatus" would have been a direct combination of the two message sets and could have been confusing. It was decided to use the term "Mass" instead as it is also not solely about securities within a list defined by SecurityListID about a much more flexible scope.

Comments received during the public review period suggested to align the name with the existing message OrderMassStatusRequest. It was therefore decided to use SecurityMassXXX instead of MassSecurityXXX as prefix for message and field names.

4 Proposed Message Flow

The message flow for the new message SecurityMassStatus is proposed to be identical to the message flow of the existing message SecurityStatus.

The usage of SecurityMassStatusRequest without specifying an individual instrument could either lead to multiple SecurityStatus messages or a single SecurityMassStatus message. Both are possible message flows which either reduce the amount of requests needed or reduce the amount of both requests and responses needed.

The other extensions do not impact the message flow.

5 FIX message tables

The following sections contain changes to two existing messages (MDSnapshotFullRefresh and MDIncrementalRefresh) and the definition of new messages (SecurityMassStatusRequest and SecurityMassStatus) based on existing messages (SecurityStatusRequest and SecurityStatus).

5.1 Market Data – Snapshot / Full Refresh

Tag	FieldName	Req'd	Comments	Action	Mapping Usage and Comments
StandardHeader		Y	MsgType = W		
component block <ApplicationSequenceControl>		N			
911	TotNumReports	N	Total number or reports returned in response to a request.		
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		

1173	MDSubBookType		N	Can be used to define a subordinate book.		
264	MarketDepth		N	Can be used to define the current depth of the book.		
1022	MDFeedType		N	Describes a class of service for a given data feed, ie Regular and Market Maker		
1683 TBD	MDSubFeedType		N		NEW	
1187	RefreshIndicator		N			
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.		
1301	MarketID		N		ADD	
1300	MarketSegmentID		N		ADD	
component block <Instrument>			Y	Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages"		
Start of Component block, expanded in line < UndInstrmtGrp >						
711	NoUnderlyings		N	Number of underlyings		
→	component block <UnderlyingInstrument>		N	Must be provided if Number of underlyings > 0		
End of Component block, expanded in line < UndInstrmtGrp >						
Start of Component block, expanded in line < InstrmtLegGrp >						
555	NoLegs		N	Number of legs		
→	component block <InstrumentLeg>		N	Must be provided if Number of legs > 0		
End of Component block, expanded in line < InstrmtLegGrp >						
291	FinancialStatus		N			
292	CorporateAction		N			
451	NetChgPrevDay		N			
1682 TBD	MDSecurityTradingStatus		N		NEW	
1684 TBD	MDHaltReason		N		NEW	
Start of Component block, expanded in line < MDFullGrp >						
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"		
→	423	PriceType	N			
→	component block <YieldData>		N	Insert here the set of YieldData (yield-related) fields defined in "Common Components of Application Messages		

→		component block <SpreadOrBenchmarkCurve Data>		N	Insert here the set of SpreadOrBenchmarkCurveData (Fixed Income spread or benchmark curve) fields defined in Common Components of Application Messages		
→	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"		
→	*Start of Component block, expanded in line < SecSizesGrp >*						
→	1177	NoOfSecSizes		N	Number of entries following. Conditionally required when MDUpdateAction = New(0) and MDEntryType = Bid(0) or Offer(1).		
→	→	1178	MDSecSizeType	N	Defines the type of secondary size specified in MDSecSize(1179). Must be first field in this repeating group		
→	→	1179	MDSecSize	N			
→	*End of Component block, expanded in line < SecSizesGrp >*						
→	1093	LotType		N	Can be used to specify the lot type of the quoted size in order depth books.		
→	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			
→	326	SecurityTradingStatus		N			
→	327	HaltReason		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			
End of Component block, expanded in line < MDFullGrp >						
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		
Start of Component block, expanded in line < RoutingGrp >						
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.		
End of Component block, expanded in line < RoutingGrp >						
StandardTrailer			Y			

5.2 Market Data –Incremental Refresh

Tag	FieldName	Req'd	Comments	Action	Mapping Usage and Comments
StandardHeader		Y	MsgType = X		
component block <ApplicationSequenceControl>		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		
1683 TBD	MDSubFeedType	N		NEW	
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.		

➜ 1301	**MarketID**		**N**		**ADD**	
➜ 1300	**MarketSegmentID**		**N**		**ADD**	
Start of Component block, expanded in line < MDIncGrp >						
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.		
➜	1173	MDSubBookType	N	Can be used to define a subordinate book.		
➜	264	MarketDepth	N	Can be used to define the current depth of the book.		
➜	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.		
➜	*Start of Component block, expanded in line < UndInstrmtGrp >*					
➜	711	NoUnderlyings	N	Number of underlyings		

→	→	component block <UnderlyingInstrument>		N	Must be provided if Number of underlyings > 0	
→	*End of Component block, expanded in line < UndInstrmtGrp >*					
→	*Start of Component block, expanded in line < InstrmtLegGrp >*					
→	555	NoLegs		N	Number of legs	
→	→	component block <InstrumentLeg>		N	Must be provided if Number of legs > 0	
→	*End of Component block, expanded in line < InstrmtLegGrp >*					
→	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"	
→	423	PriceType		N		
→	component block <YieldData>			N	Insert here the set of YieldData (yield-related) fields defined in Common Components of Application Messages	
→	component block <SpreadOrBenchmarkCurveData>			N	Insert here the set of SpreadOrBenchmarkCurveData (Fixed Income spread or benchmark curve) fields defined in Common Components of Application Messages	
→	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"	
→	*Start of Component block, expanded in line < SecSizesGrp >*					
→	1177	NoOfSecSizes		N	Number of entries following. Conditionally required when MDUpdateAction = New(0) and MDEntryType = Bid(0) or Offer(1).	
→	→	1178	MDSecSizeType	N	Defines the type of secondary size specified in MDSecSize(1179). Must be first field in this repeating group	
→	→	1179	MDSecSize	N		
→	*End of Component block, expanded in line < SecSizesGrp >*					
→	1093	LotType		N	Can be used to specify the lot type of the quoted size in order depth books.	

→	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			
→	326	SecurityTradingStatus	N			
→	327	HaltReason	N			
→	276	QuoteCondition	N	Space-delimited list of conditions describing a quote.		
→	277	TradeCondition	N	Space-delimited list of conditions describing a trade		
→	828	TrdType	N	For optional use in reporting Trades		
→	574	MatchType	N	For optional use in reporting Trades		
→	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		
→	1003	TradeID	N	For optional use in reporting Trades		
→	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	
→	483	TransBkdTime		N	For optional use in reporting Trades. Used to specify the time of trade agreement for privately negotiated trades.	
→	60	TransactTime		N	For optional use in reporting Trades. Used to specify the time of matching.	
→	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		
→	*Start of Component block, expanded in line < StatsIndGrp >*					
→	1175	NoStatsIndicators		N	Number of statistics indicators	
→	→	1176	StatsType	N	Indicates that the MD Entry is eligible for inclusion in the type of statistic specified by the StatsType. Must be provided if NoStatsIndicators greater than 0.	

➔	*End of Component block, expanded in line < StatsIndGrp >*				
➔	component block <Parties>	N			
End of Component block, expanded in line < MDIncGrp >					
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		
Start of Component block, expanded in line < RoutingGrp >					
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.		
End of Component block, expanded in line < RoutingGrp >					
StandardTrailer		Y			

5.3 Security Mass Status Request

The Security Mass Status Request message is a new message.

The Security Mass Status Request message provides for the ability to request the status of a group of securities. A single Security Mass Status message or multiple Security Status messages are returned as a result of a Security Mass Status Request message.

The Security Mass 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.

Tag	FieldName	Req'd	Comments	Action	Mapping Usage and Comments
StandardHeader		Y	MsgType = ~~TBD~~CN	ADD	
324	SecurityStatusReqID	Y	Must be unique, or the ID of previous Security Mass Status Request to disable if SubscriptionRequestType = Disable previous Snapshot + Updates Request (2).	ADD	
component block <InstrumentScope>		N		ADD	
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	ADD	

			changes. Unsubscribe will cancel any future update messages from the counter party.)		
1465	SecurityListID	N		ADD	
1301	MarketID	N		ADD	
1300	MarketSegmentID	N		ADD	
336	TradingSessionID	N		ADD	
625	TradingSessionSubID	N		ADD	
StandardTrailer		Y			

5.4 Security Mass Status

The SecurityMassStatus message is intended to provide the trading status for a well defined group of securities. This can either be a previously defined security list identified by SecurityListID or all securities of a specific market, market segment, trading session, trading sub-session or by usage of one or more fields of the <InstrumentScope> component block. Exceptions to the state change can be conveyed through a list of individual securities with a deviating trading status.

Tag	FieldName	Req'd	Comments	Action	Mapping Usage and Comments
StandardHeader		Y	MsgType = TBDCO	ADD	
component block <ApplicationSequenceControl>		N		ADD	
324	SecurityStatusReqID	N	Required when mass status is in response to a Security Mass Status Request message	ADD	
1465	SecurityListID	N	Identifies all securities for a security list identifier	ADD	
1301	MarketID	N	Identifies all securities for a market	ADD	
1300	MarketSegmentID	N	Identifies all securities for a market segment	ADD	
336	TradingSessionID	N	Identifies all securities for a trading session	ADD	
625	TradingSessionSubID	N	Identifies all securities for a trading sub-session	ADD	
component block <InstrumentScope>		N		ADD	
325	UnsolicitedIndicator	N	Set to 'Y' if message is sent as a result of a subscription request not a snapshot request	ADD	
1679 TBD	SecurityMassTradingStatus	N		NEW	
1680 TBD	SecurityMassTradingEvent	N		NEW	
1681 TBD	MassHaltReason	N		NEW	
1021	MDBookType	N	Used to relay changes in the book type	ADD	
264	MarketDepth	N	Used to relay changes in Market Depth.	ADD	
60	TransactTime	N	Time of state change for security list	ADD	
334	Adjustment	N		ADD	
Start of Component block, expanded in line < SecMassStatGrp >					

146		NoRelatedSym	N	Number of exceptions with a trading status different from SecurityMassTradingStatus (1679TBD)	ADD	
→		component block <Instrument>	N	Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages" Conditionally required if NoRelatedSym>0	ADD	
→		component block <InstrumentExtension>	N	Insert here the set of "InstrumentExtension" fields defined in "Common Components of Application Messages"	ADD	
→		*Start of Component block, expanded in line < UndInstrmtGrp >*				
→	711	NoUnderlyings	N	Number of underlyings	ADD	
→	→	component block <UnderlyingInstrument>	N	Must be provided if Number of underlyings > 0	ADD	
→		*End of Component block, expanded in line < UndInstrmtGrp >*				
→		*Start of Component block, expanded in line < InstrmtLegGrp >*				
→	555	NoLegs	N	Number of legs	ADD	
→	→	component block <InstrumentLeg>	N	Must be provided if Number of legs > 0	ADD	
→		*End of Component block, expanded in line < InstrmtLegGrp >*				
→	326	SecurityTradingStatus	N	Conditionally required if NoRelatedSym>0	ADD	
→	1174	SecurityTradingEvent	N		ADD	
→	327	HaltReason	N		ADD	
→	291	FinancialStatus	N		ADD	
→	292	CorporateAction	N		ADD	
→	58	Text	N	Comment, instructions, or other identifying information.	ADD	
→	354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.	ADD	
→	355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.	ADD	
End of Component block, expanded in line < SecMassStatGrp >						
StandardTrailer			Y			

6 FIX component blocks

6.1 Market Data Feed Types

Tag		Field Name	Req'd	Action	*Mappings and Usage Comments*	Comments
< MarketDataFeedTypes >						
1141		NoMDFeedTypes	N			The number of feed types and corresponding book depths associated with a security
→	1022	MDFeedType	N			
→	1683 TBD	MDSubFeedType	N	NEW		
→	264	MarketDepth	N			
→	1021	MDBookType	N			
</ MarketDataFeedTypes >						

6.2 Instrument Scope

This component block is only shown here for reference. It will be introduced by the Parties Reference Data Extensions gap analysis which will be approved prior to this one. In case of deviations, the Parties Reference Data Extensions version represents the final and approved version. Therefore, ~~noen~~ none of the text here is highlighted in yellow as it must not be the basis for any changes made to the FIX repository.

The InstrumentScope component category is to be changed from PartiesReferenceData to Common category.

Tag		Field Name	Req'd	Action	*Mappings and Usage Comments*	Comments
< InstrumentScope >						
1535		InstrumentScopeOperator	N	Rename		Required when NoInstrumentScopes > 0.
1536		InstrumentScopeSymbol	N	Rename		
1537		InstrumentScopeSymbolSfx	N	Rename		
1538		InstrumentScopeSecurityID	N	Rename		
1539		InstrumentScopeSecurityIDSource	N	Rename		
Begin component block <InstrScopeSecAltIDGrp>						
1540		NoInstrumentScopeSecurityAltID	N	Rename		
→	1541	InstrumentScopeSecurityAltID	new	Rename		Required when NoInstrumentScopeSecurityAltID > 0.
→	1542	InstrumentScopeSecurityAltIDSource	new	Rename		Required when NoInstrumentScopeSecurityAltID > 0.
End component block <InstrScopeSecAltIDGrp>						
1543		InstrumentScopeProduct	N	Rename		
1544		InstrumentScopeProductComplex	N	Rename		
1545		InstrumentScopeSecurityGroup	N	Rename		

1546	InstrumentScopeCFICode	N	Rename		
1547	InstrumentScopeSecurityType	N	Rename		
1548	InstrumentScopeSecuritySubType	N	Rename		
1549	InstrumentScopeMaturityMonthYear	N	Rename		
1550	InstrumentScopeMaturityTime	N	Rename		
1551	InstrumentScopeRestructuringType	N	Rename		
1552	InstrumentScopeSeniority	N	Rename		
1553	InstrumentScopePutOrCall	N	Rename		
1554	InstrumentScopeFlexibleIndicator	N	Rename		
1555	InstrumentScopeCouponRate	N	Rename		
1616	InstrumentScopeSecurityExchange	N	Rename		
1556	InstrumentScopeSecurityDesc	N	Rename		
1620	InstrumentScopeEncodedSecurityDescLen	N	Rename		
1621	InstrumentScopeEncodedSecurityDesc	N	Rename		
1557	InstrumentScopeSettlType	N	Rename		Can be used to specify FX tenors.
</ InstrumentScope>					

6.3 InstrumentScopeSecurityAltIDGrp

The InstrumentScopeSecurityAltIDGrp component category is to be changed from PartiesReferenceData to Common category.

7 Appendix A – Data Dictionary

Tag	Field Name	Action	Data type	Description	FIXML Abbreviation	Add to / Deprecate from Message type or Component block
35	MsgType	ADD ENUM	String	Defines message type ALWAYS THIRD FIELD IN MESSAGE. (Always unencrypted) Note: A "U" as the first character in the MsgType field (i.e. U, U2, etc) indicates that the message format is privately defined between the sender and receiver. *** Note the use of lower case letters *** Valid Values: 0 – Heartbeat 1 – TestRequest … y – SecurityList z – DerivativeSecurityListRequest TBD CN – SecurityMassStatusRequest TBD CO – SecurityMassStatus		
269	MDEntryType	ADD ENUM	char	Type Market Data entry. Valid Values: 0 – Bid 1 – Offer … Z – Recovery Rate for Long a – Recovery Rate for Short TBD b – Market Bid TBD c – Market Offer	Typ	
324	SecurityStatusReqID	CHANGE	String	Unique ID of a Security Status Request or a Security Mass Status Request message.	StatReqID	
TBD16 79	SecurityMassTradingStatus	NEW	int	Identifies the trading status applicable to a group of instruments. Same values as SecurityTradingStatus (326).	TrdgStat	Add to message SecurityMassStatus

TBD1680	SecurityMassTradingEvent	NEW	int	Identifies an event related to the mass trading status. Same values as SecurityTradingEvent (1174)	SecTrdEvnt	Add to message SecurityMassStatus
TBD1681	MassHaltReason	NEW	int	Denotes the reason for the Opening Delay or Trading halt of a group of securities. Same values as HaltReason (327)	HaltRsn	Add to message SecurityMassStatus
TBD1682	MDSecurityTradingStatus	NEW	int	Identifies the trading status applicable to the instrument in the market data message. Same values as SecurityTradingStatus (326).	TrdgStat	Add to message MDSnapshotFullRefresh
TBD1683	MDSubFeedType	NEW	String	Describes a sub-class for a given class of service defined by MDFeedType (1022)	MDSubFeedTyp	Add to messages MDSnapshotFullRefresh and MDIncrementalRefresh and to component block <MarketDataFeedTypes>
TBD1684	MDHaltReason	NEW	int	Denotes the reason for the Opening Delay or Trading Halt. Same values as HaltReason (327).	HaltRsn	Add to message MDSnapshotFullRefresh

8 Appendix B - Glossary Entries

Term	Definition	Field where used

9 Appendix C - Usage Examples