GENIUM® Protocol Specification

GENIUM FIX for OMX Nordic Exchange - FIX Specification

Protocol version 2.0.0

Document ID:	GENIUM-GFIX-PS-17_OMXE-2.0
Documentation Release:	a1
Release Date:	2007-06-28
Publication Date:	2007-06-28
Protocol version	2.0.0

Table of Contents

List of Tables

List of figures

1 Introduction

1.1 Purpose

This document contains the definition of the interface provided by OMX for developing applications related to the trading scope.

1.2 Audience

This is a reference document for those Members, Participants and other parties eligible for connection and ISVs that wish to develop software that can communicate with the market using the GENIUM FIX interface.

2 Document Information

2.1 References

[1]

FIX Protocol Specification, http://www.fixprotocol.org/specifications/fix5.0spec

[2]

FIX Transport Specification, http://www.fixprotocol.org/specifications/fixt1.1spec

[3]

FPL Website, http://www.fixprotocol.org/

[4]

Book Management Best Practices,
http://www.fixprotocol.org/documents/1462/MDOWG_Book_Mgt_Finaldraft.doc

[5]

FAST Specification, http://www.fixprotocol.org/fast

[6]

Exchanges/ECNs Working Group (EEWG) - Quotation Best Practices - Note: Not yet officially published

2.2 Definitions and Acronyms

2.3 Reading Instructions

2.3.1 The Main Chapters of this Specification

FIX Transport (Session) Layer

Describes how FIX sessions are established and maintained.

Business Layer Introduction

Describes general aspect for the business layer of the protocol.

Single and General Order Handling

Describes single orders, their workflows and functionality. Many aspect of those also apply to other types of orders and quotes.

Reporting Privately Negotiated Trades

Describes how user can report a trade negotiated outside the marketplace. Also includes reporting of trades formed at another execution venue.

Trade Confirmation and Management

Describes how the marketplace publishes trade information to eligible participants. Also includes the ability for users to request cancelation or amendment of confirmed trades.

Market Data

Describes how the marketplace publishes data for book views, trade tickers, trade statistics and other price indicators. Also includes trading status information,

Reference Data

Describes how the marketplace publishes reference data.

General Messages

Contains the message details for messages that are not specific to one of the above chapters.

General Component Blocks

Contains the details of the message component blocks that are not specific to one of the above chapters.

Data Dictionary

Contains all the message fields used in the messages and components of this specification. Also includes the applicable value domain for fields.

2.3.2 OMX Specification versus the FIX Standard

The interface is based on the FIX Protocol standard (Financial Information exchange). More detailed information about the standard can be found in FIX specification document see [1] or on the FPL website [3].

The interface implemented by OMX follows the FIX specifications, as far as possible. In the majority of cases the structure and semantics of the messages are identical to the standard.

In some cases, the protocol has been extended to cover functions not considered by the standard. These extensions are clearly detailed in the document.

In other cases, the standard is ambiguous or indicates that the details should be bilaterally agreed between the parties. In these cases the manual provides a detailed description to avoid any possible ambiguity.

All annotations and adaptations of the standard made by OMX have been done in accordance with the recommendations in the standard.

To avoid possible duplication in the sources of information, this document does not include explanations of those matters that comply exactly with the standard. Therefore, the standard documentation should be considered as the main source of information for any matter that is not explicitly covered in this manual.

The OMX specification tries not to repeat what is specified in the FIX standard. In many cases, however, the FIX standard is, by necessity, more generic than that required for a marketplace. In other cases OMX

has found reason to clarify matters. OMX tries to be explicit on deviations from the FIX standard specification in order to avoid confusion.

OMX is committed to follow and enhance the FIX standard. To achieve this the OMX is active in the Global Technical Committee, the Exchanges / ECNs working group, the Repository working group and other committees and working groups as required. OMX sees no meaning in having user defined Messages, Component Blocks, Fields or Enum values and works to include all such requirements in the standard. In some cases however, OMX requirements may be more specific than or outside the scope of the FIX standard and may therefore stay specific to OMX for a shorter or longer time.

2.3.3 Message and Component Tables

All Message and Component Block tables are built on the FIX Repository. Messages, Component Blocks, Fields or Enum values are filtered to the content needed by OMX. Please refer to the standard FIX specification for matters outside the scope of the OMX FIX specification.

OMX specific Messages have message identifiers, MsgType (35), prefixed by "U" and followed by a unique sequential number. OMX may also use the Non_FIX_XML (MsgType = n) message to tunnel content that is not considered worth their own messages. In areas where a FIX message exists but standardization on the individual field level is not considered possible even in the longer term, OMX may use the XML_Data tag of the Standard Header to embed additional content as specified in the individual messages. This option is especially relevant for Reference Data messages.

Component blocks pending inclusion in future versions of FIX are marked as such. The same applies to Fields (Tags) and Enum values. Such Fields are assigned tag numbers starting from 20000. Enum values are assigned "high" values to avoid conflict with other extensions. Users should be aware that those values will change when the standard is extended.

All messages, component and field descriptions are fetched from the FIX Repository. In cases where OMX has found a reason for clarification, that text starts with "OMX Comment:".

In cases where a Field or Component Block in a message is not required by FIX, but is required by OMX (or the opposite) - the "Reqd" column of the message / component table contains the FIX standard value within parenthesis.

The "Format" column of the Message and Component Block tables specifies the field format used by OMX. The format is limited by the FIX standard but in many cases is more specific.

3 FIX Transport (Session) Layer

3.1 FIX Transport 1.1 Protocol Compliance

The FIX Specification is fully compliant with the FIX Transport layer according to the standard FIX Transport 1.1 documentation with the following exception(s):

- Encryption not supported.

3.1.1 FIX Session Level Test Cases

The FIX Specification is fully compliant with the FIX session-level test cases specified in FIX Transport version 1.1, section "FIX Session-level Test Cases and Expected Behaviors" with the following exceptions:

- Encryption test cases not supported.

FIX client application developers should study the FIX session-level test cases in detail and verify their FIX client software for compliance.

3.2 Encryption Support

This feature is currently not supported.

3.3 FIX Session Establishment

A FIX session is established by sending a Logon message. The FIX session is established between two parties, the sender and the target represented by the following tags in the Standard Message Header:

- SenderCompID (tag 49). The party initiating the session
- SenderSubID (tag 50). The session identifier for the party initiating the session as per connection agreement with the market place.
- TargetCompID (tag 56). The acceptor of the session as per configuration.
- TargetSubID (tag 57). The session identifier for the party accepting the session as per connection agreement with the market place.

The FIX session is always initiated by the FIX client and accepted by the FIX server.

Only one unique combination of SenderCompID, SenderSubID, TargetCompID, and TargetSubID will be allowed per physical gateway instance.

3.4 Start of Day/End of Day Procedures

A FIX session can last forever, or until:

1. A login message that specifies that the sequence number series should be reset ResetSeqNumFlag (141).
2. A SequenceReset message is sent either by the market place or the FIX client.
3. As defined by the marketplace (e.g. in between trading days)

Subscriptions defined by user requests through FIX messages expire when the session expires. Also, all history and resend capabilities will be lost after a sequence number reset on the FIX session.

Note:

A user can lose a connection or logout, but still resume the session through the use of a logon (not resetting sequence numbers).

If a user wishes to finish his connection with a logout, this should be preceded by unsubsription requests. Note that the session is not terminated.

Note:

Messages that are pushed from the marketplace (as e.g. Execution Reports) are never lost.

3.5 Duplicate and Resend Message Handling

The FIX specification is detailed in describing required handling of potential duplicate information; the following are specific rules for this installation.

Tags 43 (PossDupFlag) and 97 (PossResend) can indicate potentially duplicate information, either contained within a message with the same sequence number, or in a message with a new sequence number. The rules in processing these messages are detailed and mostly clear; however, the following rule applies:

- No specific rules at this time

Such rules would be implemented one level above the session layer, and serve as an additional safeguard against duplicate information.

3.6 Session Level Counterparty Identification Fields

A number of tags in the FIX standard message header are used for session level counterparty identification. The following counterparty identification tags are supported:

- SenderCompID. This field is used to identify the sending (originating) party as per connection agreement with the market place.
- SenderSubID. This field is used to identify the session identifier as per connection agreement with the market place.

- TargetCompID. This field is used to identify the party at which the message is targeted as per connection agreement with the market place.
- TargetSubID. This field is used to identify the session to which a specific message is targeted as per connection agreement with the market place.
- OnBehalfOfCompID. This field is used to identify the member code for the originating participant when a third party sends a message. This field is only used in incoming business messages.
- OnBehalfOfSubID. This field is used to identify the responsible firm session identifier as per connection agreement. This field is only used in incoming business messages.
- DeliverToCompID. This field is used to identify the member code for the originating participant at which the message is targeted. The field is only used in outgoing business messages.
- DeliverToSubID. This field is used to identify the responsible firm session identifier as per connection agreement. This field is only used in incoming business messages.

3.6.1 Primary Member Messaging

The following counterparty fields are used in a business message when a member of the exchange sends a message directly to the exchange without any third party involved.

3.6.1.1 Incoming messages

- Incoming messages should identify the receiving process by specifying:
- SenderCompID (49) = The connected firm identifier as per connection agreement
- SenderSubID (50) = The connected message originator (e.g Desk, Trader etc.)
- TargetCompID (56) = The targeted system at the market place
- TargetSubID (57) = Optional field to specify the receiving FIX gateway

3.6.1.2 Outgoing messages

- SenderCompID (49) = The sending system at the market place
- SenderSubID (50) = Optional field to specify the sending FIX gateway
- TargetCompID (56) = The connected firm identifier as per connection agreement
- TargetSubID (57) = The connected message originator (e.g Desk, Trader etc.)

3.6.2 On-behalf-of Messages

Third parties may be authorized to act as routers of messages. The on-behalf-of fields are only relevant when messages are routed through a third party, not for other type of on-behalf-of functions as e.g. when a user enters an order on-behalf-of a colleague. GENIUM authorizes that the 3rd party is allowed to route the relevant message(s) and relay the fields back in responses (using the DeliverTo fields). The on-behalf-of fields are however not used in any business layer logic.

3.6.2.1 Incoming messages

The following fields must be specified, representing the party responsible for the business content of the message:

- SenderCompID (49) = The code of the participant acting as a third party
- SenderSubID (50) = The session identifier for the third party
- TargetCompID (56) = Code for targeted system at the market place
- TargetSubID (57) = Optional field to specify the receiving FIX gateway
- OnBehalfOfCompID (115) = responsible firm identifier as per that party's connection agreement.
- OnBehalfOfSubID (116) = responsible firm session identifier as per connection agreement

3.6.2.2 Outgoing messages

Outgoing messages reverse the Target and Sender fields. In addition the following fields are used in cases where the receiver is a 3rd party that is to route the messages onward:

- SenderCompID (49) = Code for sending system at the market place
- SenderSubID (50) = Optional field to specify the sending FIX gateway
- TargetCompID (56) = The code of the participant acting as a third party
- TargetSubID (57) = The session identifier for the third party
- DeliverToCompID (128) = responsible firm identifier as per that party's connection agreement.
- DeliverToSubID (129) = responsible firm session identifier as per connection agreement.

3.7 Reject vs. BusinessMessageReject vs. Application Reject

The Reject messageis issued when a message is received but cannot be properly processed due to a session-level rule violation. An example of when a Reject may be appropriate would be the receipt of a message with invalid basic data (e.g. MsgType=&) which successfully passes de-encryption, CheckSum and BodyLength checks. Note that, if a business message is received which fulfills session-level rules but the message is still rejected, then whichever reject message corresponds to that incoming business message should be sent.

Note:

Note the only exceptions to this rule are:

1. In the event a business message is received, fulfills session-level rules, however, the message cannot be communicated to the business-level processing system. In this situation a Business Message Reject with BusinessRejectReason = “Application not available at this time” can be issued if the system is unable to send the specific “reject” message listed above due to this condition.
2. In the event a valid business message is received, fulfills session-level rules, however, the message type is not supported by the receipient. In this situation a Business Message Reject with BusinessRejectReason = “Unsupported Message Type” can be issued if the system is unable to send the specific “reject” message listed above because the receiving system cannot generate the related “reject” message.
3. In the event a business message is received, fulfills session-level rules, but lacks a field conditionally required by the FIX specification.

3.8 Authorization

All FIX sessions are subject to authorization. When the FIX gateway receives a Logon message at connection startup, the session is authorized using:

- SenderCompId (49). This is the market place member code.
- SenderSubID (50). This is the market place session level identifier.
- Password (554). This is the market place session level password valid for the specified session and member.

If the authorization fails the session will be disconnected without sending any message.

3.8.1 Message Authorization

Application messages are subject to authorization. The responsible end-user, defined in the Parties component block in the application message, must be authorized to perform the operation. If not, the messages will be rejected.

3.9 Message Details

3.9.1 Heartbeat

Table 1: *Heartbeat*

Tag	FieldName	Req'd	Comments	Format
	StandardHeader	Y	MsgType = 0	
112	TestReqID	N	Required when the heartbeat is the result of a Test Request message.	String
	StandardTrailer	Y		

3.9.2 Logon

Table 2: *Logon*

Tag	FieldName	Req'd	Comments	Format
	StandardHeader	Y	MsgType = A	
98	EncryptMethod	Y	(Always unencrypted)	int
108	HeartBtInt	Y	Note same value used by both sides	int
141	ResetSeqNumFlag	N	Indicates both sides of a FIX session should reset sequence numbers	Boolean
789	NextExpectedMsgSeqNum	N	Optional, alternative via counterparty bi-lateral agreement message gap detection and recovery approach (see "Logon Message NextExpectedMsgSeqNum Processing" section)	SeqNum
554	Password	N	Note: minimal security exists without transport-level encryption.	String
1137	DefaultApplVerID	Y	The default version of FIX being carried over this FIXT session	String
	StandardTrailer	Y		

3.9.3 Logout

Table 3: *Logout*

Tag	FieldName	Req'd	Comments	Format
	StandardHeader	Y	MsgType = 5	
58	Text	N		String
	StandardTrailer	Y		

3.9.4 Reject

Table 4: *Reject*

Tag	FieldName	Req'd	Comments	Format
	StandardHeader	Y	MsgType = 3	
45	RefSeqNum	Y	MsgSeqNum of rejected message	SeqNum
371	RefTagID	N	The tag number of the FIX field being referenced.	int
372	RefMsgType	N	The MsgType of the FIX message being referenced.	String
373	SessionRejectReason	N	Code to identify reason for a session-level Reject message.	int
58	Text	N	Where possible, message to explain reason for rejection	String
	StandardTrailer	Y		

3.9.5 ResendRequest

Table 5: *ResendRequest*

Tag	FieldName	Req'd	Comments	Format
	StandardHeader	Y	MsgType = 2	
7	BeginSeqNo	Y		SeqNum
16	EndSeqNo	Y		SeqNum
	StandardTrailer	Y		

3.9.6 SequenceReset

Table 6: *SequenceReset*

Tag	FieldName	Req'd	Comments	Format
	StandardHeader	Y	MsgType = 4	
123	GapFillFlag	N		Boolean
36	NewSeqNo	Y		SeqNum
	StandardTrailer	Y		

3.9.7 TestRequest

Table 7: *TestRequest*

Tag	FieldName	Req'd	Comments	Format
	StandardHeader	Y	MsgType = 1	
112	TestReqID	Y		String
	StandardTrailer	Y		

3.10 Component Blocks

3.10.1 StandardHeader

Table 8: *StandardHeader*

Tag	FieldName	Req'd	Comments	Format
8	BeginString	Y	FIXT.1.1 (Always unencrypted, must be first field in message)	String
9	BodyLength	Y	(Always unencrypted, must be second field in message)	Length
35	MsgType	Y	(Always unencrypted, must be third field in message)	String

Tag	FieldName	Req'd	Comments	Format
1128	ApplVerID	N	Indicates application version using a service pack identifier. The ApplVerID applies to a specific message occurrence.	String
49	SenderCompID	Y	(Always unencrypted)	String
56	TargetCompID	Y	(Always unencrypted)	String
115	OnBehalfOfCompID	N	Trading partner company ID used when sending messages via a third party (Can be embedded within encrypted data section.)	String
128	DeliverToCompID	N	Trading partner company ID used when sending messages via a third party (Can be embedded within encrypted data section.)	String
34	MsgSeqNum	Y	(Can be embedded within encrypted data section.)	SeqNum
50	SenderSubID	N	(Can be embedded within encrypted data section.)	String
57	TargetSubID	N	"ADMIN" reserved for administrative messages not intended for a specific user. (Can be embedded within encrypted data section.)	String
116	OnBehalfOfSubID	N	Trading partner SubID used when delivering messages via a third party. (Can be embedded within encrypted data section.)	String
129	DeliverToSubID	N	Trading partner SubID used when delivering messages via a third party. (Can be embedded within encrypted data section.)	String
43	PossDupFlag	N	Always required for retransmitted messages, whether prompted by the sending system or as the result of a resend request. (Can be embedded within encrypted data section.)	Boolean
52	SendingTime	Y	(Can be embedded within encrypted data section.)	UTCTimestamp
122	OrigSendingTime	N	Required for message resent as a result of a ResendRequest. If data is not available set to same value as SendingTime (Can be embedded within encrypted data section.)	UTCTimestamp

3.10.2 StandardTrailer

Table 9: *StandardTrailer*

Tag	FieldName	Req'd	Comments	Format
10	CheckSum	Y	(Always unencrypted, always last field in message)	String

4 Business Layer Introduction

4.1 Identifier Definitions

4.1.1 Definition of Business Level Party Identifier Conventions

All inbound business messages are subject to marketplace authorization and must therefore specify the party responsible for the business content of the message and, whenever applicable, the party that entered the data for the message (if not the owner itself):

- PartyID (448) = identifiers as per trading (or similar) agreements
- PartyIDSource (447)
  - C = Generally accepted market participant identifier (a participant mnemonic symbol) or
  - D = Proprietary / Custom code (exchange-assigned id-number)
- PartyRole (452) = see below

Using the PartyIDSource = C means the trading engine will have to do a lookup which increases latency; users are therefore encouraged to use PartyIDSource = D whenever possible.

This specification of business level party identifiers is done at the root level of the respective message, normally in the <Parties> component block but sometimes in the <RootParties> component block. The FIX specification sometimes has the <Parties> component block as part of a repeatable structure, in such cases the <RootParties> component block is attached to the root level of the message and used for the above purpose instead.

The following table specifies the party identifiers that are specified:

Table 10: Party Identifiers

	Business Role	PartyRole (452)	Comment
Transaction owner = parties legally responsible for consequences of the message (mandatory data)	Firm	1 = Executing Firm	Mandatory for all incoming messages
	Organizational Unit	59 = Executing Unit	Optional
	Individual User	12 = Executing Trader	Mandatory for all incoming messages
Transaction enterer (if different from owner) = party doing data entry on behalf of the owner	Firm	7 = Entering Firm	Optional
	Organizational Unit	58 = Entering Unit	Optional
	Individual User	36 = Entering Trader	Optional

Please see other sections for additional information on party roles.

The trading engine will verify that the broker and trader are authorized to submit the relevant business message. In cases where another party does data entry, the engine will verify that that party is authorized to perform the action on behalf of the owner. Please note that session layer On-Behalf-Of has nothing to do with the business layer one.

4.1.2 Definition of Security ID Conventions

For any trading system, the correct identification of securities in a FIX message is of utmost importance. There are several fields within each FIX message, incoming or outgoing, that allow for identification of securities. This chapter specifies which identifiers are preferred, and, if more than one is supported, which conventions are acceptable.

What identifier is acceptable varies with the market convention applicable for various type of security. Non-identifier fields are accepted in an incoming message, but remain unused for validation or identification purposes and are not relayed back in response messages.

Instruments (order books) are identified using one, or multiple, of the following options – all supported by the <Instruments> component block (or incarnations of it). In cases where more than one identifier is specified the following applies:

- If one of the identifiers is SecurityID + SecurityIDSource = 8 (Marketplace Assigned Identifier), that is the only identifier used, all others are ignored.
- In other cases, all identifiers must refer to one and the same order book. When one field is part of multiple potential identifiers, it will only be considered for the identifier that has the largest number of fields specified. Example:
    - Symbol (55) can be used as a unique identifier in its own right, but also as part of an options symbology in combination with CFICode (461), MaturityMonthYear (200) and StrikePrice (202). In the case those fields are specified, the Symbol is only considered as part of the options identifier (and there considered as a reference to the underlying instrument).

Note:

Please note that the Symbol (55) field is mandatory. In cases where the symbol is not relevant or cannot be specified, Symbol = "[N/A]" (including brackets, but excluding quote-marks) must be entered.

FIX allows a large number of other identifiers as CUSIP, SEDOL, RIC, etc – those are currently not supported.

4.1.2.1 General Identifiers Usable for any Security Type

Table 11: *Generic Security Identifiers*

Tag	Field Name	Req'd	Comments	OMX Comment
Unique Identifier: **SecurityID**				
48	SecurityID	N	Takes precedence in identifying security to counterparty over SecurityAltID block. Requires SecurityIDSource if specified.	Unique marketplace assigned identifier number for an order book.
22	SecurityIDSource	N	Required if SecurityID is specified.	Value = 8 (Marketplace assigned identifier)

Table 12: *Generic Identifier for Security with symbolic name*

Tag	Field Name	Req'd	Comments	OMX Comment
Unique Identifier: **Symbol**				

Tag	Field Name	Req'd	Comments	OMX Comment
55	Symbol	N	Common, “human understood” representation of the security. SecurityID value can be specified if no symbol exists (e.g. non-exchange traded Collective Investment Vehicles). Use “[N/A]” for products which do not have a symbol.	A required field when the <**Instrument**> component block is mandatory.
65	SymbolSfx	N	Used in Fixed Income with a value of “WI” to indicate “When Issued” for a security to be reissued under an old CUSIP or ISIN or with a value of “CD” to indicate a EUCP with lump-sum interest rather than discount price.	Suffix for the symbol, e.g. used for different share classes (A- and B-shares, preferential shares, “when issued”, etc – when the suffix is not embedded in the Symbol field).

The following identifiers are currently not supported as identifier fields for inbound messages. The fields may however be made available in outbound messages.

Table 13: *Generic Identifier for Security with ISIN*

Tag	Field Name	Req'd	Comments	OMX Comment
Unique Identifier: **ISIN**				
48	SecurityID	N	Takes precedence in identifying security to counterparty over SecurityAltID block. Requires SecurityIDSource if specified.	ISIN code
22	SecurityIDSource	N	Required if SecurityID is specified.	Value = 4 (ISIN)

ISINs are commonly used to identify fixed income securities in trading.

Other identifiers, similar to ISIN, as e.g. CUSIP and SEDOL may be made available

Table 14: *Options Identifiers*

Tag	Field Name	Req'd	Comments	OMX Comment
Unique Identifier: **Options characteristic (Expiry Month)**				
55	Symbol	N	Common, “human understood” representation of the security. SecurityID value can be specified if no symbol exists (e.g. non-exchange traded Collective Investment Vehicles). Use “[N/A]” for products which do not have a symbol.	Symbol of underlying
65	SymbolSfx	N	Used in Fixed Income with a value of “WI” to indicate “When Issued” for a security to be reissued under an old CUSIP or ISIN or with a value of “CD” to indicate a EUCP with lump-sum interest rather than discount price.	Symbol suffix of underlying
461	CFICode	N	Indicates the type of security using ISO 10962 standard, Classification of Financial Instruments (CFI code) values. It is recommended that CFICode be used instead of SecurityType for non-Fixed Income instruments.	Part of alternative identifier for derivatives (options vs futures). First position = “O”

Tag	Field Name	Req'd	Comments	OMX Comment
200	MaturityMonthYear	N	Specifies the month and year of maturity. Applicable for standardized derivatives which are typically only referenced by month and year (e.g. S&P futures). Note MaturityDate (a full date) can also be specified.	Part of alternative identifier for derivatives.
202	StrikePrice	N	Used for derivatives, such as options and covered warrants	Part of alternative identifier for derivatives.
Unique Identifier: **Options characteristic (Expiry Date)**				
55	Symbol	N	Common, “human understood” representation of the security. SecurityID value can be specified if no symbol exists (e.g. non-exchange traded Collective Investment Vehicles). Use “[N/A]” for products which do not have a symbol.	Symbol of underlying
65	SymbolSfx	N	Used in Fixed Income with a value of “WI” to indicate “When Issued” for a security to be reissued under an old CUSIP or ISIN or with a value of “CD” to indicate a EUCP with lump-sum interest rather than discount price.	Symbol suffix of underlying
461	CFICode	N	Indicates the type of security using ISO 10962 standard, Classification of Financial Instruments (CFI code) values. It is recommended that CFICode be used instead of SecurityType for non-Fixed Income instruments.	Part of alternative identifier for derivatives (options vs futures). First position = “O”
541	MaturityDate	N	Specifies date of maturity (a full date). Note that standardized derivatives which are typically only referenced by month and year (e.g. S&P futures). May use MaturityMonthYear and/or this field. When using MaturityMonthYear, it is recommended that markets and sell sides report the MaturityDate on all outbound messages as a means of data enrichment.	Part of alternative identifier for derivatives. Used instead of MaturityMonthYear when there are multiple strikes per month.
202	StrikePrice	N	Used for derivatives, such as options and covered warrants	Part of alternative identifier for derivatives.

4.2 System Partitioning

The marketplace system may run processes that are partitioned to enhance performance and lower latency. External actors should be aware of that partitioning has certain effects on workflows.

4.2.1 Multi partition transaction

The back-end systems require that mass transactions can only affect securities within one logical partition. Single mass transactions that affect multiple partitions will be rejected. The logical partition is defined by configuration.

4.3 Trading in various Security Types

The marketplace supports trading in a large number of Security Types, e.g. equity, fixed income instruments, options, futures and cross product multi-leg instruments. Each Security Type may be associated with different rules as to what fields are used. Examples:

- The PriceType (423) field is mandatory for trading Fixed Income products. This is a strong recommendation from the Fixed Income industry and is due to the fact that prices often can be quoted in various price types.

  Note:

  Note, a security is always associated with a default price type as defined by the security definition.

- Prices in Fixed Income products may be presented as both yield and percentage at par. In those cases the percentage at par price will be relayed in the normal price field, while the yield is presented in the Yield (236) field.

4.4 Accounts, Pre-Allocation and Give Ups

Although a firm may have default accounts specified at the exchange or in clearing, sometimes the user needs to assign a specific account to an order. Broker omnibus accounts are specified using the following fields:

- **Account** (1).
- **AccountIDSource** (660) = 99 = Other (custom or proprietary).

If the broker needs to allocate the order to a certain customer, or give up the trade in downstream clearing, the following fields are used:

- AllocID (70). Used to specify an identifier that can be used in downstream processes as clearing, e.g. a client order reference.
- NoAllocs (78). Specifies the number of allocations in the following repeating group. A single pre-allocation is currently allowed, but that allocation can be done both for back-office and downstream clearing / settlement purposes.
  - AllocAccount (79). Specifies the account.
  - AllocAcctIDSource (661) = 99 = Other (custom or proprietary).
  - IndividualAllocID (467). Currently not used. Needed to access a particular allocation in downstream systems, e.g. a client order reference.
  - NestedParties. Specifies the parties to the allocation, whenever relevant.

Note:

The field names and tag numbers varies between the FIX messages. The field names and tag numbers used herein are the ones used in Single Orders, Contingent Orders and for reporting Crosses. Inbound Multileg Orders and reported Privately Negotiated Trades also include the same features, but with partly other field names (and tag numbers). The same applies to outbound Trade Capture Reports (confirmed trades).

The following fields of the <NestedParties> component block are used:

- NestedPartyID (524) = the identifier of the (clearing) firm the trade is given up to
- NestedPartyIDSource (525)
  - C = Generally accepted market participant identifier or
  - D = Proprietary/Custom code
- NestedPartyRole (538) = see below

The following table specifies the relevant party roles:

Table 15: *Supported PartyRole (452) values*

	Business Role	PartyRole (452)	Comment
The (clearing) firm the broker gives up a trade to	Clearing firm	4 = Clearing Firm (firm to which trade is given up) (PartyIDSource = D)	Relevant when the marketplace issues an id-number for those actors. Applicable e.g. when the actor has a connection to the marketplace.
		4 = Clearing Firm (firm to which trade is given up) (PartyIDSource = C)	Relevant when the marketplace supports mnemonic symbolic names for those actors. Applicable e.g. when the actor has a connection to the marketplace.
The client the broker gives up a trade to	Client	3 = ClientID (PartyIDSource = D)	Marketplaces often do not validate client identifiers, they are provided for use in downstream processes.

As in other cases, the user is encouraged to use the marketplace assigned identifier (PartyIDSource = D) as this does not require a lookup in the trading engine.

Account and pre-allocation information is passed to downstream systems in Trade Capture Reports confirming the trade and is normally not validated by the marketplace.

Note:

Note that Execution Reports echo the values back, but do not report the state of allocations.

Examples:

1. **Assigning the trade to a CSD (Central Securities Depoitory) account**:
   - NestedPartyID (524) = "NCSD" - National CSD

- NestedPartyIDSource (525) = "C"
- NestedPartyRole (538) = "10" - Settlement Location
- AllocAccount (79) = account at the CSD
- AllocAcctIDSource (661) = "99" - Other

2. **Assigning the trade to an account at the back office of the executing firm**:
   - AllocAccount (79) = account at the back office
   - AllocAcctIDSource (661) = "99" - Other
3. **Giving up the trade to a clearing firm**:
   - NestedPartyID (524) = the ID of the Clearing Firm
   - NestedPartyIDSource (525) = "C" or "D"
   - NestedPartyRole (538) = "4" - Clearing Firm
   - AllocAccount (79) = account at the Clearing Firm
   - AllocAcctIDSource (661) = "99" - Other
4. **Giving up the trade to another broker firm**:
   - NestedPartyID (524) = the ID of the Broker Firm
   - NestedPartyIDSource (525) = "C" or "D"
   - NestedPartyRole (538) = "1" - Executing Firm
   - AllocAccount (79) = account at the Broker Firm
   - AllocAcctIDSource (661) = "99" - Other
5. **Assigning the trade to an account at the CCP (Central CounterParty / Clearing House)**:
   - NestedPartyID (524) = the ID of the Clearing House
   - NestedPartyIDSource (525) = "C" or "D"
   - NestedPartyRole (538) = "21" - Clearing Organization
   - AllocAccount (79) = account at the Clearing House
   - AllocAcctIDSource (661) = "99" - Other

4.5 Queries, Subscriptions and Drop Copies

FIX supports subscription for a large variety of messages. Subscriptions however never apply to response messages in interactive workflows, in those cases the response will always be produced and returned over the FIX session from where the request came. If other actors (FIX sessions) needs the same information, a subscription may be needed.

FIX subscriptions are always based on a Request message. Request messages contain a SubscriptionRequestType (263) which offers the following subscription features:

- 1 - Snapshot + Updates (Subscribe)
- 2 - Disable previous Snapshot + Update Request (Unsubscribe)

FIX also supports queries for messages, generally the same messages for which subscriptions are supported. Again the SubscriptionRequestType (263) field is used:

- 0 - Snapshot

Both subscriptions and queries contain filtering mechanisms as defined per message type. Messages returned as the result of a query or subscription will contain the ID of the subscription request message.

In general, this implementation is based on pushing our information to the eligible actors (FIX sessions) and so has a conservative view on queries and subscriptions (pull mechanisms). For details on what queries and subscriptions are supported, please see the respective chapters describing the functionality.

FIX also supports the concept of "drop copies", i.e. copies of outbound messages to FIX sessions other than the primary receiver. Drop copies can be seen a pre-defined subscriptions, so they are not applicable in cases where this implementation supports FIX subscriptions. Drop copies are produced based on a configuration at the marketplace side and must be requested outside the scope of FIX messaging. Drop copies are indicated as such through the use of the following field:

- CopyMsgIndicator (797) = True

Drop copies are particularly relevant to support participant fail-over solutions, i.e. when a participant e.g. has a secondary Order Management System (OMS) that needs to be fed the same information as the primary OMS. Drop copies can also be relevant for trade confirmations in the case where a second party is responsible for certain back office processes.

Note:

Queries, subscriptions and drop copies are never needed when the marketplace supported work flows includes the production of separate message to the relevant receivers. Examples:

- Confimed trades (Trade Capture Reports) are automatically forwarded to the counterparties, Central Securities Depository (CSD), Clearing House (CCP), regulators and vendors.
- When reporting privately negotiated trades that are to be acknowledged by the counterparty, the Trade Capture Report is automatically, by configuration, forwarded to a FIX session of the counterparty
- In quote negotiations, the Quote Request is automatically, by configuration, forwarded to a FIX session of the relevant quote issuer(s).

4.6 Error / Warning Messages

Many inbound message contain the Text (58), EncodedTextLen (354) and EncodedText (355) fields. Users should be aware that those fields may be overwritten in responses. The fields can be used to provide additional error-/warning text.

Internal system errors not related to FIX will be returned specifying 2 = Broker / Exchange Option and the Text (58) field will contain the following text:

[SYSTEM];[ERROR CODE];[DESCRIPTION]

Where SYSTEM denotes a specific back-end system, the ERROR CODE is an internal error code, and the DESCRIPTION is a brief description of the error.

4.7 The Business Message Reject Message

The Business Message Reject is used to report rejections in situations where other reject messages are not available, e.g. when the inbound message does not reach the trading engine due to trading being closed or authorization not sufficient.

The user must be prepared to receive this message as an alternative response to all other business messages. For message details, please see General Messages on page 127.

Table 16: Business Message Reject

Reject Message	Direction	Business Message	Comment
Business Message Reject	Out	Any	Primarily used when there is no other FIX reject message. Also used: • When the trading engine is not available, i.e. the incoming message can not be compared to the state of the order book (e.g. verifying that there are no conflicting order identifiers in the book). • In response to update and status requests when the user is not authorized to see the state of the order (may e.g. happen if a user tries to update another user's order).

4.8 The Don't Know Trade DK Message

The Don't Know Trade message is used by external actors to reject Execution Reports they do not recognize (see the Execution Report message for further details). It should be noted that this is regarded as an abnormal situation and will be handled as such.

For message details, please see General Messages on page 127.

Table 17: Don't Know Trade

Reject Message	Direction	Business Message	Comment
Don't Know Trade DK	In	Execution Report	In practice only used when a user does not recognize a trade (Execution Report). The marketplace does not accept incoming Execution Reports at all.

5 Single and General Order Handling

5.1 Business Message Types

The marketplace supports the message types described in the following tables. Full details of the messages and workflows around them are available in other parts of the document.

Table 18: *Business Messages*

In/Out	Message Name	Comment
In	**New Order Single**	
In	**Order Cancel Replace Request**	Used to modify an order. Whether the order is re-ranked or not is subject to marketplace rules.
In	**Order Cancel Request**	
Out	**Execution Report**	
Out	**Order Cancel Reject**	
Out	**Business Message Reject**	Used to report rejections in situations where other reject messages are not available, e.g. when the inbound message does not reach the trading engine due to trading being closed or authorization not sufficient.

5.2 Order Routing

5.2.1 Main Workflow

5.2.1.1 New Order

The order workflow starts with the user submitting a New Order Single message. In response an Execution Report is produced. The Execution Report is a reply directed to the sender of the order and will contain details of the order. If the order is rejected the Execution Report will contain relevant error messages.

5.2.1.2 Fills

When an order is filled the Execution Report will contain details about the fill. In addition, a Trade Capture Report will be produced. The principal differences between the two are:

Execution Reports

are messages directed to the sender of the order and is primarily intended for front-office purposes. It captures order status information as well as fill information (if applicable).

Trade Capture Reports

are messages capturing the trade as such and is primarily intended for downstream processing. The Trade Capture Report is used to inform a variety of parties about a trade, e.g.: broker backoffice; clearing firms;

clearing houses; depositories and; regulators. As such downstream processing occurs at various locations and for different purposes, the Trade Capture Report message might look slightly different depending on the receiver.

Trade Capture Report messages are also used for a large number of other purposes, including reporting of privately negotiated trades and relaying trades to parties not directly involved in the trade - but this is outside the scope of this chapter.

5.2.1.3 Market Data

When an order is accepted into the order book or a trade occurs, that information will be relayed using public market data messages (Market Data Incremental Refresh and Market Data Snapshot Refresh).

For a complete description of Market Data, please see chapter Market Data on page 95.

5.2.1.4 Order Modification

Order modification is accomplished through the use of the Order Cancel Replace Request message. Despite its name, it represents a modification of the existing order, not removing the old order and replacing it with a new one. However, an order modification is not a delta change to order instructions; the message must contain all order details.

An Execution Report will relay the new state of the order.

5.2.1.5 Order Cancellation

- If the user wishes to cancel a single previously sent order, the Order Cancel Request message is used.
- Execution Reports are issued relaying the status of every canceled order.

5.2.1.6 Order Reject

There are a number of different messages used to reject order messages:

Table 19: Order Reject Messages

Reject Message	Direc-tion	Business Message	Comment
Execution Report	Out	New Order Single	
Order Cancel Reject	Out	Order Cancel Replace Request Order Cancel Request	

5.2.2 Order Features

5.2.2.1 Order Identification

5.2.2.1.1 Client Order ID

Any message related to an order (entry, cancellation, modification) sent by the client, must have a unique identifier in the ClOrdID (11) field. As the standard indicates, the uniqueness of these identifiers must be

maintained during the trading session. If orders with a duration of more than one trading session are used, the sender needs to cater for uniqueness across those.

GENIUM primarily requires that the ClOrdID is unique per FIX session, order book and among active orders only.

Once the message is accepted by the trading engine, the client receives the corresponding confirmation message with the same ClOrdID. In cases where the user immediately after sending an order wants to modify or cancel it, this can be achieved by referring to the initial order in the OrigClOrdID (41) field of the subsequent message.

In cases where a third party delivers orders to the marketplace on behalf of the broker, the ClOrdID may have to be assigned by that third party. When routing messages back to the originator, the third party must then be able to roll the respective ClOrdIDs so they reflect the IDs used by **its own** clients.

Client Order IDs when the Firm uses multiple FIX sessions

Firms using multiple front-end trading applications or multiple FIX sessions should be aware of the following:

- In cases where the exchange offers drop copies of Execution Reports to FIX sessions other than the one that submitted the order, those drop copy Execution Reports will **not** contain a ClOrdID. The reason for excluding the ClOrdID in those cases is that various FIX sessions or the underlying trading applications might use conflicting ClOrdIDs.
- The above applies also in cases where exchange business operations perform order management on behalf of the order owner.
- If an order entered through one FIX session is updated using another FIX session based on a drop copy Execution Report, the only way to identify the order to be updated is by using the exchange assigned OrderID (as the OrigClOrdID cannot be provided).

5.2.2.1.2 Order ID

The OrderID (37) field is the globally unique order identifier assigned by the marketplace. This identifier is static and stays with the order even when it is modified. It is never disclosed to counterparties in market data or elsewhere, it stays private between the order owner and the marketplace.

Users are encouraged to provide the OrderID on order updates and cancellations whenever possible, i.e. in all cases except for submitting order actions before the new order ack (Execution Report) is received. The OrderID is the preferred identifier for order modification and cancellation as it is the identifier used internally in the trading engine. Use of other identifiers requires a lookup which increases message latency. As use of the OrderID requires the user to wait for an order acknowledgement from the trading engine, immediate actions require the use of the OrigClOrdID (41) reference field.

This field could be necessary to identify the order in communications with the market by other means than FIX.

5.2.2.1.3 Secondary Order ID

The SecondaryOrderID (198) field is a public order identifier assigned by the marketplace. This means that the ID is published in Market Data in cases where the exchange uses Order Depth market data views and has the practice of disclosing an order identifier. Publication of the SecondaryOrderID is necessary in order to:

- Support Hit / Take matching models where a user selects a certain order to execute against.
- Allow firms and users to identify their own orders in the public book.

The SecondaryOrderID is unique for the entire marketplace.

The SecondaryOrderID field received in the reply to an order modification need not coincide with the one initially received. The reason for this is that the order is assigned a new SecondaryOrderID every time it is reranked. Thus, hidden properties (as reserve size, pegged prices, etc) of the order are less easy to identify for other market participants.

5.2.2.2 Trade Identification

5.2.2.2.1 Execution ID

The ExecID (17) field is not an identifier of trades. It is an identifier assigned to each unique Execution Report message produced by the marketplace, without duplicates during the entire FIX session. The ExecID is in fact globally unique.The ExecID also allows the user to connect Execution Reports for fills and partial fills with the subsequent Trade Capture Report messages.

5.2.2.2.2 Trade Match ID

The TrdMatchID (880) field contains the match identifier. This is the identifier assigned by the trading engine to the deal referred to in the message. A deal in this context is considered all orders matched against one aggressive order in one execution round. The period in which the uniqueness of this field is guaranteed is determined by the trading engine.

5.2.2.3 Order Types

An order must specify its type, or more specifically how the price of the order is expressed.

Table 20: *Supported OrdType (40) Values*

Value	Name	Description
1	Market	
2	Limit	
Q	Counter-Order Selection	An order that will execute only against a specific order in the book. See chapter Hit / Take on page 40.

5.2.2.4 Order Expiry

An order can specify various conditions for when or how it should expire or be automatically removed from the book.

Table 21: *Supported TimeInForce (59) values*

Value	Name	Description
0	Day (or Session)	Default value
1	Good Till Cancel (GTC)	
2	At the Opening (OPG)	
3	Immediate or Cancel (IOC)	
4	Fill or Kill (FOK)	
5	Good Till Crossing	The order is valid up to, but excluding the next call auction.
6	Good Till Date	

Value	Name	Description
7	At the Close	
y	Good Through Crossing	The order is valid up to and including the next call auction.
z	At Next Crossing	The order is valid in the next call auction only (activated when the auction starts if entered before)

Orders can be canceled using one of the following transaction(s)

- Order Cancel

Orders can also be canceled as defined in marketplace rules, e.g. due to events such as corporate actions processing.

An expired order will show OrdStatus (39) as 4 = Canceled or C = Expired. The former is according to FIX convention used e.g. for Immediate-Or-Cancel and Fill-Or-Kill orders while the latter is typically used when orders expire at the end of a trading day.

5.2.2.5 Price Conditions

An order can specify conditions related to its price in various ways.

Table 22: Supported OrdType (40) values

Value	Name	Description
1	Market	The Price (44) field is not used, the order executes against the best prices order on the opposite side.
2	Limit	The Price (44) field is specified and the order will execute at this price or better.

Table 23: Supported ExecInst (18) values

Value	Name	Description
i	Imbalance Only	A type of market order that is ranked last (instead of first) among orders executable at the equilibrium price of a call auction. Does not specify the Price (44) field.

Average Price Execution - An order can specify that it is willing to trade at worse prices if this is compensated by execution against better prices so the average price of the execution round is at the provided limit price or better. This is accomplished by specifying:

- DiscretionInst (388) = 7 (Average Price Guarantee).

5.2.2.6 Quantity Conditions

An order can specify various types of quantity conditions.

Minimum Quantity, MinQty (110). Means the order will not be filled unless the first execution round yields at least the quantity specified in this field.

Match Increment, MatchIncrement (1089). Allows orders to specify a minimum quantity that applies to every execution (one execution could be for multiple counter-orders). The order may still fill against smaller orders, but the cumulative quantity of the execution must be in multiples of the MatchIncrement.

Table 24: Supported ExecInst (18) values

Value	Name	Description
G	All or None - AON	Means the full order quantity must be filled in one execution round. **Note:** Note, whenever using this value to specify an All or None order the Miniumum Quantity field (if provided) must be equal to the total quantity.

Reserve Quantity (a.k.a. "Hidden" or "Iceberg" Orders)

Reserve orders allow users to hide the full size of their order and thereby potentially limit its influence on prices. The following options are available:

DisplayQty (1138): Traditionally used to indicate reserve quantity. To indicate a single level of reserve quantity, DisplayQty should be used.

Also see examples in chapter Reserve Instructions on page 64.

5.2.2.7 Hit / Take

Some markets allow users to hit / take a specific order, thereby bypassing order ranking rules. This is accomplished as follows:

1. Look up the order in the public book (as made available in market data)
2. Take the SecondaryOrderID (198) of the order in the public book and add it to the Hit/Take order field RefOrderID (1080).
3. Set the RefOrderIDSource (1081) = 0 (SecondaryOrderID)
4. Set the OrdType (40) = Q (Counter-order selection)
5. Fill in other relevant order properties and submit the order.

The order will only be able to match against the referenced order. Matching will only occur if the price and other criteria allow it. A Hit/Take order will never be stored in the book; it will be expire immediately after the matching attempt.

Also see examples in chapter Hit / Take on page 66.

5.2.2.8 Call Auction Imbalance Orders

Some markets encourage all or certain brokers to contribute to close out an imbalance at the auction clearing price (a.k.a. Equilibrium or Indicative Auction Price) of call auctions. Imbalance orders are unpriced orders that are ranked last among all orders at the auction clearing price. Among the imbalance orders, they are filled in time priority.

A user wanting to enter an imbalance order specifies:

- No Price (44)
- ExecInst (18) = i (Imbalance only). Please note the lowercase "i" is used.

5.2.2.9 Pre-Trade Anonymity or Disclosure

Some market segments allow a user to request pre-trade anonymity where the counterparty transparency is the normal case, or disclosure where counterparty transparency is not. Overriding counterparty transparency is applicable to pre-trade market data only. Applicability may be restricted to orders over a pre-defined quantity or other rules.

A user wishing to override the standard transparency rules specifies:

- PreTradeAnonymity (1091)
  - True = Request anonymity in a market where counterparties are normally disclosed
  - False = Request disclosure in a market where counterparties are normally anonymous.

5.2.2.10 Accounts, Pre-Allocation and Give Ups

Please refer to Accounts, Pre-Allocation and Give Ups on page 29.

5.2.2.11 Order Capacity

The marketplace may require that the order capacity is specified, often for regulatory reasons. The order capacity is specified in the OrderCapacity (528) field as either of:

- A = **Agency**. Used for orders on behalf of clients.
- P = **Principal**. Used for own account trading.

In some cases the basic order capacity requires further specification in the OrderRestrictions (529) field, e.g.:

- 5 = **Acting as Market Maker or Specialist in the security**
- Y = **Issuer Holding**
- Z = **Issue Price Stabilization**

5.2.2.12 Text

FIX messages provide the possibility of sending a text in two fields:

- Text (58)

Text-fields are never relayed back in confirmation messages, so they are uni-directional. Although incoming messages to the marketplace are allowed to contain text, it is not evaluated or acted upon.

Users should be aware that outgoing messages may include error or warning information in the text fields. For more information, please see chapter Business Layer Introduction on page 25.

5.2.3 Order State Changes

Order state changes are divulged in Execution Report messages. Every state change is communicated in an Execution Report. For best performance and maximum user friendliness, in cases where an order pass

through multiple state changes in a single processing round (e.g. new, partially filled and then filled), a single Execution Report message is produced. The standard FIX practice of relaying intermediary states too could lead to users acting on stale states.

An order can be in the following intermediate states:

- **New**. This state is applicable when an order is accepted by the trading engine and is not in immediately transitioned into any other state:
  - The order is put on the book but not (partially) filled
  - The order is held outside the book waiting for activation, e.g. due to a stop condition or for a session change (as e.g. for an At-the-Close order).
- **Partially filled**.
- **Done for Day**. *Currently not supported*. Relevant for order with validity for more than one trading day. In cases where the market has a closed state, the marketplace may optionally relay this state.
- **Suspended**. Relevant in markets that support suspension (and reactivation) of orders.

The following are final states, indicating that the order is no longer in the book and no longer available for updates or status requests:

- **Rejected**. The order did not pass validation rules.
- **Canceled**. The order was removed from the system due to a cancellation request.
- **Filled**. The order is completely filled.
- **Expired**. The order has expired as defined by exlicit or implicit time in force conditions.

The following diagram depicts the supported state changes applicable for orders:

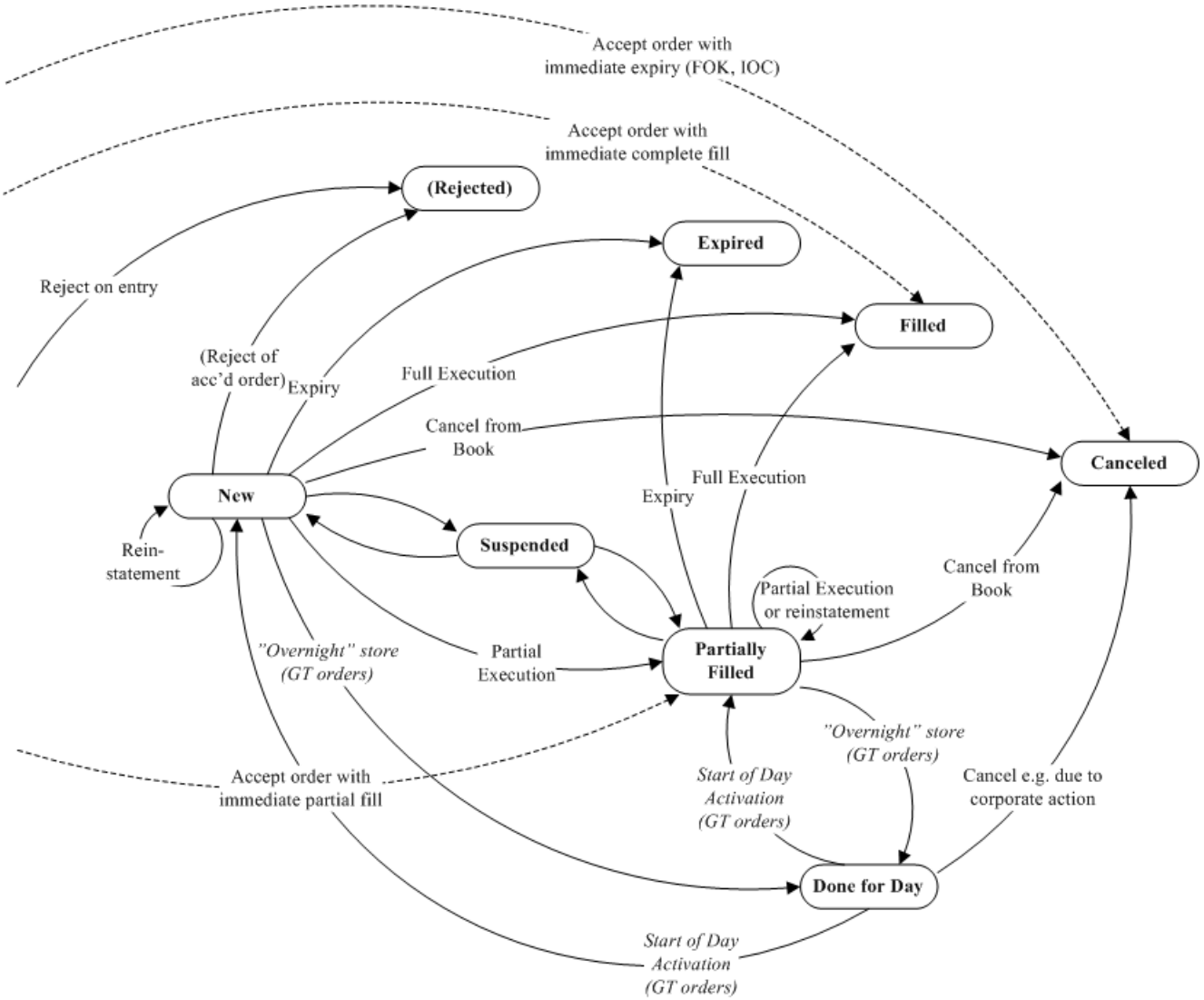

Figure 1: Order Status State Event Chart

Note:

Note that an order in the Done for Day status will not revert to Suspended as the former has a higher precedence value. See the Execution Reports chapter in the FIX standard specification [1].

5.2.4 Suspension of Orders at Connection Loss

Subject to marketplace rules, a participant can choose to specify that orders should be suspended in the event of a connection loss. A suspension means the orders are taken out of the book, are not executable and get the “Suspended” order status. The orders are not canceled, meaning they do not need to be re-entered as new orders when the connection is re-established. Orders that should be suspended on connection loss are to be marked with ExecInst (18) = z (Suspend on Connection Loss). Alternatively, the marketplace suspends all orders and do not require the indicator.

The user can choose to cancel or activate all the orders when the connection is re-established. Orders that are reactivated will return to their previous order status but may also be immediately passed into a new status, e.g. from “New” to “Partially Filled” (subject to market rules).

Activation of individual orders is done using the Order Cancel Replace Request message (see workflow in M - Connection Loss on page 64). Cancelation is done using the ordinary Cancel messages.

5.3 Message Details

For messages that are not specific for Orders, please see chapter General Messages on page 127.

5.3.1 NewOrderSingle

Table 25: *NewOrderSingle*

Tag	FieldName	Req'd	Comments	Format
	StandardHeader	Y	MsgType = D	
11	ClOrdID	Y	Unique identifier of the order as assigned by institution or by the intermediary (CIV term, not a hub/service bureau) with closest association with the investor.	String
	Parties	N	Insert here the set of "Parties" (firm identification) fields defined in "Common Components of Application Messages"	
1	Account	N		String
660	AcctIDSource	N	**OMX Comment:** Pass through to clearing, not used in trading engine	int
70	AllocID	N	Used to assign an overall allocation id to the block of preallocations	String
	PreAllocGrp	N	Number of repeating groups for pre-trade allocation	
18	ExecInst	N	Can contain multiple instructions, space delimited. If OrdType=P, exactly one of the following values (ExecInst = L, R, M, P, O, T, W, a, d) must be specified.	MultipleCharValue
110	MinQty	N		Qty
1089	MatchIncrement	N		Qty
	DisplayInstruction	N		
	TrdgSesGrp	N	Specifies the number of repeating TradingSessionIDs	
	Instrument	Y	Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages" **OMX Comment:** Messages shall use identifier fields only.	
54	Side	Y		char
60	TransactTime	Y	Time this order request was initiated/released by the trader, trading system, or intermediary.	UTCTimestamp

Tag	FieldName	Req'd	Comments	Format
	OrderQtyData	Y	Insert here the set of "OrderQtyData" fields defined in "Common Components of Application Messages"	
40	OrdType	Y		char
44	Price	N	Required for limit OrdTypes. For F/X orders, should be the "all-in" rate (spot rate adjusted for forward points). Can be used to specify a limit price for a pegged order, previously indicated, etc.	Price
59	TimeInForce	N	Absence of this field indicates Day order	char
432	ExpireDate	N	Conditionally required if TimeInForce = GTD and ExpireTime is not specified.	LocalMktDate
126	ExpireTime	N	Conditionally required if TimeInForce = GTD and ExpireDate is not specified.	UTCTimestamp
528	OrderCapacity	N		char
529	OrderRestrictions	N		MultipleCharValue
1091	PreTradeAnonymity	N		Boolean
58	Text	N	**OMX Comment:** Note that Text is not relayed back to sender!	String
	DiscretionInstructions	N	Insert here the set of "DiscretionInstruction" fields defined in "Common Components of Application Messages" **OMX Comment:** Use: - DiscretionOffsetValue (389) when submitting a "discretionary price" - DiscretionInst (388) = 7 (Average Price Guarantee)	
1080	RefOrderID	N	Required for counter-order selection / Hit / Take Orders. (OrdType = Q)	String
1081	RefOrderIDSource	N	Conditionally required if RefOrderID is specified. **OMX Comment:** Valid values (also default if unspecified for Hit/Take order, i.e. if RefOrderID is specified): - 0 = SecondaryOrderID	char
	StandardTrailer	Y		

5.3.2 ExecutionReport

Table 26: ExecutionReport

Tag	FieldName	Req'd	Comments	Format
	StandardHeader	Y	MsgType = 8	
37	OrderID	Y	OrderID is required to be unique for each chain of orders.	String

Tag	FieldName	Req'd	Comments	Format
			OMX Comment: Static id assigned by trading engine to new orders, otherwise from order. Not publicly disclosed.	
198	SecondaryOrderID	N	Can be used to provide order id used by exchange or executing system. **OMX Comment:** Marketplace assigned id that changes when order is reranked. May be dislosed in market data	String
526	SecondaryClOrdID	N	In the case of quotes can be mapped to: - QuoteID (117) of a single Quote - QuoteEntryID (299) of a Mass Quote **OMX Comment:** FIX 5.0 SP1	String
11	ClOrdID	N	Required for executions against electronically submitted orders which were assigned an ID by the institution or intermediary. Not required for orders manually entered by the broker or fund manager (for CIV orders). From order. In the case of quotes can be mapped to: - QuoteMsgID (1166) of a single Quote - QuoteID (117) of a Mass Quote **OMX Comment:** FIX 5.0 SP1. Required when referring to orders that where electronically submitted over FIX or otherwise including a ClOrdID.	String
41	OrigClOrdID	N	Conditionally required for response to an electronic Cancel or Cancel/Replace request (ExecType=PendingCancel, Replace, or Canceled). ClOrdID of the previous accepted order (NOT the initial order of the day) when canceling or replacing an order. **OMX Comment:** Conditionally required only when referring to orders electronically submitted over FIX or otherwise including a ClOrdID.	String
	Parties	N	Insert here the set of "Parties" (firm identification) fields defined in "Common Components of Application Messages"	
17	ExecID	Y	Unique identifier of execution message as assigned by sell-side (broker, exchange, ECN) (will be 0 (zero) forExecType=I (Order Status)). **OMX Comment:** Assigned by marketplace. Links Execution Report (ExecType = F [Trade]) to Trade Capture Report (TradeHandlingInstr = 0 [Trade Confirm])	String
880	TrdMatchID	N	**OMX Comment:** Currently not in FIX 5.0. OMX requests extension.	String
150	ExecType	Y	Describes the purpose of the execution report. **OMX Comment:** Assigned by marketplace	char
70	AllocID	N	**OMX Comment:** Currently not in FIX 5.0. OMX requests extension.	String
	PreAllocGrp	N	Use to echo back pre-allocation details from order. (Not used to report status of allocation.)	

Tag	FieldName	Req'd	Comments	Format
			OMX Comment: Currently not in FIX 5.0. OMX requests extension.	
39	OrdStatus	Y	Describes the current state of a CHAIN of orders, same scope as OrderQty, CumQty, LeavesQty, and AvgPx **OMX Comment:** Assigned by marketplace	char
636	WorkingIndicator	N	For optional use with OrdStatus = 0 (New) **OMX Comment:** Assigned by marketplace	Boolean
103	OrdRejReason	N	For optional use with ExecType = 8 (Rejected) **OMX Comment:** Assigned by marketplace	int
378	ExecRestatementReason	N	Required for ExecType = D (Restated). **OMX Comment:** Assigned by marketplace	int
1	Account	N	Required for executions against electronically submitted orders which were assigned an account by the institution or intermediary **OMX Comment:** From order	String
660	AcctIDSource	N	**OMX Comment:** Currently not suported Assigned by marektplace	int
574	MatchType	N	**OMX Comment:** Assigned by marketplace for ExecType = Trade	String
1115	OrderCategory	N	**OMX Comment:** Assigned by marketplace for ExecType = Trade	char
	Instrument	Y	Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages" **OMX Comment:** From order, but marketplace responds with SecurityIDSource = 8, SecurityID and one standard identifier which depends on market conventions	
54	Side	Y	**OMX Comment:** From order	char
	OrderQtyData	N	Insert here the set of "OrderQtyData" fields defined in "Common Components of Application Messages" **IMPORTANT NOTE: OrderQty field is required for Single Instrument Orders unless rejecting or acknowledging an order for a CashOrderQty or PercentOrder. ** **OMX Comment:** From order	
1093	LotType	N	**OMX Comment:** From order	char
40	OrdType	N	**OMX Comment:** From order but might change due to triggers	char
44	Price	N	Required if specified on the order **OMX Comment:** From order but might change due to triggers, pegs, etc	Price
	DiscretionInstructions	N	Insert here the set of "DiscretionInstruction" fields defined in "Common Components of Application Messages"	

Tag	FieldName	Req'd	Comments	Format
			OMX Comment: From order	
845	DiscretionPrice	N	The current discretionary price of the order **OMX Comment:** Marketplace assigned	Price
59	TimeInForce	N	Absence of this field indicates Day order **OMX Comment:** From order Marketplace assigns default (0) if left out on order	char
432	ExpireDate	N	Conditionally required if TimeInForce = GTD and ExpireTime is not specified. **OMX Comment:** From order	LocalMktDate
126	ExpireTime	N	Conditionally required if TimeInForce = GTD and ExpireDate is not specified.	UTCTimes-tamp
18	ExecInst	N	Can contain multiple instructions, space delimit-ed. **OMX Comment:** From order Marketplace might assign S = Suspend	MultipleCharVal-ue
1057	AggressorIndicator	N		Boolean
528	OrderCapacity	N	**OMX Comment:** From order	char
529	OrderRestrictions	N		MultipleCharVal-ue
1091	PreTradeAnonymity	N	**OMX Comment:** From order	Boolean
32	LastQty	N	Quantity (e.g. shares) bought/sold on this (last) fill. Required if ExecType = Trade or Trade Cor-rect. If ExecType=Stopped, represents the quantity stopped/guaranteed/protected for. **OMX Comment:** Assigned by marketplace	Qty
31	LastPx	N	Price of this (last) fill. Required if ExecType = Trade or Trade Correct. Should represent the "all-in" (LastSpotRate + LastForwardPoints) rate for F/X orders.). If ExecType=Stopped, represents the price stopped/guaranteed/protected at. Not required for FX Swap when ExecType = Trade or Trade Correct as there is no "all-in" rate that applies to both legs of the FX Swap. **OMX Comment:** Assigned by marketplace	Price
336	TradingSessionID	N	**OMX Comment:** Assigned by marketplace (Note not from order). Specifies the trading session when the Execution Report was produced. Trading Sessions from the Order are not relayed in the Execution Report!	String
625	TradingSessionSubID	N	**OMX Comment:** Assigned by marketplace (Note not from order). Specifies the trading session when the Execution Report was produced. Trading Sessions from the Order are not relayed in the Execution Report!	String
151	LeavesQty	Y	Quantity open for further execution. If the OrdSta-tus is Canceled, DoneForTheDay, Expired, Cal-culated, or Rejected (in which case the order is	Qty

Tag	FieldName	Req'd	Comments	Format
			no longer active) then LeavesQty could be 0, otherwise LeavesQty = OrderQty - CumQty. **OMX Comment:** Assigned by marketplace	
14	CumQty	Y	Currently executed quantity for chain of orders. **OMX Comment:** Assigned by marketplace	Qty
75	TradeDate	N	Used when reporting other than current day trades.	LocalMktDate
60	TransactTime	N	Time the transaction represented by this ExecutionReport occurred **OMX Comment:** Assigned by marketplace. Set by the trading engine. Can be used in order updates to avoid update on stale order information	UTCTimestamp
110	MinQty	N	**OMX Comment:** From order	Qty
1089	MatchIncrement	N	**OMX Comment:** From order	Qty
	DisplayInstruction	N	Insert here the set of "DisplayInstruction" fields defined in "common components of application messages" **OMX Comment:** From order	
58	Text	N	**OMX Comment:** Assigned by marketplace (Note not from order). May contain message from the marketplace	String
	StandardTrailer	Y		

5.3.3 OrderCancelReplaceRequest

Table 27: OrderCancelReplaceRequest

Tag	FieldName	Req'd	Comments	Format
	StandardHeader	Y	MsgType = G	
37	OrderID	N	Unique identifier of most recent order as assigned by sell-side (broker, exchange, ECN).	String
	Parties	N	Insert here the set of "Parties" (firm identification) fields defined in "Common Components of Application Messages"	
41	OrigClOrdID	N (Y)	ClOrdID of the previous non rejected order (NOT the initial order of the day) when canceling or replacing an order. **OMX Comment:** Required when referring to orders that where electronically submitted over FIX or otherwise including a ClOrdID. The Exchanges/ECNs Working Group proposes the field is made optional as a reference to the OrderIID is sufficient.	String
11	ClOrdID	Y	Unique identifier of replacement order as assigned by institution or by the intermediary with closest association with the investor.. Note that this identifier will be used in ClOrdID field of the	String

Tag	FieldName	Req'd	Comments	Format
			Cancel Reject message if the replacement request is rejected.	
1	Account	N		String
660	AcctIDSource	N		int
70	AllocID	N	Used to assign an overall allocation id to the block of preallocations	String
	PreAllocGrp	N	Number of repeating groups for pre-trade allocation	
18	ExecInst	N	Can contain multiple instructions, space delimited. Replacement order must be created with new parameters (i.e. original order values will not be brought forward to replacement order unless redefined within this message).	MultipleCharValue
110	MinQty	N		Qty
1089	MatchIncrement	N		Qty
	DisplayInstruction	N	Insert here the set of "DisplayInstruction" fields defined in "common components of application messages"	
	TrdgSesGrp	N	Specifies the number of repeating TradingSessionIDs	
	Instrument	Y	Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages" Must match original order **OMX Comment:** Messages shall use identifier fields only. Instrument / Order book data cannot be modified	
54	Side	Y	Should match original order's side, however, if bilaterally agreed to the following groups could potentially be interchanged: Buy and Buy Minus Sell, Sell Plus, Sell Short, and Sell Short Exempt Cross, Cross Short, and Cross Short Exempt **OMX Comment:** Side cannot be updated	char
60	TransactTime	Y	Time this order request was initiated/released by the trader or trading system.	UTCTimestamp
	OrderQtyData	Y	Insert here the set of "OrderQtyData" fields defined in "Common Components of Application Messages" Note: OrderQty value should be the "Total Intended Order Quantity" (including the amount already executed for this chain of orders)	
40	OrdType	Y		char
44	Price	N	Required for limit OrdTypes. For F/X orders, should be the "all-in" rate (spot rate adjusted for forward points). Can be used to specify a limit price for a pegged order, previously indicated, etc.	Price

Tag	FieldName	Req'd	Comments	Format
	DiscretionInstructions	N	Insert here the set of "DiscretionInstruction" fields defined in "Common Components of Application Messages"	
59	TimeInForce	N	Absence of this field indicates Day order	char
432	ExpireDate	N	Conditionally required if TimeInForce = GTD and ExpireTime is not specified.	LocalMktDate
126	ExpireTime	N	Conditionally required if TimeInForce = GTD and ExpireDate is not specified.	UTCTimes-tamp
528	OrderCapacity	N		char
529	OrderRestrictions	N		MultipleCharVal-ue
1091	PreTradeAnonymity	N		Boolean
58	Text	N		String
	StandardTrailer	Y		

5.3.4 OrderCancelRequest

Table 28: *OrderCancelRequest*

Tag	FieldName	Req'd	Comments	Format
	StandardHeader	Y	MsgType = F	
41	OrigClOrdID	N (Y)	ClOrdID of the previous non-rejected order (NOT the initial order of the day) when canceling or replacing an order. **OMX Comment:** Required when referring to orders that where electronically submitted over FIX or otherwise including a ClOrdID. The Exchanges/ECNs Working Group proposes the field is made optional as a reference to the OrderIID is sufficient.	String
37	OrderID	N	Unique identifier of most recent order as assigned by sell-side (broker, exchange, ECN).	String
11	ClOrdID	Y	Unique ID of cancel request as assigned by the institution.	String
	Parties	N	Insert here the set of "Parties" (firm identification) fields defined in "Common Components of Application Messages"	
	Instrument	Y	Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages" **OMX Comment:** Messages shall use identifier fields only.	
54	Side	Y		char
60	TransactTime	Y	Time this order request was initiated/released by the trader or trading system.	UTCTimes-tamp

Tag	FieldName	Req'd	Comments	Format
	OrderQtyData	Y	Insert here the set of "OrderQtyData" fields defined in "Common Components of Application Messages" Note: OrderQty = CumQty + LeavesQty (see exceptions above) **OMX Comment:** Field is mandatory in standard FIX, but will not be used by OMX	
	StandardTrailer	Y		

5.3.5 OrderCancelReject

Table 29: *OrderCancelReject*

Tag	FieldName	Req'd	Comments	Format
	StandardHeader	Y	MsgType = 9	
37	OrderID	Y	If CxlRejReason="Unknown order", specify "NONE".	String
11	ClOrdID	Y	Unique order id assigned by institution or by the intermediary with closest association with the investor. to the cancel request or to the replacement order.	String
41	OrigClOrdID	N (Y)	ClOrdID which could not be canceled/replaced. ClOrdID of the previous accepted order (NOT the initial order of the day) when canceling or replacing an order. **OMX Comment:** Required when referring to orders that where electronically submitted over FIX or otherwise including a ClOrdID. The Exchanges/ECNs Working Group proposes the field is made optional as a reference to the OrderIID is sufficient.	String
39	OrdStatus	Y	OrdStatus value after this cancel reject is applied. If CxlRejReason = "Unknown Order", specify Rejected.	char
1	Account	N		String
660	AcctIDSource	N		int
60	TransactTime	N		UTCTimestamp
434	CxlRejResponseTo	Y	**OMX Comment:** Valid values:1 = Order Cancel Request 2 = Order Cancel/Replace Request	char

Tag	FieldName	Req'd	Comments	Format
102	CxlRejReason	N	**OMX Comment:** Valid values: 0 = Too late to cancel 1 = Unknown order 2 = Broker / Exchange Option 4 = Unable to process Order Mass Cancel Request 5 = OrigOrdModTime (586) did not match last TransactTime (60) of order 6 = Duplicate ClOrdID (11) received 99 = Other	int
58	Text	N	**OMX Comment:** Contains a specified error number and message acording to separate specification	String
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.	Length
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.	data
	StandardTrailer	Y		

5.4 Component Blocks (Order Specific)

For components that are not specific for Orders, please see chapter General Component Blocks on page 129.

5.4.1 Components

5.4.1.1 DiscretionInstructions

Table 30: *DiscretionInstructions*

Tag	FieldName	Req'd	Comments	Format
388	DiscretionInst	N	What the discretionary price is related to (e.g. primary price, display price etc) **OMX Comment:** Valid values: 7 - Average Price Guarantee when you allow the order to fill against worse priced orders as long as this is compensated by fills against better priced orders Unspecified - if defining a discretionary offset value	char

5.4.1.2 DisplayInstruction

Table 31: *DisplayInstruction*

Tag	FieldName	Req'd	Comments	Format
1138	DisplayQty	N		Qty

Tag	FieldName	Req'd	Comments	Format
1084	DisplayMethod	N	**OMX Comment:** Requeired if any other field of this component is provided.	char
1088	RefreshQty	N	Required when DisplayMethod = 2	Qty

5.4.1.3 OrderQtyData

Table 32: *OrderQtyData*

Tag	FieldName	Req'd	Comments	Format
38	OrderQty	N	One of CashOrderQty, OrderQty, or (for CIV only) OrderPercent is required. Note that unless otherwise specified, only one of CashOrderQty, OrderQty, or OrderPercent should be specified.	Qty

5.4.2 Implicit Components

5.4.2.1 PreAllocGrp

Table 33: *PreAllocGrp*

Tag	FieldName	Req'd	Comments	Format
78	NoAllocs	N	Number of repeating groups for pre-trade allocation **OMX Comment:** A single pre-allocation is allowed.	NumInGroup
>79	AllocAccount	N	Required if NoAllocs > 0. Must be first field in repeating group.	String
>661	AllocAcctIDSource	N		int
>467	IndividualAllocID	N		String
	NestedParties	N	Insert here the set of "Nested Parties" (firm identification "nested" within additional repeating group) fields defined in "Common Components of Application Messages" Used for NestedPartyRole=Clearing Firm	

5.5 Workflows

5.5.1 Introduction

The following workflows describe important aspects of the FIX interaction model. The FIX Protocol Specification includes many of the workflows herein defined. OMX workflows however adhere to the Best Practices issued by the FPL Exchange / ECNs Working Group. The main differences to standard FIX and those are:

- Orders are rejected at a single point (not dual points)

- Pending states are not supported
- Orders that have been canceled, expired or filled are not accessible and never restated
- The OrderID is shown in the tables

Workflows taken from the Order State Change Matrices chapter of volume 4 of the FIX standard specification are prefixed with the same chapters number (e.g. "A.1.a").

5.5.2 A - New Order

5.5.2.1 Vanilla

Table 34: A.1.a. - Filled order

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comment
1	New order (X)					10000				
2		*Execution (X)*		*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by marketplace*
2		Execution (X)	A	New	New	10000	0	10000	0	
3		Execution (X)	A	Trade	Partially Filled	10000	2000	8000	2000	Execution for 2000
4		Execution	A	Trade	Partially Filled	10000	3000	7000	1000	Execution for 1000
5		Execution	A	Trade	Filled	10000	10000	0	7000	Execution for remainding 7000

The FIX Standard specification has an A.1.b. workflow not shown here as OMX does not support the Done for Day order status.

The following table is not part of the FIX standard specification.

Table 35: Order immediately filled (completely) on entry into the book

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comment
1	New Order (X)					10000				
2		*Execution (X)*		*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by marketplace*
2		Execution (X)	A	Trade	Filled	10000	10000	0	10000	Execution for 10000

The following table is not part of the FIX standard specification.

Table 36: Order immediately partially filled on entry into the book

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comment
1	New order (X)					10000				
2		*Execution (X)*		*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by marketplace*
2		Execution (X)	A	Trade	Partially Filled	10000	2000	8000	2000	Execution for 2000
3		Execution (X)	A	Trade	Partially Filled	10000	3000	7000	1000	Execution for 1000

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comment
4		Execution (X)	A	Trade	Filled	10000	10000	0	7000	Execution for remainding 7000

5.5.3 B - Order Cancellation

Table 37: B.1.a. - Cancel request issued for a zero-filled order

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comment
1	New Order (X)					10000				
2		*Execution (X)*		*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by marketplace*
2		Execution (X)	A	New	New	10000	0	10000	0	
3	Cancel Request (Y, X)		A							
4		*Cancel Reject (Y, X)*	*A*	*New*						*If rejected by marketplace*
4		Execution (Y, X)	A	Canceled	Canceled	10000	0	0	0	Confirm that order has been canceled

Table 38: B.1.b. - Cancel request issued for a part-filled order – executions occur whilst cancel request is active

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comment
1	New Order (X)					10000				
2		*Execution (X)*		*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by marketplace*
2		Execution (X)	A	New	New	10000	0	10000	0	
3		Execution (X)	A	Trade	Partially Filled	10000	2000	8000	2000	Execution for 2000
4	Cancel Request (Y, X)		A							
5		Execution (X)	A	Trade	Partially Filled	10000	5000	5000	3000	Execution for 3000. This execution passes the cancel request on the connection
6		*Cancel Reject (Y, X)*	*A*		*Partially Filled*					*If rejected by marketplace*
6		Execution (Y, X)	A	Canceled	Canceled	10000	5000	0	0	'Canceled' order status takes precedence over 'partially filled' order status

Table 39: B.1.c. - Cancel request issued for an order that becomes filled before cancel request can be accepted

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comment
1	New Order (X)					10000				
2		*Execution (X)*		*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by marketplace*
2		Execution (X)	A	New	New	10000	0	10000	0	
3		Execution (X)	A	Trade	Partially Filled	10000	2000	8000	2000	Execution for 2000
4	Cancel Request (Y, X)		A							

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comment
5		Execution (X)	A	Trade	Partially Filled	10000	5000	5000	3000	Execution for 3000. This execution passes the cancel request on the connection
6		Execution (X)	A	Trade	Filled	10000	10000	0	5000	Execution for 5000. This execution passes the cancel request on the connection
7		*Cancel Reject (Y, X)*	*A*		*Rejected*					*Filled orders are taken out of the book.* *CxlRejectReason = 1 (unknown order)*

Table 40: *B.1.d. / B.1.e. - Cancel request issued for an order that has not yet been acknowledged*

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comment
1	New Order (X)					10000				
2	Cancel Request (Y, X)					10000				Order sender immediately wishes to cancel the order
3		Execution (X)	A	New	New	10000	0	10000	0	Order accepted before cancel request is processed.
4		Execution (Y, X)	A	Canceled	Canceled	10000	0	0	0	Order canceled

Table 41: *B.1.f. - Cancel request issued for an unknown order*

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comment
1	Cancel Request (Y, X)					10000				
2		*Cancel Reject (Y, X)*	NONE		*Rejected*					*Cancel request rejected with reject reason of "Unknown Order", OrdStatus is "Rejected" and OrderID is "NONE"*

5.5.4 C - Order Modification

5.5.4.1 C.1 Replace to Increase Quantity

Table 42: *C.1.a. - Zero-filled order, cancel/replace request issued to increase order qty*

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comment
1	New Order (X)					10000				
2		*Execution (X)*		*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by marketplace*
2		Execution (X)	A	New	New	10000	0	10000	0	
3	Replace Request (Y, X)		A			11000				Request to increase quantity to 11000
4		*Cancel Reject (Y, X)*	*A*		*New*					*If order is rejected by marketplace*
4		Execution (Y, X)		Replace	New	11000	0	11000	0	

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comment
5		Execution (Y)		Trade	Partially Filled	11000	1000	10000	1000	Execution for 1000. Use Y as the new ClOrdID
6		Execution (Y)		Trade	Partially Filled	11000	3000	8000	2000	Execution for 2000

For the below workflow, note that the user can avoid updates on stale order information by using OrigOrdModTime.

Table 43: C.1.b. - Part-filled order, followed by cancel/replace request to increase order qty, execution occurs whilst order is pending replace

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comment
1	New Order (X)					10000				
2		*Execution (X)*		*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by marketplace*
2		Execution (X)	A	New	New	10000	0	10000	0	
3		Execution (X)	A	Trade	Partially Filled	10000	1000	9000	1000	Execution for 1000
4	Replace Request (Y, X)		A			12000				Request increase in order quantity to 12 000
5		Execution (X)	A	Trade	Partially Filled	10000	1100	8900	100	Execution for 100 before cancel/replace request is dealt with
6		*Execution (Y, X)*	*A*		*Rejected*					*If order is rejected by marketplace*
6		Execution (Y, X)	A	Replace	Partially Filled	12000	1100	10900	0	Confirm replace has been accepted
7		Execution (Y)	A	Trade	Filled	12000	12000	0	10900	Execution for 10900

Table 44: C.1.c. - Filled order, followed by cancel/replace request to increase order quantity

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comment
1	New Order (X)					10000				
2		*Execution (X)*		*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by marketplace*
2		Execution (X)	A	New	New	10000	0	10000	0	
3		Execution (X)	A	Trade	Filled	10000	10000	0	10000	Execution for 10000
4	Replace Request (Y, X)		A			12000				Request to increase quantity to 12000
5		*Cancel Reject (Y, X)*	*A*		*Rejected*					*Filled orders are not reinstated, request is rejected* *CxlRejectReason = 0 (too late to cancel) or 1 (Unknown order)*

5.5.4.2 C.2 - Replace not for Quantity Change

The C.2.a. workflow of the standard specification is considered not applicable.

5.5.4.3 C.3 - Replace to Decrease Quantity

The workflows in this section are all examples of In-Flight modifications.

Table 45: C.3.a. - Cancel/replace request sent whilst execution is being reported – the requested order qty exceeds the cum qty. Order is replaced then filled

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comment
1	New Order (X)					10000				
2		*Execution (X)*		*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by marketplace*
2		Execution (X)	A	New	New	10000	0	10000	0	
3		Execution (X)	A	Trade	Partially Filled	10000	1000	9000	1000	Execution for 1000
4	Replace Request (Y, X)		A			8000				Request a decrease order quantity to 8000 (leaving 7000 open)
5		Execution (X)	A	Trade	Partially Filled	10000	1500	8500	500	Execution for 500 sent. Replace request and this execution report pass each other on the connection
6		Execution (X)	A	Trade	Partially Filled	10000	1600	8400	100	Execution for 100 occurs before cancel/replace request is accepted
7		*Execution (Y, X)*	*A*		*Rejected*					*If order is rejected by marketplace*
7		Execution (Y, X)	A	Replace	Partially Filled	8000	1600	6400	0	Replace is accepted as requested order qty exceeds cum qty
8		Execution (Y)	A	Trade	Filled	8000	8000	0	6400	Execution for 6400

Table 46: C.3.b. - Cancel/replace request sent whilst execution is being reported – the requested order qty equals the cum qty – order qty is amended to cum qty

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comment
1	New Order (X)					10000				
2		*Execution (X)*		*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by marketplace*
2		Execution (X)	A	New	New	10000	0	10000	0	
3	Replace Request (Y, X)		A			7000				Request a decrease order quantity to 7000
4		Execution (X)	A	Trade	Partially Filled	10000	7000	3000	7000	Execution for 7000 - the replace message and this execution report pass each other on the connection
5		Execution (Y, X)	A	Replace	Filled	7000	7000	0	0	The replace request is interpreted as requiring the balance of the order to be canceled – the 'filled' order status takes precedence over 'canceled'.

Table 47: C.3.c. - Cancel/replace request sent whilst execution is being reported – the requested order qty is below cum qty – order qty is amended to cum qty

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comment
1	New Order (X)					10000				
2		*Execution (X)*		*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by marketplace*
2		Execution (X)	A	New	New	10000	0	10000	0	
3	Replace Request (Y, X)		A			7000				Request a decrease order quantity to 7000
4		Execution (X)	A	Trade	Partially Filled	10000	8000	2000	8000	Execution for 8000 - the replace message and this execution report pass each other on the connection
5		Execution (Y, X)	A	Replace	Filled	8000	8000	0	0	The replace request is interpreted as requiring the balance of the order to be canceled – the 'filled' order status takes precedence over 'canceled'.

5.5.5 D - Cancel/Replace Sequencing and Chaining

5.5.5.1 D.1 - Sequencing

The D.1.a - D.1.c. workflows of the FIX standard specification are not considered to contribute anything not otherwise included in the OMX specification.

5.5.5.2 D.2 - Chaining

Table 48: D.2.a. - One cancel/replace request is issued followed immediately by another – market place processes sequentially

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comment
1	New Order (X)					10000				
2		*Execution (X)*		*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by marketplace*
2		Execution (X)	A	New	New	10000	0	10000	0	
3		Execution (X)	A	Trade	Partially Filled	10000	1000	9000	1000	Execution for 1000
4	Replace Request (Y, X)		A			8000				Request decrease in order quantity to 8000, leaving 7000 open
5	Replace Request (Z, Y)		A			7000				Request decrease in order quantity to 7000, leaving 6000 open. Note OrigClOrdID set to last non rejected ClOrdID i.e. Y (on an 'optimistic' basis)
6		Execution (Y, X)	A	Replace	Partially Filled	8000	1000	7000	0	Marketplace processes Replace (Y,X) first
7		Execution (Z, Y)	A	Replace	Partially Filled	7000	1000	6000	0	Marketplace then processes Replace (Z,Y)
8		Execution (Z)	A	Trade	Filled	7000	7000	0	6000	Execution for 6000

The D.2.b - D.2.d workflows of the FIX standard specification are considered not contributing anything not otherwise captured by the OMX specification.

5.5.6 F - Order Reject

Table 49: F.1.a. - Order rejected due to duplicate ClOrdID

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comment
1	New Order (X)					10000				
2		*Execution (X)*		*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by marketplace*
2		Execution (X)	A	Trade	Partially Filled	10000	1000	9000	1000	Execution for 1000
3	New Order (X)					15000				Another Order submitted with the same ClOrdID
4		*Execution (X)*		*Rejected*	*Rejected*	*15000*	*0*	*0*	*0*	*OrdRejReason = duplicate order. Note that the reject does not relay the state of the initial order!*

Table 50: F.1.b. - Poss resend and duplicate ClOrdID

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comment
1	New Order (X)					10000				
2		*Execution (X)*		*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by marketplace*
2		Execution (X)	A	New	New	10000	0	10000	0	
3	New Order (X)									PossResend = Y
4		Execution (X)	A	Order Status	New	10000	0	10000		Because order X has already been received, confirm back the current state of the order. Last Qty not required when ExecType = Order Status
5	New Order (X)					20000				PossResend = N or not set
6		*Execution (X)*		*Rejected*	*Rejected*	*20000*	*0*	*0*	*0*	*OrdRejReason = duplicate order.* ***Note that the reject does not relay the state of the initial order!***
7	New Order (Y)					15000				PossResend=Y
8		Exection (Y)	B	New	New	15000	0	15000	0	Because order Y has not been received before, confirm back as a new order.

The F.1.c. workflow of the FIX standard specification is not supported by OMX (no duplicate detection beside ClOrdID).

5.5.7 H - GT

Table 51: H.1.a. - GTC Order Partially Filled, Restated (renewed) and Partially Filled the following day

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comment
1:1	New Order (X)					10000				

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comment
1:2		*Execution (X)*		*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by marketplace*
1:2		Execution (X)	A	New	New	10000	0	10000	0	Order accepted
1:3		Execution (X)	A	Trade	Partially Filled	10000	2000	8000	2000	Execution for 2000
1:4		Execution (X)	A	Done for Day	Done for Day	10000	2000	8000	0	Optional at end of day Not used by OMX!
2:1		Execution (X)	A	Restated	Partially Filled	10000	2000	8000	0	ExecRestatementReason = GTC renewal/restatement (no change) – optionally sent the following morning Not used by OMX!
2:2		Execution (X)	A	Trade	Partially Filled	10000	3000	7000	1000	Execution for 1000

The H.1.b - H.1.d workflows of the FIX standard specification are not considered to contribute anything not otherwise included.

5.5.8 I - Time in Force

Table 52: *Expiry of Day order*

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comment
1	New Order (X)					10000				
2		*Execution (X)*		*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by marketplace*
2		Execution (X)	A	New	New	10000	0	10000	0	
3		Execution (X)	A	Trade	Partially Filled	10000	2000	8000	2000	Execution for 2000
4		Execution (X)	A	Trade	Partially Filled	10000	3000	7000	1000	Execution for 1000
5		Execution (X)	A	Canceled	Expired	10000	3000	0	0	Assuming day order. See other examples which cover GT orders

Table 53: *I.1.a. - Fill-or-Kill Order cannot be filled*

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comment
1	New Order (X)					10000				TimeInForce = 4 (Fill or Kill)
2		*Execution (X)*		*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by marketplace*
2		Execution (X)	A	Canceled	Canceled	10000	0	0	0	Order expire unfilled

Table 54: *I.1.b. - Immediate-or-Cancel Order that cannot be immediately hit*

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comment
1	New Order (X)					10000				TimeInForce = 3 (Immediate Or Cancel)

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comment
2		*Execution (X)*		*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by marketplace*
2		Execution (X)	A	Trade	Canceled	10000	3000	0	3000	Order is partially filled for 3 000 and then expire

The below example applies in markets where market price orders are allowed, but cannot sit on the book (i.e. they are implicitly IOC orders).

Table 55: Market Price Order that is not filled

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comment
1	New Order (X)					10000				OrdType = 1 (Market)
2		*Execution (X)*		*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by marketplace*
2		Execution (X)	A	Canceled	Canceled	10000	0	0	0	Order expire unfilled

Table 56: Market Price Order that is partially filled

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comment
1	New Order (X)					10000				OrdType = 1 (Market)
2		*Execution (X)*		*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by marketplace*
2		Execution (X)	A	Trade	Canceled	10000	3000	0	3000	Order is partially filled for 3 000 and then expire

5.5.9 J - Execution Cancels/Corrects

OMX does not support restating orders as a consequence of a trade break or correction. The J.1.a - J.1.d workflows of the FIX standard specification are therefore not supported.

5.5.10 K - Trading Halt

Table 57: K.1.a. - Reinstatement after Trading Halt

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comment
1	New Order (X)					10000				Market convention or ExecInst set to reinstate on trading halt
2		*Execution (X)*		*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by marketplace*
2		Execution (X)	A	New	New	10000	0	10000	0	
3										Trading Halt established
4										Trading Halt lifted
5		Execution (X)	A	Trade	Filled	10000	10000	0	10000	

Table 58: *K.1.b. - Trading Halt - Cancel*

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comment
1	New Order (X)					10000				Market convention or ExecInst set to cancel on trading halt
2		*Execution (X)*		*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by marketplace*
2		Execution (X)	A	New	New	10000	0	10000	0	
3										Trading Halt established
4		Execution (X)	A	Canceled	Canceled	10000	0	0	0	Order canceled due to trading halt. ExecRestatementReason = Canceled due to trading halt

5.5.11 L - Miscellaneous

The L.1.a - L.1.b workflows of the FIX standard specification are not supported.

5.5.12 M - Connection Loss

The following workflow is an OMX proposed extension to the FIX standard specification. However, note that the functionality is available in FIX.

Table 59: *Suspend on Connection Loss*

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comment
1	New Order (X)					10000				Market convention or ExecInst set to suspend on connection loss
2		*Execution (X)*		*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by marketplace*
2		Execution (X)	A	New	New	10000	0	10000	0	
3										Connection is lost
4		Execution (X)	A	Restated	Suspended	10000	0	10000	0	Order suspended due to conenction loss. ExecInst = S (Suspend) ExecRestatementReason = Suspended due to connection loss
5										Connection is re-esttablished User must update order to remove ExecInst = S in order to activate the order again

5.5.13 Reserve Instructions

5.5.13.1 Introduction

The following workflows are copies of the FIX standard specification (volume 4).

5.5.13.2 Refresh Immediate

Table 60: *Refresh Immediate using Initial Display Quantity*

Message	Order Qty	Leaves Qty	Display Qty	Display When	Display Method	Display Low Qty	Display High Qty	Refresh Qty	Comment
New Order	1000		100	1 (Immediate)	1 (Initial)				
Execution (New)	1000	1000	100	1 (Immediate)	1 (Initial)				
Execution (Partially Filled)	1000	750	100	1 (Immediate)	1 (Initial)				Fill for 250
Execution (Partially Filled)	1000	50	50	1 (Immediate)	1 (Initial)				Fill for 700
Execution (Filled)	1000	0	0	1 (Immediate)	1 (Initial)				Fill for 50

Table 61: *Refresh Immediate using New Display Quantity*

Message	Order Qty	Leaves Qty	Display Qty	Display When	Display Method	Display Low Qty	Display High Qty	Refresh Qty	Comment
New Order	1000		100	1 (Immediate)	2 (New)			200	
Execution (New)	1000	1000	100	1 (Immediate)	2 (New)			200	
Execution (Partially Filled)	1000	750	200	1 (Immediate)	2 (New)			200	Fill for 250
Execution (Partially Filled)	1000	50	50	1 (Immediate)	2 (New)			200	Fill for 700
Execution (Filled)	1000	0	0	1 (Immediate)	2 (New)			200	Fill for 50

5.5.13.3 Refresh when Displayed Quantity is Exhausted

Table 62: *Refresh When Displayed Quantity is Exhausted using Initial Display Quantity*

Message	Order Qty	Leaves Qty	Display Qty	Display When	Display Method	Display Low Qty	Display High Qty	Refresh Qty	Comment
New Order	1000		100	2 (Exhaust)	1 (Initial)				
Execution (New)	1000	1000	100	2 (Exhaust)	1 (Initial)			(100)	
Execution (Partially Filled)	1000	950	50	2 (Exhaust)	1 (Initial)			(100)	Fill for 50
Execution (Partially Filled)	1000	900	100	2 (Exhaust)	1 (Initial)			(100)	Fill for 50
...									Subsequent fills, totalling 850
Execution (Partially Filled)	1000	50	50	2 (Exhust)	1 (Initial)			(100)	
Execution (Filled)	1000	0	0	2 (Exhaust)	1 (Initial)			(100)	Fill for 50

Table 63: *Refresh When Displayed Quantity is Exhausted using New Display Quantity*

Message	Order Qty	Leaves Qty	Display Qty	Display When	Display Method	Display Low Qty	Display High Qty	Refresh Qty	Comment
New Order	1000		100	2 (Exhaust)	2 (New)			200	
Execution (New)	1000	1000	100	2 (Exhaust)	2 (New)			200	
Execution (Partially Filled)	1000	950	50	2 (Exhaust)	2 (New)			200	Fill for 50
Execution (Partially Filled)	1000	900	200	2 (Exhaust)	2 (New)			200	Fill for 700
...									Subsequent fills, totalling 850

Message	Order Qty	Leaves Qty	Display Qty	Display When	Display Method	Display Low Qty	Display High Qty	Refresh Qty	Comment
Execution (Partially Filled)	1000	50	50	2 (Exhaust)	2 (New)			200	
Execution (Filled)	1000	0	0	2 (Exhaust)	2 (New)			200	Fill for 50

5.5.14 Hit / Take

Please note that although "Ord.id" is used in the examples, other identifiers (SecondaryOrderID or MDEntryID could be used instead).

5.5.14.1 Vanilla

Table 64: Hit / Take Order is entered, both Orders are Filled

<table>
<tr><th>Trading Party</th><th>Direc-tion</th><th>Market Place</th><th>Direc-tion</th><th>Market Data</th></tr>
<tr><td></td><td></td><td>Order Depth Market Data is published for book X</td><td>-></td><td><table><tr><th>OrdID</th><th>Price</th><th>Qty</th><th>Party</th></tr><tr><td>3</td><td>10.00</td><td>100</td><td>ABC</td></tr><tr><td>1</td><td>9.90</td><td>200</td><td>DEF</td></tr><tr><td>2</td><td>9.90</td><td>100</td><td>ABC</td></tr><tr><td>4</td><td>9.80</td><td>400</td><td>GHI</td></tr></table></td></tr>
<tr><td>Party JKL enters New Order Single to hit DEF's order (OrdType = Counter-Order Selection, RefOrdID = 1, Price = 9.90, OrderQty = 200)</td><td>-></td><td></td><td></td><td></td></tr>
<tr><td>Party JKL receives Execution Report (ExecType = Trade, OrdStatus = Filled, OrdType = Counter-Order Selection, RefOrdID = 1)
Party DEF receives Execution Report (ExecType = Trade, OrdStatus = Filled)</td><td><-</td><td>JKL's Hit / Take order is matched. Execution Report produced to involved parties and new Market Data published</td><td>-></td><td><table><tr><th>OrdID</th><th>Price</th><th>Qty</th><th>Party</th></tr><tr><td>3</td><td>10.00</td><td>100</td><td>ABC</td></tr><tr><td>2</td><td>9.90</td><td>100</td><td>ABC</td></tr><tr><td>4</td><td>9.80</td><td>400</td><td>GHI</td></tr></table></td></tr>
</table>

Table 65: Hit / Take Order is entered, it is Filled but Referred Orders is Partially Filled

<table>
<tr><th>Trading Party</th><th>Direc-tion</th><th>Market Place</th><th>Direc-tion</th><th>Market Data</th></tr>
<tr><td></td><td></td><td>Order Depth Market Data is published for book X</td><td>-></td><td><table><tr><th>OrdID</th><th>Price</th><th>Qty</th><th>Party</th></tr><tr><td>3</td><td>10.00</td><td>100</td><td>ABC</td></tr><tr><td>1</td><td>9.90</td><td>200</td><td>DEF</td></tr><tr><td>2</td><td>9.90</td><td>100</td><td>ABC</td></tr><tr><td>4</td><td>9.80</td><td>400</td><td>GHI</td></tr></table></td></tr>
<tr><td>Party JKL enters New Order Single to hit DEF's order (OrdType =</td><td>-></td><td></td><td></td><td></td></tr>
</table>

Trading Party	Direction	Market Place	Direction	Market Data
Counter-Order Selection, RefOrdID = 1, Price = 9.90, OrderQty = 100)				
Party JKL receives Execution Report (ExecType = Trade, OrdStatus = Filled, OrdType = Counter-Order Selection, RefOrdID = 1) Party DEF receives Execution Report (ExecType = Trade, OrdStatus = Partially Filled)	<-	JKL's Hit / Take order is matched. Execution Report produced to involved parties and new Market Data published	->	OrdID / Price / Qty / Party 3 / 10.00 / 100 / ABC 1 / 9.90 / 100 / DEF 2 / 9.90 / 100 / ABC 4 / 9.80 / 400 / GHI

Table 66: Hit / Take Order is entered, it is Partially Filled but Referred Order is Filled

		Order Depth Market Data is published for book X	->	OrdID / Price / Qty / Party 3 / 10.00 / 100 / ABC 1 / 9.90 / 200 / DEF 2 / 9.90 / 100 / ABC 4 / 9.80 / 400 / GHI
Party DEF enter Order Cancel Replace Request for Ord.id = 1 (OrdQty = 100)	->	Receives Cancel Replace Request		
Party JKL enters New Order Single to hit DEF's order (OrdType = Counter-Order Selection, RefOrdID = 1, Price = 9.90, OrderQty = 200)	->			
Party DEF receives Execution Report (ExecType = Replace, OrderQty = 100)	<-	DEF's Cancel Request is processed and Execution Report + new market data published	->	OrdID / Price / Qty / Party 3 / 10.00 / 100 / ABC 1 / 9.90 / 100 / DEF 2 / 9.90 / 100 / ABC 4 / 9.80 / 400 / GHI
Party JKL receives Execution Report (ExecType = Trade, OrdStatus = Partially Filled, OrdType = Counter-Order Selection, RefOrdID = 1) Party DEF receives Execution Report (ExecType = Trade, OrdStatus = Filled) Party JKL receives Execution Report (ExecType = Canceled, OrdStatus = Canceled)	<-	JKL's Hit / Take order is matched. Execution Report produced to involved parties and new Market Data published Residue quantity on JKL's order is canceled (Hit / Take can not sit on book), Execution Report is produced.	->	OrdID / Price / Qty / Party 3 / 10.00 / 100 / ABC 2 / 9.90 / 100 / ABC 4 / 9.80 / 400 / GHI

5.5.14.2 Non-Executed Hit/Take Orders

Table 67: *Hit / Take Order is Entered, Referred Order is Removed from Book when it Arrives*

<table>
<tr><th>Trading Party</th><th>Direc-tion</th><th>Market Place</th><th>Direc-tion</th><th>Market Data</th></tr>
<tr><td></td><td></td><td>Order Depth Market Data is published for book X</td><td>-></td><td><table><tr><th>OrdID</th><th>Price</th><th>Qty</th><th>Party</th></tr><tr><td>3</td><td>10.00</td><td>100</td><td>ABC</td></tr><tr><td>1</td><td>9.90</td><td>200</td><td>DEF</td></tr><tr><td>2</td><td>9.90</td><td>100</td><td>ABC</td></tr><tr><td>4</td><td>9.80</td><td>400</td><td>GHI</td></tr></table></td></tr>
<tr><td>Party DEF enter Order Cancel Request for Ord.id = 1</td><td>-></td><td>Receives Cancel Re-quest</td><td></td><td></td></tr>
<tr><td>Party JKL enters New Order Sin-gle to hit DEF's order (OrdType = Counter-Order Selection, RefOr-dID = 1, Price = 9.90, OrderQty = 200)</td><td>-></td><td></td><td></td><td></td></tr>
<tr><td>Party DEF receives Execution Report (ExecType = Canceled, OrdStatus = Canceled)</td><td><-</td><td>DEF's Cancel Re-quest is processed and Execution Report + new market data published</td><td>-></td><td><table><tr><th>OrdID</th><th>Price</th><th>Qty</th><th>Party</th></tr><tr><td>3</td><td>10.00</td><td>100</td><td>ABC</td></tr><tr><td>2</td><td>9.90</td><td>100</td><td>ABC</td></tr><tr><td>4</td><td>9.80</td><td>400</td><td>GHI</td></tr></table></td></tr>
<tr><td>Party JKL receives Execution Re-port (OrdType = Counter-Order Selection, RefOrdID = 1, Or-dRejReason = 5)</td><td><-</td><td>JKL's Hit / Take order is processed, as the referred order is out of the book, the order is rejected.</td><td></td><td></td></tr>
</table>

Table 68: *Hit / Take Order is Entered, Referred Order does not Match when it Arrives*

<table>
<tr><th>Trading Party</th><th>Direc-tion</th><th>Market Place</th><th>Direc-tion</th><th>Market Data</th></tr>
<tr><td></td><td></td><td>Order Depth Market Data is published for book X</td><td>-></td><td><table><tr><th>OrdID</th><th>Price</th><th>Qty</th><th>Party</th></tr><tr><td>3</td><td>10.00</td><td>100</td><td>ABC</td></tr><tr><td>1</td><td>9.90</td><td>200</td><td>DEF</td></tr><tr><td>2</td><td>9.90</td><td>100</td><td>ABC</td></tr><tr><td>4</td><td>9.80</td><td>400</td><td>GHI</td></tr></table></td></tr>
<tr><td>Party DEF enter Order Cancel Replace Request for Ord.id = 1 (Price = 10.00)</td><td>-></td><td>Receives Cancel Re-place Request</td><td></td><td></td></tr>
<tr><td>Party JKL enters New Order Sin-gle to hit DEF's order (OrdType = Counter-Order Selection, RefOr-</td><td>-></td><td></td><td></td><td></td></tr>
</table>

<table>
<tr><th>Trading Party</th><th>Direc-tion</th><th>Market Place</th><th>Direc-tion</th><th>Market Data</th></tr>
<tr><td>dID = 1, Price = 9.90, OrderQty = 200)</td><td></td><td></td><td></td><td></td></tr>
<tr><td>Party DEF receives Execution Report (ExecType = Replace, Price = 10.00)</td><td><-</td><td>DEF's Cancel Request is processed and Execution Report + new market data published. Note that the Ord.Id used for market data is changed (normal procedure on e.g. price update)!</td><td>-></td><td><table><tr><th>OrdID</th><th>Price</th><th>Qty</th><th>Party</th></tr><tr><td>3</td><td>10.00</td><td>100</td><td>ABC</td></tr><tr><td>5</td><td>10.00</td><td>200</td><td>DEF</td></tr><tr><td>2</td><td>9.90</td><td>100</td><td>ABC</td></tr><tr><td>4</td><td>9.80</td><td>400</td><td>GHI</td></tr></table></td></tr>
<tr><td>Party JKL receives Execution Report (
OrdType = Counter-Order Selection, RefOrdID = 1, OrdRejReason = 8)</td><td><-</td><td>JKL's Hit / Take order is processed, as the referred order does not have a matching price, the order is rejected.</td><td></td><td></td></tr>
</table>

6 Multileg Orders

This feature is currently not supported.

7 Contingent (Linked) Orders

This feature is currently not supported.

8 Continuous and General Quote Handling

This feature is currently not supported.

9 Quote Negotiation

This feature is currently not supported.

10 Indications of Interest (IOI)

This feature is currently not supported.

11 Request for One-Sided Auction

This feature is currently not supported.

12 Reporting Privately Negotiated Trades

12.1 Business Message Types

The marketplace supports the message types described in the following tables. Full details of the messages and workflows around them are available in other parts of the document.

Table 69: Business Messages

In/Out	Message Name	Comment
In	**Trade Capture Report**	Used to submit a privately negotiated trade into the marketplace.
Out	**Trade Capture Report Ack**	Used to acknowledge the receipt of a Trade Capture Report or reject one.
Out	**Trade Capture Report**	Used to publish confirmed trades. Also used to inform the counterparty that an action is required (in the Pass Thru model).

12.2 Submitting a Privately Negotiated Trade

Trades may, subject to regulations or bilateral agreement, be reported to the marketplace in the following cases:

- Trades negotiated between market participants without using execution mechanisms provided by the marketplace
- Trades formed at other execution venues but reported to the marketplace for regulatory or publication reasons. Such execution venues may include (systematic) internalizers, ECN's, ATS's, MTF's and others regulated markets.

The marketplace allows trades to be reported using a set of different mechanisms, where each mechanism is appropriate in different situations:

One-Party Report for Matching

Used when both parties report their trade half. The marketplace matches the reports on security, price, quantity and possibly other conditions.

Reporting of Locked-In trades

Used when another execution venue reports confirmed trades to the exchange for regulatory reasons or for consolidated publication. The marketplace does not normally have a role in validating such trades.

For details about general workflows and functionality, please see the following chapters in volume 5 of standard FIX Specification :

- CATEGORY: TRADE CAPTURE ("STREETSIDE") REPORTING
- Appendix B - Trade Capture Report (TCR) Work Flow and Usage Tables

Deviations and clarifications to the FIX standard specification are specified in the following chapters.

12.2.1 Main Workflow

12.2.1.1 The Trade Capture Report

See the above mentioned chapters in the FIX standard specification.

The Trade Capture Report message is used for the following purposes:

- To submit a new Trade Report
- To update a Trade Report (if allowed)
- To cancel a Trade Report (if allowed)
- For the marketplace to publish trade confirmations (see Trade Confirmation and Management on page 87 for further details)

12.2.1.2 Trade Capture Report Reject

There is a single message used to reject Trade Capture Reports.

Table 70: Trade Capture Report Reject Messages

Reject Message	Direction	Business Message	Comment
Trade Capture Report Ack	Out	Trade Capture Report	

12.2.2 Trade Reporting Features

12.2.2.1 Trade Report Identification

12.2.2.1.1 Trade Report ID

Each Trade Capture Report message must be identified with a unique identification number (TradeReportID, 571).

In cases where the marketplace relays a message from the initiator to the counterparty for approval or as a reminder, the marketplace assigns its own Trade Report ID to that message.

In cases where the initiator wants to modify or cancel his request, the message must contain a new TradeReportID and be chained to the earlier version using the TradeReportRefID.

12.2.2.1.2 Trade Report Reference ID

If a user wishes to update or cancel an Trade Capture Report, the update is assigned a new Trade Report Id (see above) and the old Trade Report Id is entered into the TradeReportRefID (572) field for reference.

12.2.2.2 Defining the Instrument

Instrument definition is done in the same manner as for orders, see chapter 5.

12.2.2.3 The Sides of a Trade

A Trade Capture Report message should contain information about both sides of the trade when a Two-Party Report or a Locked-In Trade is reported. One-Party Reports should contain only a single side.

The reporting party specifies the counterparty in the RootParties component block:

- RootPartyID = ID for the counterparty
- RootPartyIDSource = "C" or "D"
- RootPartyRole = "17" - Contra Firm

12.2.2.4 Accounts, Pre-Allocations and Give-Ups

Please refer to Accounts, Pre-Allocation and Give Ups on page 29.

12.3 Message Details

See chapter Trade Confirmation and Management on page 87 for the message specifications.

12.4 Component Blocks (Privately Negotiated Trades Specific)

See chapter Trade Confirmation and Management on page 87 for the components specifications.

12.5 Workflows

12.5.1 Introduction

See the chapters in the FIX standard specification mentioned earlier.

13 Trade Confirmation and Management

13.1 Business Message Types

The marketplace supports the message types described in the following tables. Full details of the messages and workflows around them are available in other parts of the document.

Table 71: Business Messages

In/Out	Message Name	Comment
In	**Trade Capture Report**	Used for the following purposes: • Report confirmed trades from other marketplaces • Request cancelation of a confirmed trade (a.k.a. Trade Break) • Request amendment of a confirmed trade
Out	**Trade Capture Report**	Used to publish confirmed trades
Out	**Trade Capture Report Ack**	Used to acknowledge or reject an incoming Trade Capture Report.

13.2 Reporting Confirmed Trades

13.2.1 Trade Confirmation Features

13.2.1.1 Trade Identifiers

Trades have a number of identifiers used for different purposes:

- **Trade Identifier**

  The TradeID (1003) field uniquely identifies a trade for a certain trade date (TradeDate, 75) and order book (SecurityID, 48 + SecurityIDSource, 22 = 8). Certain systems may produce trade identifiers with a broader scope of uniqueness.

- **Secondary Trade Identifier**

  The SecondaryTradeID (1040) field is an additional trade identifier that will be provided in cases where downstream system (often clearing house related), require identifiers of another nature than the ones produced by the marketplace. The scope of uniqueness for the SecondaryTradeID is separately defined.

- **Trade Match Identifier**

  The TrdMatchID (880) will uniquely identify a match round and be shared by all trades created in that round. A match round is defined by one aggressive order hitting one or multiple orders at the other side at one or multiple prices. The scope of uniqueness for the TrdMatchID is the same as the TradeID.

- **Execution Identifier**

The ExecID (17) is a reference to the Execution Report where an auto-matched fill was reported. It does not apply to privately negotiated trades and will only be specified when the details of one trade-half is published.

The ExecRefID (19) represents the same thing as the ExecID, but this field is published per side of the trade and is therefore only relevant when both sides of the trade are published together as e.g. when a trade is sent to a depository.

- **Secondary Execution Identifier**

  The SecondaryExecID (527) is not assigned by the marketplace, it is used by participants who assign their own ExecIDs and thereby roll the marketplace assigned ExecID into this field.

13.2.1.2 Accounts, Pre-Allocation and Give-Ups

Please refer to Accounts, Pre-Allocation and Give Ups on page 29.

13.3 Message Details

13.3.1 TradeCaptureReport

Table 72: *TradeCaptureReport*

Tag	FieldName	Req'd	Comments	Format
	StandardHeader	Y	MsgType = AE	
571	TradeReportID	N	TradeReportID is conditionally required in a message-chaining model in which a subsequent message may refer to a prior message via TradeReportRefID. The alternative to a message-chain model is an entity-based model in which TradeID is used to identify a trade. In this case, TradeID is required and TradeReportID can be optionally specified.	String
1003	TradeID	N		String
487	TradeReportTransType	N	Identifies Trade Report message transaction type.	int
856	TradeReportType	N		int
828	TrdType	N	**OMX Comment:** Values as separately defined by the marketplace	int
855	SecondaryTrdType	N	**OMX Comment:** Used to distinguish between On-Hours and Off-Hours trades	int
1123	TradeHandlingInstr	N	**OMX Comment:** Note the confirmed trades from marketplace = 0.	char
1126	OrigTradeID	N	Used to preserve original trade id when original trade is being referenced in a subsequent trade transaction such as a transfer	String
1127	OrigSecondaryTradeID	N	Used to preserve original secondary trade id when original trade is being referenced in a sub-sequent trade transaction such as a transfer	String
572	TradeReportRefID	N	The TradeReportID that is being referenced for some action, such as correction or cancelation	String

Tag	FieldName	Req'd	Comments	Format
			OMX Comment: Refers to a previous Trade Capture Report. Relevant for references also in workflows for Privately Negotiated Trades	
880	TrdMatchID	N		String
17	ExecID	N	Exchanged assigned Execution ID (Trade Identifier) **OMX Comment:** If specified refers to the Execution Report issued for the fill.	String
527	SecondaryExecID	N		String
	RootParties	N	Insert here the set of "Root Parties" fields defined in "common components of application messages" Used for acting parties that applies to the whole message, not individual legs, sides, etc..	
	Instrument	Y	Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages" **OMX Comment:** Inbound messages shall use identifier fields only.	
	YieldData	N	Insert here the set of "YieldData" fields defined in "Common Components of Application Messages"	
32	LastQty	Y	Trade Quantity.	Qty
31	LastPx	Y	Trade Price.	Price
75	TradeDate	Y	Used when reporting other than current day trades.	LocalMktDate
715	ClearingBusinessDate	N	**OMX Comment:** Field used on confirmed Trades only - not in reporting Privately Negotiated Trades	LocalMktDate
60	TransactTime	N	Time the transaction represented by this Trade Capture Report occurred	UTCTimestamp
64	SettlDate	N	Takes precedence over SettlType value and conditionally required/omitted for specific SettlType values. **OMX Comment:** Used when non-standard settlement applies.	LocalMktDate
574	MatchType	N		String
1115	OrderCategory	N		char
	TrdCapRptSideGrp	Y	Number of sides	
797	CopyMsgIndicator	N	Indicates drop copy.	Boolean
852	PublishTrdIndicator	N		int
	StandardTrailer	Y		

13.3.2 TradeCaptureReportAck

Table 73: *TradeCaptureReportAck*

Tag	FieldName	Req'd	Comments	Format
	StandardHeader	Y	MsgType = AR	
571	TradeReportID	N	Unique identifier for the Trade Capture Report	String
1003	TradeID	N		String
487	TradeReportTransType	N	Identifies Trade Report message transaction type.	int
856	TradeReportType	N	Indicates action to take on trade	int
828	TrdType	N		int
855	SecondaryTrdType	N		int
1123	TradeHandlingInstr	N		char
	RootParties	N	Insert here the set of "Root Parties" (firm identification) fields defined in "common components of application messages" Range of values on report:	
572	TradeReportRefID	N	The TradeReportID that is being referenced for some action, such as correction or cancelation	String
939	TrdRptStatus	N	Status of Trade Report	int
751	TradeReportRejectReason	N	Reason for Rejection of Trade Report	int
880	TrdMatchID	N		String
17	ExecID	N	Exchanged assigned Execution ID (Trade Identifier)	String
527	SecondaryExecID	N		String
570	PreviouslyReported	N		Boolean
32	LastQty	N		Qty
31	LastPx	N		Price
	Instrument	Y	Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages"	
75	TradeDate	N		LocalMktDate
715	ClearingBusinessDate	N		LocalMktDate
60	TransactTime	N	Time ACK was issued by matching system, trading system or counterparty	UTCTimestamp
573	MatchStatus	N		char
574	MatchType	N		String
797	CopyMsgIndicator	N		Boolean
852	PublishTrdIndicator	N		int
58	Text	N	May be used by the executing market to record any execution Details that are particular to that market	String

Tag	FieldName	Req'd	Comments	Format
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.	Length
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.	data
	TrdCapRptAckSideGrp	N		
64	SettlDate	N		LocalMktDate
	StandardTrailer	Y		

13.4 Component Blocks (Trade Confirmation and Management Specific)

For components that are not specific for Trade Confirmation and Management, please see General Component Blocks on page 129.

13.4.1 Components

13.4.1.1 YieldData

Table 74: *YieldData*

Tag	FieldName	Req'd	Comments	Format
236	Yield	N		Percentage

13.4.2 Implicit Components

13.4.2.1 TrdAllocGrp

Table 75: *TrdAllocGrp*

Tag	FieldName	Req'd	Comments	Format
78	NoAllocs	N	Number of repeating groups for trade allocation **OMX Comment:** A single pre-allocation is allowed.	NumInGroup
>79	AllocAccount	N	Required if NoAllocs > 0. Must be first field in repeating group.	String
>661	AllocAcctIDSource	N		int
>467	IndividualAllocID	N		String
	NestedParties2	N	Insert here the set of "NestedParties2" (firm identification "nested" within additional repeating group) fields defined in "Common Components of Application Messages"	

13.4.2.2 TrdCapRptSideGrp

Table 76: *TrdCapRptSideGrp*

Tag	FieldName	Req'd	Comments	Format
552	NoSides	Y	Number of sides	NumInGroup
>54	Side	Y		char
>37	OrderID	N	OrderID should be conditionally required when Trade Capture Report is used for back office processing.	String
>11	ClOrdID	N	Required for executions against electronically submitted orders which were assigned an ID by the institution or intermediary. Not required for orders manually entered by the broker or fund manager (for CIV orders). In the case of quotes can be mapped to: - QuoteMsgID (1166) of a single Quote - QuoteID (117) of a Mass Quote **OMX Comment:** FIX 5.0 SP1	String
>19	ExecRefID	N	**OMX Comment:** Applicable for new trades, not only for trade cancels and corrects	String
	Parties	N	Insert here the set of "Parties" (firm identification) fields defined in "Common Components of Application Messages" Range of values on report: **OMX Comment:** Enter side-related clearing party info here	
>1	Account	N	Required for executions against electronically submitted orders which were assigned an account by the institution or intermediary	String
>660	AcctIDSource	N		int
>1093	LotType	N		char
>377	SolicitedFlag	N		Boolean
>528	OrderCapacity	N	The capacity of the participant for this trade (principal or agent for example).	char
>529	OrderRestrictions	N	Restrictions associated with the participant and their capacity for this trade.	MultipleCharValue
>483	TransBkdTime	N	A date and time stamp to indicate when this order was booked. For Equities, this is the time at which an order was received by an Exchange or Marketplace. For CIV, this is the time that a Fund Manager booked an order for execution at the next valuation point. **OMX Comment:** When reporting a privately negotiated trade, shall contain the the time of trade agreement.	UTCTimestamp
>336	TradingSessionID	N		String
>625	TradingSessionSubID	N		String

Tag	FieldName	Req'd	Comments	Format
>120	SettlCurrency	N	Used to report results of forex accommodation trade	Currency
>58	Text	N	May be used by the executing market to record any execution Details that are particular to that market	String
>354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.	Length
>355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.	data
>70	AllocID	N	Used to assign an ID to the block of preallocations	String
	TrdAllocGrp	N		
>1057	AggressorIndicator	N		Boolean

13.4.2.3 TrdCapRptAckSideGrp

Table 77: *TrdCapRptAckSideGrp*

Tag	FieldName	Req'd	Comments	Format
552	NoSides	Y		NumInGroup
>54	Side	Y		char
>37	OrderID	N		String
>11	ClOrdID	N		String
	Parties	N	Insert here here the set of "Parties" fields defined in "Common Components of Application Messages"	
>1	Account	N		String
>660	AcctIDSource	N		int
>1093	LotType	N		char
>377	SolicitedFlag	N		Boolean
>528	OrderCapacity	N		char
>529	OrderRestrictions	N		MultipleCharValue
>483	TransBkdTime	N		UTCTimestamp
>336	TradingSessionID	N		String
>625	TradingSessionSubID	N		String
>120	SettlCurrency	N		Currency
>70	AllocID	N		String
	TrdAllocGrp	N		
>1057	AggressorIndicator	N		Boolean

13.5 Workflows

13.5.1 Introduction

See the chapters in the FIX standard specification mentioned earlier.

14 Market Data

14.1 Business Message Types

The marketplace supports the message types described in the following tables. Full details of the messages and workflows around them are available in other parts of the document.

Table 78: Business Messages

In/Out	Message Name	Comment
Out	**Market Data Incremental Refresh**	Used to relay continuous incremental market data entries
Out	**Market Data Statistics**	Used to relay trade statistics when those do not fit into the Market Data Snapshot Full Refresh and Market Incremental Refresh messages.
Out	**Trading Session Status**	Used to relay the state of a market segment / group of securities.
Out	**Security Status**	Used to relay the state of an order book / individual security

14.2 Market Data

14.2.1 Introduction

The systems allow the transmission of real-time quote, order, trade, trade volume, open interest, and/or other price information on a feed or subscription basis.

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.

Pre-Trade data for the order book view is associated with a Book Type (MDBookType). The following book types are applicable:

Order Depth

The Exchange may provide full or a limited Order Depth, herafter called Market Depth. The Market Depth may change during the different trading states of a trading day. The value of the distributed Market Depth is relayed for the first Market Data Entry, and in case of a change. For an Order Depth book view, the Bid and Offer side may each have several Market Data Entries. Several Market Data Entries at one price tier could each represent a broker, Market Maker, external marketplace or the Exchange's quote in a security, or individual orders in a book. This is a **Non-Aggregated** book.

Price Depth

The Price Depth view of an Order Book represents an aggregate for each price tier, and there will be one Market Data Entry per side and per price active at a time. This is referred to as an **Aggregated** book. The

Price Depth view is also associated with a Market Depth, which can change just like for the Order Depth view.

The Book Type may change during the different trading states of a a trading day.

Post-Trade data for the Trade Ticker is another category of market data. A Market Data Entry can therefore represent a completed trade in a security. Trades can either be distributed on individual trade basis, or per deal basis. Deal basis implies the total volume executed at the same price will be distributed.

A third category of market data is **statistics**, including 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.

State information is also disseminated as part of market data. State information comes in two messages:

Trading Session Status

Relays state changes related to a market segment or group of securities. What trading schedules and states are available is defined in the Reference Data chapter (see chapter Reference Data on page 113 for details).

Security Status

Relays state changes for individual order books / securities.

Note:

Note: The Trading Session Status and Security Status messages relays snapshots of the state of the market segment and order book respectively. The state is a combination of the TradingSessionId, TradingSessionSubId and TradSesStatus for the Trading Session Status message and the TradingSessionId, TradingSessionSubId, SecurityTradingStatus and HaltReason for the SecurityStatus message.

Please refer to the Book Management Recommended Practices published FPL.

14.2.2 Unsolicited Feeds versus Subscriptions

Market Data information can either be sent on a subscription basis (using Market Data Request), or unsolicited per agreement with the exchange.

14.2.2.1 Subscriptions

Currently not supported.

14.2.2.2 Unsolicited Feeds

For unsolicted feeds the information is sent unsolicited, that is no subscription is required.

14.2.3 Main Workflow

14.2.3.1 Market Data - Incremental Refresh

The second Market Data message format (Market Data Incremental Refresh) is used for incremental updates. With the incremental format the exchange has the responsibility to provide all Market Data messages needed by the Client user in order to build an order book copy, populate a Trade Ticker, etc.

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.

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.

14.2.3.2 Security State Changes

The Security Status message is provided in Market Data feeds to relay changes in primarily the trading state of a security. A security will pass through a set of scheduled states during a trading day (or other period defined by the marketplace) e.g.:

- Closed
- Pre-Open
- Opening
- Continuous Trading
- Closing
- Post-Close
- Closed

The security may also be subject to non-scheduled states as Trading Halts.

The following fields are used to relay the different state levels:

- TradingSessionID (336). Defines the main trading sessions of the day, e.g. Day and Evening trading sessions. Values:
  - DAY
  - (SESSION <n>)
- TradingSessionSubID (625). Defines the main trading phases within a trading session, typically according to the bullet list shown above.
  - TRADING
  - PRE-OPEN
  - PRE-CLOSE
  - PRE-TRADING
  - POST-TRADING
  - OPENING
  - CLOSING
  - OPENING AUCTION
  - INTRADAY AUCTION
  - CLOSING AUCTION
  - CLOSED
- SecurityTradingStatus (326). Defines any subordinate states when such applies,, e.g. subdividing a Pre-Auction phase into sub-phases.
- HaltReason (327). Defines that a halt is instated and the reason for the halt.

Please refer to the Data Dictionary chapter for examples of field values for the SecurityTradingStatus (326) and HaltReason (327) fields.

The Security Status message is also used to relay events in cases where such are divulged by the marketplace. Events include warnings as e.g. "Market opening in 30 seconds, please remove indicative quotes". Events are published through the following field:

- SecurityTradingEvent (1174)

14.2.4 Market Data Features

14.2.4.1 Introduction

Please refer to [3] for recommended practices.

14.2.4.2 Market Data Identifiers

14.2.4.2.1 Feeds (MDFeedType, 1022)

Streaming market data is packaged into “feeds”. In some cases users can subscribe to one or more feeds, subject to marketplace rules and connectivity agreements. In other cases the marketplace offers feeds in a push mode where users simply listen in.

The contents of a feed are defined by the marketplace.

A feed is identified by the MDFeedType (1022) field.

14.2.4.2.2 Books (MDBookType, 1021)

Pre-trade market data, i.e. prices from orders and quotes, are presented in various aggregation levels (a.k.a. book types). The marketplace may choose to publish different book types in various trading sessions but also simultaneously publish multiple book types and let users choose what fits them best.

A book type is identified by the MDBoookType (1021) field.

14.2.4.2.3 Sub Books (MDSubBookType, 1173)

In order to rank orders in more complex situations, pre-trade market data may also be presented in various sub book types.

Marketplaces trading securities in different lot types stipulate rules for how orders may trade between these lots. In some markets, lot types are integrated whereas other markets keep lot types separated. This is commonly referred to as integrated vs non-integrated matching.

In the non-integrated matching model, orders are still received for one order book, and the marketplace will determine the appropriate lot type. This is done mainly to hide complexity of selecting the appropriate lot type for clients, and the order flow will still be distributed for just one order book. However, since they are not integrated, the marketplace will have to rank orders of different lot types independently. It would make no sense to provide a common ranking since the orders are not allowed to trade between lot types.

This constitutes a problem if using only MDPriceLevel (1023) and MDEntryPositionNo (290) to identify the position/ranking of an order. Round Lot, Odd Lot and Block Lot will all have separate ranking and individual MDEntryPositions per price level. Even if Lot Type were to be added to communicate lot, the client would not necessarily know if the market is operating a non-integrated or integrated matching. Hence, a client cannot rely on just MDPriceLevel and MDEntryPositionNo to sort orders.

In order to divulge the sorting/ranking of orders when that cannot be derived from MDPriceLevel and MDEntryPosition, the MDSubBookType tag is added to market data messages. The field is optional and FIX clients must always consider MDSubBookType, MDPriceLevel, and MDEntryPositionNo to be able to sort orders and quotes accordingly.

A sub book type is identified by the MDSubBookType (1173) field.

14.2.4.2.4 Market Data Entries (MDEntryID, 278)

Pre-trade incremental market data entries are identified by the unique MDEntryID (278) field. This allows users to index book updates for easy reference.

When adding a market data entry contains the EntryPositionNumber within the PriceLevel at which the order is to be inserted. Change and Delete instructions will reference the order by its MDEntryID. MDEntryID is an identifier assigned by the GENIUM FIX to track this order in the market data feed. FIX requires that the MDEntryID be unique among all other active entries. The order keeps the same MDEntryID for the lifetime of the entity. An MDEntryID can be reused within a day only if it has first been deleted.

14.2.4.2.5 Sorting of Market Data Entries

Pre-trade market data entries for a certain order book are published according to the ranking in the book. In true and simple Price-Time scenarios no additional ranking information is needed. However, some markets are characterized by more complex ranking algorithms, e.g. inverse ranking on price. Orders and market maker quotes may be ranked separately on the same price, etc. To accommodate more complex ranking rules, the market data messages use:

- MDSubBookType (1173). This fields contains a sequential number indicating a sub book type. Each sub book type will have its separate ranking as illustrated below.
- MDPriceLevel (1023). This field contains a sequential number allowing entries to be sorted in ascending order even if prices are sorted in inverse order or if there are different categories of orders at a certain price level that are to be sorted seperately.
- MDEntryPositionNo (290). This field contains a sequential number allowing entries to be sorted in ascending order (per MDPriceLevel).

The following diagram illustrates the usage of of the fields:

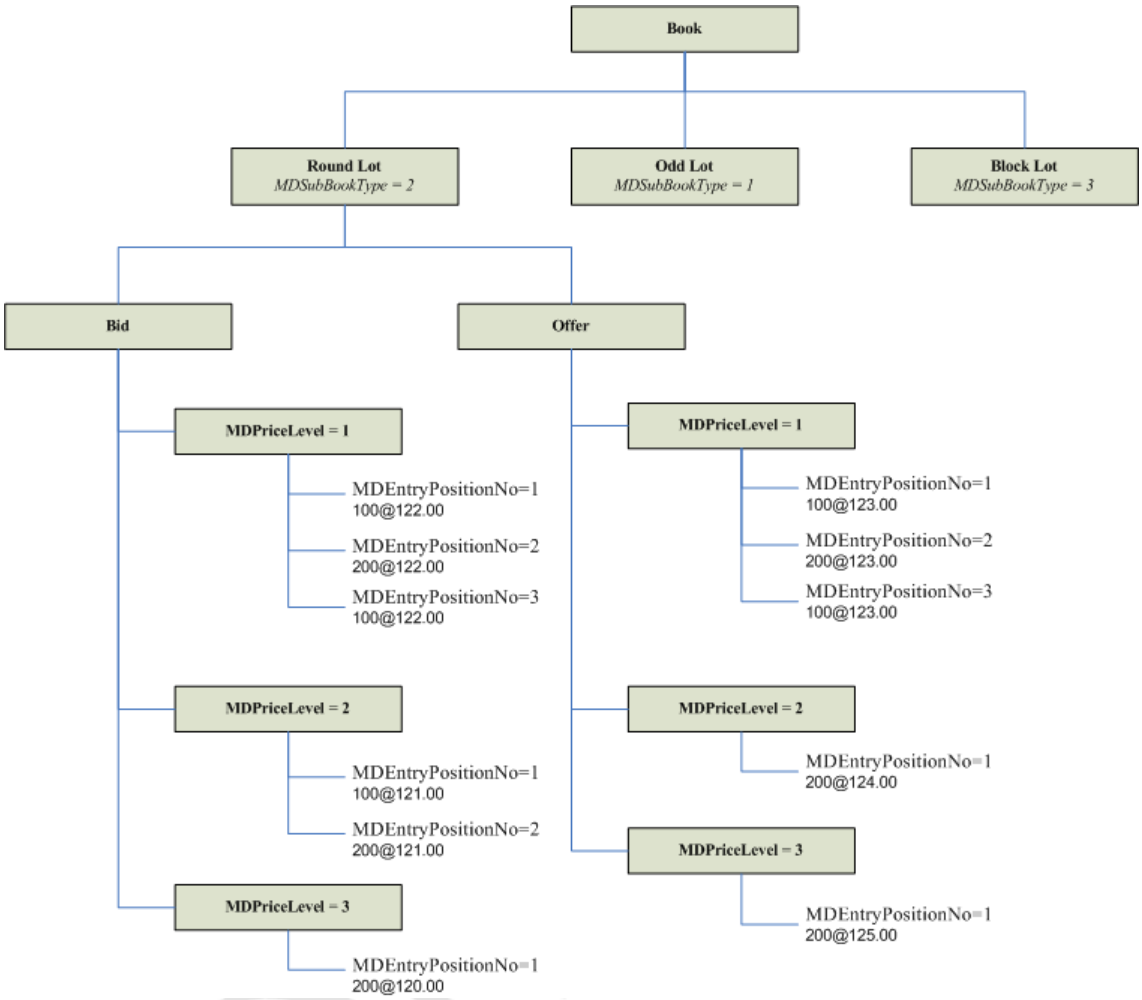

Figure 2: Book Sorting and Ranking Example

14.2.4.3 Exchange Trading Day and other Timestamps

The exchange Trading Day is relayed using the TradeDate (75) field. The TradeDate delimits the feed for a specific exchange trading day. The exchange trading day may differ from the trade date of an individual trade.

Market Data Entries commonly provide the MDEntryDate (272) and MDEntryTime (273) fields. Those fields represent either the time the entry is made public by the marketplace, or the time the source of the entry was created / updated.

Trade entries (MDEntryType = 2) may, subject to marketplace rules, also contain additional timestamps:

- TransactTime (60). The time the trade was matched at the marketplace. In the case of privately negotiated trades and other reported trades, the time the trade was confirmed as complete.
- TransBkdTime (463). The time the parties agreed on the trade. Only relevant for privately negotiated trades and optionally for other reported trades.

14.2.4.4 Trade Statistics

Trade Statistics are relayed using the market data messages. The marketplace can send out statistics using one of the following option(s):

- Statistic indicators. Used to allow receivers to calculate statistics themselves.

The following chapters discuss the respective alternatives.

14.2.4.4.1 Statistics Indicators

The marketplace may provide statistics indicators in order to help external actors calculate statistics themselves. The indicators are typically relevant when certain trades are excluded from ceratin statistics. A market data entry marked with a certain statistics indicator is eligible for the specified type of statistic, but the user must compared the entry to the current value of that statistic in order to determine if the statistic should be updated.

Usage of statistics indicators requires users to receive an uninterrupted sequence of Market Data Incremental Refresh messages. Note that a trade cancellation could mean the statistics calculation must revert back to an earlier statistics value.

The following fields are used for the statistics indicators:

- NoStatsIndicators (1175) captures the number of indicators relevant for the entry
- StatsType (1176) specifies the type of statistic the entry is eligible for. Example values:
  - 1 - Exchange Last
  - 2 - High / Low Price
  - 3 - Average Price
  - 4 - Turnover (Price * Quantity)

14.2.4.5 Fixed Income Considerations

Fixed Income securities are traded using various types of prices as e.g. percentage at par and yield. The fixed income industry recommends that the PriceType (423) field is always used in order to explicitly show the type of price relayed and thereby avoid misinterpretation.

Some markets relay more than one price for certain fixed income instruments. The following FIX component blocks are used to provide additional prices whenever this is relevant:

- Yield Data
- Spread or Benchmark Curve Data

14.3 Message Details

14.3.1 MarketDataIncrementalRefresh

Table 79: *MarketDataIncrementalRefresh*

Tag	FieldName	Req'd	Comments	Format
	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 **OMX Comment:** Used when multiple book types are simultaneously disseminated	int
1022	MDFeedType	N	Describes a class of service for a given data feed, ie Regular and Market Maker **OMX Comment:** Use when multiple feeds are simultaneously disseminated, e.g. a bandwidth conservative one vss a bandwidth intensive one	String
75	TradeDate	N	Used to specify the trading date for which a set of market data applies	LocalMktDate
	MDIncGrp	Y	Number of entries following.	
	StandardTrailer	Y		

14.3.2 MarketDataStatistics

Table 80: *MarketDataStatistics*

Tag	FieldName	Req'd	Comments	Format
	StandardHeader	Y	MsgType = U0	
20034	MDReportID	N	Unique indentifier for Market Data Report	String
715	ClearingBusinessDate	N		LocalMktDate
1022	MDFeedType	N	Describes a class of service for a given data feed, ie Regular and Market Maker	String
75	TradeDate	N	Used to specify the trading date for which a set of market data applies	LocalMktDate
262	MDReqID	N	Conditionally required if this message is in response to a Market Data Request.	String
	MDStatInstrsGrp	N	Repeating group of instruments for which statistics is defined	
813	ApplQueueDepth	N	Depth of application messages queued for transmission as of delivery of this message	int
814	ApplQueueResolution	N	Action taken to resolve application queuing	int
	StandardTrailer	Y		

14.3.3 TradingSessionStatus

Table 81: *TradingSessionStatus*

Tag	FieldName	Req'd	Comments	Format
	StandardHeader	Y	MsgType = h (lowercase)	
207	SecurityExchange	N		Exchange
20036	MarketSegmentID	N		String
336	TradingSessionID	Y	Identifier for Trading Session	String
625	TradingSessionSubID	N		String
339	TradSesMode	N	Trading Session Mode **OMX Comment:** T.b.d.	int
340	TradSesStatus	Y	State of the trading session	int
58	Text	N		String
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.	Length
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.	data
	StandardTrailer	Y		

14.3.4 SecurityStatus

Table 82: *SecurityStatus*

Tag	FieldName	Req'd	Comments	Format
	StandardHeader	Y	MsgType = f (lowercase)	
20036	MarketSegmentID	N		String
	Instrument	Y	Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages"	
336	TradingSessionID	N		String
625	TradingSessionSubID	N		String
326	SecurityTradingStatus	N	Identifies the trading status applicable to the transaction.	int
1174	SecurityTradingEvent	N	Identifies an event related to the trading status **OMX Comment:** FIX 5.0 SP1	int
292	CorporateAction	N	**OMX Comment:** T.b.d.	MultipleCharValue
327	HaltReason	N	Denotes the reason for the Opening Delay or Trading Halt.	char
328	InViewOfCommon	N	**OMX Comment:** T.b.d.	Boolean

Tag	FieldName	Req'd	Comments	Format
329	DueToRelated	N	**OMX Comment:** T.b.d.	Boolean
1021	MDBookType	N	Used to relay changes in the book type **OMX Comment:** FIX 5.0 SP1	int
264	MarketDepth	N	Used to relay changes in Market Depth. **OMX Comment:** FIX 5.0 SP1	int
60	TransactTime	N	Trade Dissemination Time	UTCTimes-tamp
58	Text	N	Comment, instructions, or other identifying information.	String
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.	Length
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.	data
	StandardTrailer	Y		

14.4 Component Blocks (Market Data Specific)

For components that are not specific for Market Data, please see General Component Blocks on page 129.

14.4.1 Components

N/A

14.4.2 Implicit Components

14.4.2.1 MDIncGrp

Table 83: *MDIncGrp*

Tag	FieldName	Req'd	Comments	Format
268	NoMDEntries	Y	Number of entries following.	NumInGroup
>279	MDUpdateAction	Y	Must be first field in this repeating group.	char
>1173	MDSubBookType	N	**OMX Comment:** FIX 5.0 SP1	String
>264	MarketDepth	N	**OMX Comment:** FIX 5.0 SP1	int
>269	MDEntryType	N	Conditionally required if MDUpdateAction = New(0). Cannot be changed. **OMX Comment:** Currently supports values 0-2 only	char
>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 =	String

Tag	FieldName	Req'd	Comments	Format
			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.	String
	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.	
>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"	Price
>423	PriceType	N	**OMX Comment:** FIX 5.0 SP1	int
	YieldData	N	Insert here the set of "YieldData" (yield-related) fields defined in "Common Components of Application Messages" **OMX Comment:** FIX 5.0 SP1	
	SpreadOrBenchmarkCurveData	N	Insert here the set of "SpreadOrBenchmarkCurveData" (Fixed Income spread or benchmark curve) fields defined in "Common Components of Application Messages" **OMX Comment:** FIX 5.0 SP1	
>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"	Qty
>1093	LotType	N	**OMX Comment:** FIX 5.0 SP1	char
>272	MDEntryDate	N		UTCDateOnly
>273	MDEntryTime	N		UTCTimeOnly
>336	TradingSessionID	N		String
>625	TradingSessionSubID	N		String
>326	SecurityTradingStatus	N	**OMX Comment:** FIX 5.0 SP1	int

Tag	FieldName	Req'd	Comments	Format
>276	QuoteCondition	N	Space-delimited list of conditions describing a quote.	MultipleString-Value
>277	TradeCondition	N	Space-delimited list of conditions describing a trade **OMX Comment:** Currently not supported	MultipleString-Value
>828	TrdType	N	For optional use in reporting Trades **OMX Comment:** FIX 5.0 SP1	int
>574	MatchType	N	For optional use in reporting Trades **OMX Comment:** FIX 5.0 SP1	String
>110	MinQty	N	For optional use when this Bid or Offer represents an order	Qty
>198	SecondaryOrderID	N	For optional use to support Hit/Take (selecting a specific order from the feed) without disclosing a private order id.	String
>1003	TradeID	N	For optional use in reporting Trades **OMX Comment:** FIX 5.0 SP1	String
>288	MDEntryBuyer	N	For optional use in reporting Trades	String
>289	MDEntrySeller	N	For optional use in reporting Trades	String
>346	NumberOfOrders	N	In an Aggregated Book, used to show how many individual orders make up an MDEntry	int
>290	MDEntryPositionNo	N	Display position of a bid or offer, numbered from most competitive to least competitive, per market side, beginning with 1 **OMX Comment:** Display position of a bid or offer numbered from most competitive to least competitive, per market side, price level, and potentially book sub type. Starts at 1.	int
>1023	MDPriceLevel	N		int
>64	SettlDate	N	Indicates date on which instrument will settle	LocalMktDate
>483	TransBkdTime	N	For optional use in reporting Trades. Used to specify the time of trade agreement for privately negotiated trades. **OMX Comment:** FIX 5.0 SP1	UTCTimes-tamp
>60	TransactTime	N	For optional use in reporting Trades. Used to specify the time of matching. **OMX Comment:** FIX 5.0 SP1	UTCTimes-tamp
>1070	MDQuoteType	N		int
	StatsIndGrp	N	**OMX Comment:** FIX 5.0 SP1	

14.4.2.2 MDStatInstrsGrp

Table 84: *MDStatInstrsGrp*

Tag	FieldName	Req'd	Comments	Format
20030	NoMDStatInstruments	Y	Number of instruments following	NumInGroup
	Instrument	Y	Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages"	
	UndInstrmtGrp	N		
	LegInstrmtGrp	N		
>291	FinancialStatus	N		MultipleCharValue
>292	CorporateAction	N		MultipleCharValue
>451	NetChgPrevDay	N		PriceOffset
	MDStatsGrp	Y	Repeating group of statistics	

14.4.2.3 MDStatsGrp

Table 85: *MDStatsGrp*

Tag	FieldName	Req'd	Comments	Format
>20031	NoMDStats	Y	Number of statistics following	NumInGroup
>>20032	MDStatType	Y	Must be the first field in the repeating group	String
>>20035	MDStatScope	N	Defines the scope of the statistics in periods of time.	String
>>336	TradingSessionID	N	When specifies means the provided statistic relates to this session only	String
>>625	TradingSessionSubID	N	When specifies means the provided statistic relates to this sub-session only	String
>>270	MDEntryPx	N		Price
>>271	MDEntrySize	N		Qty
>>274	TickDirection	N		char

14.4.2.4 SecSizesGrp

Table 86: *SecSizesGrp*

Tag	FieldName	Req'd	Comments	Format
>1177	NoSecSizes	N	Number of entries following. Conditionally required when MDUpdateAction = New(0) and MDEntryType = Bid(0) or Offer(1). **OMX Comment:** FIX 5.0 SP1	NumInGroup

Tag	FieldName	Req'd	Comments	Format
>>1178	MDSecSizeType	N	Defines the type of secondary size specified in MDSecSize (tbd). Must be first field in this repeating group **OMX Comment:** FIX 5.0 SP1	int
>>1179	MDSecSize	N	**OMX Comment:** FIX 5.0 SP1	Qty

14.4.2.5 StatsIndGrp

Table 87: *StatsIndGrp*

Tag	FieldName	Req'd	Comments	Format
1175	NoStatsIndicators	N	Number of statistics indicators **OMX Comment:** FIX 5.0 SP1	NumInGroup
>1176	StatsType	N	Indicates that the MD Entry is eligible for inclusion in the type of statistic specified by the StatsType. Must be provided id NoStatsIndicators > 0. **OMX Comment:** FIX 5.0 SP1	Int

14.5 Workflows

14.5.1 Introduction

The following workflows describe important aspects of the FIX interaction model.

The FIX Protocol Specification includes many of the workflows defined in this document; differences between the GENIUM and standard FIX specification are described.

14.5.2 Market DataWorkflows

14.5.2.1 Introduction

T.b.d

15 News Management

This feature is currently not supported.

16 Reference Data

Note:

Note, OMX is currently engaged in discussions with the FPL GTC and Exchanges/ECNs Working Group to expand and clarify the scope of Reference Data.This chapter is provided as best effort to provide an indication of what the specification will look like.

16.1 Business Message Types

The marketplace supports the message types described in the following tables. Full details of the messages and workflows around them are available in other parts of the document.

Table 88: *Business Messages*

In/Out	Message Name	Comment
Out	**Market Segment**	Used to relay market segments
Out	**Security List**	Used to relay a list of securities
Out	**Derivative Security List**	Used to relay a list of derivatives
Out	**Security Definition**	Used to acknowledge or reject a Security Definition Request
Out	**Security Definition Update Report**	Used to relay individual updates to a securities master file
Out	**Security List Update Report**	Used to relay master file updates to a securities list
Out	**Trading Session List**	Used to relay a list of Trading Sessions
Out	**Participant List**	Used to relay a list of Participants

16.2 Publication of Reference Data

16.2.1 Main Workflow

16.2.1.1 “Start of Day” Upload of Securities

Securities are uploaded the following way:

- Push. The marketplace publishes traded instruments using the Market Segment and Security Definition messages. The Security Type message may also be used.
- Pull. The user requests a list of securities through the Security List Request (or the Derivative Security List Request) message. The marketplace responds with a Security List (or Derivative Security List) message. If the user needs further details on individual securities he issues a Security Definition Request for those. The marketplace responds with Security Definition messages.

16.2.1.2 Continuous Updates of Securities

Updates are published in the following way:

- Push. The marketplace publishes Security Definition Update Report messages when a security is updated.

16.2.1.3 User-Defined Instruments

A user requests the registration of a user-defined security using the Security Definition Request message. The marketplace responds with a Security Definition message and publishes a Security Update Report (and Security List Update Report) to relevant subscribers.

16.2.1.4 Security Status

Security Status messages are published over Market Data feeds when the trading state of a security changes.

16.2.2 Reference Data Features

16.2.2.1 Extended Attributes

Extended attributes for reference data entities will be distributed in the XMLData field in XML format.

16.2.2.2 Note Codes

So called "note codes" are published using the EvntGrp component in the Instrument component block. Users should be aware that note codes are included in Security Defintion messges but can be modified with the Security Status message intraday.

16.3 Message Details

16.3.1 DerivativeSecurityList

Table 89: *DerivativeSecurityList*

Tag	FieldName	Req'd	Comments	Format
	StandardHeader	Y	MsgType = AA (2 A's)	
320	SecurityReqID	Y		String
322	SecurityResponseID	Y	Identifier for the Derivative Security List message	String
560	SecurityRequestResult	Y	Result of the Security Request identified by SecurityReqID	int
20036	MarketSegmentID	N		String
	UnderlyingInstrument	N	Underlying security for which derivatives are being returned	

Tag	FieldName	Req'd	Comments	Format
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.	int
893	LastFragment	N	Indicates whether this is the last fragment in a sequence of message fragments. Only required where message has been fragmented.	Boolean
	RelSymDerivSecGrp	N	Specifies the number of repeating symbols (instruments) specified	
	StandardTrailer	Y		

16.3.2 MarketSegment

Table 90: *MarketSegment*

Tag	FieldName	Req'd	Comments	Format
	StandardHeader	Y	MsgType = U1	
20036	MarketSegmentID	N		String
20037	MarketSegmentDesc	N		String
20038	EncodedMktSegmDescLen	N	Must be set if EncodedMktSegmDesc field is specified and must immediately precede it.	Length
20039	EncodedMktSegmDesc	N	Encoded (non-ASCII characters) representation of the MarketSegmDesc field in the encoded format specified via the MessageEncoding field.	data
20040	ParentMktSegmID	N	Specifies that the Market Segment is a sub segment of the Market Segment defined in this field.	String
20078	MktSegmSecurityExchange	N		String
20041	NoInstruments	N	Number of Instruments	NumINGroup
	Instrument	N	Must be provided if Number of instruments > 0	
15	Currency	N	The default trading currency	Currency
423	PriceType	N	The default price type for trading	int
386	NoTradingSessions	N	Specifies the number of repeating TradingSessionIDs	NumInGroup
>336	TradingSessionID	N	Required if NoTradingSessions is > 0.	String
>625	TradingSessionSubID	N		String
>341	TradSesStartTime	N	Scheduled start time of trading session or subsession	UTCTimestamp
>342	TradSesOpenTime	N	Scheduled open time of trading session or subsession	UTCTimestamp
>343	TradSesPreCloseTime	N	Scheduled pre-close time of trading session or subsession	UTCTimestamp
>344	TradSesCloseTime	N	Scheduled close time of trading session or subsession	UTCTimestamp
>345	TradSesEndTime	N	Scheduled end time of trading session or subsession	UTCTimestamp

Tag	FieldName	Req'd	Comments	Format
20042	NoLotTypes	N	Specifies the number of repeating LotTypes	NumInGroup
>1093	LotType	N	Required if NoLotSizes is > 0.	char
>20043	LotSize	N	The default lot size for trading	Qty
20044	NoTickSizes	N	Number of TickSize entries	NumInGroup
>20045	MinPx	N	Minimum price for which the price increment (tick size) applies. (The maximum price is one tick below the next price tier if such exist). Required if number of Tick Sizes > 0.	Price
>20046	PriceIncrement	N	The price increment (tick size) applicable to the price tier. Required if number of Tick Sizes > 0.	Price
60	TransactTime	N		UTCTimes-tamp
58	Text	N	Comment, instructions, or other identifying information.	String
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.	Length
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.	data
893	LastFragment	N	Indicates whether this is the last fragment in a sequence of message fragments. Only required where message has been fragmented.	Boolean
	StandardTrailer	Y		

16.3.3 ParticipantList

Table 91: *ParticipantList*

Tag	FieldName	Req'd	Comments	Format
	StandardHeader	N	MsgType = U9	
20079	PartyReqID	N		String
20080	PartyReportID	N		String
20081	PartyRequestResult	N		int
20082	ToNoRelParties	N	Used to indicate the total number of participants being returned for this request. Used in the event that message fragmentation is required.	int
893	LastFragment	N	Indicates whether this is the last fragment in a sequence of message fragments. Only required where message has been fragmented.	Boolean
	PartyLstGrp	N	Includes the list of parties	
	StandardTrailer	N		

16.3.4 SecurityDefinition

Table 92: *SecurityDefinition*

Tag	FieldName	Req'd	Comments	Format
	StandardHeader	Y	MsgType = d (lowercase)	
964	SecurityReportID	N	Identifier for Security Definition message	int
715	ClearingBusinessDate	N		LocalMktDate
320	SecurityReqID	N		String
323	SecurityResponseType	N	Response to the Security Definition Request	int
20036	MarketSegmentID	N		String
	Instrument	N	Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages" of the requested Security	
	InstrumentExtension	N	Insert here the set of "InstrumentExtension" fields defined in "Common Components of Application Messages"	
	UndInstrmtGrp	N	Number of underlyings	
15	Currency	N	Currency in which the price is denominated	Currency
58	Text	N	Comment, instructions, or other identifying information.	String
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.	Length
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.	data
	InstrmtLegGrp	N	Number of legs that make up the Security	
	SecurityTradingDefinition	N	Insert here the set of "SecurityTradingDefintion" fields defined in "Common Components of Application Messages"	
	StandardTrailer	Y		

16.3.5 SecurityDefinitionUpdateReport

Table 93: *SecurityDefinitionUpdateReport*

Tag	FieldName	Req'd	Comments	Format
	StandardHeader	Y	MsgType = BP	
964	SecurityReportID	N	Identifier for the Security Definition Update message in a bulk transfer environment (No Request/Response)	int
320	SecurityReqID	N		String
323	SecurityResponseType	N	Response to the Security Definition Request.	int

Tag	FieldName	Req'd	Comments	Format
980	SecurityUpdateAction	N	Specifies New (0), Cancel (1) or Replace (2)	char
292	CorporateAction	N	Identifies the type of Corporate Action	MultipleCharVal-ue
20036	MarketSegmentID	N		String
	Instrument	N		
	UnderlyingInstrument	N		
15	Currency	N		Currency
58	Text	N	Comment, instructions, or other identifying infor-mation.	String
354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.	Length
355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.	data
	InstrmtLegGrp	N		
	SecurityTradingDefinition	N	Insert here the set of " SecurityTradingDefintion" fields defined in " COMMON COMPONENTS OF APPLICATION MESSAGES "	
	StandardTrailer	Y		

16.3.6 SecurityList

Table 94: *SecurityList*

Tag	FieldName	Req'd	Comments	Format
	StandardHeader	Y	MsgType = y (lowercase Y)	
320	SecurityReqID	N		String
322	SecurityResponseID	N	Identifier for the Security List message	String
560	SecurityRequestResult	N	Result of the Security Request identified by the SecurityReqID	int
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.	int
893	LastFragment	N	Indicates whether this is the last fragment in a sequence of message fragments. Only required where message has been fragmented.	Boolean
20036	MarketSegmentID	N		String
	SecListGrp	N	Specifies the number of repeating symbols (in-struments) specified	
	StandardTrailer	Y		

16.3.7 SecurityListUpdateReport

Table 95: *SecurityListUpdateReport*

Tag	FieldName	Req'd	Comments	Format
	StandardHeader	Y	MsgType = BK	
964	SecurityReportID	N	Identifier for the Security List Update message in a bulk transfer environment (No Request/Response)	int
320	SecurityReqID	N		String
322	SecurityResponseID	N	Identifier for the Security List message.	String
560	SecurityRequestResult	N	Result of the Security Request identified by the SecurityReqID.	int
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.	int
980	SecurityUpdateAction	N	Specifies New (0), Cancel (1) or Replace (2)	char
292	CorporateAction	N	Identifies the type of Corporate Action that triggered the update	MultipleCharValue
893	LastFragment	N	Indicates whether this is the last fragment in a sequence of message fragments. Only required where message has been fragmented.	Boolean
20036	MarketSegmentID	N		String
	SecLstUpdRelSymGrp	N	Specifies the number of repeating symbols (instruments) specified	
	StandardTrailer	Y		

16.3.8 TradingSessionList

Table 96: *TradingSessionList*

Tag	FieldName	Req'd	Comments	Format
	StandardHeader	Y	MsgType = BJ	
335	TradSesReqID	N	Provided for a response to a specific Trading Session List Request message (snapshot).	String
	TrdSessLstGrp	Y		
	StandardTrailer	Y		

16.3.9 TradingSessionListRequest

Table 97: *TradingSessionListRequest*

Tag	FieldName	Req'd	Comments	Format
	StandardHeader	Y	MsgType = BI	

Tag	FieldName	Req'd	Comments	Format
207	SecurityExchange	N		Exchange
20036	MarketSegmentID	N		String
263	SubscriptionRequestType	Y		char
	StandardTrailer	Y		

16.4 Component Blocks (Reference Data Specific)

For components that are not specific for Reference Data, please see General Component Blocks on page 129.

16.4.1 Components

16.4.1.1 InstrumentExtension

Table 98: *InstrumentExtension*

Tag	FieldName	Req'd	Comments	Format
	AttrbGrp	N	Number of repeating InstrAttrib group entries.	

16.4.2 Implicit Components

16.4.2.1 AttrbGrp

Table 99: *AttrbGrp*

Tag	FieldName	Req'd	Comments	Format
870	NoInstrAttrib	N		NumInGroup
>871	InstrAttribType	N		int
>872	InstrAttribValue	N		String

16.4.2.2 RelSymDerivSecGrp

Table 100: *RelSymDerivSecGrp*

Tag	FieldName	Req'd	Comments	Format
146	NoRelatedSym	N	Specifies the number of repeating symbols (instruments) specified	NumInGroup
	Instrument	N	Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages" of the requested Security	
>15	Currency	N		Currency

Tag	FieldName	Req'd	Comments	Format
	InstrumentExtension	N	Insert here the set of "InstrumentExtension" fields defined in "Common Components of Application Messages" **OMX Comment:** Currently not supported	
	InstrmtLegGrp	N		
>58	Text	N	Comment, instructions, or other identifying information.	String
>354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.	Length
>355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.	data

16.4.2.3 InstrmtLegSecListGrp

Table 101: *InstrmtLegSecListGrp*

Tag	FieldName	Req'd	Comments	Format
555	NoLegs	N	Number of legs that make up the Security	NumInGroup
	InstrumentLeg	N	Insert here the set of "Instrument Legs" (leg symbology) fields defined in "Common Components of Application Messages" Required if NoLegs > 0	
	LegStipulations	N	Insert here the set of "LegStipulations" (leg symbology) fields defined in "Common Components of Application Messages" Required if NoLegs > 0 **OMX Comment:** Currently not supported	
	LegBenchmarkCurveData	N	Insert here the set of "LegBenchmarkCurveData" (leg symbology) fields defined in "Common Components of Application Messages" Required if NoLegs > 0 **OMX Comment:** Currently not supported	

16.4.2.4 PartyLstGrp

Table 102: *PartyLstGrp*

Tag	FieldName	Req'd	Comments	Format
453	NoPartyIDs	N	Repeating group below should contain unique combinations of PartyID, PartyIDSource, and PartyRole	NumInGroup
>448	PartyID	N	Used to identify Party. Required if PartyIDSource is specified. Required if NoPartyIDs > 0. **OMX Comment:** Contains the ID recommended for use by the marketplace	String
>447	PartyIDSource	N	Used to identify class source of PartyID value (e.g. BIC). Required if PartyID is specified. Required if NoPartyIDs > 0.	char

Tag	FieldName	Req'd	Comments	Format
>452	PartyRole	N	Identifies the type of PartyID (e.g. Executing Broker). Required if NoPartyIDs > 0. **OMX Comment:** Specifies the role the party has at the marketplace. In cases where a party has multiple roes, this is the primary role.	int
	PtysSubGrp	N	Repeating group of Party sub-identifiers. **OMX Comment:** Includes additional alternate identifiers as e.g. "short name" (1 - Firm), may also contain other contact information as full name (5 - Full legal name of firm) etc.	
>20083	Suspended	N	Indicates if the party is suspended (or not)	Boolean

16.4.2.5 PtysSubGrp

Table 103: *PtysSubGrp*

Tag	FieldName	Req'd	Comments	Format
802	NoPartySubIDs	N		NumInGroup
>523	PartySubID	N		String
>803	PartySubIDType	N		int

16.4.2.6 SecListGrp

Table 104: *SecListGrp*

Tag	FieldName	Req'd	Comments	Format
146	NoRelatedSym	N	Specifies the number of repeating symbols (instruments) specified	NumInGroup
	Instrument	N	Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages" of the requested Security	
	InstrumentExtension	N	Insert here the set of "InstrumentExtension" fields defined in "Common Components of Application Messages" **OMX Comment:** Currently not supported	
	FinancingDetails	N	Insert here the set of "FinancingDetails" fields defined in "Common Components of Application Messages" **OMX Comment:** FIXME: potentially relevant for Repo's	
	UndInstrmtGrp	N		
>15	Currency	N		Currency
	Stipulations	N	Insert here the set of "Stipulations" fields defined in "Common Components of Application Messages" **OMX Comment:** Currently not supported	

Tag	FieldName	Req'd	Comments	Format
	InstrmtLegSecListGrp	N		
	SpreadOrBench-markCurveData	N	Insert here the set of "SpreadOrBenchmarkCurve-Data" fields defined in "Common Components of Application Messages" **OMX Comment:** Currently not supported	
>58	Text	N	Comment, instructions, or other identifying information.	String
>354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.	Length
>355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.	data

16.4.2.7 SecLstUpdRelSymGrp

Table 105: *SecLstUpdRelSymGrp*

Tag	FieldName	Req'd	Comments	Format
146	NoRelatedSym	N	Specifies the number of repeating symbols (instruments) specified	NumInGroup
	Instrument	N	Insert here the set of "Instrument" (symbology) fields defined in "common components of application messages" of the requested Security	
	InstrumentExtension	N	Insert here the set of " InstrumentExtension " fields defined in " COMMON COMPONENTS OF APPLICATION MESSAGES "	
	FinancingDetails	N	Insert here the set of " FinancingDetails " fields defined in " COMMON COMPONENTS OF APPLICATION MESSAGES "	
	SecLstUpdRelSymsLeg-Grp	N		
	SpreadOrBench-markCurveData	N	Insert here the set of " SpreadOrBenchmarkCurve-Data " fields defined in " COMMON COMPONENTS OF APPLICATION MESSAGES "	
	SecurityTradingDefinition	N	Insert here the set of " SecurityTradignDefintions " fields defined in " COMMON COMPONENTS OF APPLICATION MESSAGES "	
>58	Text	N	Comment, instructions, or other identifying information.	String
>354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.	Length
>355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.	data

16.4.2.8 SecLstUpdRelSymsLegGrp

Table 106: *SecLstUpdRelSymsLegGrp*

Tag	FieldName	Req'd	Comments	Format
555	NoLegs	N	Number of legs that make up the Security	NumInGroup
	InstrumentLeg	N	Insert here the set of "Instrument Legs" (leg symbology) fields defined in "common components of application messages" Required if NoLegs > 0	
	LegStipulations	N	Insert here the set of "LegStipulations" (leg symbology) fields defined in "common components of application messages" Required if NoLegs > 0	
	LegBenchmarkCurveData	N	Insert here the set of "LegBenchmarkCurveData" (leg symbology) fields defined in "common components of application messages" Required if NoLegs > 0	

16.4.2.9 SecurityTradingDefinition

Table 107: *SecurityTradingDefinition*

Tag	FieldName	Req'd	Comments	Format
827	ExpirationCycle	N	Manner in which the security trading eligibility will expire.	int
561	RoundLot	N	Trading lot size of security	Qty
562	MinTradeVol	N	The minimum order quantity that can be submitted for an order	Qty
1140	MaxTradeVol	N	The maximum order quantity that can be submitted for a security	Qty
1141	NoMDFeedTypes	N	The number of feed types and corresponding book depths associated with a security	NumInGroup
>1022	MDFeedType	N	Describes a class of service for a given data feed	String
>264	MarketDepth	N	The depth of book associated with a particular feed type	int
1142	MatchAlgorithm	N	The type of algorithm used to match orders in a specific security on an electronic trading platform.	String
1143	MaxPriceVariation	N	The maximum price variation of an execution from one event to the next for a given security	Float
423	PriceType	N	The default price type used for trading	int
20004	MultilegPriceMethod	N	The default price method used for multilegs	int
20016	MultilegModel	N	Defines whether the instrument is a Pre-defined or User-defined multileg. **OMX Comment:** Currently not in FIX 5.0. OMX requests extension.	int

16.4.2.10 TrdSessLstGrp

Table 108: *TrdSessLstGrp*

Tag	FieldName	Req'd	Comments	Format
386	NoTradingSessions	Y		NumInGroup
>336	TradingSessionID	Y	Identifier for Trading Session	String
>625	TradingSessionSubID	N		String
>207	SecurityExchange	N		Exchange
>20036	MarketSegmentID	N		String
>339	TradSesMode	N	Trading Session Mode	int
>340	TradSesStatus	Y	State of trading session.	int
>567	TradSesStatusRejReason	N	Used with TradSesStatus = "Request Rejected"	int
>341	TradSesStartTime	N	Starting time of trading session	UTCTimes-tamp
>342	TradSesOpenTime	N	Time of the opening of the trading session	UTCTimes-tamp
>343	TradSesPreCloseTime	N	Time of pre-close of trading session	UTCTimes-tamp
>344	TradSesCloseTime	N	Closing time of trading session	UTCTimes-tamp
>345	TradSesEndTime	N	End time of trading session	UTCTimes-tamp
>58	Text	N		String
>354	EncodedTextLen	N	Must be set if EncodedText field is specified and must immediately precede it.	Length
>355	EncodedText	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.	data

16.5 Workflows

16.5.1 Introduction

The following workflows describe important aspects of the FIX interaction model.

The FIX Protocol Specification includes many of the workflows defined in this document; differences between the GENIUM and standard FIX specification are described.

16.5.2 Reference Data

T.b.d.

17 General Messages

17.1 BusinessMessageReject

Table 109: BusinessMessageReject

Tag	FieldName	Req'd	Comments	Format
	StandardHeader	Y	MsgType = j (lowercase)	
45	RefSeqNum	N	MsgSeqNum of rejected message	SeqNum
372	RefMsgType	Y	The MsgType of the FIX message being referenced.	String
379	BusinessRejectRefID	N	The value of the business-level "ID" field on the message being referenced. Required unless the corresponding ID field (see list above) was not specified.	String
380	BusinessRejectReason	Y	Code to identify reason for a Business Message Reject message.	int
58	Text	N	Where possible, message to explain reason for rejection	String
	StandardTrailer	Y		

18 General Component Blocks

18.1 Components

18.1.1 Instrument

Table 110: *Instrument*

Tag	FieldName	Req'd	Comments	Format
55	Symbol	N	Common, "human understood" representation of the security. SecurityID value can be specified if no symbol exists (e.g. non-exchange traded Collective Investment Vehicles) Use "[N/A]" for products which do not have a symbol.	String
48	SecurityID	N	Takes precedence in identifying security to counterparty over SecurityAltID block. Requires SecurityIDSource if specified.	String
22	SecurityIDSource	N	Required if SecurityID is specified.	String
1151	SecurityGroup	N	An exchange specific name assigned to a group of related securities which may be concurrently affected by market events and actions. **OMX Comment:** Used in reference data messages only	String
1079	MaturityTime	N		TZTimeOnly
965	SecurityStatus	N	Gives the current state of the instrument	String
967	StrikeMultiplier	N	Used for derivatives. Multiplier applied to the strike price for the purpose of calculating the settlement value.	float
968	StrikeValue	N	Used for derivatives. The number of shares/units for the financial instrument involved in the option trade.	float
969	MinPriceIncrement	N	Minimum price increment for the instrument. Could also be used to represent tick value.	float
	EvntGrp	N	Number of repeating EventType group entries.	

18.1.2 InstrumentLeg

Table 111: *InstrumentLeg*

Tag	FieldName	Req'd	Comments	Format
600	LegSymbol	N		String
601	LegSymbolSfx	N		String
602	LegSecurityID	N		String

Tag	FieldName	Req'd	Comments	Format
603	LegSecurityIDSource	N		String
	LegSecAltIDGrp	N		
608	LegCFICode	N		String
616	LegSecurityExchange	N		Exchange
620	LegSecurityDesc	N		String
621	EncodedLegSecurityDe-scLen	N		Length
622	EncodedLegSecurityDesc	N		data
623	LegRatioQty	N	Specific to the <InstrumentLeg> (not in <Instru-ment>)	float
624	LegSide	N	Specific to the <InstrumentLeg> (not in <Instru-ment>)	char

18.1.3 NestedParties

Table 112: *NestedParties*

Tag	FieldName	Req'd	Comments	Format
539	NoNestedPartyIDs	N	Repeating group below should contain unique combinations of NestedPartyID, NestedPartyID-Source, and NestedPartyRole	NumInGroup
>524	NestedPartyID	N	Used to identify source of NestedPartyID. Re-quired if NestedPartyIDSource is specified. Re-quired if NoNestedPartyIDs > 0.	String
>525	NestedPartyIDSource	N	Used to identify class source of NestedPartyID value (e.g. BIC). Required if NestedPartyID is specified. Required if NoNestedPartyIDs > 0.	char
>538	NestedPartyRole	N	Identifies the type of NestedPartyID (e.g. Execut-ing Broker). Required if NoNestedPartyIDs > 0. **OMX Comment:** Valid values: 4; 14.	int

18.1.4 NestedParties2

Table 113: *NestedParties2*

Tag	FieldName	Req'd	Comments	Format
756	NoNested2PartyIDs	N	Repeating group below should contain unique combinations of Nested2PartyID, Nested2PartyID-Source, and Nested2PartyRole	NumInGroup
>757	Nested2PartyID	N	Used to identify source of Nested2PartyID. Re-quired if Nested2PartyIDSource is specified. Re-quired if NoNested2PartyIDs > 0.	String
>758	Nested2PartyIDSource	N	Used to identify class source of Nested2PartyID value (e.g. BIC). Required if Nested2PartyID is specified. Required if NoNested2PartyIDs > 0.	char

Tag	FieldName	Req'd	Comments	Format
>759	Nested2PartyRole	N	Identifies the type of Nested2PartyID (e.g. Executing Broker). Required if NoNested2PartyIDs > 0.	int

18.1.5 Parties

Table 114: *Parties*

Tag	FieldName	Req'd	Comments	Format
453	NoPartyIDs	N	Repeating group below should contain unique combinations of PartyID, PartyIDSource, and PartyRole	NumInGroup
>448	PartyID	N	Used to identify source of PartyID. Required if PartyIDSource is specified. Required if NoPartyIDs > 0.	String
>447	PartyIDSource	N	Used to identify class source of PartyID value (e.g. BIC). Required if PartyID is specified. Required if NoPartyIDs > 0.	char
>452	PartyRole	N	Identifies the type of PartyID (e.g. Executing Broker). Required if NoPartyIDs > 0.	int

18.1.6 RootParties

Table 115: *RootParties*

Tag	FieldName	Req'd	Comments	Format
1116	NoRootPartyIDs	N	Repeating group below should contain unique combinations of RootPartyID, RootPartyIDSource, and RootPartyRole	NumInGroup
>1117	RootPartyID	N	Used to identify source of RootPartyID. Required if RootPartyIDSource is specified. Required if NoRootPartyIDs > 0.	String
>1118	RootPartyIDSource	N	Used to identify class source of RootPartyID value (e.g. BIC). Required if RootPartyID is specified. Required if NoRootPartyIDs > 0.	char
>1119	RootPartyRole	N	Identifies the type of RootPartyID (e.g. Executing Broker). Required if NoRootPartyIDs > 0.	int

18.1.7 UnderlyingInstrument

Table 116: *UnderlyingInstrument*

Tag	FieldName	Req'd	Comments	Format
311	UnderlyingSymbol	N		String
312	UnderlyingSymbolSfx	N		String
309	UnderlyingSecurityID	N		String
305	UnderlyingSecurityIDSource	N		String

Tag	FieldName	Req'd	Comments	Format
	UndSecAltIDGrp	N		
463	UnderlyingCFICode	N		String
307	UnderlyingSecurityDesc	N		String
364	EncodedUnderlyingSecurity-DescLen	N		Length
365	EncodedUnderlyingSecurity-Desc	N		data

18.2 Implicit Components

18.2.1 EvntGrp

Table 117: *EvntGrp*

Tag	FieldName	Req'd	Comments	Format
864	NoEvents	N		NumInGroup
>865	EventType	N		int
>866	EventDate	N		LocalMktDate

18.2.2 InstrmtLegGrp

Table 118: *InstrmtLegGrp*

Tag	FieldName	Req'd	Comments	Format
555	NoLegs	N	Number of legs Identifies a Multi-leg Execution if present and non-zero.	NumInGroup
	InstrumentLeg	N	Must be provided if Number of legs > 0	

18.2.3 LegSecAltIDGrp

Table 119: *LegSecAltIDGrp*

Tag	FieldName	Req'd	Comments	Format
604	NoLegSecurityAltID	N		String
>605	LegSecurityAltID	N		String
>606	LegSecurityAltIDSource	N		String

18.2.4 TrdgSesGrp

Table 120: *TrdgSesGrp*

Tag	FieldName	Req'd	Comments	Format
386	NoTradingSessions	N	Specifies the number of repeating TradingSessionIDs **OMX Comment:** Only allowed for TimeInForce = 0. Valid values: 1	NumInGroup
>336	TradingSessionID	N	Required if NoTradingSessions is > 0.	String
>625	TradingSessionSubID	N		String

18.2.5 UndInstrmtGrp

Table 121: *UndInstrmtGrp*

Tag	FieldName	Req'd	Comments	Format
711	NoUnderlyings	N	Number of underlyings	NumInGroup
	UnderlyingInstrument	N	Must be provided if Number of underlyings > 0	

18.2.6 UndSecAltIDGrp

Table 122: *UndSecAltIDGrp*

Tag	FieldName	Req'd	Comments	Format
457	NoUnderlyingSecurityAltID	N		NumInGroup
>458	UnderlyingSecurityAltID	N		String
>459	UnderlyingSecurityAltID-Source	N		String

19 Data Dictionary

19.1 Fields per Field Name

Table 123:

FieldName	Tag	Type	OMXLen	Desc	Valid values
Account	1	String		Account mnemonic as agreed between buy and sell sides, e.g. broker and institution or investor/intermediary and fund manager.	
AcctID-Source	660	int		Used to identify the source of the Account (1) code. This is especially useful if the account is a new account that the Respondent may not have setup yet in their system.	**Value** / **Description** 99 / Other (custom or proprietary)
Aggressor-Indicator	1057	Boolean		Used to identify whether the order initiator is an aggressor or not in the trade.	**Value** / **Description** Y / Order initiator is aggressor N / Order initiator is passive
AllocAccount	79	String		Sub-account mnemonic	
AllocAcctID-Source	661	int		Used to identify the source of the AllocAccount (79) code. See AcctIDSource (660) for valid values.	
AllocID	70	String		Unique identifier for allocation message. (Prior to FIX 4.1 this field was of type int)	
ApplVerID	1128	String		Specifies the service pack release being applied at message level. Enumerated field with values assigned at time of service pack release	**Value** / **Description** 7 / FIX50
BeginSeqNo	7	SeqNum		Message sequence number of first message in range to be resent	
BeginString	8	String		Identifies beginning of new message and protocol version. ALWAYS FIRST FIELD IN MESSAGE. (Always unencrypted) Valid values: FIXT.1.1	

FieldName	Tag	Type	OMXLen	Desc	Valid values
BodyLength	9	Length		Message length, in bytes, forward to the CheckSum field. ALWAYS SECOND FIELD IN MESSAGE. (Always unencrypted)	
BusinessRejectReason	380	int		Code to identify reason for a Business Message Reject message.	**Value** / **Description** 0 Other 1 Unknown ID 2 Unknown Security 3 Unknown Message Type 4 Application not available 5 Conditionally required field missing
BusinessRejectRefID	379	String		The value of the business-level "ID" field on the message being referenced.	
CheckSum	10	String		Three byte, simple checksum (see Volume 2: "Checksum Calculation" for description). ALWAYS LAST FIELD IN MESSAGE; i.e. serves, with the trailing <SOH>, as the end-of-message delimiter. Always defined as three characters. (Always unencrypted)	
ClearingBusinessDate	715	LocalMktDate		The "Clearing Business Date" referred to by this maintenance request.	
ClOrdID	11	String		Unique identifier for Order as assigned by the buy-side (institution, broker, intermediary etc.) (identified by SenderCompID (49) or OnBehalfOfCompID (5) as appropriate). Uniqueness must be guaranteed within a single trading day. Firms, particularly those which electronically submit multi-day orders, trade globally or throughout market close periods, should ensure uniqueness across days, for example by embedding a date within the ClOrdID field.	
CopyMsgIndicator	797	Boolean		Indicates whether or not this message is a drop copy of another message.	

FieldName	Tag	Type	OMXLen	Desc	Valid values
CorporateAction	292	MultipleCharValue		Identifies the type of Corporate Action. **OMX Comment:** Valid values t.b.d.	**Value** / **Description** A Ex-Dividend B Ex-Distribution C Ex-Rights E Ex-Interest F Cash Dividend G Stock Dividend H Non-Integer Stock Split I Reverse Stock Split J Standard-Integer Stock Split M Merger Reorganization N Rights Offering O Shareholder Meeting P Spinoff Q Tender Offer U CUSIP / Name Change
CumQty	14	Qty		Total quantity (e.g. number of shares) filled. (Prior to FIX 4.2 this field was of type int)	
Currency	15	Currency		Identifies currency used for price. Absence of this field is interpreted as the default for the security. It is recommended that systems provide the currency value whenever possible. See "Appendix 6-A: Valid Currency Codes" for information on obtaining valid values.	
CxlRejReason	102	int		Code to identify reason for cancel rejection.	**Value** / **Description** 0 Too late to cancel 1 Unknown order 2 Broker / Exchange Option 6 Duplicate ClOrdID (11) received 18 Invalid price increment

FieldName	Tag	Type	OMXLen	Desc	Valid values
					Value: 99 — Description: Other
CxlRejResponseTo	434	char		Identifies the type of request that a Cancel Reject is in response to.	Value: 1 — Description: Order cancel request Value: 2 — Description: Order cancel/replace request
DefaultApplVerID	1137	String		Specifies the service pack release being applied, by default, to message at the session level. Enumerated field with values assigned at time of service pack release. Uses same values as ApplVerID	
DeliverToCompID	128	String		Assigned value used to identify the firm targeted to receive the message if the message is delivered by a third party i.e. the third party firm identifier would be delivered in the TargetCompID (56) field and the ultimate receiver firm ID in this field.	
DeliverToSubID	129	String		Assigned value used to identify specific message recipient (i.e. trader) if the message is delivered by a third party	
DiscretionInst	388	char		Code to identify the price a DiscretionOffsetValue (389) is related to and should be mathematically added to.	Value: 7 — Description: Average Price Guarantee
DiscretionPrice	845	Price		The current discretionary price of the order	
DisplayMethod	1084	char		Defines what value to use in DisplayQty (1138). If not specified the default DisplayMethod is "1"	Value: 1 — Description: Initial (use original DisplayQty) Value: 2 — Description: New (use RefreshQty)
DisplayQty	1138	Qty		The quantity to be displayed . Required for reserve orders. On orders specifies the qty to be displayed, on execution reports the currently displayed quantity.	

FieldName	Tag	Type	OMXLen	Desc	Valid values
DueToRelated	329	Boolean		Indicates whether or not the halt was due to the Related Security being halted.	Value: N — Halt was not related to a halt of the related security; Y — Halt was due to related security being halted
EncodedLegSecurityDesc	622	data		Multileg instrument's individual security's EncodedSecurityDesc. See EncodedSecurityDesc (35) field for description	
EncodedLegSecurityDescLen	621	Length		Multileg instrument's individual security's EncodedSecurityDescLen. See EncodedSecurityDescLen (350) field for description	
EncodedMktSegmDesc	20039	data		Encoded (non-ASCII characters) representation of the MarketSegmDesc (tbd) field in the encoded format specified via the MessageEncoding (347) field. If used, the ASCII (English) representation should also be specified in the MarketSegmDesc field.	
EncodedMktSegmDescLen	20038	Length		Byte length of encoded (non-ASCII characters) EncodedSecurityDesc (351) field.	
EncodedText	355	data		Encoded (non-ASCII characters) representation of the Text (58) field in the encoded format specified via the MessageEncoding (347) field. If used, the ASCII (English) representation should also be specified in the Text field.	
EncodedTextLen	354	Length		Byte length of encoded (non-ASCII characters) EncodedText (355) field.	
EncodedUnderlyingSecurityDesc	365	data		Encoded (non-ASCII characters) representation of the UnderlyingSecurityDesc (307) field in the encoded format specified via the MessageEncoding (347) field. If used, the ASCII (English) representation should also be specified in the UnderlyingSecurityeDesc field.	
EncodedUnderlyingSecurityDescLen	364	Length		Byte length of encoded (non-ASCII characters) Encode-	

FieldName	Tag	Type	OMXLen	Desc	Valid values
				dUnderlyingSecurityDesc (365) field.	
Encrypt-Method	98	int		Method of encryption.	**Value** / **Description** 0 / None / Other
EndSeqNo	16	Se-qNum		Message sequence number of last message in range to be resent. If request is for a single message BeginSeqNo (7) = EndSeqNo. If request is for all messages subsequent to a particular message, EndSeqNo = "0" (represent-ing infinity).	
EventDate	866	Lo-calMkt-Date		Date of event	
EventType	865	int		Code to represent the type of event	**Value** / **Description** 5 / Activation 6 / Inactiviation 93 / Excluding combined Split and Issue Rights 94 / Under Drawing 95 / Company subject to Public Offer 96 / Excluding Participat-ing in Split 97 / Excluding Participat-ing in Rights 98 / Excluding Dividend 99 / On the Surveillance List
ExecID	17	String		Unique identifier of execution message as assigned by sell-side (broker, exchange, ECN) (will be 0 (zero) for ExecType (50) =I (Order Status)). Uniqueness must be guaran-teed within a single trading day or the life of a multi-day order. Firms which accept multi-day orders should con-sider embedding a date within the ExecID field to assure uniqueness across days. (Prior to FIX 4.1 this field was of type int)	

FieldName	Tag	Type	OMXLen	Desc	Valid values
ExecInst	18	MultipleCharValue		Instructions for order handling on exchange trading floor. If more than one instruction is applicable to an order, this field can contain multiple instructions separated by space. *** SOME VALUES HAVE BEEN REPLACED - See "Replaced Features and Supported Approach" *** (see Volume : "Glossary" for value definitions)	**Value** / **Description** G — All or none - AON **OMX Comment:** Currently not supported - use MinQty (110) = QrderQty (38) S — Suspend **OMX Comment:** Used to report an order as suspended in Execution Reports. Also used to send in an order as suspended. i — Imbalance Only z — Suspend on Connection Loss
ExecRefID	19	String		Reference identifier used with Trade Cancel and Trade Correct execution types. (Prior to FIX 4.1 this field was of type int)	
ExecRestatementReason	378	int		Code to identify reason for an ExecutionRpt message sent with ExecType=Restated or used when communicating an unsolicited cancel.	**Value** / **Description** 0 — GT corporate action 1 — GT renewal / restatement (no corporate action) 2 — Verbal change 3 — Repricing of order 6 — Cancel on Trading Halt 7 — Cancel on System Failure 9 — Canceled, not best 99 — Other
ExecType	150	char		Describes the specific ExecutionRpt (i.e. Pending Cancel) while OrdStatus (39) will always identify the current order status (i.e. Partially Filled) *** SOME VALUES HAVE BEEN REPLACED - See "Replaced Features and Supported Approach" ***	**Value** / **Description** 0 — New 4 — Canceled 5 — Replaced 8 — Rejected 9 — Suspended **OMX Comment:** Currently not used

FieldName	Tag	Type	OMXLen	Desc	Valid values
					Value / **Description** C – Expired D – Restated (Execution Report sent unsolicited by sellside, with ExecRestatementReason (378) set) F – Trade (partial fill or fill)
ExpirationCycle	827	int		Part of trading cycle when an instrument expires. Field is applicable for derivatives.	**Value** / **Description** 0 – Expire on trading session close (default) 1 – Expire on trading session open 2 – Expires at specified expiration as the eligibility
ExpireDate	432	LocalMktDate		Date of order expiration (last day the order can trade), always expressed in terms of the local market date. The time at which the order expires is determined by the local market's business practices	
ExpireTime	126	UTCTimestamp		Time/Date of order expiration (always expressed in UTC (Universal Time Coordinated, also known as "GMT") The meaning of expiration is specific to the context where the field is used. For orders, this is the expiration time of a Good Til Date TimeInForce. For Quotes - this is the expiration of the quote. Expiration time is provided across the quote message dialog to control the length of time of the overall quoting process. For collateral requests, this is the time by which collateral must be assigned. For collateral assignments, this is the time by which a response to the assignment is expected.	

FieldName	Tag	Type	OMXLen	Desc	Valid values
GapFillFlag	123	Boolean		Indicates that the Sequence Reset message is replacing administrative or application messages which will not be resent.	**Value** / **Description** N – Sequence Reset, Ignore Msg Seq Num (N/A For FIXML - Not Used) Y – Gap Fill Message, Msg Seq Num Field Valid
HaltReason	327	char		Denotes the reason for the Opening Delay or Trading Halt. **OMX Comment:** Valid values t.b.d.	**Value** / **Description** D – News Dissemination E – Order Influx M – Additional Information
HeartBtInt	108	int		Heartbeat interval (seconds)	
IndividualAllocID	467	String		Unique identifier for a specific NoAllocs (78) repeating group instance (e.g. for an AllocAccount).	
InstrAttribType	871	int		Code to represent the type of instrument attribute	**Value** / **Description** 1 – Flat (securities pay interest on a current basis but are traded without interest) 2 – Zero coupon 3 – Interest bearing (for Euro commercial paper when not issued at discount) 4 – No periodic payments 5 – Variable rate 6 – Less fee for put 7 – Stepped coupon 8 – Coupon period (if not semi-annual). Supply redemption date in the InstrAttribValue (872) field. 9 – When [and if] issued 10 – Original issue discount 11 – Callable, puttable 12 – Escrowed to Maturity

FieldName	Tag	Type	OMXLen	Desc	Valid values
					Value / **Description** 13 – Escrowed to redemption date - callable. Supply redemption date in the InstrAttribValue (872) field 14 – Pre-refunded 15 – In default 16 – Unrated 17 – Taxable 18 – Indexed 19 – Subject To Alternative Minimum Tax 20 – Original issue discount price. Supply price in the InstrAttribValue (872) field 21 – Callable below maturity value 22 – Callable without notice by mail to holder unless registered 23 – Price tick rules for security. Tick rule values to be expressed using InstrAttribValue [872] 24 – Trade type eligibility details for security. Trade types to be expressed using InstrAttribValue [872] 99 – Text. Supply the text of the attribute or disclaimer in the InstrAttribValue (872) field.
InstrAttribValue	872	String		Attribute value appropriate to the InstrAttribType (87) field.	
InViewOfCommon	328	Boolean		Indicates whether or not the halt was due to Common Stock trading being halted.	**Value** / **Description** N – Halt was not related to a halt of the common stock Y – Half was due to common stock being halted

FieldName	Tag	Type	OMXLen	Desc	Valid values
LastFragment	893	Boolean		Indicates whether this message is the last in a sequence of messages for those messages that support fragmentation, such as Allocation Instruction, Mass Quote, Security List, Derivative Security List	Value: N – Not Last Message; Y – Last Message
LastPx	31	Price		Price of this (last) fill.	
LastQty	32	Qty		Quantity (e.g. shares) bought/sold on this (last) fill. (Prior to FIX 4.2 this field was of type int)	
LeavesQty	151	Qty		Quantity open for further execution. If the OrdStatus (39) is Canceled, DoneForTheDay, Expired, Calculated, or Rejected (in which case the order is no longer active) then LeavesQty could be 0, otherwise LeavesQty = OrderQty (38) – CumQty (14). (Prior to FIX 4.2 this field was of type int)	
LegCFICode	608	String		Multileg instrument's individual security's CFICode. See CFICode (461) field for description	
LegRatioQty	623	float		The ratio of quantity for this individual leg relative to the entire multileg security.	
LegSecurityAltID	605	String		Multileg instrument's individual security's SecurityAltID. See SecurityAltID (455) field for description	
LegSecurityAltID-Source	606	String		Multileg instrument's individual security's SecurityAltID-Source. See SecurityAltIDSource (456) field for description	
LegSecurity-Desc	620	String		Multileg instrument's individual security's SecurityDesc. See SecurityDesc (07) field for description	
LegSecurityExchange	616	Exchange		Multileg instrument's individual security's SecurityExchange. See SecurityExchange (207) field for description	
LegSecurityID	602	String		Multileg instrument's individual security's SecurityID. See SecurityID (48) field for description	

FieldName	Tag	Type	OMXLen	Desc	Valid values
LegSecuri-tyIDSource	603	String		Multileg instrument's individual security's SecurityID-Source. See SecurityIDSource (22) field for description	
LegSide	624	char		The side of this individual leg (multileg security). See Side (54) field for description and values	
LegSymbol	600	String		Multileg instrument's individual security's Symbol. See Symbol (55) field for description	
LegSymbolS-fx	601	String		Multileg instrument's individual security's SymbolSfx. See SymbolSfx (65) field for description	
LotSize	20043	Qty		Quantity Increment for trading	
LotType	1093	char		Defines the lot type assigned to the order.	**Value** / **Description** 1 — Odd Lot 2 — Round Lot 3 — Block Lot
MarketDepth	264	int		Depth of market for Book Snapshot	**Value** / **Description** 0 — Full Book **OMX Comment:** FIX 5.0 SP1 1 — Top of Book **OMX Comment:** FIX 5.0 SP1
MarketSeg-mentDesc	20037	String		Description or name of Market Segment	
MarketSeg-mentID	20036	String		Market Segment identifier value	
MatchAlgo-rithm	1142	String		The type of algorithm used to match orders in a specific security Possible values are FIFO, Allocation, Pro-rata, Lead Market Maker, Currency Calendar	
MatchIncre-ment	1089	Qty		Allows orders to specify a minimum quantity that applies to every execution (one execution could be for multiple counter-orders). The order may still fill against smaller orders, but the cumulative	

FieldName	Tag	Type	OMXLen	Desc	Valid values
				quantity of the execution must be in multiples of the MatchIncrement.	
MatchStatus	573	char		The status of this trade with respect to matching or comparison.	**Value** / **Description** 0 – Compared, matched or affirmed 1 – Uncompared, unmatched, or unaffired
MatchType	574	String		The point in the matching process at which this trade was matched.	**Value** / **Description** 1 – One-Party Trade Report (privately negotiated trade) 4 – Auto-match 5 – Cross Auction 6 – Counter-Order Selection 7 – Call Auction 8 – Issuing/Buy-Back Auction **OMX Comment:** FIX 5.0 SP1
MaturityTime	1079	TZTimeOnly		Time of security's maturity expressed in local time with offset to UTC specified	
MaxPriceVariation	1143	Float		The maximum price variation of an execution from one event to the next for a given security	
MaxTradeVol	1140	Qty		The maximum order quantity that can be submitted for a security	
MDBookType	1021	int		Describes the type of book for which the feed is intended. Used when multiple feeds are provided over the same connection	**Value** / **Description** 2 – Price Depth 3 – Order Depth
MDEntryBuyer	288	String		Buying party in a trade	
MDEntryDate	272	UTCDateOnly		Date of Market Data Entry. (prior to FIX 4.4 field was of type UTCDate)	
MDEntryID	278	String		Unique Market Data Entry identifier.	

FieldName	Tag	Type	OMXLen	Desc	Valid values
MDEntryPositionNo	290	int		Display position of a bid or offer, numbered from most competitive to least competitive, per market side, beginning with .	
MDEntryPx	270	Price		Price of the Market Data Entry.	
MDEntryRefID	280	String		Refers to a previous MDEntryID (278).	
MDEntrySeller	289	String		Selling party in a trade	
MDEntrySize	271	Qty		Quantity or volume represented by the Market Data Entry.	
MDEntryTime	273	UTCTimeOnly		Time of Market Data Entry.	
MDEntryType	269	char		Type Market Data entry.	Value: 0 Description: Bid Value: 1 Description: Offer Value: 2 Description: Trade Value: 3 Description: Index Value
MDFeedType	1022	String		Describes a class of service for a given data feed, ie Regular and Market Maker, Bandwidth Intensive or Bandwidth Conservative	
MDPriceLevel	1023	int		Integer to convey the level of a bid or offer at a given price level. This is in contrast to MDEntryPositionNo which is used to convey the position of an order within a Price level	
MDQuoteType	1070	int		Identifies market data quote type.	Value: 0 Description: Indicative Value: 1 Description: Tradeable
MDSubBookType	1173	String		Describes a class of sub book, e.g. for the separation of various lot types. The Sub Book Type indicates that the following Market Data Entries belong to a non-integrated Sub Book. Whenever provided the Sub Book must be used together with MDPriceLevel and MDEntryPosi-	

FieldName	Tag	Type	OMXLen	Desc	Valid values
				tionNo in order to sort the order properly. **OMX Comment:** FIX 5.0 SP1	
MDUpdateAction	279	char		Type of Market Data update action.	**Value** / **Description** 0 New 1 Change 2 Delete 3 Delete Thru 4 Delete From 5 Overlay
MinPriceIncrement	969	float		Minimum price increase for a given exchange-traded Instrument	
MinPx	20045	Price		A minimum price	
MinQty	110	Qty		Minimum quantity of an order to be executed. (Prior to FIX 4.2 this field was of type int)	
MinTradeVol	562	Qty		The minimum trading volume for a security	
MktSegmSecurityExchange	20078	String		Marketplace used to identify the Marekt Segment. Same values as SecurityExchange (207).	
MsgSeqNum	34	SeqNum		Integer message sequence number.	
MsgType	35	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 ***	**Value** / **Description** 0 Heartbeat 1 Test Request 2 Resend Request 3 Reject 4 Sequence Reset 5 Logout 8 Execution Report 9 Order Cancel Reject A Logon D New Order - Single G Order Cancel/Replace Request (a.k.a. Order Modification Request)

FieldName	Tag	Type	OMXLen	Desc	Valid values
					Value: Q — Description: Don't Know Trade (DK) Value: X — Description: Market Data - Incremental Refresh Value: d — Description: Security Definition Value: f — Description: Security Status Value: j — Description: Business Message Reject Value: y — Description: Security List Value: AA — Description: Derivative Security List Value: AE — Description: Trade Capture Report Value: AR — Description: Trade Capture Report Ack Value: BJ — Description: Trading Session List Value: U3 — Description: SyncHeartbeat Value: U0 — Description: Market Data Statistics Value: U1 — Description: Market Segment Value: U2 — Description: One Sided Auction Request Value: U3 — Description: One Sided Auction Request Ack Value: U4 — Description: One Sided Auction Result Value: U5 — Description: NewsDataRequest Value: U6 — Description: NewsDataRequestReject Value: U7 — Description: NewsPublicationRequest Value: U8 — Description: NewsPublicationRequestReject
MultilegModel	20016	int		Specifies the type of multileg model the user is targeting.	Value: 0 — Description: Predefined Multileg Security Value: 2 — Description: Strategy Order
MultilegPriceMethod	20004	int		Defines the type of combination price the multileg uses	Value: 1 — Description: Net Price Value: 2 — Description: Reversed Net Price

FieldName	Tag	Type	OMXLen	Desc	Valid values
					Value / Description 3 Yield Difference 4 Individual 5 Weighted Average Price 6 Multiplied Price
Nested2PartyID	757	String		PartyID value within a "second instance" Nested repeating group. Same values as PartyID (448)	
Nested2PartyIDSource	758	char		PartyIDSource value within a "second instance" Nested repeating group. Same values as PartyIDSource (447)	
Nested2PartyRole	759	int		PartyRole value within a "second instance" Nested repeating group. Same values as PartyRole (452)	
NestedPartyID	524	String		PartyID value within a nested repeating group. Same values as PartyID (448)	
NestedPartyIDSource	525	char		PartyIDSource value within a nested repeating group. Same values as PartyIDSource (447)	
NestedPartyRole	538	int		PartyRole value within a nested repeating group. Same values as PartyRole (452)	
NewSeqNo	36	SeqNum		New sequence number	
NextExpectedMsgSeqNum	789	SeqNum		Next expected MsgSeqNum value to be received.	
NoAllocs	78	NumInGroup		Number of repeating AllocAccount (79)/AllocPrice (366) entries.	
NoEvents	864	NumInGroup		Number of repeating EventType entries.	
NoInstrAttrib	870	NumInGroup		Number of repeating InstrAttribType entries.	
NoInstruments	20041	NumINGroup		The number of Instrument entries.	
NoLegs	555	NumInGroup		Number of InstrumentLeg repeating group instances.	

FieldName	Tag	Type	OMXLen	Desc	Valid values
NoLegSecurityAltID	604	String		Multileg instrument's individual security's NoSecurityAltID. See NoSecurityAltID (454) field for description	
NoLotTypes	20042	NumInGroup		The number of Lot Type entries.	
NoMDEntries	268	NumInGroup		Number of entries in Market Data message.	
NoMDFeedTypes	1141	NumInGroup		The number of feed types and corresponding book depths associated with a security	
NoNested2PartyIDs	756	NumInGroup		Number of Nested2PartyID (757), Nested2PartyIDSource (758), and Nested2PartyRole (759) entries	
NoNestedPartyIDs	539	NumInGroup		Number of NestedPartyID (524), NestedPartyIDSource (525), and NestedPartyRole (538) entries	
NoPartyIDs	453	NumInGroup		Number of PartyID (448), PartyIDSource (447), and PartyRole (452) entries	
NoPartySubIDs	802	NumInGroup		Number of PartySubID (523)and PartySubIDType (803) entries	
NoRelatedSym	146	NumInGroup		Specifies the number of repeating symbols specified.	
NoRootPartyIDs	1116	NumInGroup		Number of RootPartyID (1117), RootPartyIDSource (1118), and RootPartyRole (1119) entries	
NoSides	552	NumInGroup		Number of Side repeating group instances.	**Value** / **Description** 1 — One Side 2 — Both Sides
NoStatsIndicators	1175	NumInGroup		Number of statistics indicator repeating group entries **OMX Comment:** FIX 5.0 SP1	
NoTickSizes	20044	NumInGroup		The number of Tick Size entries.	
NoTradingSessions	386	NumInGroup		Number of TradingSessionIDs (336) in repeating group.	
NoUnderlyings	711	NumInGroup		Number of underlying legs that make up the security.	
NoUnderlyingSecurityAltID	457	NumInGroup		Number of UnderlyingSecurityAltID (458) entries.	

FieldName	Tag	Type	OMXLen	Desc	Valid values
NumberOfOrders	346	int		Number of orders in the market.	
OnBehalfOfCompID	115	String		Assigned value used to identify firm originating message if the message was delivered by a third party i.e. the third party firm identifier would be delivered in the SenderCompID field and the firm originating the message in this field.	
OnBehalfOfSubID	116	String		Assigned value used to identify specific message originator (i.e. trader) if the message was delivered by a third party	
OrderCapacity	528	char		Designates the capacity of the firm placing the order. (as of FIX 4.3, this field replaced Rule80A (tag 47) -- used in conjunction with OrderRestrictions (529) field) (see Volume : "Glossary" for value definitions)	**Value** / **Description** A – Agency P – Principal (Note for CMS purposes, "Principal" includes "Proprietary")
OrderCategory	1115	char		Defines the type of interest behind a trade (fill or partial fill).	**Value** / **Description** 1 – Order 2 – Quote 3 – Privately Negotiated Trade 4 – Multileg order 5 – Linked order 6 – Quote Request 7 – Implied Order 8 – Cross Order
OrderID	37	String		Unique identifier for Order as assigned by sell-side (broker, exchange, ECN). Uniqueness must be guaranteed within a single trading day. Firms which accept multi-day orders should consider embedding a date within the OrderID field to assure uniqueness across days.	
OrderQty	38	Qty		Quantity ordered. This represents the number of shares for equities or par, face or nominal value for FI instruments. (Prior to FIX 4.2 this field was of type int)	

FieldName	Tag	Type	OMXLen	Desc	Valid values
OrderRestrictions	529	MultipleCharValue		Restrictions associated with an order. If more than one restriction is applicable to an order, this field can contain multiple instructions separated by space.	**Value** / **Description** 5 – Acting as Market Maker or Specialist in the security Y – Issuer Holding Z – Issue Price Stabilization Y – Issuer Holding Z – Issue Price Stabilization
OrdRejReason	103	int		Code to identify reason for order rejection. Note: Values 3, 4, and 5 will be used when rejecting an order due to pre-allocation information errors.	**Value** / **Description** 0 – Broker / Exchange option 1 – Unknown symbol 2 – Exchange closed 3 – Order exceeds limit 4 – Too late to enter 5 – Unknown order 6 – Duplicate Order (e.g. dupe ClOrdID) 7 – Duplicate of a verbally communicated order 8 – Stale order 9 – Trade along required 10 – Invalid Investor ID 11 – Unsupported order characteristic 12 – Surveillence Option 13 – Incorrect quantity 14 – Incorrect allocated quantity 15 – Unknown account(s) 18 – Invalid price increment 99 – Other
OrdStatus	39	char		Identifies current status of order. *** SOME VALUES HAVE BEEN REPLACED - See "Replaced Features and Supported Approach" *** (see	**Value** / **Description** 0 – New 1 – Partially filled

FieldName	Tag	Type	OMXLen	Desc	Valid values
				Volume : "Glossary" for value definitions)	**Value** / **Description** 2 Filled 4 Canceled 8 Rejected 9 Suspended C Expired
OrdType	40	char		Order type. *** SOME VALUES ARE NO LONGER USED - See "Deprecated (Phased-out) Features and Supported Approach" *** (see Volume : "Glossary" for value definitions)	**Value** / **Description** 1 Market 2 Limit Q Counter-order selection
OrigClOrdID	41	String		ClOrdID (11) of the previous order (NOT the initial order of the day) as assigned by the institution, used to identify the previous order in cancel and cancel/replace requests.	
OrigSecondaryTradeID	1127	String		Used to preserve original secondary trade id when original trade is being referenced in a subsequent trade transaction such as a transfer	
OrigSendingTime	122	UTCTimestamp		Original time of message transmission (always expressed in UTC (Universal Time Coordinated, also known as "GMT") when transmitting orders as the result of a resend request.	
OrigTradeID	1126	String		Used to preserve original trade id when original trade is being referenced in a subsequent trade transaction such as a transfer	
ParentMktSegmID	20040	String		Reference to a parent Market Segment. See MarketSegmentID (tbd)	
PartyID	448	String		Party identifier/code. See PartyIDSource (447) and PartyRole (452). See "Appendix 6-G – Use of <Parties> Component Block"	
PartyIDSource	447	char		Identifies class or source of the PartyID (448) value. Required if PartyID is specified. Note: applicable values de-	**Value** / **Description** C Generally accepted market participant

FieldName	Tag	Type	OMXLen	Desc	Valid values
				pend upon PartyRole (452) specified. See "Appendix 6-G – Use of <Parties> Component Block"	**Value** / **Description** identifier (e.g. NASD mnemonic) D – Proprietary / Custom code
PartyReportID	20080	String		Unique message ID for a Participant list	
PartyReqID	20079	String		Unique ID for a Participant request	
PartyRequestResult	20081	int		Return code for a Participant request	**Value** / **Description** 0 – Valid request 1 – Invalid or unsupported request 2 – No data found that match selection criteria 3 – Not authorized to retrieve data
PartyRole	452	int		Identifies the type or role of the PartyID (448) specified. See "Appendix 6-G – Use of <Parties> Component Block" (see Volume : "Glossary" for value definitions)	**Value** / **Description** 1 – Executing Firm (formerly FIX 4.2 ExecBroker) 7 – Entering Firm 12 – Executing Trader (associated with Executing Firm - actually executes) 14 – Giveup Clearing Firm (firm to which trade is given up) 17 – Contra Firm **OMX Comment:** Used in Trade Capture Reports for trade confirmations divulding a single side only. Also used in Quote Requests to specify receiving parties. 21 – Clearing Organization 36 – Entering trader 37 – Contra trader 58 – Entering Unit 59 – Executing Unit

FieldName	Tag	Type	OMXLen	Desc	Valid values
PartySubID	523	String		Sub-identifier (e.g. Clearing Account for PartyRole (452)=Clearing Firm, Locate ID # for PartyRole=Locate/Lending Firm, etc). Not required when using PartyID (448), PartyIDSource (447), and PartyRole.	
PartySubID-Type	803	int		Type of PartySubID (523) value 4000+ = Reserved and available for bi-laterally agreed upon user defined values	**Value** / **Description** 1 Firm 5 Full legal name of firm 6 Postal address 7 Phone number 8 Email address 9 Contact name 16 BIC 18 Registered address 21 Fax number
Password	554	String		Password or passphrase.	
PossDupFlag	43	Boolean		Indicates possible retransmission of message with this sequence number	**Value** / **Description** N Original transmission Y Possible duplicate
PreTradeAnonymity	1091	Boolean		Allows trader to explicitly request anonymity or disclosure in pre-trade market data feeds. Anonymity is relevant in markets where counterparties are regularly disclosed in order depth feeds. Disclosure is relevant when counterparties are not normally visible.	
PreviouslyReported	570	Boolean		Indicates if the trade capture report was previously reported to the counterparty	**Value** / **Description** N Not reported to counterparty Y Perviously reported to counterparty
Price	44	Price		Price per unit of quantity (e.g. per share)	
PriceIncrement	20046	Price		Price increment (tick size)	

FieldName	Tag	Type	OMXLen	Desc	Valid values
PriceType	423	int		Code to represent the price type. (For Financing transactions PriceType implies the "repo type" – Fixed or Floating – 9 (Yield) or 6 (Spread) respectively - and Price (44) gives the corresponding "repo rate". See Volume : "Glossary" for further value definitions)	**Value** / **Description** 1 – Percentage (i.e. percent of par) (often called "dollar price" for fixed income) **OMX Comment:** Only relevant for Fixed Income trading 2 – Per unit (i.e. per share or contract) **OMX Comment:** Default value, should be specified (if applicable) for Fixed Income trading 3 – Fixed amount (absolute value) **OMX Comment:** Only allowed for IOI's 4 – Discount - percentage points below par **OMX Comment:** Only allowed for IOI's 5 – Premium - percentage points over par **OMX Comment:** Only allowed for IOI's 6 – Spread (basis points spread) **OMX Comment:** Only allowed for IOI's 7 – TED Price **OMX Comment:** Only allowed for IOI's 8 – TED Yield **OMX Comment:** Only allowed for IOI's 9 – Yield **OMX Comment:** Only relevant for Fixed Income trading
PublishTrdIndicator	852	int		Indicates if a trade should be reported via a market reporting service. **OMX Comment:** Field is Boolean in FIX 5.0, OMX requests change	**Value** / **Description** 0 – Do Not Report Trade 1 – Report Trade 2 – Deferred Publication

FieldName	Tag	Type	OMXLen	Desc	Valid values
QuoteCondition	276	MultipleStringValue		Space-delimited list of conditions describing a quote.	**Value** / **Description** Z — Order Imbalance 4 — Better Prices in Conditional Orders **OMX Comment:** FIX 5.0 SP1 3 — Rest of Book VWAP **OMX Comment:** FIX 5.0 SP1
RefMsgType	372	String		The MsgType (35) of the FIX message being referenced.	
RefOrderID	1080	String		The ID reference to the order being hit or taken	
RefOrderIDSource	1081	char		Used to specify what identifier, provided in order depth market data, to use when hitting (taking) a specific order.	**Value** / **Description** 0 — SecondaryOrdeID (198)
RefreshQty	1088	Qty		Defines the quantity used to refresh DisplayQty.	
RefSeqNum	45	SeqNum		Reference message sequence number	
RefTagID	371	int		The tag number of the FIX field being referenced.	
ResetSeqNumFlag	141	Boolean		Indicates that the both sides of the FIX session should reset sequence numbers.	**Value** / **Description** N — No Y — Yes, reset sequence numbers
RootPartyID	1117	String		PartyID value within a root parties component. Same values as PartyID (448)	
RootPartyIDSource	1118	char		PartyIDSource value within a root parties component. Same values as PartyIDSource (447)	
RootPartyRole	1119	int		PartyRole value within a root parties component. Same values as PartyRole (452)	
RoundLot	561	Qty		The trading lot size of a security	
SecondaryClOrdID	526	String		Assigned by the party which originates the order. Can be used to provide the ClOrdID (11) used by an exchange or executing system.	

FieldName	Tag	Type	OMXLen	Desc	Valid values
SecondaryExecID	527	String		Assigned by the party which accepts the order. Can be used to provide the ExecID (17) used by an exchange or executing system. **OMX Comment:** Can be used by participants who assign their own ExecID (17) and then roll the ExecID assigned by an exchange into this field.	
SecondaryOrderID	198	String		Assigned by the party which accepts the order. Can be used to provide the OrderID (37) used by an exchange or executing system.	
SecondaryTrdType	855	int		Additional TrdType (see tag 828) assigned to a trade by trade match system.	**Value** / **Description** 98 — On-Hours **OMX Comment:** Not in FIX Standard 99 — Off-hours **OMX Comment:** Not in FIX Standard
SecurityExchange	207	Exchange		Market used to help identify a security. Valid values: See "Appendix 6-C"	
SecurityGroup	1151	String		An exchange specific name assigned to a group of related securities which may be concurrently affected by market events and actions.	
SecurityID	48	String		Security identifier value of SecurityIDSource (22) type (e.g. CUSIP, SEDOL, ISIN, etc). Requires SecurityIDSource.	
SecurityIDSource	22	String		Identifies class or source of the SecurityID (48) value. Required if SecurityID is specified. 100+ are reserved for private security identifications	**Value** / **Description** 8 — Exchange Symbol
SecurityReportID	964	int		Security Report ID. Unique identifier for the Security Report.	
SecurityReqID	320	String		Unique ID of a Security Definition Request.	
SecurityRequestResult	560	int		The results returned to a Security Request message	**Value** / **Description** 0 — Valid request

FieldName	Tag	Type	OMXLen	Desc	Valid values
					Value / **Description** 1 – Invalid or unsupported request 2 – No instruments found that match selection criteria 3 – Not authorized to retrieve instrument data
SecurityResponseID	322	String		Unique ID of a Security Definition message.	
SecurityResponseType	323	int		Type of Security Definition message response.	**Value** / **Description** 1 – Accept security proposal as-is 2 – Accept security proposal with revisions as indicated in the message 5 – Reject security proposal
SecurityStatus	965	String		Used for derivatives. Denotes the current state of the Instrument.	**Value** / **Description** 1 – Active 2 – Inactive
SecurityTradingEvent	1174	int		Identifies an event related to a SecurityTradingStatus (326). An event occurs and is gone, it is not a state that applies for a period of time. **OMX Comment:** FIX 5.0 SP1	**Value** / **Description** 1 – Order imbalance, auction is extended 2 – Price Volatility Interruption 3 – Trading resumes (after Halt) 4 – Change of Trading Session 5 – Change of Trading Subsession 6 – Change of Security Status 7 – Change of Book Type 8 – Change of Market Depth

FieldName	Tag	Type	OMXLen	Desc	Valid values
SecurityTradingStatus	326	int		Identifies the trading status applicable to the transaction. **OMX Comment:** Valid values t.b.d.	**Value** / **Description** 1 Opening delay 2 Trading halt 7 Market Imbalance Buy 8 Market Imbalance Sell 17 Ready to trade (start of session) 21 Pre-open 22 Opening Rotation 23 Fast Market
SecurityUpdateAction	980	char			**Value** / **Description** A Add D Delete M Modify
SenderCompID	49	String		Assigned value used to identify firm sending message.	
SenderSubID	50	String		Assigned value used to identify specific message originator (desk, trader, etc.)	
SendingTime	52	UTC-Timestamp		Time of message transmission (always expressed in UTC (Universal Time Coordinated, also known as "GMT")	
SessionRejectReason	373	int		Code to identify reason for a session-level Reject message.	**Value** / **Description** 0 Invalid Tag Number 1 Required Tag Missing 2 Tag not defined for this message type 3 Undefined tag 4 Tag specified without a value 5 Value is incorrect (out of range) for this tag 6 Incorrect data format for value 7 Decryption problem 8 Signature problem

FieldName	Tag	Type	OMXLen	Desc	Valid values
					Value / **Description** 9 — CompID problem 10 — SendingTime Accuracy Problem 11 — Invalid MsgType 12 — XML Validation Error 13 — Tag appears more than once 14 — Tag specified out of required order 15 — Repeating group fields out of order 16 — Incorrect NumInGroup count for repeating group 17 — Non "Data" value includes field delimiter (<SOH> character)
SettlCurrency	120	Currency		Currency code of settlement denomination.	
SettlDate	64	LocalMktDate		Specific date of trade settlement (SettlementDate) in YYYYMMDD format. If present, this field overrides SettlType (63). This field is required if the value of SettlType (63) is 6 (Future) or 8 (Sellers Option). This field must be omitted if the value of SettlType (63) is 7 (When and If Issued) (expressed in local time at place of settlement)	
Side	54	char		Side of order (see Volume : "Glossary" for value definitions)	**Value** / **Description** 1 — Buy 2 — Sell 7 — Undisclosed (valid for IOI and List Order messages only) **OMX Comment:** Valid for IOI messages only B — "As Defined" (for use with multileg instruments)

FieldName	Tag	Type	OMXLen	Desc	Valid values
					Value / **Description** **OMX Comment:** Valid for multileg Orders only
SolicitedFlag	377	Boolean		Indicates whether or not the order was solicited.	**Value** / **Description** N — Was not solicited Y — Was solicited
StatsType	1176	Int		Type of statistics the MDEntry is eligible to be included in **OMX Comment:** FIX 5.0 SP1	**Value** / **Description** 1 — Exchange Last 2 — High / Low Price 3 — Average Price (VWAP, TWAP...) 4 — Turnover (Px * Qty)
StrikeMultiplier	967	float		Used for derivatives. Multiplier applied to the strike price for the purpose of calculating the settlement value.	
StrikeValue	968	float		Used for derivatives. The number of shares/units for the financial instrument involved in the option trade.	
SubscriptionRequestType	263	char		Subscription Request Type	**Value** / **Description** 0 — Snapshot 1 — Snapshot + Updates (Subscribe) 2 — Disable previous Snapshot + Update Request (Unsubscribe)
Suspended	20083	Boolean		Indicates whether a suspension applies or not.	**Value** / **Description** 0 — Not suspended 1 — Suspended
Symbol	55	String		Ticker symbol. Common, "human understood" representation of the security. SecurityID (48) value can be specified if no symbol exists (e.g. non-exchange traded Collective Investment Vehicles)	

FieldName	Tag	Type	OMXLen	Desc	Valid values
				Use "[N/A]" for products which do not have a symbol.	
TargetCompID	56	String		Assigned value used to identify receiving firm.	
TargetSubID	57	String		Assigned value used to identify specific individual or unit intended to receive message. "ADMIN" reserved for administrative messages not intended for a specific user.	
TestReqID	112	String		Identifier included in Test Request message to be returned in resulting Heartbeat	
Text	58	String		Free format text string (Note: this field does not have a specified maximum length)	
TimeInForce	59	char		Specifies how long the order remains in effect. Absence of this field is interpreted as DAY. NOTE not applicable to CIV Orders. (see Volume : "Glossary" for value definitions)	**Value** / **Description** 0 Day (or session) 1 Good Till Cancel (GTC) 2 At the Opening (OPG) 3 Immediate Or Cancel (IOC) 4 Fill Or Kill (FOK) 5 Good Till Crossing (GTX) 6 Good Till Date (GTD) 7 At the Close y Good Through Crossing z At Next Crossing
ToNoRelParties	20082	int		Total number of participants	
TotNoRelatedSym	393	int		Total number of securities. (Prior to FIX 4.4 this field was named TotalNumSecurities)	
TradeCondition	277	MultipleStringValue		Space-delimited list of conditions describing a trade	**Value** / **Description** X Crossed AV Outside Spread **OMX Comment:** FIX 5.0 SP1

FieldName	Tag	Type	OMXLen	Desc	Valid values
TradeDate	75	LocalMktDate		Indicates date of trade referenced in this message in YYYYMMDD format. Absence of this field indicates current day (expressed in local time at place of trade).	
TradeHandlingInstr	1123	char		Specified how the Trade Capture Report should be handled by the Respondent.	Value: 0 — Description: Trade Confirmation Value: 2 — Description: One-Party Report for Matching
TradeID	1003	String		The unique ID assigned to the trade entity once it is received or matched by the exchange or central counterparty.	
TradeReportID	571	String		Unique identifier of trade capture report	
TradeReportRefID	572	String		Reference identifier used with CANCEL and REPLACE transaction types.	
TradeReportRejectReason	751	int		Reason Trade Capture Request was rejected. 4000+ Reserved and available for bi-laterally agreed upon user-defined values	Value: 0 — Description: Successful (default) Value: 1 — Description: Invalid party onformation Value: 2 — Description: Unknown instrument Value: 3 — Description: Unauthorized to report trades Value: 4 — Description: Invalid trade type Value: 99 — Description: Other
TradeReportTransType	487	int		Identifies Trade Report message transaction type (Prior to FIX 4.4 this field was of type char)	Value: 0 — Description: New Value: 1 — Description: Cancel Value: 2 — Description: Replace
TradeReportType	856	int		Type of Trade Report	Value: 0 — Description: Submit Value: 1 — Description: Alleged Value: 2 — Description: Accept Value: 3 — Description: Decline Value: 4 — Description: Addendum

FieldName	Tag	Type	OMXLen	Desc	Valid values
					Value: 5 — No/Was 7 — (Locked-In) Trade Break 11 — Alleged New 12 — Alleged Addendum 14 — Alleged Trade Report Cancel 15 — Alleged (Locked-In) Trade Break
TradingSessionID	336	String		Identifier for Trading Session Can be used to represent a specific market trading session (e.g. "PRE-OPEN", "CROSS_2", "AFTER-HOURS", "TOSTNET", "TOSTNET2", etc). To specify good for session where session spans more than one calendar day, use TimeInForce = Day in conjunction with TradingSessionID. Values should be bi-laterally agreed to between counterparties. Firms may register Trading Session values on the FIX website (presently a document maintained within "ECN and Exchanges" working group section).	
TradingSessionSubID	625	String		Optional market assigned sub identifier for a trading session. Usage is determined by market or counterparties. Used by US based futures markets to identify exchange specific execution time bracket codes as required by US market regulations.	
TradSesCloseTime	344	UTC-Timestamp		Closing time of the trading session	
TradSesEndTime	345	UTC-Timestamp		End time of the trading session	
TradSesMode	339	int		Trading Session Mode	Value: 1 — Testing 3 — Production

FieldName	Tag	Type	OMXLen	Desc	Valid values
TradSesOpenTime	342	UTCTimestamp		Time of the opening of the trading session	
TradSesPreCloseTime	343	UTCTimestamp		Time of the pre-closed of the trading session	
TradSesReqID	335	String		Unique ID of a Trading Session Status message.	
TradSesStartTime	341	UTCTimestamp		Starting time of the trading session	
TradSesStatus	340	int		State of the trading session.	**Value** / **Description** 1 — Halted **OMX Comment:** T.b.d. 2 — Open **OMX Comment:** T.b.d. 3 — Closed **OMX Comment:** T.b.d.
TradSesStatusRejReason	567	int		Indicates the reason a Trading Session Status Request was rejected.	**Value** / **Description** 1 — Unknown or invalid TradingSessionID 99 — Other
TransactTime	60	UTCTimestamp		Time of execution/order creation (expressed in UTC (Universal Time Coordinated, also known as "GMT")	
TransBkdTime	483	UTCTimestamp		For CIV A date and time stamp to indicate the time a CIV order was booked by the fund manager. For derivatives a date and time stamp to indicate when this order was booked with the agent prior to submission to the VMU. Indicates the time at which the order was finalized between the buyer and seller prior to submission.	
TrdMatchID	880	String		Identifier assigned to a trade by a matching system. **OMX Comment:** Identifies a group of fills matched in the same execution round and at the same price. Established for a single aggressive order	

FieldName	Tag	Type	OMXLen	Desc	Valid values
				but can include many contra orders.	
TrdRptStatus	939	int		Trade Report Status	**Value** / **Description** 0 Accepted 1 Rejected
TrdType	828	int		Type of Trade:	**Value** / **Description** 0 Regular Trade 4 Late Trade 6 Weighted Average Price Trade 38 Block trade (same as large trade) 47 Financing transaction (includes repo and stock lending) 48 Non-Standard Settlment **OMX Comment:** FIX 5.0 SP1 49 Derivative Related Transaction **OMX Comment:** FIX 5.0 SP1 50 Portfolio Trade **OMX Comment:** FIX 5.0 SP1 51 Volume Weighted Average Trade **OMX Comment:** FIX 5.0 SP1 52 Exchange Granted Trade **OMX Comment:** FIX 5.0 SP1 53 Repurchase Agreement **OMX Comment:** FIX 5.0 SP1 54 OTC **OMX Comment:** FIX 5.0 SP1

FieldName	Tag	Type	OMXLen	Desc	Valid values
UnderlyingCFICode	463	String		Underlying security's CFICode. Valid values: see CFICode (461) field	
UnderlyingSecurityAltID	458	String		Alternate Security identifier value for this underlying security of UnderlyingSecurityAltIDSource (459) type (e.g. CUSIP, SEDOL, ISIN, etc). Requires UnderlyingSecurityAltIDSource.	
UnderlyingSecurityAltIDSource	459	String		Identifies class or source of the UnderlyingSecurityAltID (458) value. Required if UnderlyingSecurityAltID is specified. Valid values: Same valid values as the SecurityIDSource (22) field	
UnderlyingSecurityDesc	307	String		Underlying security's SecurityDesc. See SecurityDesc (07) field for description	
UnderlyingSecurityID	309	String		Underlying security's SecurityID. See SecurityID (48) field for description	
UnderlyingSecurityIDSource	305	String		Underlying security's SecurityIDSource. Valid values: see SecurityIDSource (22) field	
UnderlyingSymbol	311	String		Underlying security's Symbol. See Symbol (55) field for description	
UnderlyingSymbolSfx	312	String		Underlying security's SymbolSfx. See SymbolSfx (65) field for description	
WorkingIndicator	636	Boolean		Indicates if the order is currently being worked. Applicable only for OrdStatus = "New". For open outcry markets this indicates that the order is being worked in the crowd. For electronic markets it indicates that the order has transitioned from a contingent order to a market order.	**Value** / **Description** N – Order has been accepted but not yet in a working state Y – Order is currently being worked
Yield	236	Percentage		Yield percentage. (Note tag # was reserved in FIX 4.1, added in FIX 4.3)	

19.2 Fields per Tag Number

Table 124:

Tag	FieldName	Type	OMXLen	Desc	Valid values
1	Account	String		Account mnemonic as agreed between buy and sell sides, e.g. broker and institution or investor/intermediary and fund manager.	
7	BeginSeqNo	SeqNum		Message sequence number of first message in range to be resent	
8	BeginString	String		Identifies beginning of new message and protocol version. ALWAYS FIRST FIELD IN MESSAGE. (Always unencrypted) Valid values: FIXT.1.1	
9	BodyLength	Length		Message length, in bytes, forward to the CheckSum field. ALWAYS SECOND FIELD IN MESSAGE. (Always unencrypted)	
10	CheckSum	String		Three byte, simple checksum (see Volume 2: "Checksum Calculation" for description). ALWAYS LAST FIELD IN MESSAGE; i.e. serves, with the trailing <SOH>, as the end-of-message delimiter. Always defined as three characters. (Always unencrypted)	
11	ClOrdID	String		Unique identifier for Order as assigned by the buy-side (institution, broker, intermediary etc.) (identified by SenderCompID (49) or OnBehalfOfCompID (5) as appropriate). Uniqueness must be guaranteed within a single trading day. Firms, particularly those which electronically submit multi-day orders, trade globally or throughout market close periods, should ensure uniqueness across days, for example by embedding a date within the ClOrdID field.	
14	CumQty	Qty		Total quantity (e.g. number of shares) filled. (Prior to FIX 4.2 this field was of type int)	

Tag	FieldName	Type	OMXLen	Desc	Valid values
15	Currency	Currency		Identifies currency used for price. Absence of this field is interpreted as the default for the security. It is recommended that systems provide the currency value whenever possible. See "Appendix 6-A: Valid Currency Codes" for information on obtaining valid values.	
16	EndSeqNo	SeqNum		Message sequence number of last message in range to be resent. If request is for a single message BeginSeqNo (7) = EndSeqNo. If request is for all messages subsequent to a particular message, EndSeqNo = "0" (representing infinity).	
17	ExecID	String		Unique identifier of execution message as assigned by sell-side (broker, exchange, ECN) (will be 0 (zero) for ExecType (50) =I (Order Status)). Uniqueness must be guaranteed within a single trading day or the life of a multi-day order. Firms which accept multi-day orders should consider embedding a date within the ExecID field to assure uniqueness across days. (Prior to FIX 4.1 this field was of type int)	
18	ExecInst	MultipleCharValue		Instructions for order handling on exchange trading floor. If more than one instruction is applicable to an order, this field can contain multiple instructions separated by space. *** SOME VALUES HAVE BEEN REPLACED - See "Replaced Features and Supported Approach" *** (see Volume : "Glossary" for value definitions)	**Value** / **Description** G — All or none - AON **OMX Comment:** Currently not supported - use MinQty (110) = QrderQty (38) S — Suspend **OMX Comment:** Used to report an order as suspended in Execution Reports. Also used to send in an order as suspended. i — Imbalance Only z — Suspend on Connection Loss

Tag	FieldName	Type	OMXLen	Desc	Valid values
19	ExecRefID	String		Reference identifier used with Trade Cancel and Trade Correct execution types. (Prior to FIX 4.1 this field was of type int)	
22	SecurityID-Source	String		Identifies class or source of the SecurityID (48) value. Required if SecurityID is specified. 100+ are reserved for private security identifications	**Value** / **Description**: 8 Exchange Symbol
31	LastPx	Price		Price of this (last) fill.	
32	LastQty	Qty		Quantity (e.g. shares) bought/sold on this (last) fill. (Prior to FIX 4.2 this field was of type int)	
34	MsgSeqNum	SeqNum		Integer message sequence number.	
35	MsgType	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 ***	**Value** / **Description**: 0 Heartbeat; 1 Test Request; 2 Resend Request; 3 Reject; 4 Sequence Reset; 5 Logout; 8 Execution Report; 9 Order Cancel Reject; A Logon; D New Order - Single; G Order Cancel/Replace Request (a.k.a. Order Modification Request); Q Don't Know Trade (DK); X Market Data - Incremental Refresh; d Security Definition; f Security Status; j Business Message Reject; y Security List; AA Derivative Security List; AE Trade Capture Report

Tag	FieldName	Type	OMXLen	Desc	Valid values
					Value: AR — Trade Capture Report Ack BJ — Trading Session List U3 — SyncHeartbeat U0 — Market Data Statistics U1 — Market Segment U2 — One Sided Auction Request U3 — One Sided Auction Request Ack U4 — One Sided Auction Result U5 — NewsDataRequest U6 — NewsDataRequestReject U7 — NewsPublicationRequest U8 — NewsPublicationRequestReject
36	NewSeqNo	SeqNum		New sequence number	
37	OrderID	String		Unique identifier for Order as assigned by sell-side (broker, exchange, ECN). Uniqueness must be guaranteed within a single trading day. Firms which accept multi-day orders should consider embedding a date within the OrderID field to assure uniqueness across days.	
38	OrderQty	Qty		Quantity ordered. This represents the number of shares for equities or par, face or nominal value for FI instruments. (Prior to FIX 4.2 this field was of type int)	
39	OrdStatus	char		Identifies current status of order. *** SOME VALUES HAVE BEEN REPLACED - See "Replaced Features and Supported Approach" *** (see Volume : "Glossary" for value definitions)	Value: 0 — New 1 — Partially filled 2 — Filled 4 — Canceled 8 — Rejected

Tag	FieldName	Type	OMXLen	Desc	Valid values
					Value / **Description** 9 Suspended C Expired
40	OrdType	char		Order type. *** SOME VALUES ARE NO LONGER USED - See "Deprecated (Phased-out) Features and Supported Approach" *** (see Volume : "Glossary" for value definitions)	**Value** / **Description** 1 Market 2 Limit Q Counter-order selection
41	OrigClOrdID	String		ClOrdID (11) of the previous order (NOT the initial order of the day) as assigned by the institution, used to identify the previous order in cancel and cancel/replace requests.	
43	PossDupFlag	Boolean		Indicates possible retransmission of message with this sequence number	**Value** / **Description** N Original transmission Y Possible duplicate
44	Price	Price		Price per unit of quantity (e.g. per share)	
45	RefSeqNum	SeqNum		Reference message sequence number	
48	SecurityID	String		Security identifier value of SecurityIDSource (22) type (e.g. CUSIP, SEDOL, ISIN, etc). Requires SecurityIDSource.	
49	SenderCompID	String		Assigned value used to identify firm sending message.	
50	SenderSubID	String		Assigned value used to identify specific message originator (desk, trader, etc.)	
52	SendingTime	UTCTimestamp		Time of message transmission (always expressed in UTC (Universal Time Coordinated, also known as "GMT")	
54	Side	char		Side of order (see Volume : "Glossary" for value definitions)	**Value** / **Description** 1 Buy 2 Sell 7 Undisclosed (valid for IOI and List Order messages only)

Tag	FieldName	Type	OMXLen	Desc	Valid values
					Value / **Description** **OMX Comment:** Valid for IOI messages only B — "As Defined" (for use with multileg instruments) **OMX Comment:** Valid for multileg Orders only
55	Symbol	String		Ticker symbol. Common, "human understood" representation of the security. SecurityID (48) value can be specified if no symbol exists (e.g. non-exchange traded Collective Investment Vehicles) Use "[N/A]" for products which do not have a symbol.	
56	TargetCompID	String		Assigned value used to identify receiving firm.	
57	TargetSubID	String		Assigned value used to identify specific individual or unit intended to receive message. "ADMIN" reserved for administrative messages not intended for a specific user.	
58	Text	String		Free format text string (Note: this field does not have a specified maximum length)	
59	TimeInForce	char		Specifies how long the order remains in effect. Absence of this field is interpreted as DAY. NOTE not applicable to CIV Orders. (see Volume : "Glossary" for value definitions)	**Value** / **Description** 0 — Day (or session) 1 — Good Till Cancel (GTC) 2 — At the Opening (OPG) 3 — Immediate Or Cancel (IOC) 4 — Fill Or Kill (FOK) 5 — Good Till Crossing (GTX) 6 — Good Till Date (GTD) 7 — At the Close y — Good Through Crossing z — At Next Crossing

Tag	FieldName	Type	OMXLen	Desc	Valid values
60	TransactTime	UTCTimestamp		Time of execution/order creation (expressed in UTC (Universal Time Coordinated, also known as "GMT")	
64	SettlDate	LocalMktDate		Specific date of trade settlement (SettlementDate) in YYYYMMDD format. If present, this field overrides SettlType (63). This field is required if the value of SettlType (63) is 6 (Future) or 8 (Sellers Option). This field must be omitted if the value of SettlType (63) is 7 (When and If Issued) (expressed in local time at place of settlement)	
70	AllocID	String		Unique identifier for allocation message. (Prior to FIX 4.1 this field was of type int)	
75	TradeDate	LocalMktDate		Indicates date of trade referenced in this message in YYYYMMDD format. Absence of this field indicates current day (expressed in local time at place of trade).	
78	NoAllocs	NumInGroup		Number of repeating AllocAccount (79)/AllocPrice (366) entries.	
79	AllocAccount	String		Sub-account mnemonic	
98	EncryptMethod	int		Method of encryption.	**Value** / **Description** 0 / None / Other
102	CxlRejReason	int		Code to identify reason for cancel rejection.	**Value** / **Description** 0 / Too late to cancel 1 / Unknown order 2 / Broker / Exchange Option 6 / Duplicate ClOrdID (11) received 18 / Invalid price increment 99 / Other
103	OrdRejReason	int		Code to identify reason for order rejection. Note: Values 3, 4, and 5 will be used when rejecting an order due to preallocation information errors.	**Value** / **Description** 0 / Broker / Exchange option

Tag	FieldName	Type	OMXLen	Desc	Valid values
					Value / **Description** 1 Unknown symbol 2 Exchange closed 3 Order exceeds limit 4 Too late to enter 5 Unknown order 6 Duplicate Order (e.g. dupe ClOrdID) 7 Duplicate of a verbally communicated order 8 Stale order 9 Trade along required 10 Invalid Investor ID 11 Unsupported order characteristic 12 Surveillence Option 13 Incorrect quantity 14 Incorrect allocated quantity 15 Unknown account(s) 18 Invalid price increment 99 Other
108	HeartBtInt	int		Heartbeat interval (seconds)	
110	MinQty	Qty		Minimum quantity of an order to be executed. (Prior to FIX 4.2 this field was of type int)	
112	TestReqID	String		Identifier included in Test Request message to be returned in resulting Heartbeat	
115	OnBehalfOfCompID	String		Assigned value used to identify firm originating message if the message was delivered by a third party i.e. the third party firm identifier would be delivered in the SenderCompID field and the firm originating the message in this field.	
116	OnBehalfOfSubID	String		Assigned value used to identify specific message originator (i.e. trader) if the message was delivered by a third party	

Tag	FieldName	Type	OMXLen	Desc	Valid values
120	SettlCurrency	Currency		Currency code of settlement denomination.	
122	OrigSendingTime	UTCTimestamp		Original time of message transmission (always expressed in UTC (Universal Time Coordinated, also known as "GMT") when transmitting orders as the result of a resend request.	
123	GapFillFlag	Boolean		Indicates that the Sequence Reset message is replacing administrative or application messages which will not be resent.	**Value** / **Description** N – Sequence Reset, Ignore Msg Seq Num (N/A For FIXML - Not Used) Y – Gap Fill Message, Msg Seq Num Field Valid
126	ExpireTime	UTCTimestamp		Time/Date of order expiration (always expressed in UTC (Universal Time Coordinated, also known as "GMT") The meaning of expiration is specific to the context where the field is used. For orders, this is the expiration time of a Good Til Date TimeInForce. For Quotes - this is the expiration of the quote. Expiration time is provided across the quote message dialog to control the length of time of the overall quoting process. For collateral requests, this is the time by which collateral must be assigned. For collateral assignments, this is the time by which a response to the assignment is expected.	
128	DeliverToCompID	String		Assigned value used to identify the firm targeted to receive the message if the message is delivered by a third party i.e. the third party firm identifier would be delivered in the TargetCompID (56) field and the ultimate receiver firm ID in this field.	
129	DeliverToSubID	String		Assigned value used to identify specific message recipient	

Tag	FieldName	Type	OMXLen	Desc	Valid values
				(i.e. trader) if the message is delivered by a third party	
141	ResetSeqNumFlag	Boolean		Indicates that the both sides of the FIX session should reset sequence numbers.	**Value** / **Description** N – No Y – Yes, reset sequence numbers
146	NoRelatedSym	NumInGroup		Specifies the number of repeating symbols specified.	
150	ExecType	char		Describes the specific ExecutionRpt (i.e. Pending Cancel) while OrdStatus (39) will always identify the current order status (i.e. Partially Filled) *** SOME VALUES HAVE BEEN REPLACED - See "Replaced Features and Supported Approach" ***	**Value** / **Description** 0 – New 4 – Canceled 5 – Replaced 8 – Rejected 9 – Suspended **OMX Comment:** Currently not used C – Expired D – Restated (Execution Report sent unsolicited by sellside, with ExecRestatementReason (378) set) F – Trade (partial fill or fill)
151	LeavesQty	Qty		Quantity open for further execution. If the OrdStatus (39) is Canceled, DoneForTheDay, Expired, Calculated, or Rejected (in which case the order is no longer active) then LeavesQty could be 0, otherwise LeavesQty = OrderQty (38) – CumQty (14). (Prior to FIX 4.2 this field was of type int)	
198	SecondaryOrderID	String		Assigned by the party which accepts the order. Can be used to provide the OrderID (37) used by an exchange or executing system.	
207	SecurityExchange	Exchange		Market used to help identify a security. Valid values: See "Appendix 6-C"	

Tag	FieldName	Type	OMXLen	Desc	Valid values
236	Yield	Percentage		Yield percentage. (Note tag # was reserved in FIX 4.1, added in FIX 4.3)	
263	SubscriptionRequestType	char		Subscription Request Type	**Value** / **Description** 0 – Snapshot 1 – Snapshot + Updates (Subscribe) 2 – Disable previous Snapshot + Update Request (Unsubscribe)
264	MarketDepth	int		Depth of market for Book Snapshot	**Value** / **Description** 0 – Full Book **OMX Comment:** FIX 5.0 SP1 1 – Top of Book **OMX Comment:** FIX 5.0 SP1
268	NoMDEntries	NumInGroup		Number of entries in Market Data message.	
269	MDEntryType	char		Type Market Data entry.	**Value** / **Description** 0 – Bid 1 – Offer 2 – Trade 3 – Index Value
270	MDEntryPx	Price		Price of the Market Data Entry.	
271	MDEntrySize	Qty		Quantity or volume represented by the Market Data Entry.	
272	MDEntryDate	UTCDateOnly		Date of Market Data Entry. (prior to FIX 4.4 field was of type UTCDate)	
273	MDEntryTime	UTCTimeOnly		Time of Market Data Entry.	
276	QuoteCondition	MultipleStringValue		Space-delimited list of conditions describing a quote.	**Value** / **Description** Z – Order Imbalance 4 – Better Prices in Conditional Orders **OMX Comment:** FIX 5.0 SP1

Tag	FieldName	Type	OMXLen	Desc	Valid values
					Value / **Description** 3 / Rest of Book VWAP **OMX Comment:** FIX 5.0 SP1
277	TradeCondition	MultipleStringValue		Space-delimited list of conditions describing a trade	**Value** / **Description** X / Crossed AV / Outside Spread **OMX Comment:** FIX 5.0 SP1
278	MDEntryID	String		Unique Market Data Entry identifier.	
279	MDUpdateAction	char		Type of Market Data update action.	**Value** / **Description** 0 / New 1 / Change 2 / Delete 3 / Delete Thru 4 / Delete From 5 / Overlay
280	MDEntryRefID	String		Refers to a previous MDEntryID (278).	
288	MDEntryBuyer	String		Buying party in a trade	
289	MDEntrySeller	String		Selling party in a trade	
290	MDEntryPositionNo	int		Display position of a bid or offer, numbered from most competitive to least competitive, per market side, beginning with .	
292	CorporateAction	MultipleCharValue		Identifies the type of Corporate Action. **OMX Comment:** Valid values t.b.d.	**Value** / **Description** A / Ex-Dividend B / Ex-Distribution C / Ex-Rights E / Ex-Interest F / Cash Dividend G / Stock Dividend H / Non-Integer Stock Split

Tag	FieldName	Type	OMXLen	Desc	Valid values
					Value: I — Description: Reverse Stock Split Value: J — Description: Standard-Integer Stock Split Value: M — Description: Merger Reorganization Value: N — Description: Rights Offering Value: O — Description: Shareholder Meeting Value: P — Description: Spinoff Value: Q — Description: Tender Offer Value: U — Description: CUSIP / Name Change
305	UnderlyingSecurityID-Source	String		Underlying security's SecurityIDSource. Valid values: see SecurityID-Source (22) field	
307	UnderlyingSecurityDesc	String		Underlying security's SecurityDesc. See SecurityDesc (07) field for description	
309	UnderlyingSecurityID	String		Underlying security's SecurityID. See SecurityID (48) field for description	
311	UnderlyingSymbol	String		Underlying security's Symbol. See Symbol (55) field for description	
312	UnderlyingSymbolSfx	String		Underlying security's SymbolSfx. See SymbolSfx (65) field for description	
320	SecurityReqID	String		Unique ID of a Security Definition Request.	
322	SecurityResponseID	String		Unique ID of a Security Definition message.	
323	SecurityResponseType	int		Type of Security Definition message response.	Value: 1 — Description: Accept security proposal as-is Value: 2 — Description: Accept security proposal with revisions as indicated in the message Value: 5 — Description: Reject security proposal

Tag	FieldName	Type	OMXLen	Desc	Valid values
326	SecurityTradingStatus	int		Identifies the trading status applicable to the transaction. **OMX Comment:** Valid values t.b.d.	**Value** / **Description** 1 Opening delay 2 Trading halt 7 Market Imbalance Buy 8 Market Imbalance Sell 17 Ready to trade (start of session) 21 Pre-open 22 Opening Rotation 23 Fast Market
327	HaltReason	char		Denotes the reason for the Opening Delay or Trading Halt. **OMX Comment:** Valid values t.b.d.	**Value** / **Description** D News Dissemination E Order Influx M Additional Information
328	InViewOfCommon	Boolean		Indicates whether or not the halt was due to Common Stock trading being halted.	**Value** / **Description** N Halt was not related to a halt of the common stock Y Half was due to common stock being halted
329	DueToRelated	Boolean		Indicates whether or not the halt was due to the Related Security being halted.	**Value** / **Description** N Halt was not related to a halt of the related security Y Halt was due to related security being halted
335	TradSesReqID	String		Unique ID of a Trading Session Status message.	
336	TradingSessionID	String		Identifier for Trading Session Can be used to represent a specific market trading session (e.g. "PRE-OPEN", "CROSS_2", "AFTER-HOURS", "TOSTNET", "TOSTNET2", etc). To specify good for session where session spans more	

Tag	FieldName	Type	OMXLen	Desc	Valid values
				than one calendar day, use TimeInForce = Day in conjunction with TradingSessionID. Values should be bi-laterally agreed to between counterparties. Firms may register Trading Session values on the FIX website (presently a document maintained within "ECN and Exchanges" working group section).	
339	TradSesMode	int		Trading Session Mode	**Value** **Description** 1 Testing 3 Production
340	TradSesStatus	int		State of the trading session.	**Value** **Description** 1 Halted **OMX Comment:** T.b.d. 2 Open **OMX Comment:** T.b.d. 3 Closed **OMX Comment:** T.b.d.
341	TradSesStartTime	UTCTimestamp		Starting time of the trading session	
342	TradSesOpenTime	UTCTimestamp		Time of the opening of the trading session	
343	TradSesPreCloseTime	UTCTimestamp		Time of the pre-closed of the trading session	
344	TradSesCloseTime	UTCTimestamp		Closing time of the trading session	
345	TradSesEndTime	UTCTimestamp		End time of the trading session	
346	NumberOfOrders	int		Number of orders in the market.	
354	EncodedTextLen	Length		Byte length of encoded (non-ASCII characters) EncodedText (355) field.	

Tag	FieldName	Type	OMXLen	Desc	Valid values
355	EncodedText	data		Encoded (non-ASCII characters) representation of the Text (58) field in the encoded format specified via the MessageEncoding (347) field. If used, the ASCII (English) representation should also be specified in the Text field.	
364	EncodedUnderlyingSecurityDescLen	Length		Byte length of encoded (non-ASCII characters) EncodedUnderlyingSecurityDesc (365) field.	
365	EncodedUnderlyingSecurityDesc	data		Encoded (non-ASCII characters) representation of the UnderlyingSecurityDesc (307) field in the encoded format specified via the MessageEncoding (347) field. If used, the ASCII (English) representation should also be specified in the UnderlyingSecurityeDesc field.	
371	RefTagID	int		The tag number of the FIX field being referenced.	
372	RefMsgType	String		The MsgType (35) of the FIX message being referenced.	
373	SessionRejectReason	int		Code to identify reason for a session-level Reject message.	**Value** / **Description** 0 Invalid Tag Number 1 Required Tag Missing 2 Tag not defined for this message type 3 Undefined tag 4 Tag specified without a value 5 Value is incorrect (out of range) for this tag 6 Incorrect data format for value 7 Decryption problem 8 Signature problem 9 CompID problem 10 SendingTime Accuracy Problem 11 Invalid MsgType 12 XML Validation Error 13 Tag appears more than once

Tag	FieldName	Type	OMXLen	Desc	Valid values
					Value / **Description** 14 — Tag specified out of required order 15 — Repeating group fields out of order 16 — Incorrect NumIn-Group count for repeating group 17 — Non "Data" value includes field delimiter (<SOH> character)
377	SolicitedFlag	Boolean		Indicates whether or not the order was solicited.	**Value** / **Description** N — Was not solicited Y — Was solicited
378	ExecRestate-mentReason	int		Code to identify reason for an ExecutionRpt message sent with ExecType=Restated or used when communicating an unsolicited cancel.	**Value** / **Description** 0 — GT corporate action 1 — GT renewal / restatement (no corporate action) 2 — Verbal change 3 — Repricing of order 6 — Cancel on Trading Halt 7 — Cancel on System Failure 9 — Canceled, not best 99 — Other
379	BusinessRe-jectRefID	String		The value of the business-level "ID" field on the message being referenced.	
380	BusinessRe-jectReason	int		Code to identify reason for a Business Message Reject message.	**Value** / **Description** 0 — Other 1 — Unknown ID 2 — Unknown Security 3 — Unknown Message Type 4 — Application not available

Tag	FieldName	Type	OMXLen	Desc	Valid values
					Value / **Description** 5 — Conditionally required field missing
386	NoTradingSessions	NumInGroup		Number of TradingSessionIDs (336) in repeating group.	
388	DiscretionInst	char		Code to identify the price a DiscretionOffsetValue (389) is related to and should be mathematically added to.	**Value** / **Description** 7 — Average Price Guarantee
393	TotNoRelatedSym	int		Total number of securities. (Prior to FIX 4.4 this field was named TotalNumSecurities)	
423	PriceType	int		Code to represent the price type. (For Financing transactions PriceType implies the "repo type" – Fixed or Floating – 9 (Yield) or 6 (Spread) respectively - and Price (44) gives the corresponding "repo rate". See Volume : "Glossary" for further value definitions)	**Value** / **Description** 1 — Percentage (i.e. percent of par) (often called "dollar price" for fixed income) **OMX Comment:** Only relevant for Fixed Income trading 2 — Per unit (i.e. per share or contract) **OMX Comment:** Default value, should be specified (if applicable) for Fixed Income trading 3 — Fixed amount (absolute value) **OMX Comment:** Only allowed for IOI's 4 — Discount - percentage points below par **OMX Comment:** Only allowed for IOI's 5 — Premium - percentage points over par **OMX Comment:** Only allowed for IOI's 6 — Spread (basis points spread) **OMX Comment:** Only allowed for IOI's 7 — TED Price **OMX Comment:** Only allowed for IOI's

Tag	FieldName	Type	OMXLen	Desc	Valid values
					Value / **Description** 8 — TED Yield **OMX Comment:** Only allowed for IOI's 9 — Yield **OMX Comment:** Only relevant for Fixed Income trading
432	ExpireDate	LocalMktDate		Date of order expiration (last day the order can trade), always expressed in terms of the local market date. The time at which the order expires is determined by the local market's business practices	
434	CxlRejResponseTo	char		Identifies the type of request that a Cancel Reject is in response to.	**Value** / **Description** 1 — Order cancel request 2 — Order cancel/replace request
447	PartyIDSource	char		Identifies class or source of the PartyID (448) value. Required if PartyID is specified. Note: applicable values depend upon PartyRole (452) specified. See "Appendix 6-G – Use of <Parties> Component Block"	**Value** / **Description** C — Generally accepted market participant identifier (e.g. NASD mnemonic) D — Proprietary / Custom code
448	PartyID	String		Party identifier/code. See PartyIDSource (447) and PartyRole (452). See "Appendix 6-G – Use of <Parties> Component Block"	
452	PartyRole	int		Identifies the type or role of the PartyID (448) specified. See "Appendix 6-G – Use of <Parties> Component Block" (see Volume : "Glossary" for value definitions)	**Value** / **Description** 1 — Executing Firm (formerly FIX 4.2 ExecBroker) 7 — Entering Firm 12 — Executing Trader (associated with Executing Firm - actually executes) 14 — Giveup Clearing Firm (firm to which trade is given up) 17 — Contra Firm

Tag	FieldName	Type	OMXLen	Desc	Valid values
					Value / **Description** **OMX Comment:** Used in Trade Capture Reports for trade confirmations divulding a single side only. Also used in Quote Requests to specify receiving parties. 21 Clearing Organization 36 Entering trader 37 Contra trader 58 Entering Unit 59 Executing Unit
453	NoPartyIDs	NumInGroup		Number of PartyID (448), PartyIDSource (447), and PartyRole (452) entries	
457	NoUnderlyingSecurityAltID	NumInGroup		Number of UnderlyingSecurityAltID (458) entries.	
458	UnderlyingSecurityAltID	String		Alternate Security identifier value for this underlying security of UnderlyingSecurityAltIDSource (459) type (e.g. CUSIP, SEDOL, ISIN, etc). Requires UnderlyingSecurityAltIDSource.	
459	UnderlyingSecurityAltIDSource	String		Identifies class or source of the UnderlyingSecurityAltID (458) value. Required if UnderlyingSecurityAltID is specified. Valid values: Same valid values as the SecurityIDSource (22) field	
463	UnderlyingCFICode	String		Underlying security's CFICode. Valid values: see CFICode (461) field	
467	IndividualAllocID	String		Unique identifier for a specific NoAllocs (78) repeating group instance (e.g. for an AllocAccount).	
483	TransBkdTime	UTCTimestamp		For CIV A date and time stamp to indicate the time a CIV order was booked by the fund manager. For derivatives a date and time stamp to indicate when this order was booked with the agent prior to submission	

Tag	FieldName	Type	OMXLen	Desc	Valid values
				to the VMU. Indicates the time at which the order was finalized between the buyer and seller prior to submission.	
487	TradeReport-TransType	int		Identifies Trade Report message transaction type (Prior to FIX 4.4 this field was of type char)	**Value** / **Description** 0 New 1 Cancel 2 Replace
523	PartySubID	String		Sub-identifier (e.g. Clearing Account for PartyRole (452)=Clearing Firm, Locate ID # for PartyRole=Locate/Lending Firm, etc). Not required when using PartyID (448), PartyIDSource (447), and PartyRole.	
524	NestedPartyID	String		PartyID value within a nested repeating group. Same values as PartyID (448)	
525	NestedPartyIDSource	char		PartyIDSource value within a nested repeating group. Same values as PartyID-Source (447)	
526	SecondaryClOrdID	String		Assigned by the party which originates the order. Can be used to provide the ClOrdID (11) used by an exchange or executing system.	
527	SecondaryExecID	String		Assigned by the party which accepts the order. Can be used to provide the ExecID (17) used by an exchange or executing system. **OMX Comment:** Can be used by participants who assign their own ExecID (17) and then roll the ExecID assigned by an exchange into this field.	
528	OrderCapacity	char		Designates the capacity of the firm placing the order. (as of FIX 4.3, this field replaced Rule80A (tag 47) -- used in conjunction with OrderRestrictions (529) field) (see Volume : "Glossary" for value definitions)	**Value** / **Description** A Agency P Principal (Note for CMS purposes, "Principal" includes "Proprietary")

Tag	FieldName	Type	OMXLen	Desc	Valid values
529	OrderRestrictions	MultipleCharValue		Restrictions associated with an order. If more than one restriction is applicable to an order, this field can contain multiple instructions separated by space.	**Value** / **Description** 5 – Acting as Market Maker or Specialist in the security Y – Issuer Holding Z – Issue Price Stabilization Y – Issuer Holding Z – Issue Price Stabilization
538	NestedPartyRole	int		PartyRole value within a nested repeating group. Same values as PartyRole (452)	
539	NoNestedPartyIDs	NumInGroup		Number of NestedPartyID (524), NestedPartyIDSource (525), and NestedPartyRole (538) entries	
552	NoSides	NumInGroup		Number of Side repeating group instances.	**Value** / **Description** 1 – One Side 2 – Both Sides
554	Password	String		Password or passphrase.	
555	NoLegs	NumInGroup		Number of InstrumentLeg repeating group instances.	
560	SecurityRequestResult	int		The results returned to a Security Request message	**Value** / **Description** 0 – Valid request 1 – Invalid or unsupported request 2 – No instruments found that match selection criteria 3 – Not authorized to retrieve instrument data
561	RoundLot	Qty		The trading lot size of a security	
562	MinTradeVol	Qty		The minimum trading volume for a security	
567	TradSesStatusRejReason	int		Indicates the reason a Trading Session Status Request was rejected.	**Value** / **Description** 1 – Unknown or invalid TradingSessionID 99 – Other

Tag	FieldName	Type	OMXLen	Desc	Valid values
570	PreviouslyReported	Boolean		Indicates if the trade capture report was previously reported to the counterparty	**Value** / **Description** N – Not reported to counterparty Y – Perviously reported to counterparty
571	TradeReportID	String		Unique identifier of trade capture report	
572	TradeReportRefID	String		Reference identifier used with CANCEL and REPLACE transaction types.	
573	MatchStatus	char		The status of this trade with respect to matching or comparison.	**Value** / **Description** 0 – Compared, matched or affirmed 1 – Uncompared, unmatched, or unaffired
574	MatchType	String		The point in the matching process at which this trade was matched.	**Value** / **Description** 1 – One-Party Trade Report (privately negotiated trade) 4 – Auto-match 5 – Cross Auction 6 – Counter-Order Selection 7 – Call Auction 8 – Issuing/Buy-Back Auction **OMX Comment:** FIX 5.0 SP1
600	LegSymbol	String		Multileg instrument's individual security's Symbol. See Symbol (55) field for description	
601	LegSymbolSfx	String		Multileg instrument's individual security's SymbolSfx. See SymbolSfx (65) field for description	
602	LegSecurityID	String		Multileg instrument's individual security's SecurityID. See SecurityID (48) field for description	

Tag	FieldName	Type	OMXLen	Desc	Valid values
603	LegSecuri-tyIDSource	String		Multileg instrument's individual security's SecurityID-Source. See SecurityIDSource (22) field for description	
604	NoLegSecu-rityAltID	String		Multileg instrument's individual security's NoSecurityAltID. See NoSecurityAltID (454) field for description	
605	LegSecu-rityAltID	String		Multileg instrument's individual security's SecurityAltID. See SecurityAltID (455) field for description	
606	LegSecu-rityAltID-Source	String		Multileg instrument's individual security's SecurityAltID-Source. See SecurityAltIDSource (456) field for description	
608	LegCFICode	String		Multileg instrument's individual security's CFICode. See CFICode (461) field for description	
616	LegSecuri-tyExchange	Ex-change		Multileg instrument's individual security's SecurityEx-change. See SecurityExchange (207) field for description	
620	LegSecurity-Desc	String		Multileg instrument's individual security's SecurityDesc. See SecurityDesc (07) field for description	
621	Encod-edLegSecuri-tyDescLen	Length		Multileg instrument's individual security's EncodedSecuri-tyDescLen. See EncodedSecurityDe-scLen (350) field for description	
622	Encod-edLegSecuri-tyDesc	data		Multileg instrument's individual security's EncodedSecuri-tyDesc. See EncodedSecurityDesc (35) field for description	
623	LegRatioQty	float		The ratio of quantity for this individual leg relative to the entire multileg security.	
624	LegSide	char		The side of this individual leg (multileg security). See Side (54) field for description and values	
625	TradingSes-sionSubID	String		Optional market assigned sub identifier for a trading session. Usage is determined by market or counterparties. Used by US based futures markets to identify exchange specific execution time	

Tag	FieldName	Type	OMXLen	Desc	Valid values
				bracket codes as required by US market regulations.	
636	WorkingIndicator	Boolean		Indicates if the order is currently being worked. Applicable only for OrdStatus = "New". For open outcry markets this indicates that the order is being worked in the crowd. For electronic markets it indicates that the order has transitioned from a contingent order to a market order.	**Value** / **Description** N – Order has been accepted but not yet in a working state Y – Order is currently being worked
660	AcctIDSource	int		Used to identify the source of the Account (1) code. This is especially useful if the account is a new account that the Respondent may not have setup yet in their system.	**Value** / **Description** 99 – Other (custom or proprietary)
661	AllocAcctIDSource	int		Used to identify the source of the AllocAccount (79) code. See AcctIDSource (660) for valid values.	
711	NoUnderlyings	NumInGroup		Number of underlying legs that make up the security.	
715	ClearingBusinessDate	LocalMktDate		The "Clearing Business Date" referred to by this maintenance request.	
751	TradeReportRejectReason	int		Reason Trade Capture Request was rejected. 4000+ Reserved and available for bi-laterally agreed upon user-defined values	**Value** / **Description** 0 – Successful (default) 1 – Invalid party onformation 2 – Unknown instrument 3 – Unauthorized to report trades 4 – Invalid trade type 99 – Other
756	NoNested2PartyIDs	NumInGroup		Number of Nested2PartyID (757), Nested2PartyIDSource (758), and Nested2PartyRole (759) entries	
757	Nested2PartyID	String		PartyID value within a "second instance" Nested repeating group. Same values as PartyID (448)	

Tag	FieldName	Type	OMXLen	Desc	Valid values
758	Nested2PartyIDSource	char		PartyIDSource value within a "second instance" Nested repeating group. Same values as PartyIDSource (447)	
759	Nested2PartyRole	int		PartyRole value within a "second instance" Nested repeating group. Same values as PartyRole (452)	
789	NextExpectedMsgSeqNum	SeqNum		Next expected MsgSeqNum value to be received.	
797	CopyMsgIndicator	Boolean		Indicates whether or not this message is a drop copy of another message.	
802	NoPartySubIDs	NumInGroup		Number of PartySubID (523)and PartySubIDType (803) entries	
803	PartySubIDType	int		Type of PartySubID (523) value 4000+ = Reserved and available for bi-laterally agreed upon user defined values	**Value** / **Description** 1 Firm 5 Full legal name of firm 6 Postal address 7 Phone number 8 Email address 9 Contact name 16 BIC 18 Registered address 21 Fax number
827	ExpirationCycle	int		Part of trading cycle when an instrument expires. Field is applicable for derivatives.	**Value** / **Description** 0 Expire on trading session close (default) 1 Expire on trading session open 2 Expires at specified expiration as the eligibility
828	TrdType	int		Type of Trade:	**Value** / **Description** 0 Regular Trade 4 Late Trade

Tag	FieldName	Type	OMXLen	Desc	Valid values
					Value / **Description** 6 — Weighted Average Price Trade 38 — Block trade (same as large trade) 47 — Financing transaction (includes repo and stock lending) 48 — Non-Standard Settlement **OMX Comment:** FIX 5.0 SP1 49 — Derivative Related Transaction **OMX Comment:** FIX 5.0 SP1 50 — Portfolio Trade **OMX Comment:** FIX 5.0 SP1 51 — Volume Weighted Average Trade **OMX Comment:** FIX 5.0 SP1 52 — Exchange Granted Trade **OMX Comment:** FIX 5.0 SP1 53 — Repurchase Agreement **OMX Comment:** FIX 5.0 SP1 54 — OTC **OMX Comment:** FIX 5.0 SP1
845	DiscretionPrice	Price		The current discretionary price of the order	
852	PublishTrdIndicator	int		Indicates if a trade should be reported via a market reporting service. **OMX Comment:** Field is Boolean in FIX 5.0, OMX requests change	**Value** / **Description** 0 — Do Not Report Trade 1 — Report Trade 2 — Deferred Publication
855	SecondaryTrdType	int		Additional TrdType (see tag 828) assigned to a trade by trade match system.	**Value** / **Description** 98 — On-Hours

Tag	FieldName	Type	OMXLen	Desc	Valid values
					Value / **Description** **OMX Comment:** Not in FIX Standard 99 — Off-hours **OMX Comment:** Not in FIX Standard
856	TradeReport-Type	int		Type of Trade Report	**Value** / **Description** 0 — Submit 1 — Alleged 2 — Accept 3 — Decline 4 — Addendum 5 — No/Was 7 — (Locked-In) Trade Break 11 — Alleged New 12 — Alleged Addendum 14 — Alleged Trade Report Cancel 15 — Alleged (Locked-In) Trade Break
864	NoEvents	NumIn-Group		Number of repeating Event-Type entries.	
865	EventType	int		Code to represent the type of event	**Value** / **Description** 5 — Activation 6 — Inactiviation 93 — Excluding combined Split and Issue Rights 94 — Under Drawing 95 — Company subject to Public Offer 96 — Excluding Participat-ing in Split 97 — Excluding Participat-ing in Rights 98 — Excluding Dividend 99 — On the Surveillance List

Tag	FieldName	Type	OMXLen	Desc	Valid values
866	EventDate	LocalMktDate		Date of event	
870	NoInstrAttrib	NumInGroup		Number of repeating InstrAttribType entries.	
871	InstrAttribType	int		Code to represent the type of instrument attribute	**Value** / **Description** 1 – Flat (securities pay interest on a current basis but are traded without interest) 2 – Zero coupon 3 – Interest bearing (for Euro commercial paper when not issued at discount) 4 – No periodic payments 5 – Variable rate 6 – Less fee for put 7 – Stepped coupon 8 – Coupon period (if not semi-annual). Supply redemption date in the InstrAttribValue (872) field. 9 – When [and if] issued 10 – Original issue discount 11 – Callable, puttable 12 – Escrowed to Maturity 13 – Escrowed to redemption date - callable. Supply redemption date in the InstrAttribValue (872) field 14 – Pre-refunded 15 – In default 16 – Unrated 17 – Taxable 18 – Indexed 19 – Subject To Alternative Minimum Tax 20 – Original issue discount price. Supply price in the InstrAttribValue (872) field

Tag	FieldName	Type	OMXLen	Desc	Valid values
					Value / **Description** 21 – Callable below maturity value 22 – Callable without notice by mail to holder unless registered 23 – Price tick rules for security. Tick rule values to be expressed using InstrAttribValue [872] 24 – Trade type eligibility details for security. Trade types to be expressed using InstrAttribValue [872] 99 – Text. Supply the text of the attribute or disclaimer in the InstrAttribValue (872) field.
872	InstrAttribValue	String		Attribute value appropriate to the InstrAttribType (87) field.	
880	TrdMatchID	String		Identifier assigned to a trade by a matching system. **OMX Comment:** Identifies a group of fills matched in the same execution round and at the same price. Established for a single aggressive order but can include many contra orders.	
893	LastFragment	Boolean		Indicates whether this message is the last in a sequence of messages for those messages that support fragmentation, such as Allocation Instruction, Mass Quote, Security List, Derivative Security List	**Value** / **Description** N – Not Last Message Y – Last Message
939	TrdRptStatus	int		Trade Report Status	**Value** / **Description** 0 – Accepted 1 – Rejected
964	SecurityReportID	int		Security Report ID. Unique identifier for the Security Report.	
965	SecurityStatus	String		Used for derivatives. Denotes the current state of the Instrument.	**Value** / **Description** 1 – Active

Tag	FieldName	Type	OMXLen	Desc	Valid values
					Value: 2 — Description: Inactive
967	StrikeMultiplier	float		Used for derivatives. Multiplier applied to the strike price for the purpose of calculating the settlement value.	
968	StrikeValue	float		Used for derivatives. The number of shares/units for the financial instrument involved in the option trade.	
969	MinPriceIncrement	float		Minimum price increase for a given exchange-traded Instrument	
980	SecurityUpdateAction	char			Value: A — Description: Add Value: D — Description: Delete Value: M — Description: Modify
1003	TradeID	String		The unique ID assigned to the trade entity once it is received or matched by the exchange or central counterparty.	
1021	MDBookType	int		Describes the type of book for which the feed is intended. Used when multiple feeds are provided over the same connection	Value: 2 — Description: Price Depth Value: 3 — Description: Order Depth
1022	MDFeedType	String		Describes a class of service for a given data feed, ie Regular and Market Maker, Bandwidth Intensive or Bandwidth Conservative	
1023	MDPriceLevel	int		Integer to convey the level of a bid or offer at a given price level. This is in contrast to MDEntryPositionNo which is used to convey the position of an order within a Price level	
1057	AggressorIndicator	Boolean		Used to identify whether the order initiator is an aggressor or not in the trade.	Value: Y — Description: Order initiator is aggressor Value: N — Description: Order initiator is passive

Tag	FieldName	Type	OMXLen	Desc	Valid values
1070	MDQuote-Type	int		Identifies market data quote type.	**Value** / **Description** 0 Indicative 1 Tradeable
1079	MaturityTime	TZTime-Only		Time of security's maturity expressed in local time with offset to UTC specified	
1080	RefOrderID	String		The ID reference to the order being hit or taken	
1081	RefOrderID-Source	char		Used to specify what identifier, provided in order depth market data, to use when hitting (taking) a specific order.	**Value** / **Description** 0 SecondaryOrdeID (198)
1084	Dis-playMethod	char		Defines what value to use in DisplayQty (1138). If not specified the default DisplayMethod is "1"	**Value** / **Description** 1 Initial (use original DisplayQty) 2 New (use RefreshQty)
1088	RefreshQty	Qty		Defines the quantity used to refresh DisplayQty.	
1089	MatchIncre-ment	Qty		Allows orders to specify a minimum quantity that applies to every execution (one execution could be for multiple counter-orders). The order may still fill against smaller orders, but the cumulative quantity of the execution must be in multiples of the MatchIncrement.	
1091	Pre-TradeAnonymi-ty	Boolean		Allows trader to explicitly request anonymity or disclosure in pre-trade market data feeds. Anonymity is relevant in markets where counterparties are regularly disclosed in order depth feeds. Disclosure is relevant when counterparties are not normally visible.	
1093	LotType	char		Defines the lot type assigned to the order.	**Value** / **Description** 1 Odd Lot 2 Round Lot 3 Block Lot

Tag	FieldName	Type	OMXLen	Desc	Valid values
1115	OrderCategory	char		Defines the type of interest behind a trade (fill or partial fill).	**Value** / **Description** 1 Order 2 Quote 3 Privately Negotiated Trade 4 Multileg order 5 Linked order 6 Quote Request 7 Implied Order 8 Cross Order
1116	NoRootPartyIDs	NumInGroup		Number of RootPartyID (1117), RootPartyIDSource (1118), and RootPartyRole (1119) entries	
1117	RootPartyID	String		PartyID value within a root parties component. Same values as PartyID (448)	
1118	RootPartyIDSource	char		PartyIDSource value within a root parties component. Same values as PartyIDSource (447)	
1119	RootPartyRole	int		PartyRole value within a root parties component. Same values as PartyRole (452)	
1123	TradeHandlingInstr	char		Specified how the Trade Capture Report should be handled by the Respondent.	**Value** / **Description** 0 Trade Confirmation 2 One-Party Report for Matching
1126	OrigTradeID	String		Used to preserve original trade id when original trade is being referenced in a subsequent trade transaction such as a transfer	
1127	OrigSecondaryTradeID	String		Used to preserve original secondary trade id when original trade is being referenced in a subsequent trade transaction such as a transfer	
1128	ApplVerID	String		Specifies the service pack release being applied at message level. Enumerated field with values assigned at time of service pack release	**Value** / **Description** 7 FIX50
1137	DefaultApplVerID	String		Specifies the service pack release being applied, by de-	

Tag	FieldName	Type	OMXLen	Desc	Valid values
				fault, to message at the session level. Enumerated field with values assigned at time of service pack release. Uses same values as ApplVerID	
1138	DisplayQty	Qty		The quantity to be displayed . Required for reserve orders. On orders specifies the qty to be displayed, on execution reports the currently displayed quantity.	
1140	MaxTradeVol	Qty		The maximum order quantity that can be submitted for a security	
1141	NoMDFeedTypes	NumInGroup		The number of feed types and corresponding book depths associated with a security	
1142	MatchAlgorithm	String		The type of algorithm used to match orders in a specific security Possible values are FIFO, Allocation, Pro-rata, Lead Market Maker, Currency Calendar	
1143	MaxPriceVariation	Float		The maximum price variation of an execution from one event to the next for a given security	
1151	SecurityGroup	String		An exchange specific name assigned to a group of related securities which may be concurrently affected by market events and actions.	
1173	MDSubBookType	String		Describes a class of sub book, e.g. for the separation of various lot types. The Sub Book Type indicates that the following Market Data Entries belong to a non-integrated Sub Book. Whenever provided the Sub Book must be used together with MDPriceLevel and MDEntryPositionNo in order to sort the order properly. **OMX Comment:** FIX 5.0 SP1	
1174	SecurityTradingEvent	int		Identifies an event related to a SecurityTradingStatus (326). An event occurs and is gone, it is not a state that applies for a period of time. **OMX Comment:** FIX 5.0 SP1	**Value** / **Description** 1 — Order imbalance, auction is extended 2 — Price Volatility Interruption

Tag	FieldName	Type	OMXLen	Desc	Valid values
					Value / **Description** 3 – Trading resumes (after Halt) 4 – Change of Trading Session 5 – Change of Trading Subsession 6 – Change of Security Status 7 – Change of Book Type 8 – Change of Market Depth
1175	NoStatsIndicators	NumInGroup		Number of statistics indicator repeating group entries **OMX Comment:** FIX 5.0 SP1	
1176	StatsType	Int		Type of statistics the MDEntry is eligible to be included in **OMX Comment:** FIX 5.0 SP1	**Value** / **Description** 1 – Exchange Last 2 – High / Low Price 3 – Average Price (VWAP, TWAP...) 4 – Turnover (Px * Qty)
20004	MultilegPriceMethod	int		Defines the type of combination price the multileg uses	**Value** / **Description** 1 – Net Price 2 – Reversed Net Price 3 – Yield Difference 4 – Individual 5 – Weighted Average Price 6 – Multiplied Price
20016	MultilegModel	int		Specifies the type of multileg model the user is targeting.	**Value** / **Description** 0 – Predefined Multileg Security 2 – Strategy Order
20036	MarketSegmentID	String		Market Segment identifier value	
20037	MarketSegmentDesc	String		Description or name of Market Segment	

Tag	FieldName	Type	OMXLen	Desc	Valid values
20038	EncodedMktSegmDescLen	Length		Byte length of encoded (non-ASCII characters) EncodedSecurityDesc (351) field.	
20039	EncodedMktSegmDesc	data		Encoded (non-ASCII characters) representation of the MarketSegmDesc (tbd) field in the encoded format specified via the MessageEncoding (347) field. If used, the ASCII (English) representation should also be specified in the MarketSegmDesc field.	
20040	ParentMktSegmID	String		Reference to a parent Market Segment. See MarketSegmentID (tbd)	
20041	NoInstruments	NumINGroup		The number of Instrument entries.	
20042	NoLotTypes	NumInGroup		The number of Lot Type entries.	
20043	LotSize	Qty		Quantity Increment for trading	
20044	NoTickSizes	NumInGroup		The number of Tick Size entries.	
20045	MinPx	Price		A minimum price	
20046	PriceIncrement	Price		Price increment (tick size)	
20078	MktSegmSecurityExchange	String		Marketplace used to identify the Marekt Segment. Same values as SecurityExchange (207).	
20079	PartyReqID	String		Unique ID for a Participant request	
20080	PartyReportID	String		Unique message ID for a Participant list	
20081	PartyRequestResult	int		Return code for a Participant request	**Value** / **Description** 0 Valid request 1 Invalid or unsupported request 2 No data found that match selection criteria 3 Not authorized to retrieve data
20082	ToNoRelParties	int		Total number of participants	
20083	Suspended	Boolean		Indicates whether a suspension applies or not.	**Value** / **Description** 0 Not suspended

Tag	FieldName	Type	OMXLen	Desc	Valid values
					Value: 1 — Description: Suspended