# GENIUM® Protocol Specification

## GENIUM FIX for OMX Nordic Exchange - FIX Specification

Protocol version 2.0.2

| | |
|---|---|
| Document ID: | GENIUM-GFIX-PS-17_OMXE-2.0 |
| Documentation Release: | a5 |
| Release Date: | 2007-10-30 |
| Publication Date: | 2007-10-30 |
| Protocol version | 2.0.2 |

# 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]**
FPL Market Data Optimization Working Group (MDOWG) - Book Management Best Practices
http://www.fixprotocol.org/documents/1462/MDOWG_Book_Mgt_Finaldraft.doc

**[5]**
FAST Specification
http://www.fixprotocol.org/fast

**[6]**
FPL Global Exchanges and Markets Committee - Recommended Practices for Continuous Mass Quoting
http://www.fixprotocol.org/documents/3496/EEWG%20Quotation%20Best%20Practices%20V1_00.pdf

**[7]**
FIXimate for FIX 5.0 - An HTML tool to view FIX standard messages and fields
http://www.fixprotocol.org/specifications/fix5.0fiximate/index.html

**[8]**
OMXimate - FIXimate adapted to the messages and fields used in this specification. Provided separately.

**[9]**

FIX Repository - A set of xml files specifying the messages, component blocks, fields and enums of this specification. Provided separately.

## 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.

#### Multileg Orders

Describes multileg orders that are used to trade multileg instruments and user-defined strategies.

#### Contingent (Linked) Orders

Describes contingent orders, i.e. the submission of orders whose execution is conditional on each other.

#### Continuous and General Quote Handling

Describes quotes as typically used by participants to maintain a continuous presence in the market. Some aspects of those also apply to Quote Negotiations.

#### Quote Negotiation

Describes how a user can encourage another participant to quote, either in the public market or privately by responding to the initiator.

#### Indication of Interest (IOI)

Describes Indications of Interest and how such indicative interest can be used for a "bulletin board" market or an IOI-hub.

#### Request for One-Sided Auction

Describes how an eligible user can request a call auction. Used primarily for issuing or buy-back auctions.

#### 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 cancellation or amendment of confirmed trades.

#### Market Data

Describes how the marketplace publishes data for book views, trade tickers, trade statistics and other price indicators.

#### Trading Sessions and States

Describes how the marketplace publishes trading session and state changes for the market segments and securities.

#### News Management

Describes how the marketplace publishes company and marketplace news.

#### 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] on page 21 or on the FPL website [3] on page 21.

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 this 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 specific 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 FPL Global Technical Committee, the Global Exchanges and Markets Committee, the Global Derivatives Committee, 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 and messages pending inclusion in a FIX extension pack 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 and Refined Market 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.

**Note:**

Value range patterns (ReservedNNNPlus and Tenor) used in field descriptions of the Data Dictionary section (Section 20 on page 273) represents the FIX standard only. Users should be prepared for additions in the allowed enumerations, but messages are not allowed to use enumerations that are not explicitly listed.

### 2.3.4 Additional Tools

**FIXimate** is an online or downloadable tool where the FIX standard messages, component blocks and fields can be viewed in an internet browser. Refer to [7] on page 21.

**OMXimate** is separately provided to complement this specification and is an adapted FIXimate version. Refer to [8] on page 21.

The **FIX Repository** is a set of xml files including the messages, component blocks, fields and enums of this specification. Refer to [9] on page 21.

# 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.

**Note:**

The DefaultApplVerID (1137) of the Logon message is expected to be removed as an errata to FIX 5.0 / FIXT 1.1. Other significant changes are expected to be introduced in the not yet named coming FIXT version, e.g. extensions to the tag-equals-value representation:

- Allowing non-ASCII string values
- Much improved support for delimiters of repeating groups and their entries

This specification may be changed to support the coming FIXT version.

### 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 (49). The party initiating the session
- TargetCompID (56). The acceptor of the session as per configuration.

**Note:**

CompIDs are used to identify a FIX Session - not only, as the names indicates, to identify the firms involved. As there often will be multiple FIX sessions established between the marketplace and a participant, the CompIDs will be assigned per the necessary granularity.

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

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

## 3.4 Start of Day/End of Day Procedures

**Note:**

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

A FIX session can last forever, or until:

1. A login message that specifies that the sequence number series should be reset using the 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 normally expire when the session expires. Also, all history and re-send capabilities will be lost after a sequence number reset on the FIX session.

**Note:**

Subscription Request messages could also be used to define persistent subscriptions ("configured" subscriptions).

If a user wishes to finish his connection with a logout, this should be preceded by un-subscription 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.

Fields PossDupFlag (43) and PossResend (97) can indicate potentially duplicate information, either contained within a message with the same sequence number, or in a message with a new sequence number.

## 3.6 Non ASCII Text and Language Support

FIX currently has limited support for non-ASCII text. Fields prefixed by "Encoded", e.g. EncodedSecurityDesc (351) however allow extended character sets as defined by the MessageEncoding (347) field of the StandardHeader.

**Note:**

There is a development towards extending FIX to allow non-ASCII content for any String field in the coming version of FIX Transport (FIXT 2.0). OMX foresees that this version will be used in areas where plain ASCII is not an option.

## 3.7 Session Level Counterparty Identification Fields

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

- SenderCompID (49). This field is used to identify the sending (originating) party as per connection agreement with the market place.
- TargetCompID (56). This field is used to identify the party for which the message is targeted as per connection agreement with the market place.

The SenderSubID (50), TargetSubID (57), SenderLocationID (142) and TargetLocationID (143) can be used for other purposes in the future. Presently they have no relevance and are not echoed back if provided in a message.

### 3.7.1 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.

#### 3.7.1.1 Incoming messages

Inbound messages (to the marketplace) should identify the receiving process by specifying:

- SenderCompID (49) = The originator as per connection agreement
- TargetCompID (56) = The receiving FIX Gateway at the market place

#### 3.7.1.2 Outgoing messages

Outbound messages (from the marketplace) identifies the receiving process by specifying:

- SenderCompID (49) = The sending FIX Gateway at the market place
- TargetCompID (56) = The message originator (responses to requests) or the configured receiver of unsolicited messages.

### 3.7.2 On-behalf-of Messages

Parties connected to the marketplace, as in the case of access providers, may act as routers of requests to the marketplace. The result of those requests are routed back to the requestors. The on-behalf-of fields of the FIX Standard Header can be used to support such behavior although it is recommended that on-behalf-of routing is solved using mechanisms that do not add to message size. The on-behalf-of fields are treated as pass-through values by the marketplace and are not validated. In cases where the intermediary is subject to marketplace authorization - please refer to Section 4.2.1 on page 37.

#### 3.7.2.1 Incoming messages

The following fields can be specified:

- OnBehalfOfCompID (115)
- OnBehalfOfSubID (116)
- OnBehalfOfLocationID (144)

#### 3.7.2.2 Outgoing messages

Outbound messages reverse the OnBehalfOf and DeliverTo fields:

- DeliverToCompID (128) = the value of the inbound OnBehalfOfCompID (115)
- DeliverToSubID (129) = the value of the inbound OnBehalfOfSubID (116)
- DeliverToLocationID (145) = the value of the inbound OnBehalfOfLocationID (144)

## 3.8 Reject vs. BusinessMessageReject vs. Application Reject

The Reject message is 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:**

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 recipient. 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.

## 3.9 Drop Copy and other dedicated FIX Sessions

Participants connected to a marketplace have many reasons for wanting copies of messages sent to a variety of applications and thereby separate FIX sessions. Trade confirmations are a prime example:

- Orders are entered through the Order Management System (OMS) over FIX session X
- Execution Reports and Trade Capture Reports are thereby returned to FIX session X
- The back office system needs copies of Trade Capture Reports over FIX session Y

There are also cases where the participant can need separate FIX sessions based either on subscriptions or marketplace defined configurations for messages as:

- Receipt of unsolicited Quote Requests - refer to Section 9 on page 143.
- Receipt of counterparty-initiated privately negotiated trade reports (Trade Capture Reports) - refer to Section 12 on page 171.
- Market Data - refer to Section 14 on page 189.

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. Drop copy sessions depend on how the participant is organized, its system architecture and whether it uses service provides for certain business processes.

Also refer to Section 4.6 on page 45.

## 3.10 Message Fragmentation

*Currently supported for Reference Data messages only.*

FIX message fragmentation support permits, but does not require, a receiving application to react in a stateful manner where it can determine if it has received all entries for a certain message before carrying out some action. However, the overall approach to fragmentation is to permit each message to be processed in a stateless manner as it is received. Each mass message should contain enough information to have the entries applied without requiring the next message if fragmentation has occurred. Also, a continued message should not require any information from the previous message.

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

In fragmentation cases (and only then) the following two fields are used (and required):

- **TotNoNnn** (nnn). The field indicates the total number of Entries for the particular message.
- **LastFragment** (893). The field indicates whether the message contains the last entry of the relevant message or not.

## 3.11 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).
- Password (554). This is the marketplace session level password valid for the specified session and member.

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

### 3.11.1 Message Authorization

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

## 3.12 Message Details

### 3.12.1 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 | | |

### 3.12.2 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 |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 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 | | |

### 3.12.3 Logout

*Table 3: Logout*

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

### 3.12.4 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 | | |

### 3.12.5 ResendRequest

*Table 5: ResendRequest*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | **StandardHeader** | Y | MsgType = 2 | |
| 7 | BeginSeqNo | Y | | SeqNum |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 16 | EndSeqNo | Y | | SeqNum |
| | **StandardTrailer** | Y | | |

### 3.12.6 SequenceReset

*Table 6: SequenceReset*

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

### 3.12.7 TestRequest

*Table 7: TestRequest*

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

## 3.13 Component Blocks

### 3.13.1 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 |
| 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 |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 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 |

### 3.13.2 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 FIX 5.0 Protocol Compliance

The FIX Specification is fully compliant with the FIX 5.0 SP1 specification according to the standard documentation with the following exception(s):

- None.

The contents of FIX 5.0 SP1 is tentatively defined and the specification is expected to become publicly available in December 2007. It is expected to include six extension packs (EPs) required by OMX:

- OMX Continuous Quoting Req's
- OMX Quote Negotiation Req's
- OMX Market Data Req's
- OMX Order Routing Phase II Req's
- OMX Multileg Order Req's
- Contingent Orders

OMX is also actively working together with other FPL Committees and Working Groups. Extensions from those groups include:

- EEWG Phase I Req's
- Market Segmentation Req's
- GDC Product Reference Req's

(EEWG = Exchanges/ECN's Working Group, elevated to Global Exchanges & Markets Committee in October 2007)

(GDC = Global Derivatives Committee)

**Note:**

Certain aspects of messages, fields, etc can change during the FIX standard extension process. This process includes a public comment period and finally approval by the FPL Global Technical Committee. The final values of the MsgType (35) of new messages, the tag numbers of fields and the enum values are not assigned until after approval. Refer to Section 2.3.3 on page 24.

## 4.2 Identifier Definitions

### 4.2.1 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 main 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 information on additional 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 (see Section 3.7.2 on page 30) has nothing to do with the business layer one.

## 4.2.2 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 may not be 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 (48) + SecurityIDSource (22) = 99 (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). An alternative to using the CFICode is using the SecurityType (167), PutOrCall (201) and ExerciseStyle (20096) fields. 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).

The SecurityExchange (207) field can be used to qualify the SecurityID (48) or Symbol (55) fields in cases where those are not globally unique. In such cases the issuer of the identifier, normally the primary listing exchange, should be defined in the SecurityExchange (207) field. A GENIUM marketplace will always have unique SecurityIDs.

**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.

#### 4.2.2.1 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 (48) | 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 (22) | N | Required if SecurityID is specified. | Value = 99 (Marketplace assigned identifier) |

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

| Tag | Field Name | Req'd | Comments | OMX Comment |
|---|---|---|---|---|
| Unique Identifier: **Symbol** | | | | |
| 55 | Symbol (55) | 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 (65) | 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” | Suffix for the symbol, e.g. used for different share classes (A- and B-shares, preferential shares, “when issued”, |

| Tag | Field Name | Req'd | Comments | OMX Comment |
|---|---|---|---|---|
| | | | to indicate a EUCP with lump-sum interest rather than discount price. | 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 (48) | N | Takes precedence in identifying security to counterparty over SecurityAltID block. Requires SecurityIDSource if specified. | ISIN code |
| 22 | SecurityIDSource (22) | 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 (55) | 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 (65) | 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 (461) or<br>SecurityType (167) + PutOrCall (201) + ExerciseStyle (20096) | N | Indicates the type of security using ISO 10962 standard, Classification of Financial Instruments (CFI code) values. | Part of alternative identifier for derivatives (options vs futures).<br>First CFICode position = "O" |
| 200 | MaturityMonthYear (200) | 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 (202) | N | Used for derivatives, such as options and covered warrants | Part of alternative identifier for derivatives. |
| Unique Identifier: **Options characteristic (Expiry Date)** | | | | |

| Tag | Field Name | Req'd | Comments | OMX Comment |
|---|---|---|---|---|
| 55 | Symbol (55) | 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 (65) | 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 (461) or<br>SecurityType (167) + PutOrCall (201) + ExerciseStyle (20096) | N | Indicates the type of security using ISO 10962 standard, Classification of Financial Instruments (CFI code) values. | Part of alternative identifier for derivatives (options vs futures).<br>First CFICode position = “O” |
| 541 | MaturityDate (541) | 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 (202) | N | Used for derivatives, such as options and covered warrants | Part of alternative identifier for derivatives. |

## 4.3 System Partitioning

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

### 4.3.1 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.

## 4.4 Trading in various Security Types

The marketplace supports quoting and 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:**

  A security is always associated with a default price type as specified 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.

## 4.5 Accounts, Pre-Allocation and Give Ups

A broker may have an agreement with separate entities for clearing. Such clearing firms can have separate connections to the marketplace and be eligible to receive trade confirmations for trades they are clearing. Standard clearing arrangements can be supported by configuration, but there may also be cases when the user needs to provide the clearing firm with the order or trade transaction. 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. Pre-allocation entries are passed to downstream actors based on the following rules:

- CSDs
  - NestedPartyIDRole (538) = "10" - Settlement Location
  - NestedPartyIDRole (538) = "4" - Clearing Firm
  - NestedPartyIDRole (538) = "14" - Give Up Clearing Firm (which is really the take up firm)
- Clearing Houses
  - NestedPartyIDRole (538) = "21" - Clearing Organization
  - NestedPartyIDRole (538) = "4" - Clearing Firm
  - NestedPartyIDRole (538) = "14" - Give Up Clearing Firm (which is really the take up firm)
- Counterparties to the trade
  - All entries

Default accounts and clearing firms can be kept in the records of the marketplace or relevant clearing / depository bodies, meaning those actors can automatically complement trades with that information. Although a firm may have those default accounts specified, sometimes the user needs to assign a specific account to an order or give the trade up.

**Note:**

In FIX, broker omnibus accounts are specified using the following fields:

- **Account** (1)
- **AccountIDSource** (660)

In this specification, those fields are not used as the marketplace and/or clearing systems support default accounts for clearing members.

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

- AllocID (70). Used to specify an account 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 (i.e. 100% of the quantity), 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 supported. Relevant when splitting the trade into different accounts. Needed to access a particular allocation in downstream systems, e.g. a client order reference.
  - NestedParties. Specifies the parties to the allocation, whenever relevant. Further discussed below.

**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 can also be used:

- NestedPartyID (524). Necessary when the AllocAccount (79) must be qualified by the identifier of the firm that manages the account (e.g. the Clearing House or CSD) - i.e. when multiple accounting parties are involved in the transation chain (e.g. both a broker back office and a clearing entity)
- NestedPartyIDSource (525). Used when the NestedPartyID is specified and specifies the type of identifier used.
  - C = Generally accepted market participant identifier or
  - D = Proprietary/Custom code
  - H = CSD or Clearing House participant/member identifier or code. Not validated by marketplace!
- NestedPartyRole (538) = Used when the NestedPartyID (524) is used and specifies the role of that party.

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

**Note:**

Execution Reports echo the values back, but do not report the state of allocations. If the pre-allocation instructions of an order is modified, that change is only applies to fills done after the modification, not to any previously done ones.

Allocation specific FIX messages are used to report the state, the allocation process is however outside the scope of this specification.

Examples:

*Table 15: Vanilla Allocation to Accounts*

| Case | Alloc Account (79) | Alloc Acct ID Source (661) | Nested Party ID (524) | Nested Party ID Source (525) | Nested Party Role (538) |
|---|---|---|---|---|---|
| Assigning the trade to an internal account (at the back office) of the executing firm | Account at the broker | 99" - Other | N/A | N/A | N/A |
| Assigning the trade to a CSD (Central Securities Depository) account | Account at the CSD | "99" - Other | "NCSD" - National CSD (or an applicable ID for a non-national CSD) | "C" | "10" - Settlement Location |
| Assigning the trade to an account at the CCP (Central Counter Party / Clearing House) | Account at the Clearing House | "99" - Other | The ID of the Clearing House | "C" or "D" | "21" - Clearing Organization |

The following table shows examples of where a trade is given up to a clearing firm.

*Table 16: Give Ups*

| Case | Alloc Account (79) | Alloc Acct ID Source (661) | Nested Party ID (524) | Nested Party ID Source (525) | Nested Party Role (538) |
|---|---|---|---|---|---|
| Specifying the clearing firm used | Account at the Clearing Firm | "99" - Other | The ID of the Clearing Firm | "C", "D" or "H" | "4" - Clearing Firm |
| Giving up the trade to another clearing firm | Account at the take-up Clearing Firm | "99" - Other | The ID of the take-up Clearing Firm | "C", "D" or "H" | "14" - Give Up Clearing Firm |

In cases where the broker needs to identify its clearing firm **and** give the trade up to another clearing firm, two Pre-Allocation entries must be specified (i.e. both rows in the above table).

Clearing houses (and other actors) may offer the ability to identify the default account without it being specified. This is exemplied in the below table:

*Table 17: Allocation without using an Account*

| Case | Alloc Account (79) | Alloc Acct ID Source (661) | Nested Party ID (524) | Nested Party ID Source (525) | Nested Party Role (538) |
|---|---|---|---|---|---|
| Allocation and Give up to default accounts | "[N/A]" | N/A | as in above examples | as in above examples | as in above examples |

## 4.6 Queries, Subscriptions and Drop Copies

Participants connected to a marketplace have many reasons for wanting copies of messages sent to a variety of applications and thereby separate FIX sessions. Reasons include failover procedures but also how the participant is organized and whether it uses service provides for certain business processes.

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)

Subscription Request messages could be used to establish **persistent subscriptions**, i.e. suscriptions that need not be reestablished after a logout. *Such mechanisms are however currently not supported*.

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 out 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

**Note:**

- Drop copies and subscriptions / queries are based on the messages sent out as the result of a process and may not contain every individual piece of information part of the request that initiated the process.
- Drop copies / subscriptions / queries never return rejects for transactions.
- Certain messages do not have "ack" messages, i.e. there is no response message that can be copied (the IOI, Section 10 on page 159, is an example of this).

**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:

- Confirmed 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).

Also refer to Section 3.9 on page 31.

## 4.7 The Text (58) Field and Error / Warning Messages

Many inbound messages 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 e.g. be used to provide additional error-/warning text in the case a message is rejected.

Error codes not representable using standard FIX enumerations will be returned specifying "Broker / Exchange Option" (e.g. OrdRejReason [103] =" 0" in Execution Reports) 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.

## 4.8 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 Section 18 on page 259.

*Table 18: Business Message Reject*

| Reject Message | Direc-tion | Business Message | Comment |
|---|---|---|---|
| **Business Message Reject** | Out | Any | Primarily used when there is no other FIX reject message. Also used:<br>• 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).<br>• 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 |

| Reject Message | Direc-tion | Business Message | Comment |
|---|---|---|---|
| | | | tries to update another user's or-der). |

## 4.9 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 Section 18 on page 259.

*Table 19: Don't Know Trade*

| Reject Message | Direc-tion | Business Message | Comment |
|---|---|---|---|
| **Don't Know Trade DK** | In | Execution Report | In practice only used when a user does not recognize a trade (Execution Re-port). The marketplace does not accept inbound 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 table. Full details of the messages and workflows around them are available in other parts of the document.

*Table 20: 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** | |
| In | **Order Mass Suspend Or Release Request** | Used to suspend or release orders en masse.<br>Currently not in FIX, the Global Exchanges and Markets Committee proposes an extension |
| Out | **Execution Report** | |
| Out | **Order Cancel Reject** | |
| Out | **Order Mass Suspend Or Release Report** | Used to reject or acknowledge an Order Mass Suspend Or Release Request.<br>Currently not in FIX, the Global Exchanges and Markets Committee proposes an extension |
| 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 are 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 are primarily intended for downstream processing. The Trade Capture Report is used to report a trade to a variety of parties that have a role in the transaction chain of trades, e.g.: broker back-office; 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 Section 14 on page 189.

#### 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.

Orders can also be suspended or released from suspension en masse using the Order Mass Suspend Or Release Request message. The Order Mass Suspend Or Release Report is used to acknowledge or reject the request and individual Execution Reports relay the new state of the orders.

#### 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 21: Order Reject Messages*

| Reject Message | Direc-tion | Business Message | Comment |
|---|---|---|---|
| **Execution Report** | Out | New Order Single | |
| **Order Cancel Reject** | Out | Order Cancel Replace Request<br>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. In those cases the SecondaryClOrdID (526) can be used to represent the ClOrdID used when communicating with the marketplace. Note however that this field is not supported by this specification.

**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 re-ranked. 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 22: 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 Section 5.2.2.7 on page 55. |

### 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 23: 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 | |
| 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

Unsolicited cancellation of orders, subject to marketplace rules, can also occur - 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 24: 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. |

| Value | Name | Description |
|---|---|---|
| 2 | Limit | The Price (44) field is specified and the order will execute at this price or better. |

*Table 25: 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.<br>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 26: Supported ExecInst (18) values*

| Value | Name | Description |
|---|---|---|
| G | All or None - AON | Means the full order quantity must be filled in one execution round.<br>**Note:**<br>Whenever using this value to specify an All or None order the Minimum 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 Section 5.5.13 on page 81.

### 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 Section 5.5.14 on page 82.

### 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 un-priced 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 Section 4.5 on page 42.

#### 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 Section 4.7 on page 46.

### 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 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 explicit or implicit time in force conditions.

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

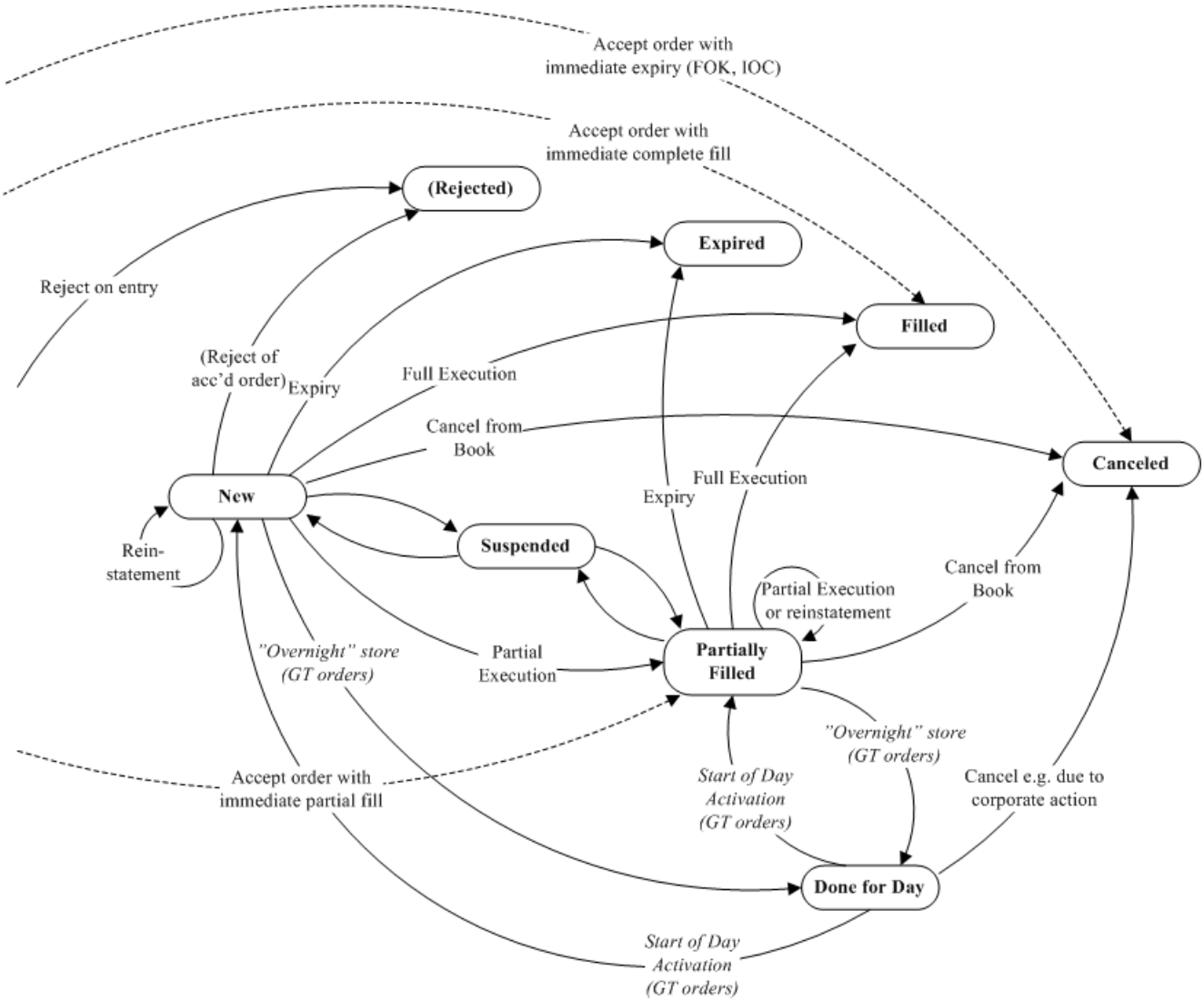


*Figure 1: Order Status State Event Chart*

**Note:**

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] on page 21.

### 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 Section 5.5.12 on page 81). Cancellation is done using the ordinary Cancel messages.

## 5.3 Message Details

For messages that are not specific for Orders, please see chapter Section 18 on page 259.

### 5.3.1 NewOrderSingle

*Table 27: 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" | |
| 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" | |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | | | **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. | UTCTimes-tamp |
| | **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. | UTCTimes-tamp |
| 528 | OrderCapacity | N | | char |
| 529 | OrderRestrictions | N | | MultipleCharVal-ue |
| 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"<br>**OMX Comment:** Use:<br>- DiscretionOffsetValue (389) when submitting a "discretionary price"<br>- 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.<br>**OMX Comment:** Valid values (also default if unspecified for Hit/Take order, i.e. if RefOrderID is specified):<br>- 0 = SecondaryOrderID | char |
| | **StandardTrailer** | Y | | |

## 5.3.2 ExecutionReport

*Table 28: 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.<br>**OMX Comment:** Static id assigned by trading engine to new orders, otherwise from order. Not publicly disclosed. | String |
| 198 | SecondaryOrderID | N | Can be used to provide order id used by exchange or executing system.<br>**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:<br>- QuoteID (117) of a single Quote<br>- QuoteEntryID (299) of a Mass Quote<br>**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:<br>- QuoteMsgID (1166) of a single Quote<br>- QuoteID (117) of a Mass Quote<br>**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.<br>**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" | |
| 229 | TradeOriginationDate | N | | LocalMktDate |
| 66 | ListID | N | Required for executions against orders which were submitted as part of a list.<br>**OMX Comment:** Identifies a set of Contingent Orders | String |
| 17 | ExecID | Y | Unique identifier of execution message as assigned by sell-side (broker, exchange, ECN) (will be 0 (zero) forExecType=I (Order Status)). | String |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | | | **OMX Comment:** Assigned by marketplace. Links Execution Report (ExecType = F [Trade]) to Trade Capture Report (TradeHandlingInstr = 0 [Trade Confirm]) | |
| 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.<br>**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.)<br>**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<br>**OMX Comment:** Assigned by marketplace | char |
| 636 | WorkingIndicator | N | For optional use with OrdStatus = 0 (New)<br>**OMX Comment:** Assigned by marketplace | Boolean |
| 103 | OrdRejReason | N | For optional use with ExecType = 8 (Rejected)<br>**OMX Comment:** Assigned by marketplace | int |
| 378 | ExecRestatementReason | N | Required for ExecType = D (Restated).<br>**OMX Comment:** Assigned by marketplace | 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"<br>**OMX Comment:** From order, but marketplace responds with SecurityIDSource = 99, 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"<br>**IMPORTANT NOTE: OrderQty field is required for Single Instrument Orders unless rejecting or acknowledging an order for a CashOrderQty or PercentOrder. **<br>**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 |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 44 | Price | N | Required if specified on the order<br>**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"<br>**OMX Comment:** From order | |
| 845 | DiscretionPrice | N | The current discretionary price of the order<br>**OMX Comment:** Marketplace assigned | Price |
| 59 | TimeInForce | N | Absence of this field indicates Day order<br>**OMX Comment:** From order<br>Marketplace assigns default (0) if left out on order | char |
| 432 | ExpireDate | N | Conditionally required if TimeInForce = GTD and ExpireTime is not specified.<br>**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.<br>**OMX Comment:** From order<br>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.<br>If ExecType=Stopped, represents the quantity stopped/guaranteed/protected for.<br>**OMX Comment:** Assigned by marketplace | Qty |
| 31 | LastPx | N | Price of this (last) fill. Required if ExecType = Trade or Trade Correct.<br>Should represent the "all-in" (LastSpotRate + LastForwardPoints) rate for F/X orders. ).<br>If ExecType=Stopped, represents the price stopped/guaranteed/protected at.<br>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.<br>**OMX Comment:** Assigned by marketplace | Price |
| 336 | TradingSessionID | N | **OMX Comment:** Assigned by marketplace (Not not from order).<br>Specifies the trading session when the Execution Report was produced. Trading Sessions from the Order are not relayed in the Execution Report! | String |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 625 | TradingSessionSubID | N | **OMX Comment:** Assigned by marketplace (Not not from order).<br>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 OrdStatus is Canceled, DoneForTheDay, Expired, Calculated, or Rejected (in which case the order is no longer active) then LeavesQty could be 0, otherwise LeavesQty = OrderQty - CumQty.<br>**OMX Comment:** Assigned by marketplace | Qty |
| 14 | CumQty | Y | Currently executed quantity for chain of orders.<br>**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<br>**OMX Comment:** Assigned by marketplace.<br>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"<br>**OMX Comment:** From order | |
| 58 | Text | N | **OMX Comment:** Assigned by marketplace (Note not from order).<br>May contain message from the marketplace | String |
| | **InstrmtLegExecGrp** | N | Number of legs<br>Identifies a Multi-leg Execution if present and non-zero.<br>**OMX Comment:** Only used to report the status of Multileg Orders | |
| 20228 | Volatility | N | **OMX Comment:** Not in FIX. OMX request addition | float |
| 20230 | DividendYield | N | **OMX Comment:** Not in FIX. OMX request addition | float |
| 20232 | RiskfreeRate | N | **OMX Comment:** Not in FIX. OMX request addition | float |
| | **StandardTrailer** | Y | | |

## 5.3.3 OrderCancelReplaceRequest

*Table 29: 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.<br>**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 Cancel Reject message if the replacement request is rejected. | String |
| 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"<br>Must match original order<br>**OMX Comment:** Messages shall use identifier fields only.<br>Instrument / Order book data cannot be modified | |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 54 | Side | Y | Should match original order's side, however, if bilaterally agreed to the following groups could potentially be interchanged:<br>Buy and Buy Minus<br>Sell, Sell Plus, Sell Short, and Sell Short Exempt<br>Cross, Cross Short, and Cross Short Exempt<br>**OMX Comment:** Side cannot be updated | char |
| 60 | TransactTime | Y | Time this order request was initiated/released by the trader or trading system. | UTCTimes-tamp |
| | **OrderQtyData** | Y | Insert here the set of "OrderQtyData" fields defined in "Common Components of Application Messages"<br>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 |
| | **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 30: 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.<br>**OMX Comment:** Required when referring to orders that where electronically submitted over FIX | String |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | | | or otherwise including a ClOrdID. The Exchanges/ECNs Working Group proposes the field is made optional as a reference to the OrderIID is sufficient. | |
| 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"<br>**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. | UTCTimestamp |
| | **OrderQtyData** | Y | Insert here the set of "OrderQtyData" fields defined in "Common Components of Application Messages"<br>Note: OrderQty = CumQty + LeavesQty (see exceptions above)<br>**OMX Comment:** Field is mandatory in standard FIX, but will not be used by OMX | |
| | **StandardTrailer** | Y | | |

## 5.3.5 OrderCancelReject

*Table 31: 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.<br>**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 |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 39 | OrdStatus | Y | OrdStatus value after this cancel reject is applied. If CxlRejReason = "Unknown Order", specify Rejected. | char |
| 60 | TransactTime | N | | UTCTimes-tamp |
| 434 | CxlRejResponseTo | Y | **OMX Comment:** Valid values:1 = Order Cancel Request<br>2 = Order Cancel/Replace Request | char |
| 102 | CxlRejReason | N | **OMX Comment:** Valid values:<br>0 = Too late to cancel<br>1 = Unknown order<br>2 = Broker / Exchange Option<br>4 = Unable to process Order Mass Cancel Request<br>5 = OrigOrdModTime (586) did not match last TransactTime (60) of order<br>6 = Duplicate ClOrdID (11) received<br>99 = Other | int |
| 58 | Text | N | **OMX Comment:** Contains a specified error number and message acording to separate specification | String |
| | **StandardTrailer** | Y | | |

## 5.3.6 OrderMassSuspendOrReleaseRequest

*Table 32: OrderMassSuspendOrReleaseRequest*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | **StandardHeader** | Y | MsgType = UF | |
| 11 | ClOrdID | Y | Unique ID of Order Mass Activate Request as assigned by the institution. | String |
| 526 | SecondaryClOrdID | N | | String |
| 20239 | MassActionType | Y | Specifies the type of activation requested | int |
| 20240 | MassActionRequestScope | Y | Specifies the scope of the action | int |
| 20084 | MarketID | N | MarketD for which orders are to suspended or released | Exchange |
| 20036 | MarketSegmentID | N | MarketSegmentD for which orders are to suspended or released | String |
| 336 | TradingSessionID | N | Trading Session in which orders are to be suspended or relased | String |
| 625 | TradingSessionSubID | N | | String |
| | **Parties** | N | Insert here the set of "Parties" (firm identification) fields defined in "common components of application messages" | |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | **Instrument** | N | Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages" | |
| | **UnderlyingInstrument** | N | Insert here the set of "UnderlyingInstrument" (underlying symbology) fields defined in "Common Components of Application Messages" | |
| 54 | Side | N | | char |
| 60 | TransactTime | N | | UTCTimestamp |
| 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 | | |

## 5.3.7 OrderMassSuspendOrReleaseReport

*Table 33: OrderMassSuspendOrReleaseReport*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | **StandardHeader** | Y | MsgType = UG | |
| 11 | ClOrdID | N | ClOrdID provided on the Order Mass Activation Request. | String |
| 526 | SecondaryClOrdID | N | | String |
| 20243 | MassSuspRelReportID | Y | | String |
| 20239 | MassActionType | Y | Order Mass Activation Request Type accepted by the system | int |
| 20240 | MassActionRequestScope | Y | Specifies the scope of the action | int |
| 20241 | MassActionResponse | Y | Indicates the action taken by the counterparty order handling system as a result of the Activation Request<br>0 - Indicates Order Mass Activation Request was rejected. | int |
| 20242 | MassActionRejectReason | N | Indicates why Order Mass Activation Request was rejected Required if MassActivationResponse = 0 | int |
| 533 | TotalAffectedOrders | N | Optional field used to indicate the total number of orders affected by the Order Mass Activation Request | int |
| | **SuspRelAffectedOrdGrp** | N | | |
| 20084 | MarketID | N | MarketD for which orders are to suspended or released | Exchange |
| 20036 | MarketSegmentID | N | MarketSegmentD for which orders are to suspended or released | String |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 336 | TradingSessionID | N | Trading Session in which orders are to be suspended or released | String |
| 625 | TradingSessionSubID | N | | String |
| | **Parties** | N | Insert here the set of "Parties" (firm identification) fields defined in "common components of application messages" | |
| | **Instrument** | N | Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages" | |
| | **UnderlyingInstrument** | N | Insert here the set of "UnderlyingInstrument" (underlying symbology) fields defined in "Common Components of Application Messages" | |
| 54 | Side | N | Side of the market specified on the Order Mass Activation Request | char |
| 60 | TransactTime | N | Time this report was initiated/released by the sells-side (broker, exchange, ECN) or sell-side executing system | UTCTimestamp |
| 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 | | |

# 5.4 Component Blocks (Order Specific)

For components that are not specific for Orders, please see chapter Section 19 on page 261.

## 5.4.1 Components

### 5.4.1.1 DiscretionInstructions

*Table 34: DiscretionInstructions*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 388 | DiscretionInst | N | What the discretionary price is related to (e.g. primary price, display price etc)<br>**OMX Comment:** Valid values:<br>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<br>Unspecified - if defining a discretionary offset value | char |

#### 5.4.1.2 DisplayInstruction

*Table 35: DisplayInstruction*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 1138 | DisplayQty | N | | Qty |
| 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 36: 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 37: PreAllocGrp*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 78 | NoAllocs | N | Number of repeating groups for pre-trade allocation<br>**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"<br>Used for NestedPartyRole=Clearing Firm | |

#### 5.4.2.2 SuspRelAffectedOrdGrp

*Table 38: SuspRelAffectedOrdGrp*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 534 | NoAffectedOrders | N | Optional field used to indicate the number of order identifiers for orders affected by the Order | NumInGroup |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | | | Mass Activation Request. Must be followed with OrigClOrdID as the next field | |
| >41 | OrigClOrdID | N | Required if NoAffectedOrders > 0<br>Indicates the client order id of an order affected by the Order Mass Activation Request. Required when referring to orders that where electronically submitted over FIX or otherwise including a ClOrdID. “[N/A]” used in cases where there is no ClOrdID to refer to. | String |
| >535 | AffectedOrderID | N | Contains the OrderID assigned by the counterparty of an affected order. Not required as part of the repeating group. | String |
| >536 | AffectedSecondaryOrderID | N | Contains the SecondaryOrderID assigned by the counterparty of an affected order. Not required as part of the repeating group | String |

# 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 Global Exchanges and Markets Committee. 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 39: 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 |

| 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 | 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 40: 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 41: 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 |
| 4 | | Execution (X) | A | Trade | Filled | 10000 | 10000 | 0 | 7000 | Execution for remaining 7000 |

## 5.5.3 B - Order Cancellation

*Table 42: 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 43: 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 44: 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 | | | | | | | |
| 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.*<br>*CxlRejectReason = 1 (unknown order)* |

*Table 45: 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 46: 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 47: 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 | |
| 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 48: 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* |

| Time | Message Received (ClOrdID, OrigClOrdID) | Message Sent (ClOrdID, OrigClOrdID) | Exchange (OrderID) | Exec Type | Ord Status | Order Qty | Cum Qty | Leaves Qty | Last Qty | Comment |
|---|---|---|---|---|---|---|---|---|---|---|
| 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 49: 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*<br>*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 50: 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* |

| Time | Message Received (ClOrdID, OrigClOrdID) | Message Sent (ClOrdID, OrigClOrdID) | Exchange (OrderID) | Exec Type | Ord Status | Order Qty | Cum Qty | Leaves Qty | Last Qty | Comment |
|---|---|---|---|---|---|---|---|---|---|---|
| 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 51: 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 52: 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 53: 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 54: 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 55: 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 56: 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 | | | | |
| *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<br>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<br>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 57: 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 58: 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 59: 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) |
| *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 60: 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 61: 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 62: 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 63: 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 64: 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.<br>ExecInst = "S" (Suspend)<br>ExecRestatementReason = "Suspended due to connection loss" |
| 5 | | | | | | | | | | Connection is re-esttablished<br>User must update order to remove ExecInst = "S" or replace it with ExecInst = "v" 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 65: 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 66: 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 |

| 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 | 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 67: 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 68: 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 |
| 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 69: Hit / Take Order is entered, both Orders are Filled*

| Trading Party | Direc-tion | Market Place | Direc-tion | Market Data |
|---|---|---|---|---|
| | | Order Depth Market Data is published for book X | -> | OrdID / Price / Qty / Party<br>3 / 10.00 / 100 / ABC<br>1 / 9.90 / 200 / DEF |

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

*Table 70: Hit / Take Order is entered, it is Filled but Referred Orders is Partially Filled*

| Trading Party | Direction | Market Place | Direction | Market Data |
|---|---|---|---|---|
| | | Order Depth Market Data is published for book X | -> | OrdID \| Price \| Qty \| Party<br>3 \| 10.00 \| 100 \| ABC<br>1 \| 9.90 \| 200 \| DEF<br>2 \| 9.90 \| 100 \| ABC<br>4 \| 9.80 \| 400 \| GHI |
| Party JKL enters New Order Single to hit DEF's order (OrdType = 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)<br>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<br>3 \| 10.00 \| 100 \| ABC<br>1 \| 9.90 \| 100 \| DEF<br>2 \| 9.90 \| 100 \| ABC<br>4 \| 9.80 \| 400 \| GHI |

*Table 71: 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<br>3 \| 10.00 \| 100 \| ABC<br>1 \| 9.90 \| 200 \| DEF |

| | | | | |
|---|---|---|---|---|
| | | | | OrdID \| Price \| Qty \| Party<br>2 \| 9.90 \| 100 \| ABC<br>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<br>3 \| 10.00 \| 100 \| ABC<br>1 \| 9.90 \| 100 \| DEF<br>2 \| 9.90 \| 100 \| ABC<br>4 \| 9.80 \| 400 \| GHI |
| Party JKL receives Execution Report (ExecType = Trade, OrdStatus = Partially Filled, OrdType = Counter-Order Selection, RefOrdID = 1)<br>Party DEF receives Execution Report (ExecType = Trade, OrdStatus = Filled)<br>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<br>Residue quantity on JKL's order is canceled (Hit / Take can not sit on book), Execution Report is produced. | -> | OrdID \| Price \| Qty \| Party<br>3 \| 10.00 \| 100 \| ABC<br>2 \| 9.90 \| 100 \| ABC<br>4 \| 9.80 \| 400 \| GHI |

### 5.5.14.2 Non-Executed Hit/Take Orders

*Table 72: Hit / Take Order is Entered, Referred Order is Removed from Book when it Arrives*

| Trading Party | Direction | Market Place | Direction | Market Data |
|---|---|---|---|---|
| | | Order Depth Market Data is published for book X | -> | OrdID \| Price \| Qty \| Party<br>3 \| 10.00 \| 100 \| ABC<br>1 \| 9.90 \| 200 \| DEF<br>2 \| 9.90 \| 100 \| ABC<br>4 \| 9.80 \| 400 \| GHI |
| Party DEF enter Order Cancel Request for Ord.id = 1 | -> | Receives Cancel Request | | |
| Party JKL enters New Order Single to hit DEF's order (OrdType = | -> | | | |

<table>
<tr><th>Trading Party</th><th>Direction</th><th>Market Place</th><th>Direction</th><th>Market Data</th></tr>
<tr><td>Counter-Order Selection, RefOrdID = 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 Request 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 Report (OrdType = Counter-Order Selection, RefOrdID = 1, OrdRejReason = 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 73: Hit / Take Order is Entered, Referred Order does not Match when it Arrives*

<table>
<tr><th>Trading Party</th><th>Direction</th><th>Market Place</th><th>Direction</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 Replace Request</td><td></td><td></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 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 (<br>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

## 6.1 Business Message Types

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

*Table 74: Business Messages*

| In/Out | Message Name | Comment |
|---|---|---|
| In | **New Order Multileg** | Used to submit a multileg order to a market place |
| In | **Multileg Order Cancel Replace** | Used to modify a multileg order. Whether the order is re-ranked or not is subject to marketplace rules. |
| In | **Order Cancel Request** | Used to request cancellation of the complete multileg order |
| Out | **Execution Report** | |
| Out | **Order Cancel Reject** | Used to reject cancellation of the complete multileg order |

## 6.2 Multileg Order Routing

### 6.2.1 Introduction

A multileg security is made up of multiple securities that are traded atomically. Swaps, option strategies and futures spreads are a few examples of multileg securities. The requirement that all legs be traded in the quantities that make up the multlileg security is the important distinction between a multileg order and a list order.

Two generalized approaches to trading multileg securities are supported by FIX. The first approach involves a market maintaining multileg securities as separate products for which markets can be created. This “product approach” is often used in electronic trading systems. The second approach is to trade the multileg security as a group of separate securities.

The multileg order can be traded using one of the following FIX trading models. The first two models are variations on the multileg security as a separate tradeable product. The last models permits trading of multileg securities in environments where the multileg securities are not productized.

#### Pre-defined Multileg Security Model

A.k.a. Standardized Combinations. Marketplace-defined multileg securities made available for trading.

#### User-defined Multileg Security Model

A.k.a. Tailor-made Combinations. User-defined multileg securities made available for trading.

#### Strategy orders

Multileg orders for combinations of security where a product is not defined or made available for others to trade.

**Note:**

OMX does currently not support the User-defined Multileg Security Model

**Note:**

The multileg models defined in the FIX standard specification are not consistent with the definitions of this specification.

**Note:**

So called "skewed" multileg orders are not supported. A skewed multileg occurs if the quantity of the legs differ when calculated as a number of round lots. I.e. all legs of a multileg order must have the same size counted as a number of round lots.

## 6.2.2 Main Workflow

### 6.2.2.1 Introduction

Multileg order work just as single orders, i.e. they:

- Have the same types of trading instructions, although the set of possibilities is limited.
- Use the same response messages, e.g. Execution Reports
- Are canceled using the Order Cancel Request or Order Mass Cancel Request messages
- Share the same type of workflows as New Order Single and Order Cancel Replace Request (although the inbound messages are different)

Please see Section 5 on page 49 for information on aspects shared with single order messages.

Multileg orders are however associated with a number of restrictions:

- Cannot have Triggering instructions or other more advanced order conditions.

### 6.2.2.2 New Order

The multileg order workflow starts with the user submitting a New Order Multileg message. In response at least one Execution Report is produced.

### 6.2.2.3 Order Modification

Order modification is accomplished using the Multileg Order Cancel Replace message. As applicable for other Cancel Replace messages, the message is used to modify an existing order and does not support delta updates (all relevant fields must be supplied).

In response at least one Execution Report is produced.

#### 6.2.2.4 Fills

When multileg orders are filled, Execution Reports are issued.

## 6.2.3 Multileg Order Features

### 6.2.3.1 Identifiers

#### 6.2.3.1.1 Order Identifiers

All Execution Reports representing a multileg order will share the same **OrderID** (37) and **ClOrdID** (11) values, those fields are echoed from the multileg order messages.

In cases where the status of each leg is reported, the Execution Report for a leg will have a **SecondaryOrderID** (198) with the following characteristics:

- When the multileg is added or updated by a user action (New Order Multileg or Multileg Order Cancel Replace) the SecondaryOrderID of all legs have the same value as the SecondaryOrderID of the multileg itself.
- When the leg is updated due to a recalculation of the implied price, a fill or any other reason leading to re-ranking of the implied price - the SecondaryOrderID of that leg is reassigned.

#### 6.2.3.1.2 Instruments

The instrument for which the order is aimed is defined as in other messages with one exception. When entering a Strategy Order, as no pre-defined product is there, the Symbol (55) field must be set to "[N/A]" (without the quote marks) and no SecurityID (48) be provided at the main level of the message. Other Instrument properties are not relevant either (even if FIX will allow the SecurityType (167) = "MLEG") at this level.

When leg parameters are provided with a multileg order, the identifier for the instrument of the leg is provided in the same way as usual (please refer to Section 4.2.2 on page 38), but using fields available in the <Instrument Leg> component block.

### 6.2.3.2 Multileg Reporting Type

*Currently supported through bilateral agreement only*.

Multileg orders may require various types of reporting depending on how they are implemented across the entire transaction chain (from order through clearing and settlement). For that reason the ability for market participants (and the market) to be able to set the level of response requested to an order for a Multileg is specified using the MultiLegRptTypeReq (563) field. The MultiLegRptTypeReq can also be bilaterally agreed in which case the message field is used to temporarily override that default for an individual multileg message.

*Table 75: Multileg Status Reporting*

| MultiLegRptTypeReq (563) | Execution Report for Multileg | Execution Report for each leg | Comment |
|---|---|---|---|
| 0 - Report by mulitleg security only (do not report legs) | Yes | No | In this case a single Execution Report is produced for the multileg order. The |

| MultiLegRptTypeReq (563) | Execution Report for Multileg | Execution Report for each leg | Comment |
|---|---|---|---|
| | | | <InstrmtLegExecGrp> component block specifies details of the legs. |
| 1 - Report by multileg security and by instrument legs belonging to the multileg security | Yes | Yes | In this case a single Execution Report is sent for the multiileg order and one for each leg. |
| 2 - Report by instrument legs belonging to the multileg security only (do not report status of multileg security) | No (except in response to an update) | Yes | In this case no Execution Report is sent for the multileg (except in the case of responding to an inbound order message), instead one Execution Report is produced for each leg. |

### 6.2.3.3 Implied Prices

Some markets support dissemination of implied prices in Market Data (see Section 14 on page 189 for details). There are two types of those:

#### Implied-In Prices

Implied-In prices are shown in a multileg book and are based on orders in the outright books of the legs of the multileg security. Implied-In prices mean the multileg book view will show the combined underlying liquidity and prices in the multileg book. *Implied-In prices are currently not supported*.

#### Implied-Out prices

Implied-Out prices are shown in the outright books of the legs of the multileg and are based on orders in the multileg security. Implied-Out prices are sometimes known as "baits".

Users can relate their multileg orders to market data in cases where the marketplace publishes order depth market data including an order identifier (SecondaryOrderID, 198).

### 6.2.3.4 Multileg Model

Multileg orders comes in two types:

- For a **Pre-defined Multileg Security**
- A **Strategy Order** for a combination of securities not represented as a product.

The type is defined in the MultilegModel (20016) field.

### 6.2.3.5 Multileg Price Method

Multileg orders support a variety of price methods (MultilegPriceMethod, 20004) not available for other types of order. The price method is used to determine how the price of non-securitized multileg order is to be interpreted when applied to the legs. The following options are available for use subject to marketplace rules for various product groups:

#### Net Price (1)

The price is given as the sum of the Price * Ratio for all legs.

- If buying the strategy, the price of a bought leg (which is a buy-leg in the multileg definition) is added, and the price of a sold leg is subtracted.
- If selling the strategy, the price of a bought leg (which is a sell-leg in the multileg definition) is subtracted, and the price of a sold leg is added.

### 6.2.3.6 Price

The price for a multileg can be defined according to the following:

#### A single price for the multileg

The Price (44) field is defined at the root level of the multileg. No further price details are provided. Applicable for MultilegPriceMethod (20004) = 1, 2, 3, 5 and 6.

### 6.2.3.7 Quantity

The quantity of a multileg order is defined in two ways:

#### Productized Multilegs

Here the quantity is defined at the root level only, using the OrderQty (38) field.

#### Strategy Orders

Here the quantity is defined at the leg level only, using the the LegOrderQty (685) field.

### 6.2.3.8 Duration Neutral Multileg Orders

Markets supporting duration neutral multileg orders, e.g. in trading fixed income swaps, allow users to indicate they want the marketplace to calculate the quantity of a certain leg by using the standard formula. The following parameter is used:

- LegExecInst (20247) = "z" - Execute as duration neutral

### 6.2.3.9 FX Neutral Multileg Orders

Certain markets may support the quoting of prices in other than the standard currency through providing a multileg order with one FX-leg and one normal security leg.

Example trade for the Ericsson stock (normally traded in SEK) by quoting in USD:

- Leg 1 = Buy 1 Ericsson
- Leg 2 = Buy SEK/USD
  - LegExecInst (20247) = "y" - Execute as FX neutral
- Price (44) = The product of the Price for all legs.
- MultilegPriceMethod (20004) = "6" - Multiplied Price

The FX-leg cannot have a fixed ratio/quantity, it is instead specified as "FX neutral". The quantity of the FX leg is calculated by the trading engine.

### 6.2.3.10 Legs Traded in Different Currencies

A multileg order could identify two legs which are traded in different currencies (or both legs traded in currencies different from the one used in quoting the multileg). Provided the price of the multileg is defined in the Currency (15) field, the relevant price for the other leg could be calculated if a currency conversion ratio is provided with the multileg order. This functionality is especially relevant in markets where trading is done in diverse currencies. Multi-currency multileg products may not be common, but the the functionality is relevant when non-securitized multileg orders are supported.

Example:

- Ericsson is quoted in SEK (Swedish krona), the current bid is SEK 25.00
- Nokia is quoted in EUR, the current ask is EUR 25.00
- The FX ratio is SEK 10.00 = EUR 1.00
- A user submits a multileg order to sell 100 Ericsson and buy 100 Nokia for a net price of EUR 22.50 per unit. The user provides the currency conversion ratio of 10.0 for the Ericsson leg and 1.0 for the Nokia leg.
- The marketplace tries to execute against the books:
  - anchoring in Ericsson at SEK 25.00 and finding a Nokia price at EUR 25.00 (EUR 22.50 of order + EUR 2.50 from sale of Ericsson at 1/10 ratio) or
  - anchoring in Nokia at EUR 25.00 and finding an Ericsson price at SEK 25.00 (EUR 2.50 left to cover from difference of order price and Nokia price – converted to SEK 25.00 at 1/10 ratio)

The applicable parameters are:

- Currency (15)
- LegCurrencyRatio (20234)

### 6.2.3.11 Other Multileg Order Properties

Some Order properties can be defined both at the root level and the leg level of the message. The root level value will be used as the default value to be overriden by any value specified by an individual leg. One example of this is that pre-allocation can be done either for the entire multileg or per leg:

- Use the AllocID (70) and the <PreAllocMlegGrp> component block to define pre-allocation properties for the entire multileg order
- Use the LegAllocID (20090) and the <LegPreAllocGrp> component block to define per-allocation properties for a specific leg.

Please refer to Section 4.5 on page 42 for full details on accounts, pre-allocation and give-ups.

# 6.3 Message Details

## 6.3.1 NewOrderMultileg

*Table 76: NewOrderMultileg*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | **StandardHeader** | Y | MsgType = AB | |
| 11 | ClOrdID | Y | Unique identifier of the order as assigned by institution or by the intermediary 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" | |
| 70 | AllocID | N | Used to assign an identifier to the block of individual preallocations | String |
| | **PreAllocMlegGrp** | 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, or W) must be specified. | MultipleCharValue |
| | **DisplayInstruction** | N | Insert here the set of "ReserveInstruction" fields defined in "common components of application messages"<br>**OMX Comment:** Allowed for productized multileg seurities but not for Strategy Orders | |
| | **TrdgSesGrp** | N | Specifies the number of repeating TradingSessionIDs<br>**OMX Comment:** Allowed for productized multileg seurities but not for Strategy Orders | |
| 54 | Side | Y | Additional enumeration that indicates this is an order for a multileg order and that the sides are specified in the Instrument Leg component block.<br>**OMX Comment:** Valid values for productized multilegs ( standardized and tailor-made combinations):<br>1 = Buy<br>2 = Sell<br><br>Valid values for Strategy Orders (a.k.a. free combinations):<br>B = As defined<br><br>Note that sides for a Strategy Order is specified in the InstrumentLeg component block | char |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | **Instrument** | Y | Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages"<br>SecurityType[167] = "MLEG"<br>CFICode should be set to the type of multileg product, such as "O" - options, "F" - Future or Swap.<br>**OMX Comment:** Messages shall use identifier fields only.<br>Instrument must not be specified for Strategy Orders. Mandatory for productized security. | |
| | **LegOrdGrp** | Y | Number of legs<br>Can be zero (e.g. standardized multileg instrument such as an Option strategy) - must be provided even if zero | |
| 60 | TransactTime | Y | Time this order request was initiated/released by the trader, trading system, or intermediary. | UTCTimestamp |
| | **OrderQtyData** | N | Insert here the set of "OrderQtyData" fields defined in "Common Components of Application Messages" Conditionally required when the multileg order is not for a FX Swap, or any other swap transaction where having OrderQty is irrelevant as the amounts are expressed in the LegQty. | |
| 40 | OrdType | Y | | char |
| 20016 | MultilegModel | N | **OMX Comment:** Currently not in FIX 5.0. OMX requests extension. | int |
| 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 |
| 20004 | MultilegPriceMethod | N | **OMX Comment:** Currently not in FIX 5.0. OMX requests extension. | int |
| 15 | Currency | N | **OMX Comment:** Only applicable in for "strategy orders". Used in cases where the legs are quoted in different currencies or when the multileg price is quoted in a currency other than legs. | Currency |
| 1080 | RefOrderID | N | Required for counter-order selection / Hit / Take Orders. (OrdType = Q)<br>**OMX Comment:** Allowed for productized multileg seurities but not for Strategy Orders | String |
| 1081 | RefOrderIDSource | N | Conditionally required if RefOrderID is specified.<br>**OMX Comment:** Allowed for productized multileg seurities but not for Strategy Orders | char |
| 59 | TimeInForce | N | Absence of this field indicates Day order<br>**OMX Comment:** Strategy orders limited to 4 - Fill Or Kill (FOK) | char |
| 432 | ExpireDate | N | Conditionally required if TimeInForce = GTD and ExpireTime is not specified. | LocalMktDate |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 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 |
| 77 | PositionEffect | N | For use in derivatives omnibus accounting<br>**OMX Comment:** Note that clearing related tags can be overriden at the Leg level | char |
| 20232 | RiskfreeRate | N | **OMX Comment:** Not in FIX. OMX request addi-tion | float |
| | **StandardTrailer** | Y | | |

## 6.3.2 MultilegOrderCancelReplace

*Table 77: MultilegOrderCancelReplace*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | **StandardHeader** | Y | MsgType = AC | |
| 37 | OrderID | N | Unique identifier of most recent order as assigned by sell-side (broker, exchange, ECN). | String |
| 41 | OrigClOrdID | Y | ClOrdID of the previous order (NOT the initial order of the day) when canceling or replacing an order. | String |
| 11 | ClOrdID | Y | Unique identifier of replacement order as as-signed by institution or by the intermediary with closest association with the investor.. Note that this identifier will be used in ClOrdID field of the Cancel Reject message if the replacement re-quest is rejected. | String |
| 586 | OrigOrdModTime | N | | UTCTimes-tamp |
| | **Parties** | N | Insert here the set of "Parties" (firm identification) fields defined in "Common Components of Appli-cation Messages" | |
| 70 | AllocID | N | Used to assign an identifier to the block of individ-ual preallocations | String |
| | **PreAllocMlegGrp** | N | Number of repeating groups for pre-trade alloca-tion | |
| 18 | ExecInst | N | Can contain multiple instructions, space delimit-ed. If OrdType=P, exactly one of the following values (ExecInst = L, R, M, P, O, T, or W) must be specified. | MultipleCharVal-ue |
| | **DisplayInstruction** | N | Insert here the set of "DisplayInstruction" fields defined in "common components of application messages" | |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | | | **OMX Comment:** Allowed for productized multileg seurities but not for Strategy Orders | |
| | **TrdgSesGrp** | N | Specifies the number of repeating TradingSessionIDs<br>**OMX Comment:** Allowed for productized multileg seurities but not for Strategy Orders | |
| 54 | Side | Y | Additional enumeration that indicates this is an order for a multileg order and that the sides are specified in the Instrument Leg component block. | char |
| | **Instrument** | Y | Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages"<br>SecurityType[167] = "MLEG"<br>CFICode should be set to the type of multileg product, such as "O" - options, "F" - Future or Swap.<br>**OMX Comment:** Messages shall use identifier fields only.<br>Instrument must not be specified for Strategy Orders. Mandatory for productized security. | |
| | **LegOrdGrp** | Y | Number of legs<br>Can be zero (e.g. standardized multileg instrument such as an Option strategy) - must be provided even if zero | |
| 60 | TransactTime | Y | Time this order request was initiated/released by the trader, trading system, or intermediary. | UTCTimestamp |
| 20004 | MultilegPriceMethod | N | **OMX Comment:** Currently not in FIX 5.0. OMX requests extension. | int |
| | **OrderQtyData** | Y | Insert here the set of "OrderQtyData" fields defined in "Common Components of Application Messages" | |
| 40 | OrdType | Y | | char |
| 20016 | MultilegModel | N | **OMX Comment:** Currently not in FIX 5.0. OMX requests extension. | int |
| 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 |
| 15 | Currency | N | | Currency |
| 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 |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 1091 | PreTradeAnonymity | N | | Boolean |
| 58 | Text | N | | String |
| 355 | EncodedText | N | Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field. | data |
| 77 | PositionEffect | N | For use in derivatives omnibus accounting | char |
| 20232 | RiskfreeRate | N | **OMX Comment:** Not in FIX. OMX request addition | float |
| | **StandardTrailer** | Y | | |

## 6.4 Component Blocks (Multileg Order Specific)

For components that are not specific for Multileg Orders, please see Section 19 on page 261.

### 6.4.1 Components

None.

### 6.4.2 Implicit Components

#### 6.4.2.1 InstrmtLegExecGrp

*Table 78: InstrmtLegExecGrp*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 555 | NoLegs | N | Number of legs<br>Identifies a Multi-leg Execution if present and non-zero. | NumInGroup |
| | **InstrumentLeg** | N | Must be provided if Number of legs > 0 | |
| >685 | LegOrderQty | N | When reporting an Execution, LegOrderQty may be used on Execution Report to echo back original LegOrderQty submission.<br>This field should be used to specify OrderQty at the leg level rather than LegQty (deprecated). | Qty |
| | **LegPreAllocGrp** | N | **OMX Comment:** Currently not in FIX 5.0. OMX requests extension. | |
| >20090 | LegAllocID | N | **OMX Comment:** Currently not in FIX 5.0. OMX requests extension. | String |
| >564 | LegPositionEffect | N | Provide if the PositionEffect for the leg is different from that specified for the overall multileg security | char |
| | **NestedParties** | N | Insert here the set of "Nested Parties" (firm identification "nested" within additional repeating group) fields defined in "Common Components of Application Messages"<br>Used for NestedPartyRole=Leg Clearing Firm/Account, Leg Account/Account Type | |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| >566 | LegPrice | N | Provide only if a Price is required for a specific leg. Used for anchoring the overall multileg security price to a specific leg Price. | Price |
| >20229 | LegVolatility | N | **OMX Comment:** Not in FIX. OMX request addition | float |
| >20231 | LegDividendYield | N | **OMX Comment:** Not in FIX. OMX request addition | float |
| >20234 | LegCurrencyRatio | N | **OMX Comment:** Not in FIX. OMX request addition | float |
| >20247 | LegExecInst | N | **OMX Comment:** Not in FIX. OMX request addition | MultipleCharValue |

### 6.4.2.2 LegOrdGrp

*Table 79: LegOrdGrp*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 555 | NoLegs | Y | Number of legs<br>Can be zero (e.g. standardized multileg instrument such as an Option strategy) – must be provided even if zero | NumInGroup |
| | **InstrumentLeg** | N | Must be provided if Number of legs > 0 | |
| >20090 | LegAllocID | N | Used to assign an an allocation id to the leg level block of preallocations<br>**OMX Comment:** Currently not in FIX 5.0. OMX requests extension. | String |
| | **LegPreAllocGrp** | N | | |
| >564 | LegPositionEffect | N | Provide if the PositionEffect for the leg is different from that specified for the overall multileg security | char |
| | **NestedParties** | N | Insert here the set of "Nested Parties" (firm identification "nested" within additional repeating group) fields defined in "Common Components of Application Messages"<br>Used for NestedPartyRole=Leg Clearing Firm/Account, Leg Account/Account Type | |
| >566 | LegPrice | N | Provide only if a price is required for a specific leg. Used for anchoring the overall multileg security price to a specific leg price. | Price |
| >685 | LegOrderQty | N | | Qty |
| >20229 | LegVolatility | N | **OMX Comment:** Not in FIX. OMX request addition | float |
| >20231 | LegDividendYield | N | **OMX Comment:** Not in FIX. OMX request addition | float |
| >20234 | LegCurrencyRatio | N | **OMX Comment:** Not in FIX. OMX request addition | float |
| >20247 | LegExecInst | N | **OMX Comment:** Not in FIX. OMX request addition | MultipleCharValue |

#### 6.4.2.3 LegPreAllocGrp

*Table 80: LegPreAllocGrp*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 670 | NoLegAllocs | N | | NumInGroup |
| >671 | LegAllocAccount | N | | String |
| >672 | LegIndividualAllocID | N | | String |
| | **NestedParties2** | N | Insert here the set of "Nested Parties #2" (firm identification "second instance of nesting" within additional repeating group) fields defined in "Common Components of Application Messages" | |
| >673 | LegAllocQty | N | | Qty |
| >674 | LegAllocAcctIDSource | N | | String |
| >675 | LegSettlCurrency | N | | Currency |

#### 6.4.2.4 PreAllocMlegGrp

*Table 81: PreAllocMlegGrp*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 78 | NoAllocs | N | Number of repeating groups for pre-trade allocation<br>**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 |
| >736 | AllocSettlCurrency | N | **OMX Comment:** Currently not supported | Currency |
| >467 | IndividualAllocID | N | | String |
| | **NestedParties3** | N | Insert here the set of "NestedParties3" (firm identification "nested" within additional repeating group) fields defined in "Common Components of Application Messages" | |
| >80 | AllocQty | N | **OMX Comment:** Currently not supported | Qty |

## 6.5 Workflows

### 6.5.1 Introduction

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

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

Again, note that the workflow defined in chapter Section 5 on page 49 also applies to multileg orders (although the inbound messages are different when inserting and modifying the multileg order).

### 6.5.2 Strategy Order

No product definition is used. The message flow is the same as the Predefined Multileg Security models – the difference being that the market receiving the order does not create nor maintain product information for the multileg security. Pre- and post-trade information are only visible to the public (in market data) through implied prices and trades in the books of the legs.

*Table 82: Product Definition using New Order Multileg*

| | **Counterparty intersted in trading a multileg security** | | **Marketplace** |
|---|---|---|---|
| 1 | Send New Order – Multileg that includes the multileg security definition in the Leg Instrument Block | -> | Accepts order for processing<br>Products are identified for the legs.<br>If the multileg security is not a valid multileg strategy in the market – the order is rejected. The order is rejected using an Execution Report – indicating an invalid product was encountered. |
| 2a | | <- | If MultilegReportTypeRequest =0 or =1 or if market rules require reporting by multileg security:<br>Send Execution Report for the overall multileg security (MultilegReportType=2) |
| 2b | | <- | If MultliegReportTypeRequest =1 or =2 or if market rules require reporting by multileg security<br>Send Execution Reports for each instrument leg defined previously for the multileg security (MultilegReportType=3) |

# 7 Contingent (Linked) Orders

**Note:**

Contingent orders are not explicitly supported in FIX. OMX is currently engaged in discussions with the FPL Global Technical Committee to expand FIX with contingent order support.

## 7.1 Business Message Types

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

*Table 83: Business Messages*

| In/Out | Message Name | Comment |
|---|---|---|
| In | **New Order List** | Used to enter a set of contingent orders. |
| In | **Order Cancel Replace Request** | Used to update an individual order part of a contingency. |
| In | **List Cancel Request** | Used to delete a set of contingent orders. |
| Out | **List Status** | Used to acknowledge / reject a set of contingency orders and a cancellation of the set. |
| Out | **Execution Report** | |
| 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. |

## 7.2 Contingent Orders

### 7.2.1 Introduction

Contingent orders (a.k.a. Linked Orders or Alternative Orders) are orders that have a dependency on other orders. The orders of a contingency are entered together in a single message. A Contingent Order can be regarded as a multileg order where a fill in one leg affects the other legs. You could say it is a multileg order with an OR condition between the legs, instead of an AND condition. In the Contingent Order case, the multileg itself is generally not a product but individual securities. As the legs of a Contingent order is in fact separate orders, they should also be treated as separate from an Execution Report etc point of view.

**Note:**

Note: The List Order messages of the FIX Standard are also used for the trading of baskets and similar - that functionality is currently not supported!

The Contingent Order (or rather the individual orders of it) is allowed to sit on the book; it is made public by displaying each individual order as a separate order over market data. There will be no resulting trade for the Contingent Order as such; all trades are in the individual security.

There are various kinds of contingent orders:

#### One Cancels the Other (OCO)

*Currently not supported!*

An OCO order is an order whose execution results in the immediate cancellation of another order linked to it. Cancellation of the Contingent Order happens on a best efforts basis. In an OCO order, both orders are live in the marketplace at the same time. The execution of either order triggers an attempt to cancel the un-executed order. Partial executions will also trigger an attempt to cancel the other order.

#### One Triggers the Other (OTO)

*Currently not supported!*

An OTO orders involves two orders - a primary order and a secondary order. The primary order is a live marketplace order. The secondary order, held in a separate order file, is not. If the primary order executes in full, the secondary order is released to the marketplace and becomes live. An OTO order can be made up of e.g. stock orders, option orders, or a combination of both.

#### One Updates the Other (OUO)

An OUO order is an order whose execution results in the immediate reduction of quantity in another order linked to it. The quantity reduction happens on a best effort basis. In an OUO order both orders are live in the marketplace at the same time. The execution of either order triggers an attempt to reduce the remaining quantity of the other order, partial executions included. Variants:

- **Proportional Quantity Reduction**. Instead of canceling the other Contingent Order(s), their quantity is reduced in proportion to the filled quantity. Example:
  - Order A is for 100; Order B is for 50.
  - When order B is partially filled for 25 (50 %), order A is restated to a leaves quantity of 50 (50 %).
- **Absolute Quantity Reduction**. *Currently not supported!* Instead of canceling the other Contingent Order(s), their quantity is reduced with the same partially filled value.
  - Order A is for 100; Order B is for 50.
  - When order B is partially filled for 25, order A is restated to a leaves quantity of 75.

> **Note:**
>
> OMX currently only supports OUO contingency orders with proportional quantity reduction. OCO's, OTO's and OUO's with absolute quantity reduction are not supported.

### 7.2.2 Main Workflow

A set of contingent orders are entered using the New Order List message. As the contingency is accepted or rejected, a List Status message is returned including the reason for a reject if applicable. All the included individual orders must be valid for the contingency to be accepted. State changes for the individual contingent orders are relayed using the Execution Report message.

All other actions follow the messages and rules specified in chapter Section 5 on page 49, but note that updating the individual contingent orders is subject to restrictions not applicable for non-contingent orders. If the user whishes to cancel the entire contingency, a List Cancel Request specifying the relevant ListID should be sent.

Note that contingent orders may be subject to limitations regarding what order conditions apply. Complete such rules are outside the scope of this specification.

## 7.2.3 Contingent Order Features

### 7.2.3.1 Order Identifiers

Individual orders of the contingency are identified with the ClOrdID and OrderID fields as defined in Section 5 on page 49.

The contingent order has its own identifier, ListID (66). The ListID (66) is relayed on Execution Reports for all orders of the contingency and on trade confirms (Trade Capture Reports).

### 7.2.3.2 Type of Contingency

The type of contingency is defined in the ContingencyType (20077) field:

- 4 – One Updates the Other (OUO) - Proportional Quantity Reduction

### 7.2.3.3 Common Properties

Certain properties of a Contingent Order need not be defined for every individual order part of the contingency. Those properties can be defined for the first order in the list of order and are automatically copied to all individual orders that do not contain defined values for the fields. Common property fields are:

- Account and allocation instructions
  - Account (1) - *currently not supported*.
  - AccountIDSource (660) = 99 = Other (custom or proprietary) - *currently not supported*.
  - AllocID (70)
  - <PreAllocGrp> (all fields in this component block)
- Order expiration instructions:
  - TimeInForce (59)
  - <TrdgSesGrp> (all fields in this component block)
  - ExpireDate (432)
  - ExpireTime (126)
- Order capacity instructions:
  - OrderCapacity (528)
  - OrderRestrictions (529)
- ExecInst (18)
- PreTradeAnonymity (1091)

### 7.2.3.4 Restrictions

There is no update message for a list of Contingent Orders, if the user wishes to update the entire list he must cancel the whole Contingency and insert a new one.

An order part of a Contingent Order cannot be removed from the contingency - i.e. an Order Cancel Request is not allowed against individual orders.

An order cannot be added to the contingency - there is no ListID (66) in the New Order Single message.

Contingent orders are implicitly good for continuous trading sessions only.

The number of legs allowed for each type of contingency is bilaterally agreed.

The following order features are not allowed for orders part of a contingency:

- Pegs
- Triggers
- Reserve size

### 7.2.3.5 One Updates the Other (OUO) and Execution Reports

Standard FIX has the following rule:

- LeavesQty (151) = OrderQty (38) - CumQty (14)

A fill in one order of an OUO means the accumulated executed quantity, CumQty (14), of that order is increased and the remaining quantity, LeavesQty (151) reduced, However, the fill also leads to a reduction of the remaining quantity of the other orders of the same contingency:

- CumQty (14), the value is retained as there has been no fill for the order
- LeavesQty (151), the value is reduced according to the OUO instruction (proportionally or absolute)
- OrderQty (39), the value is reduced in order to maintain the rule above.

**Note:**

The marketplace does not do unsolicited updates of the order quantity (OrderQty, 39) in cases other than OUO.

State changes for orders in the contingency are relayed using Execution Report messages.

### 7.2.3.6 Unsolicited Cancellation of an Individual Order in the Contingency

Subject to marketplace rules, an individual order in a One Updates the Other (OUO) contingency may be canceled if its order book is made unavailable for automatic matching (e.g. due to a trade halt). The other orders of the contingency remain executable.

# 7.3 Message Details

## 7.3.1 NewOrderList

*Table 84: NewOrderList*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | **StandardHeader** | Y | MsgType = E | |
| 66 | ListID | Y | Must be unique, by customer, for the day | String |
| 394 | BidType | Y | e.g. Non Disclosed Model, Disclosed Model, No Bidding Process<br>**OMX Comment:** Contingent orders must use 3 | int |
| 20077 | ContingencyType | N | **OMX Comment:** Required field | int |
| | **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 orders. | |
| | **ListOrdGrp** | Y | Number of orders in this message (number of repeating groups to follow) | |
| | **StandardTrailer** | Y | | |

## 7.3.2 ListCancelRequest

*Table 85: ListCancelRequest*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | **StandardHeader** | Y | MsgType = K | |
| 66 | ListID | Y | | String |
| | **Parties** | N | Insert here the set of "Parties" (firm identification) fields defined in "common components of application messages" | |
| 60 | TransactTime | Y | Time this order request was initiated/released by the trader or trading system. | UTCTimestamp |
| 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 | | |

### 7.3.3 ListStatus

*Table 86: ListStatus*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | **StandardHeader** | Y | MsgType = N | |
| 66 | ListID | Y | | String |
| 429 | ListStatusType | Y | | int |
| 82 | NoRpts | Y | Total number of messages required to status complete list.<br>**OMX Comment:** Always = 1 | int |
| 431 | ListOrderStatus | Y | | int |
| 20077 | ContingencyType | N | **OMX Comment:** Currently not in FIX 5.0. OMX requests extension. | int |
| 20244 | ListRejectReason | N | **OMX Comment:** Currently not in FIX 5.0. OMX requests extension. | String |
| 83 | RptSeq | Y | Sequence number of this report message. | int |
| 60 | TransactTime | N | | UTCTimes-tamp |
| | **OrdListStatGrp** | Y | Number of orders statused in this message, i.e. number of repeating groups to follow. | |
| | **StandardTrailer** | Y | | |

## 7.4 Component Blocks (Contingent Order Specific)

For components that are not specific for Contingent Orders, please see Section 19 on page 261.

### 7.4.1 Components

None

### 7.4.2 Implicit Components

#### 7.4.2.1 ListOrdGrp

*Table 87: ListOrdGrp*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 73 | NoOrders | Y | Number of orders in this message (number of repeating groups to follow) | NumInGroup |
| >11 | ClOrdID | Y | Must be the first field in the repeating group. | String |
| >67 | ListSeqNo | Y | Order number within the list<br>**OMX Comment:** A primary contingent order should have seq.no = 1 | int |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| >70 | AllocID | N | Use to assign an ID to the block of individual preallocations | String |
| | **PreAllocGrp** | N | | |
| >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, or W) must be specified. | MultipleCharValue |
| | **Instrument** | Y | Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages" | |
| >54 | Side | Y | Note: to indicate the side of SideValue1 or SideValue2, specify Side=Undisclosed and SideValueInd=either the SideValue1 or SideValue2 indicator. | char |
| >60 | TransactTime | N | | UTCTimestamp |
| | **OrderQtyData** | Y | Insert here the set of "OrderQtyData" fields defined in "Common Components of Application Messages" | |
| >40 | OrdType | N | | char |
| >44 | Price | N | | Price |
| >59 | TimeInForce | N | | char |
| >528 | OrderCapacity | N | | char |
| >529 | OrderRestrictions | N | | MultipleCharValue |
| >1091 | PreTradeAnonymity | N | | Boolean |
| >58 | Text | N | | String |

### 7.4.2.2 OrdListStatGrp

*Table 88: OrdListStatGrp*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 73 | NoOrders | Y | Number of orders statused in this message, i.e. number of repeating groups to follow. | NumInGroup |
| >11 | ClOrdID | Y | | String |
| >103 | OrdRejReason | N | Used if the order is rejected | int |
| >58 | Text | N | | String |

## 7.5 Workflows

### 7.5.1 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 in the following tables.

## 7.5.2 Contingent Orders

### 7.5.2.1 Vanilla One Updates the Other (OUO)

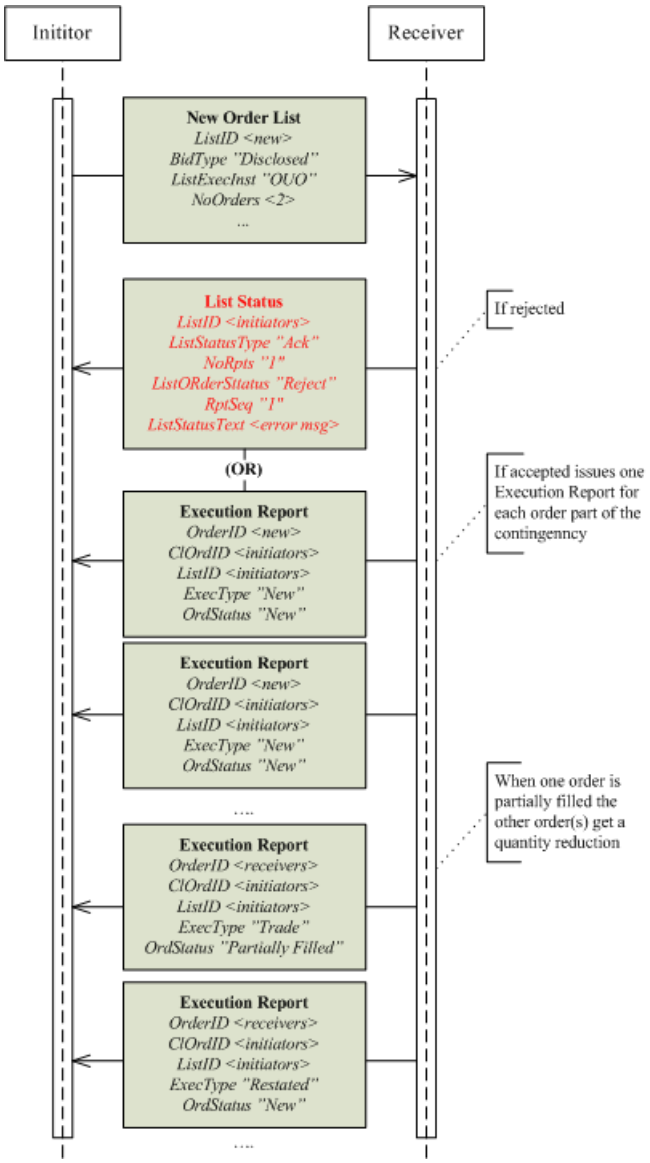


*Figure 2: Vanilla One Update the Other (OUO)*

Note that the List Status message may also be relayed to confirm an accepted List Order.

There are two variants to how quantity reduction can be done:

- Proportional Quantity Reduction. Instead of canceling the other Contingent Order(s), their quantity is reduced in proportion to the filled quantity. Example:
  - Order A is for 100; Order B is for 50.

    - When order B is partially filled for 25 (50 %), order A is restated to a leaves quantity of 50 (50 %).
- Absolute Quantity Reduction. Instead of canceling the other Contingent Order(s), their quantity is reduced with the same partially filled value.
    - Order A is for 100; Order B is for 50.
    - When order B is partially filled for 25, order A is restated to a leaves quantity of 75.

#### 7.5.2.2 Contingent Order Updates

The Root level properties of a set of contingent orders cannot be updated. The individual contingent orders can be updated by using the Order Cancel Replace Request message. Bilateral agreement governs the behavior.

#### 7.5.2.3 Contingent Order Cancellation

The whole list, including its contingent orders, can be canceled using the List Cancel Request message specifying the ListID.

# 8 Continuous and General Quote Handling

## 8.1 Business Message Types

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

*Table 89: Business Messages*

| In/Out | Message Name | Comment |
|---|---|---|
| In | **Quote** | Used to insert, modify or remove a single quote |
| In | **Mass Quote** | Used to insert, modify or remove many quotes in a single transaction. |
| In | **Quote Cancel** | Used to cancel (one or) many quotes |
| Out | **Execution Report** | Used to relay fills for quotes |
| Out | **Quote Status Report** | Used to relay the state of an individual quote, including acknowledging or rejecting a (single) Quote.<br>Also used as the response to a Quote Cancel or Quote Status Request when those identify a single quote. |
| Out | **Mass Quote Acknowledgement** | Used to acknowledge a Mass Quote. Includes rejection of whole message as well as individual quotes<br>Also used as the response to a Quote Cancel or Quote Status Request when those identify a set of a quotes. |

## 8.2 Quoting

### 8.2.1 Introduction

The Quote message is used by market makers and other actors with similar responsibilities to send quotes into a market. The same message type is also used in Quote Negotiation workflows, see chapter Section 9 on page 143 for further details on this. The quote messages, as described in this section, are typically used to send continuous unsolicited quotes in markets with tradable quoting. Such quotes are sent by quote issuers (market makers, specialists, liquidity providers or similar), i.e. actors that have an obligation to provide continuous liquidity in the market.

A quote is two-sided, i.e. normally contains both bid and offer price and size. Many marketplaces limit market makers to one (two-sided) quote in each security.

The Quote message contains a quote for a single product but there is also the Mass Quote message that allows the user to submit multiple quotes in a single message.

Responses (acks / rejects) for Quote, Mass Quote and Quote Cancel messages are subject to bilateral agreement between parties and/or as specified in the QuoteResponseLevel (301) field of the request message.

The Time-in-Force for continuous quotes is determined by agreement between counterparties.

Please refer to [6] on page 21 for best practices published by the FPL Global Exchanges and Markets Committee.

**Note:**

Continuous quoting is currently not supported. Some of the messages described in this chapter are also used for Quote Negotiation.

## 8.2.2 Main Workflow

### 8.2.2.1 The Quote

The quote workflow starts with the user submitting a Quote or Mass Quote message. In response a Quote Status Report (or Mass Quote Acknowledgement) is optionally produced. Response messages are directed to the sender of the quote and will contain details of the quote. If the quote is rejected the Quote Status Report (or Mass Quote Acknowledgement) will contain relevant error messages.

A Mass Quote message can be limited in scope subject to rules defined by the marketplace.

### 8.2.2.2 Fills

When a quote is filled, the Execution Report message is used to report details about the fill. In addition, a Trade Capture Report will be produced. The principal differences between the two are described in Section 5 on page 49.

**Note:**

Execution Reports are not used to report non-fill status changes. Such are relayed using the Mass Quote Acknowledgement or Quote Status Report messages.

Because quotes are usually replaced or replenished it is not always possible, nor does it necessarily make sense, for markets to keep track and transmit the detailed quantity information required on the quote. Execution Reports for trades against a tradable quote use the quantity fields in the following manner.

*Table 90: Quote Fill Fields*

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

Users should use the LastQty (32) or LeavesQty (151) to update their status of their quotes.

### 8.2.2.3 Market Data

When a quote 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).

### 8.2.2.4 Quote Modification

Quote modification is accomplished through the use of the same messages as when adding a quote, i.e. through the Quote or Mass Quote messages.

#### Markets allowing a Single Quote per Order book

Replacing a quote in a single quote market is straightforward as every update replaces the old one based on the quote issuer, security (series) and side.

#### Markets allowing Multiple Quotes per Order book

Replacing a quote in a multiple quote market requires attention when prices are moved. See Section 8.2.3.3 on page 117 on Multiple quote markets.

### 8.2.2.5 Quote Cancellation

A quote can be canceled (or withdrawn) either using the Quote Cancel message or by sending a Single or Mass Quote message with bid and offer prices and sizes all set to zero:

- BidPx (132) = 0
- OfferPx (133) = 0
- BidSize (134) = 0
- OfferSize (135) = 0

The Quote Cancel message is used by the quote issuer to cancel quotes (a cancel applies to quotes made by the same quote issuer). Canceling a Quote is accomplished by indicating the type of cancellation in the QuoteCancelType (298) field. The following types are supported:

- All quotes for an underlying
- Single Quote as specified in Quote ID

Note that the quote sides (bid / offer) can be canceled individually.

When a single quote is canceled, this is reported through the Quote Status Report message. If a single side is canceled the Quote Status Report either shows only the canceled side or both sides (subject to marketplace preferences).

In the case a set of quotes are canceled, it is reported using the Mass Quote Acknowledgement. The Mass Quote Acknowledgement will contain either the identifiers of the canceled quotes or the total number of quotes canceled per underlying. What method is used is subject to bilateral agreement or specification in the QuoteResponseLevel (301) of the Quote Cancel.

### 8.2.2.6 Quote Status Report

The quote status report message is used:

- As the optional response to a Quote message
- As a response to a Quote Cancel message for a single quote
- When the marketplace or trading engine does an unsolicited change to a quote (e.g. automatic replenishment of quote size after the quote is exhausted).

### 8.2.2.7 Mass Quotes

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

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

Each QuoteSet contains a repeating group of QuoteEntries where each entry represents an individual two-sided quote.

It is possible that the number of Quote Entries for a Quote Set could exceed one's physical or practical message size. It may be necessary to fragment a message across multiple quote messages. Message size limits are bilaterally agreed.

The grouping of quotes is as follows:

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

### 8.2.2.8 Mass Quote Acknowledgement

Mass Quote Acknowledgement is used as the application level response to a Mass Quote message.

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

The Mass Quote Acknowledgement message is also used to respond to:

- A Quote Status Request that identifies many Quotes.
- A Quote Cancel that identifies many Quotes.

### 8.2.2.9 Quote Rejects

There are two different messages used to reject quote messages:

*Table 91: Quote Reject Messages*

| Reject Message | Direc-tion | Business Message | Comment |
|---|---|---|---|
| **Quote Status Re-port** | Out | Quote | |

| Reject Message | Direction | Business Message | Comment |
|---|---|---|---|
| **Mass Quote Acknowledgement** | Out | Mass Quote | |

## 8.2.3 Quote Features

### 8.2.3.1 Quote Identifiers

#### 8.2.3.1.1 Quote Message Identifier

Every inbound quote message must be associated with a unique message identifier per FIX session. The message identifier can be used to keep an audit trail of quote updates and is used to link a request message to responses. The message identifier is echoed back on response, fill and other messages that are sent out based on a quote. For historical reasons the FIX protocol names those identifiers differently in the various quote messages:

- **Quote ID**

  The QuoteID (117) is the message identifier used in Mass Quote messages.
- **Quote Message ID**

  The QuoteMsgID (1166) is the message identifier used in Quote and Quote Cancel messages.
- **Quote Status Request ID**

  The QuoteStatusReqID (649) is the message identifier used in Quote Status Requests.

The message identifier is relayed back in the following messages:

*Table 92: Quote Message Identifier in Response Messages*

| Request Message | Response Message | Message Identifier Mapping |
|---|---|---|
| Quote | Quote Status Report | Q.QuoteMsgID -> QSR.QuoteMsgID |
| Mass Quote | Mass Quote Acknowledgement | MQ.QuoteID -> MQA.QuoteID |
| Quote Cancel | Quote Status Report | QC.QuoteMsgID -> QSR.QuoteMsgID |
| | Mass Quote Acknowledgement (cancel of set of quotes) | QC.QuoteMsgID -> MQA.QuoteID |
| N/A | Quote Status Report (unsolicited restatement) | Q.QuoteMsgID or MQ.QuoteID -> QSR.QuoteID |
| N/A | Execution Report, Trade Capture Report (on fills) | Q.QuoteMsgID or MQ.QuoteID -> ClOrdID |

When alternative fields ("or") are shown in the table, the field to use depends on what message was last used to update the quote.

It is recommended that the Quote message identifiers are taken from the same numbering series as the ClOrdID in cases where Orders and Quotes are submitted through the same FIX session. Quote issuers using multiple sessions or even trading applications should cater for appropriate QuoteID uniqueness.

#### 8.2.3.1.2 Quote Entity Identifier

Every quote must be associated with a unique entity identifier. The identifier is used to identify an individual quote when updating quotes. The identifier can be compared to the OrderID (37) of orders, but is normally static over time as the same quote is continuously updated. Another difference from the OrderID is that the quote issuer includes the identifier in the Quote messages, it is not returned by the marketplace in responses to quotes. For historical reasons the FIX protocol names those identifiers differently in the various quote messages:

- **Quote ID**

  The QuoteID (117) is the entity identifier used in Quote, Quote Cancel and Quote Status Request messages.

- **Quote Entry ID**

  The QuoteEntryID (299) is the entity identifier used in Mass Quote messages. The ID must be unique per Quote Issuer and Security meaning the value "1" is sufficient in cases where a single quote is allowed.

It should be noted that a quote issuer is never allowed to exceed the number of quotes allowed for a single book (often one) - irrespective of what identifiers are used.

The Quote entity ID (QuoteID or QuoteEntryID) is echoed back in the following messages:

*Table 93: Quote Entity Identifier in Response Messages*

| Request Message | Response Message | Message Identifier Mapping |
|---|---|---|
| Quote | Quote Status Report | Q.QuoteID -> QSR.QuoteID |
| Mass Quote | Mass Quote Acknowledgement | MQ.QuoteEntryID -> MQA.QuoteEntryID |
| Quote Cancel | Quote Status Report (cancel of single quote) | Q.QuoteID or MQ.QuoteEntryID -> QSR.QuoteID |
| | Mass Quote Acknowledgement (cancel of set of quotes) | Q.QuoteID or MQ.QuoteEntryID -> MQA.QuoteEntryID |
| N/A | Quote Status Report (unsolicited restatement) | Q.QuoteID or MQ.QuoteEntryID -> QSR.QuoteID |
| N/A | Execution Report, Trade Capture Report (on fills) | Q.QuoteID or MQ.QuoteEntryID -> SecondaryClOrdID<br>(if OrderID is not used for other purpose it is assigned "N/A" or the same value as SecondaryClOrdID) |

When alternative fields ("or") are shown in the table, the field to use depends on what message was last used to update the quote.

### 8.2.3.2 Quote Response Level

*Currently supported through bilateral agreement only.*

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

QuoteResponseLevel (301) field. The QuoteResponseLevel can also be bilaterally agreed in which case the message field is used to temporarily override that default for an individual quote message.

The QuoteResponseLevel is used to specify the level of acknowledgement requested from the counterparty. A QuoteResponseLevel of:

- 0 = Indicates that no acknowledgement is requested (neither positive nor negative)
- 1 = Requests acknowledgement of invalid or erroneous quote messages only (negative)
- 2 = Requests acknowledgement of each quote message (both positive and negative)
- 3 = Summary Acknowledgement. For a Mass Quote, requests acknowledgement of each message (positive response without supplying Quote Set/Quote Entry data). For a Quote Cancel, requests acknowledgement of total number of canceled quotes, not the individual quote entries.

### 8.2.3.3 Multiple Quote Markets

Some markets allow a single quote issuer to simultaneously have multiple two-sided quotes in a single instrument. The feature allows the market maker to stay in the market even if one quote is traded out and can be used as an alternative to automatic quote replenishment.

- Each quote must have a unique quote entity identifier (QuoteEntryID [299] for Mass Quote messages or QuoteID [117] for single Quote messages). As the quotes only have to be unique within the quote issuer and instrument, IDs are recommended to start at 1 and increment with 1 up to the maximum allowed number of quotes.

Example:

- A Quote Issuer sends the first quote into the IBM June 80 Call
  - QuoteID (117) = "100", QuoteEntryID (299) = "1", IBM June 80 Call, Buy 10 @ $1.80, Sell 10 @ $1.90
- Submit a second quote in the same instrument
  - QuoteID (117) = "102", QuoteEntryID (299) = "2", IBM June 80 Call, Buy 10 @ $1.82, Sell 10 @ 1.92
- Update the first Quote
  - QuoteID"= "105", QuoteEntryID (299) = "1", IBM June 80 Call, Buy 10 @ $1.84, Sell 10 @ 1.94
- Update Second Quote while first quote is still at $1.82
  - QuoteID (117) = "110", QuoteEntryID (299) = "2", IBM June 80 Call, Buy 10 @ $1.86, Sell 10 @ 1.96
- At this point, this Quote Issuer has two quotes in IBM June 80 Calls
  - QuoteID (117) = "105", QuoteEntryID (299) = "1", Buy 10 @ 1.84, Sell 10 @ 1.94
  - QuoteID (117) = "110", QuoteEntryID (299) = "2", Buy 10 @ 1.86, Sell 10 @ 1.96

Note that the above is a simple example where the bid and offers are moved together. However, the quote issuer will be moving the bids and offers independently. When either side changes then the quote must be updated and both sides of that pair must be sent together. The Quote issuer must know which bids and offers are paired.

- The Quote Issuer can send one side only but it is treated as though both sides were sent.

    - QuoteID (117) = "120", QuoteEntryID (299) = "2", IBM June 80 Call, Sell 10 @ 1.92
- At this point, this Quote Issuer has two quotes in IBM June 80 Calls
    - QuoteID (117) = "105", QuoteEntryID (299) = "1", Buy 10 @ 1.84, Sell 10 @ 1.94
    - QuoteID (117) = "120", QuoteEntryID (299) = "2", Buy 10 @ 1.86, Sell 10 @ 1.92

## 8.2.4 Quote State Changes

A quote is, in principle, regarded as a permanent representation of interest from the relevant market maker, even though it may not always be externally visible (or implemented in the trading engine). This means the quote always has a state: it can only be “not found” when the market maker does not have any side quoted for a security. Empty (or “zero”) quote sides are represented using BidPx, OfferPx, BidSize and OfferSize = 0. However, zero quotes can also be implemented as non-existing quotes.

Quote state changes are divulged by:

- The Quote Status Report or Mass Quote Acknowledgement message after a quote update (excluding fills) and subject to the specified or bilaterally agreed QuoteReponseLevel
- Execution Reports after fills

A quote can move from any state to any other state.

The following table shows the recommended use of the QuoteStatus field in the Quote Status Report and the Mass Quote Acknowledgment messages. The list of status’s shown includes the main values.

*Table 94: Quote Status Usage*

| | **Response** | | |
|---|---|---|---|
| | **(Single) Quote: Quote Status Report** | **Set of Quotes: Mass Quote Acknowledgement** | |
| **Request message** | **Quote Status** | **Quote Status** | **Quote Entry Status** |
| Quote | • Accepted<br>• Rejected<br>• Canceled (if both sides = 0) | N/A | N/A |
| Mass Quote | N/A | • Accepted<br>• Rejected | • Accepted<br>• Rejected<br>• Canceled (if both sides = 0) |
| Quote Cancel | • Canceled<br>• Rejected | • Accepted<br>• Rejected | • Canceled<br>• Rejected (if "locked") |
| Unsolicited | • Removed from Market<br>• Unsolicited Quote Replenishment | • Removed from Market | • Removed from Market |

# 8.3 Message Details

## 8.3.1 Quote

*Table 95: Quote*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | **StandardHeader** | Y | MsgType = S | |
| 131 | QuoteReqID | N | Required when quote is in response to a Quote Request message<br>**OMX Comment:** Ony relevant in Quote Negotiations | String |
| 117 | QuoteID | Y | | String |
| 1166 | QuoteMsgID | N | Optionally used to supply a message identifier for a quote<br>**OMX Comment:** FIX 5.0 SP1 | String |
| 537 | QuoteType | N | Quote Type<br>If not specified, the default is an indicative quote | int |
| 1171 | PrivateQuote | N | Used to indicate whether a private negotiation is requested or if the response should be public. If field is not provided in message, the model must be bilaterally agreed.<br>**OMX Comment:** FIX 5.0 SP1 | Boolean |
| 301 | QuoteResponseLevel | N | Level of Response requested from receiver of quote messages.<br>**OMX Comment:** Field presence in message and default value subject to marketplace rules. | int |
| | **Parties** | N | Insert here the set of "Parties" (firm identification) fields defined in "Common Components of Application Messages"<br>**OMX Comment:** Assume CCP etc are se up by configuration | |
| | **Instrument** | Y | Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages"<br>**OMX Comment:** Inbound messages shall use identifier fields only. | |
| 132 | BidPx | N | If F/X quote, should be the "all-in" rate (spot rate adjusted for forward points). Note that either BidPx, OfferPx or both must be specified. | Price |
| 133 | OfferPx | N | If F/X quote, should be the "all-in" rate (spot rate adjusted for forward points). Note that either BidPx, OfferPx or both must be specified. | Price |
| 134 | BidSize | N | Specifies the bid size. If MinBidSize is specified, BidSize is interpreted to contain the maximum bid size. | Qty |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 135 | OfferSize | N | Specified the offer size. If MinOfferSize is specified, OfferSize is interpreted to contain the maximum offer size. | Qty |
| 110 | MinQty | N | For use in private/directed quote negotiations.<br>**OMX Comment:** Used to limit fills to a certain minimum size.<br>Only relevant for Private Quotes used in Quote Negotiations.<br>FIX 5.0 SP1 | Qty |
| 62 | ValidUntilTime | N | The time when the quote will expire<br>**OMX Comment:** Only relevant for Private Quotes used in Quote Negotiations | UTCTimestamp |
| 60 | TransactTime | N | | UTCTimestamp |
| | **StandardTrailer** | Y | | |

## 8.3.2 QuoteCancel

*Table 96: QuoteCancel*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | **StandardHeader** | Y | MsgType = Z | |
| 1166 | QuoteMsgID | N | Optionally used to supply a message identifier for a quote cancel<br>**OMX Comment:** FIX 5.0 SP1 | String |
| 117 | QuoteID | N | Conditionally required when QuoteCancelType = 5 (cancel quote specified in QuoteID). Maps to:<br>- QuoteID (117) of a single Quote<br>- QuoteEntryID (299) of a Mass Quote<br>**OMX Comment:** FIX 5.0 SP1 | String |
| 298 | QuoteCancelType | Y | Identifies the type of Quote Cancel request. | int |
| 301 | QuoteResponseLevel | N | Level of Response requested from receiver of quote messages.<br>**OMX Comment:** Field presence in message and default value subject to marketplace rules. | int |
| | **Parties** | N | Insert here the set of "Parties" (firm identification) fields defined in "Common Components of Application Messages" | |
| | **QuotCxlEntriesGrp** | N | The number of securities (instruments) whose quotes are to be canceled<br>Not required when cancelling all quotes. | |
| | **StandardTrailer** | Y | | |

## 8.3.3 QuoteStatusReport

*Table 97: QuoteStatusReport*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | **StandardHeader** | Y | MsgType = AI | |
| 649 | QuoteStatusReqID | N | | String |
| 131 | QuoteReqID | N | Required when quote is in response to a Quote Request message | String |
| 117 | QuoteID | Y | Maps to:<br>- QuoteID (117) of a single Quote<br>- QuoteEntryID (299) of a Mass Quote<br>**OMX Comment:** FIX 5.0 SP1 | String |
| 1166 | QuoteMsgID | N | Maps to:<br>- QuoteMsgID (1166) of a single Quote or Quote Cancel<br>- QuoteID (117) of a Mass Quote<br>**OMX Comment:** FIX 5.0 SP1 | String |
| 693 | QuoteRespID | N | Required when responding to a Quote Response message. | String |
| 537 | QuoteType | N | Quote Type<br>If not specified, the default is an indicative quote | int |
| 298 | QuoteCancelType | N | For use in response to quote cancel | int |
| | **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" | |
| 132 | BidPx | N | If F/X quote, should be the "all-in" rate (spot rate adjusted for forward points). Note that either BidPx, OfferPx or both must be specified.<br>**OMX Comment:** In a pure ack to a Quote, specifying the bid price is subject to bilateral agreement | Price |
| 133 | OfferPx | N | If F/X quote, should be the "all-in" rate (spot rate adjusted for forward points). Note that either BidPx, OfferPx or both must be specified.<br>**OMX Comment:** In a pure ack to a Quote, specifying the offer prices is subject to bilateral agreement | Price |
| 134 | BidSize | N | Specifies the bid size. If MinBidSize is specified, BidSize is interpreted to contain the maximum bid size. | Qty |
| 135 | OfferSize | N | Specified the offer size. If MinOfferSize is specified, OfferSize is interpreted to contain the maximum offer size. | Qty |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 110 | MinQty | N | **OMX Comment:** Only applicablle for Private Quotes (in Quote Negotiations).<br>FIX 5.0 SP1 | Qty |
| 60 | TransactTime | N | | UTCTimes-tamp |
| 297 | QuoteStatus | N | Quote Status | int |
| 300 | QuoteRejectReason | N | **OMX Comment:** FIX 5.0 SP1 | int |
| 58 | Text | N | | String |
| | **StandardTrailer** | Y | | |

## 8.3.4 MassQuote

*Table 98: MassQuote*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | **StandardHeader** | Y | MsgType = i (lowercase) | |
| 117 | QuoteID | Y | | String |
| 537 | QuoteType | N | Type of Quote<br>Default is Indicative if not specified | int |
| 301 | QuoteResponseLevel | N | Level of Response requested from receiver of quote messages.<br>**OMX Comment:** Field presence in message and default value subject to marketplace rules. | int |
| | **Parties** | N | Insert here the set of "Parties" (firm identification) fields defined in "Common Components of Appli-cation Messages" | |
| 293 | DefBidSize | N | Default Bid Size for quote contained within this quote message - if not explicitly provided. | Qty |
| 294 | DefOfferSize | N | Default Offer Size for quotes contained within this quote message - if not explicitly provided. | Qty |
| | **QuotSetGrp** | Y | The number of sets of quotes in the message | |
| | **StandardTrailer** | Y | | |

## 8.3.5 MassQuoteAcknowledgement

*Table 99: MassQuoteAcknowledgement*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | **StandardHeader** | Y | MsgType = b (lowercase) | |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 117 | QuoteID | N | Required when acknowledgment is in response to a Mass Quote, mass Quote Cancel or mass Quote Status Request message. Maps to:<br>- QuoteID (117) of a Mass Quote<br>- QuoteMsgID (1166) of Quote Cancel<br>- QuoteStatusReqID (649) of Quote Status Request<br>**OMX Comment:** FIX 5.0 SP1 | String |
| 297 | QuoteStatus | Y | Status of the mass quote acknowledgement. | int |
| 300 | QuoteRejectReason | N | Reason Quote was rejected. | int |
| 301 | QuoteResponseLevel | N | Level of Response requested from receiver of quote messages. Is echoed back to the counterparty.<br>**OMX Comment:** Field presence in message and default value subject to marketplace rules. | int |
| 537 | QuoteType | N | Type of Quote | int |
| 298 | QuoteCancelType | N | For use in response to quote cancel | int |
| | **Parties** | N | Insert here the set of "Parties" (firm identification) fields defined in "Common Components of Application Messages" | |
| | **QuotSetAckGrp** | N | The number of sets of quotes in the message | |
| | **StandardTrailer** | Y | | |

# 8.4 Component Blocks (Continuous Quote Specific)

For components that are not specific for Continuous Quotes, please see Section 19 on page 261.

## 8.4.1 Components

None

## 8.4.2 Implicit Components

### 8.4.2.1 QuotCxlEntriesGrp

*Table 100: QuotCxlEntriesGrp*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 295 | NoQuoteEntries | N | The number of securities (instruments) whose quotes are to be canceled<br>Not required when cancelling all quotes. | NumInGroup |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | **Instrument** | N | Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages"<br>**OMX Comment:** Messages shall use identifier fields only. | |
| | **UndInstrmtGrp** | N | | |

### 8.4.2.2 QuotEntryAckGrp

*Table 101: QuotEntryAckGrp*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 295 | NoQuoteEntries | N | The number of quotes for this Symbol (QuoteSet) that follow in this message. | NumInGroup |
| >299 | QuoteEntryID | N | Uniquely identifies the quote across the complete set of all quotes for a given quote provider. First field in repeating group. Required if NoQuoteEntries > 0.<br>**OMX Comment:** FIX 5.0 SP1 | String |
| | **Instrument** | N | Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages" | |
| >132 | BidPx | N | If F/X quote, should be the "all-in" rate (spot rate adjusted for forward points). Note that either BidPx, OfferPx or both must be specified.<br>**OMX Comment:** In a pure ack to a Quote, specifying the bid price is subject to bilateral agreement | Price |
| >133 | OfferPx | N | If F/X quote, should be the "all-in" rate (spot rate adjusted for forward points). Note that either BidPx, OfferPx or both must be specified.<br>**OMX Comment:** In a pure ack to a Quote, specifying the offer price is subject to bilateral agreement | Price |
| >134 | BidSize | N | | Qty |
| >135 | OfferSize | N | | Qty |
| >60 | TransactTime | N | | UTCTimestamp |
| 1167 | QuoteEntryStatus | N | **OMX Comment:** FIX 5.0 SP1 | int |
| >368 | QuoteEntryRejectReason | N | Reason Quote Entry was rejected. | int |

### 8.4.2.3 QuotEntryGrp

*Table 102: QuotEntryGrp*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 295 | NoQuoteEntries | Y | The number of quotes for this Symbol (instrument) (QuoteSet) that follow in this message. ** Nested Repeating Group follows ** | NumInGroup |
| >299 | QuoteEntryID | Y | Uniquely identifies the quote across the complete set of all quotes for a given quote provider.<br>**OMX Comment:** Must be unique per quote issuer and security. The value "1" is recommended in cases where a single quote is allowed.<br>FIX 5.0 SP1 | String |
| | **Instrument** | N | Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages" | |
| >132 | BidPx | N | If F/X quote, should be the "all-in" rate (spot rate adjusted for forward points). Note that either BidPx, OfferPx or both must be specified. | Price |
| >133 | OfferPx | N | If F/X quote, should be the "all-in" rate (spot rate adjusted for forward points). Note that either BidPx, OfferPx or both must be specified. | Price |
| >134 | BidSize | N | | Qty |
| >135 | OfferSize | N | | Qty |
| >60 | TransactTime | N | | UTCTimestamp |

### 8.4.2.4 QuotSetAckGrp

*Table 103: QuotSetAckGrp*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 296 | NoQuoteSets | N | The number of sets of quotes in the message | NumInGroup |
| >302 | QuoteSetID | N | First field in repeating group. Required if NoQuoteSets > 0 | String |
| | **UnderlyingInstrument** | N | Insert here the set of "UnderlyingInstrument" (underlying symbology) fields defined in "Common Components of Application Messages" Required if NoQuoteSets > 0 | |
| >304 | TotNoQuoteEntries | N | Total number of quotes for the quote set across all messages. Should be the sum of all NoQuoteEntries in each message that has repeating quotes that are part of the same quote set.<br>Required if NoQuoteEntries > 0<br>**OMX Comment:** Field is required when LastFragment (893) is used, otherwise not relevant. | int |
| >1168 | TotNoCxlQuotes | N | Total number of quotes canceled for the quote set across all messages. | int |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | | | **OMX Comment:** FIX 5.0 SP1 | |
| >1169 | TotNoAccQuotes | N | Total number of quotes accepted for the quote set across all messages.<br>**OMX Comment:** FIX 5.0 SP1 | int |
| >1170 | TotNoRejQuotes | N | Total number of quotes rejected for the quote set across all messages.<br>**OMX Comment:** FIX 5.0 SP1 | int |
| | **QuotEntryAckGrp** | N | | |

#### 8.4.2.5 QuotSetGrp

*Table 104: QuotSetGrp*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 296 | NoQuoteSets | Y | The number of sets of quotes in the message | NumInGroup |
| >302 | QuoteSetID | Y | Sequential number for the Quote Set. For a given QuoteID – assumed to start at 1.<br>Must be the first field in the repeating group. | String |
| | **UnderlyingInstrument** | N | Insert here the set of "UnderlyingInstrument" (underlying symbology) fields defined in "Common Components of Application Messages" | |
| >304 | TotNoQuoteEntries | Y | Total number of quotes for the quote set across all messages. Should be the sum of all NoQuoteEntries in each message that has repeating quotes that are part of the same quote set. | int |
| | **QuotEntryGrp** | Y | | |

## 8.5 Workflows

### 8.5.1 Introduction

The following rules are used as the basis for the workflows:

1. A Mass Quote always results in a Mass Quote Acknowledgement unless QuoteResponseLevel = 0 (No Acknowledgement) has been specified. A Mass Quote should not result in multiple Quote Status Reports in response.
   - The only exception to this rule occurs if restatements are needed due to automatic quantity refreshes or something similar. In this case, the QuoteID of that Quote Status Report would carry the QuoteEntryID of the previously submitted Mass Quote.
2. A Quote Cancel can result in a Mass Quote Acknowledgement under the following conditions:
   - Multiple quotes are affected, i.e. the QuoteCancelType is set to 1 (Cancel for Symbol[s]), 2 (Cancel for Security Type[s]), 3 (Cancel for Underlying Symbol) or 4 (Cancel all Quotes)

- QuoteResponseLevel has been set to 2 (Acknowledge each quote messages)

If both conditions are not met then the Quote Status Report should be used. An exception to the rule would be a bilateral agreement to always do one or the other.

3. A Quote Status Request can result in a Mass Quote Acknowledgement under the following conditions:

   - Multiple quotes are affected. This means that QuoteID should not be provided and <UndInstrmtGrp> or other filters are specified, meaning e.g. that all strikes in a series should be returned. Since this is a query it is assumed that any qualified quote will be reported.

   If the condition is not met then the Quote Status Report should be used. An exception to the rule would be a bilateral agreement to always do one or the other.

The below table defines what messages can be used to relay request responses and unsolicited actions back to the quote issuer. The table also shows how identifiers are mapped from incoming to outgoing messages.

*Table 105: Quote Response Messages*

| Incoming Request | Action | Outgoing Response | Comment |
|---|---|---|---|
| (Single) **Quote** | New | **Quote Status Report** | QSR.QuoteID := Q.QuoteID<br>QSR.QuoteReportID := Q.QuoteReportID |
| **Mass Quote** | New | **Mass Quote Acknowledgement** | MQA.QuoteID := MQ.QuoteID<br>MQA.QuoteEntryID := MQ.QuoteEntryID |
| (Single) **Quote** | Update | **Quote Status Report** | QSR.QuoteID := Q.QuoteID<br>QSR.QuoteReportID := Q.QuoteReportID |
| **Mass Quote** | Update | **Mass Quote Acknowledgement** | MQA.QuoteID := MQ.QuoteID<br>MQA.QuoteEntryID := MQ.QuoteEntryID |
| (Single) **Quote** | Cancel<br>Px/Qty = 0 | **Quote Status Report** | QSR.QuoteID := Q.QuoteID<br>QSR.QuoteReportID := Q.QuoteReportID |
| **Mass Quote** | Cancel<br>Px/Qty = 0 | **Mass Quote Acknowledgement** | MQA.QuoteID := MQ.QuoteID<br>MQA.QuoteEntryID := MQ.QuoteEntryID |
| **Quote Cancel** | Cancel | **Quote Status Report** | Canceling a single quote and (subject to bilateral agreement) when canceling multiple quotes<br>QSR.QuoteID := Q.QuoteID or MQ.QuoteEntryID<br>QSR.QuoteReportID := QC.QuoteReportID |
| | | **Mass Quote Acknowledgement** | Canceling multiple quotes (subject to bilateral agreement)<br>MQA.QuoteID := QC.QuoteReportID<br>MQA.QuoteEntryID := Q.QuoteID or MQ.QuoteEntryID |
| N/A | Unsolicited State Change | **Quote Status Report** | Used for unsolicited replenishment of exhausted quote size (subject to bilateral agreement)<br>QSR.QuoteID := Q.QuoteID or MQ.QuoteEntryID |

| Incoming Request | Action | Outgoing Response | Comment |
|---|---|---|---|
| | | | QSR.QuoteReportID := Q.QuoteReportID or MQ.QuoteID |
| N/A | Fills | **Execution Report**<br>(and **Trade Capture Report**) | SecondaryClOrdID := Q.QuoteID or MQ.QuoteEntryID<br>ClOrdID := Q.QuoteReportID or MQ.QuoteID |

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

## 8.5.2 Continuous Quote Model

### 8.5.2.1 Quoting

The Continuous Quote Model in a tradeable quote market – requires market makers or specialists to maintain market compliant two sided markets.

The following diagram (and the table following) depicts the overall workflow for quote entry messages, detailed in the following chapters.

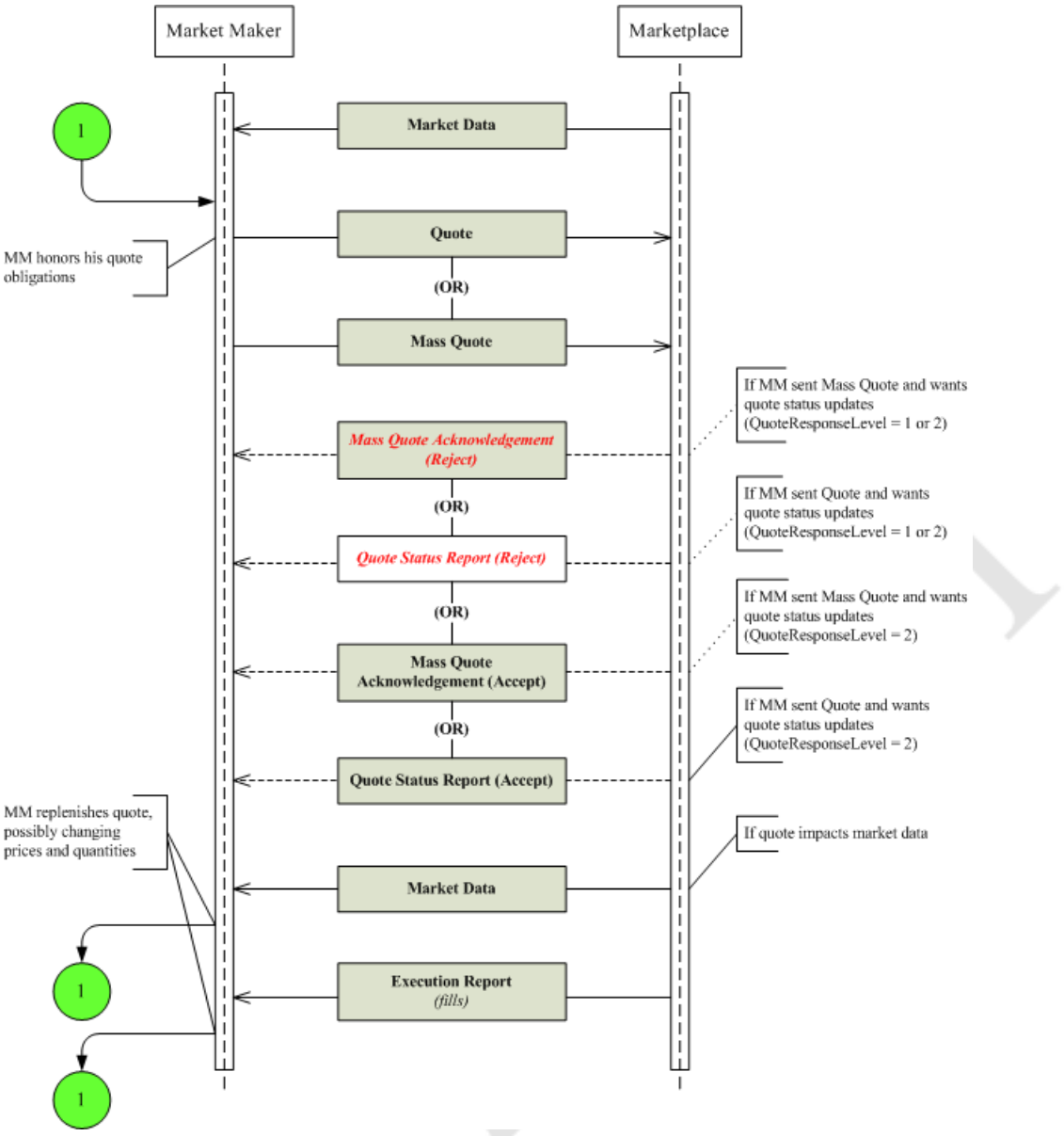


*Figure 3: Mass Quote Workflow*

*Table 106: Continuous Quote Overview*

| Non-Quote Issuer | | Marketplace | | Quote Issuer |
|---|---|---|---|---|
| Uses market data to determine market participation and pricing on orders | <- | Market Data is disseminated<br>(NOTE: This may include the need to transmit expected opening prices based upon current state of the book at the opening) | -> | Uses market data to create subsequent quotes |
| | | Interprets Quotes and applies them to a market<br>Interprets Quote Response Level to determine if quote status should | <- | Quote Market Makers / Specialist are expected to maintain two sided quotes that comply with market re- |

| Non-Quote Issuer | | Marketplace | | Quote Issuer |
|---|---|---|---|---|
| | | be sent back to the quote issuer using a Quote Status Report message with the QuoteStatus field set appropriately<br>Market Data will be generated to report state of the book is changed by the quote | | quired bid-ask spread and minimum quantities |
| | | Optional Quote Status Report | -> | |
| Receives Market Data<br>Will use Market Data to make market participation and pricing decision | <- | If the Quote is valid and has an impact on the market, Market Data is published<br>(NOTE: The process of subscribing for market data is omitted from this example) | -> | Receives Market Data<br>Used to create subsequent quotes |
| Sends New Order – Single | -> | Order is matched against other orders and quotes according to market rules. | | |
| Received Execution Report – reporting New | <- | See order workflows for responses to orders! | | |
| Receipt of Execution Report – Reporting Fill or Partial Fill | <- | If the order is matched against the tradable quote resulting in a trade – Execution Reports are sent to the counterparties of the trade<br>(NOTE: Orders can match against orders and quotes against quotes too - subject to market rules) | -> | Receipt of Market Maker side Execution Report reporting Fill against the previously submitted tradable Quote<br>(Optionally can choose to replenish market or wait) |
| | | Quote is processed as above – market data is generated – an optional Quote Status Report message is generated | <- | Replenishes Quote – possibly changing prices and quantities |

## 8.5.2.2 Single Quote Scenarios

Note that the (Single) Quote message supports:

- Adding individual quotes (if there was no previous quote in the market)
- Updating individual quotes (if there already was a quote in the market)
- Withdrawing (canceling) individual quotes – if the bid / offer prices and sizes are set to zero in the message

### 8.5.2.2.1 Single Quote - No Ack

In the first example a Quote is sent from the quote issuer to the marketplace. The quote has the QuoteResponseLevel = 0 or omitted. The marketplace does not acknowledge the receipt of the quote. If the quote is later hit, resulting in a trade, an Execution Report is sent to the first party. The following diagram depicts the workflow:

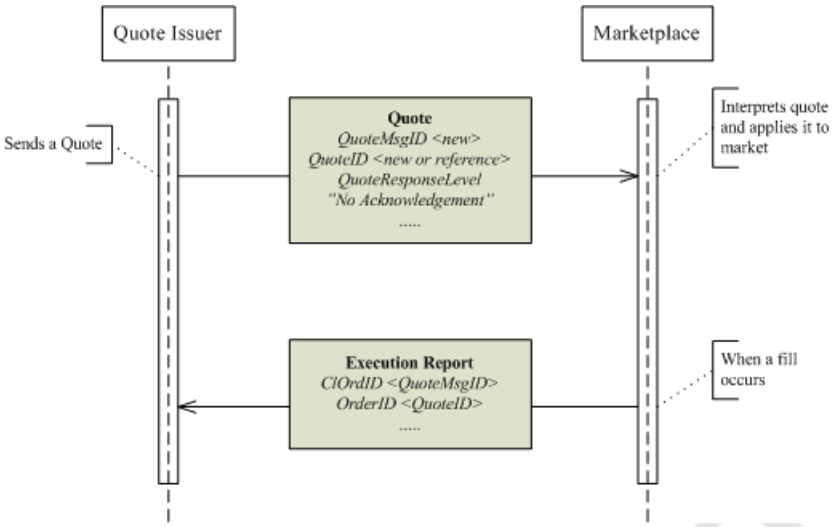


*Figure 4: (Single) Quote with no Ack*

Note that:

- The QuoteMsgID (if used) is renewed for every message sent.
- The QuoteID will contain a new value when a quote is first inserted and that id is then referenced for subsequent updates. The same id can be reused in cases where both sides of the quote are cancelled or exhausted – so a quote issuer can assign a static QuoteID to every quote responsibility (security or options strike).

*Table 107: (Single) Quote with no Ack*

| Quote Issuer | | Marketplace |
|---|---|---|
| Sends Quote message<br>Options:<br>• QuoteResponseLevel is set to 0 or omitted | -> | Interprets Quote and applies it to a market<br>Interprets QuoteResponseLevel – provides response accordingly<br>No response is sent |
| | <- | Execution Report<br>If Quote Results in Trade |

#### 8.5.2.2.2 Single Quote - Negative Ack

In the second example a Quote is again sent from the quote issuer to the marketplace. The quote has the QuoteResponseLevel = 1. The marketplace only acknowledges the quote if there is an error. If the marketplace encounters an error while processing the quote, a Quote Status Report message is sent with the QuoteRejectReason set to the error encountered.

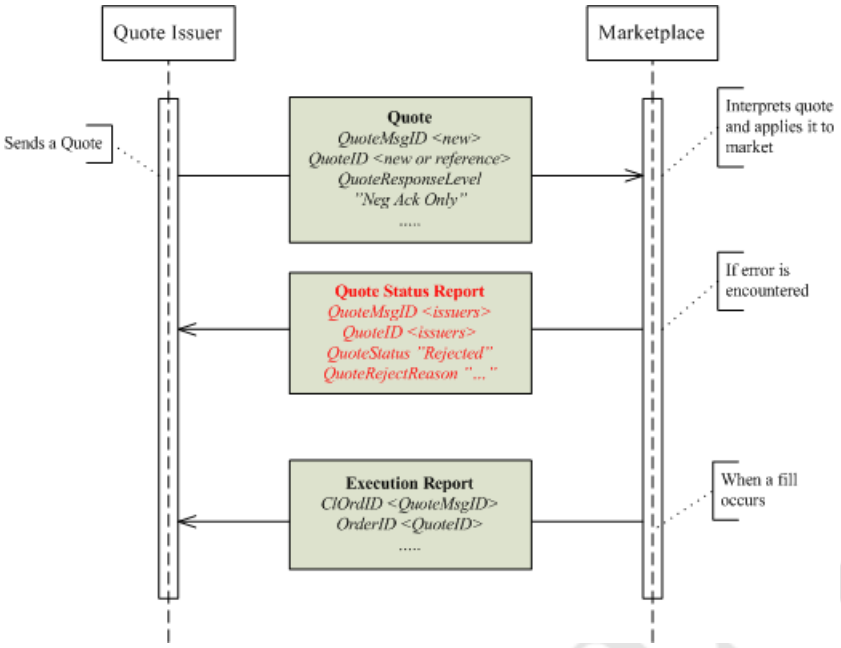


*Figure 5: (Single) Quote with Negative Ack only*

*Table 108: (Single) Quote with Negative Ack only*

| Quote Issuer | | Marketplace |
|---|---|---|
| Sends Quote message<br>Options:<br>• QuoteResponseLevel is set to 1 | -> | Interprets Quote and applies it to a market if correct |
| Interprets Quote Status Report<br>If error – then send revised quote | <- | Sends Quote Status Report - if an error is encountered |
| Sends Quote message | -> | Interprets Quote and applies it to a market |

#### 8.5.2.2.3 Single Quote - Ack All

In the third example a (Single) Quote is sent from the quote issuer to the marketplace. The quote has the QuoteResponseLevel = 2. The marketplace acknowledges each quote.

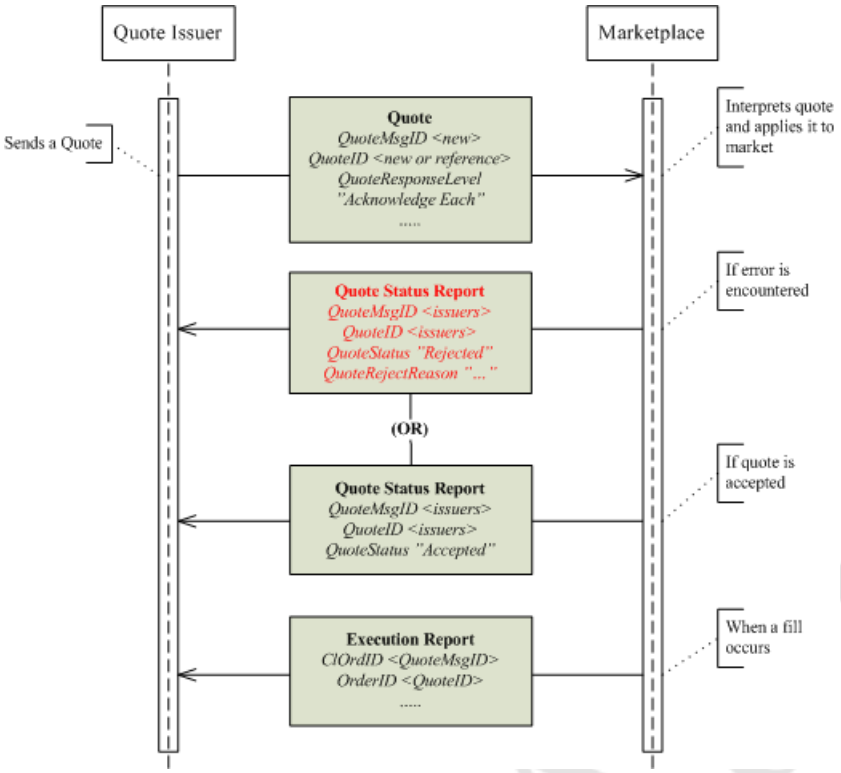


*Figure 6: (Single) Quote with Ack All*

*Table 109: (Single) Quote with Ack All*

| Quote Issuer | | Marketplace |
|---|---|---|
| Sends Quote message<br>Options:<br>• QuoteResponseLevel is set to 2 | -> | Interprets Quote and applies it to a market if correct |
| Interprets Quote Status Report | <- | Sends Quote Status Report |

#### 8.5.2.2.4 Cancel Single Quote using Quote Cancel

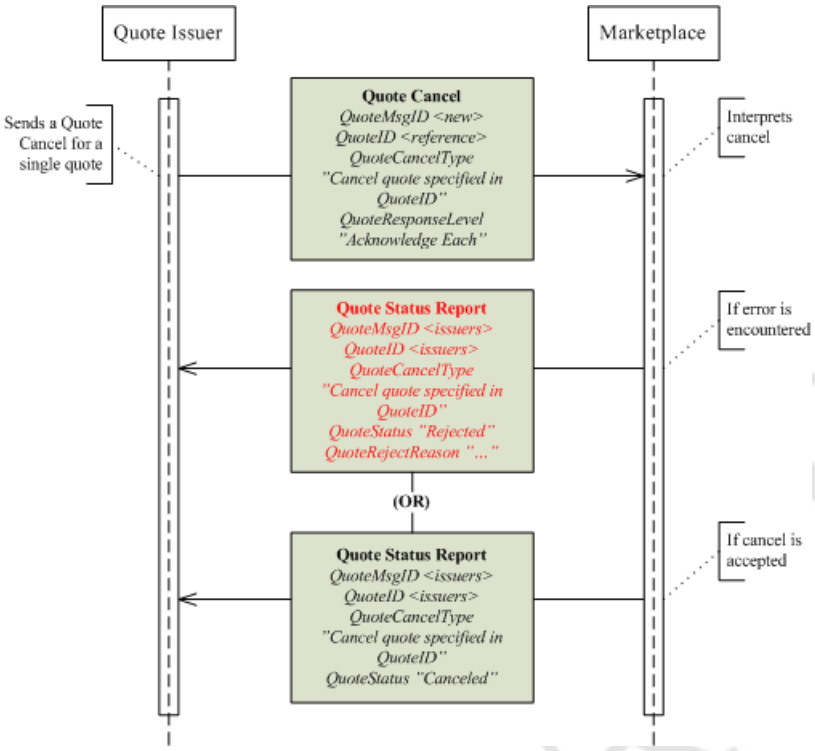


*Figure 7: Cancel (Single) Quote using Quote Cancel*

*Table 110: Cancel (Single) Quote using Quote Cancel*

| Quote Issuer | | Marketplace |
|---|---|---|
| Sends Quote Cancel message<br>Options:<br>• QuoteCancelType = 5 (Cancel quote specified in QuoteID) | -> | Interprets Quote Cancel message and cancels quotes. |
| Interprets Quote Status Report | <- | Sends Quote Status Report |

#### 8.5.2.2.5 Unsolicited Restatement of Single Quote

Some marketplaces, when the quote size is exhausted, support the automatic replenishment with a pre-defined quantity (and moving the price). In such cases a restatement of the quote is appropriate.

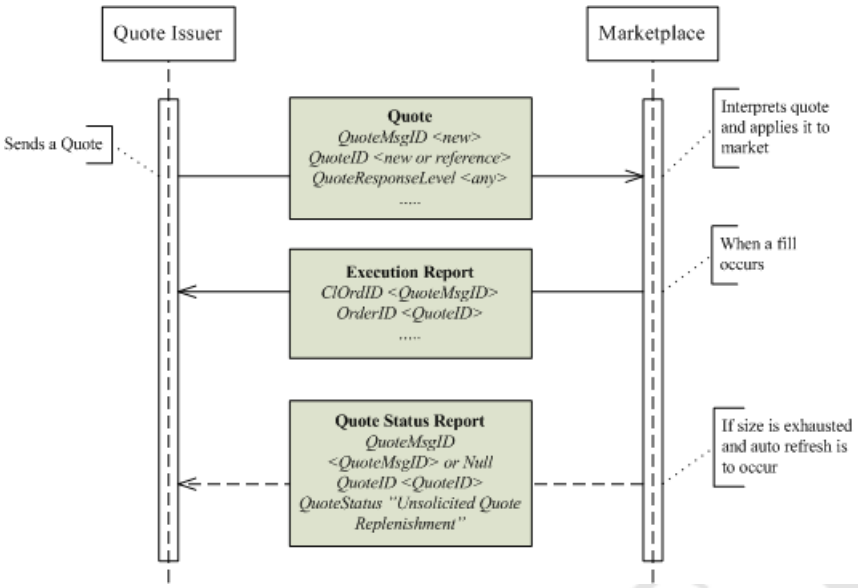


*Figure 8: Unsolicited Restatement of a (Single) Quote*

*Table 111: Unsolicited Restatement of a (Single) Quote*

| Quote Issuer | | Marketplace |
|---|---|---|
| Sends Quote message | -> | Interprets Quote and applies it to a market |
| Interprets Execution Report | <- | Execution Report on fill |
| Interprets Quote Status Report | <- | A Quote Status Report is issued when a side is exhausted and subsequently restated |

#### 8.5.2.2.6 Query for Single Quote

*This workflow is currently not supported!*

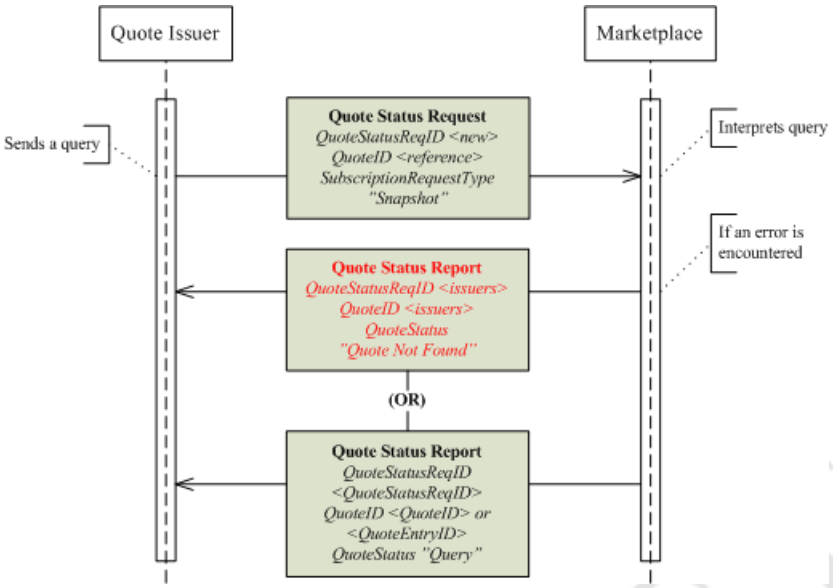


*Figure 9: Querying for (Single) Quote Status*

*Table 112: Querying for (Single) Quote Status*

| Quote Issuer | | Marketplace |
|---|---|---|
| Quote Status Request<br><br>Contains the QuoteID of a previously submitted (Single) Quote. | -> | Accepts Quote Status Request |
| Accepts Quote and updates trading system | <- | Sends Quote Status Report messages with the QuoteStatus field set, bid and ask prices, and quantities for each quote belonging to the request issuer that meet the criteria in the request.<br><br>If there is a current quote in the market – the Quote Status Report in response to a Quote Status Request should be sent with a QuoteStatus of “Query”.<br><br>The Quote Status Report message can also contain a QuoteStatus of “Quote Not Found” if no quote currently exists. |

### 8.5.2.3 Mass Quote Message Scenarios

Note that the Mass Quote message supports:

- Adding individual quotes (if there was no previous quote in the market)
- Updating individual quotes (if there already was a quote in the market)
- Withdrawing (canceling) individual quotes – if the bid / offer prices and sizes are set to zero in the message

#### 8.5.2.3.1 Mass Quote - No Ack

In the first example a Mass Quote is sent from the quote issuer to the marketplace. The quote has the QuoteResponseLevel = 0 or omitted. The marketplace does not acknowledge the receipt of the quote. If the quote is later hit, resulting in a trade, an Execution Report is sent to the first party.

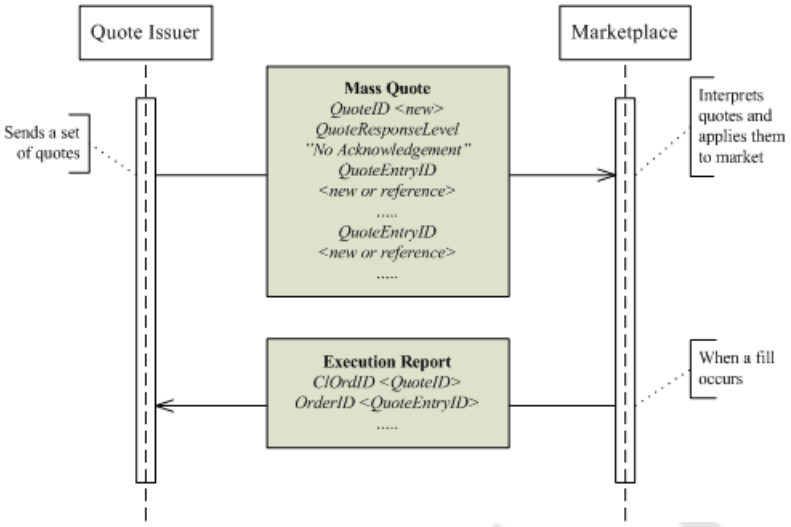


*Figure 10: Mass Quote with no Ack*

*Table 113: Mass Quote with no Ack*

| Quote Issuer | | Marketplace |
|---|---|---|
| Mass Quote message<br>Options:<br>• One or more sets of quotes<br>• QuoteResponseLevel is set to 0 or omitted. | -> | Interprets quotes applies them to a market<br>Interprets Response Level – provides response accordingly<br>No response is sent |
| | <- | Execution Report<br>Quote Results in Trade |

#### 8.5.2.3.2 Mass Quote - Negative Ack

In the second example a Mass Quote is sent from the quote issuer to the marketplace. The quote has the QuoteResponseLevel = 1. The marketplace only acknowledges the quote if there is an error. If the marketplace encounters an error while processing the quote, a Mass Quote Acknowledgement message is sent with the QuoteRejectReason set to the error encountered.

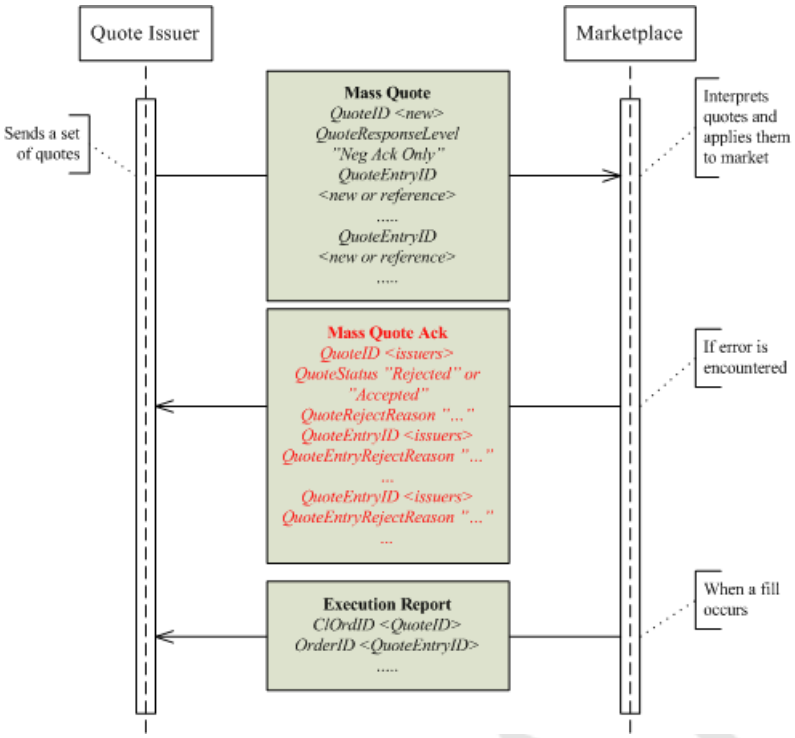


*Figure 11: Mass Quote with Negative Ack only*

*Table 114: Mass Quote with Negative Ack only*

| Quote Issuer | | Marketplace |
|---|---|---|
| Mass Quote message<br>Options:<br>• One or more sets of quotes<br>• QuoteResponseLevel is set to 1 | -> | Interprets quotes applies them to a market |
| Interprets Mass Quote Acknowledgement<br>If error – then send revised quote | <- | Mass Quote Acknowledgement<br>If an error is encountered |
| Mass Quote message | -> | Interprets quotes applies them to a market |

### 8.5.2.3.3 Mass Quote - Ack All

In the third example a Mass Quote is sent from the quote issuer to the marketplace. The quote has the QuoteResponseLevel = 2. The marketplace acknowledges all quote entries with a single Mass Quote message.

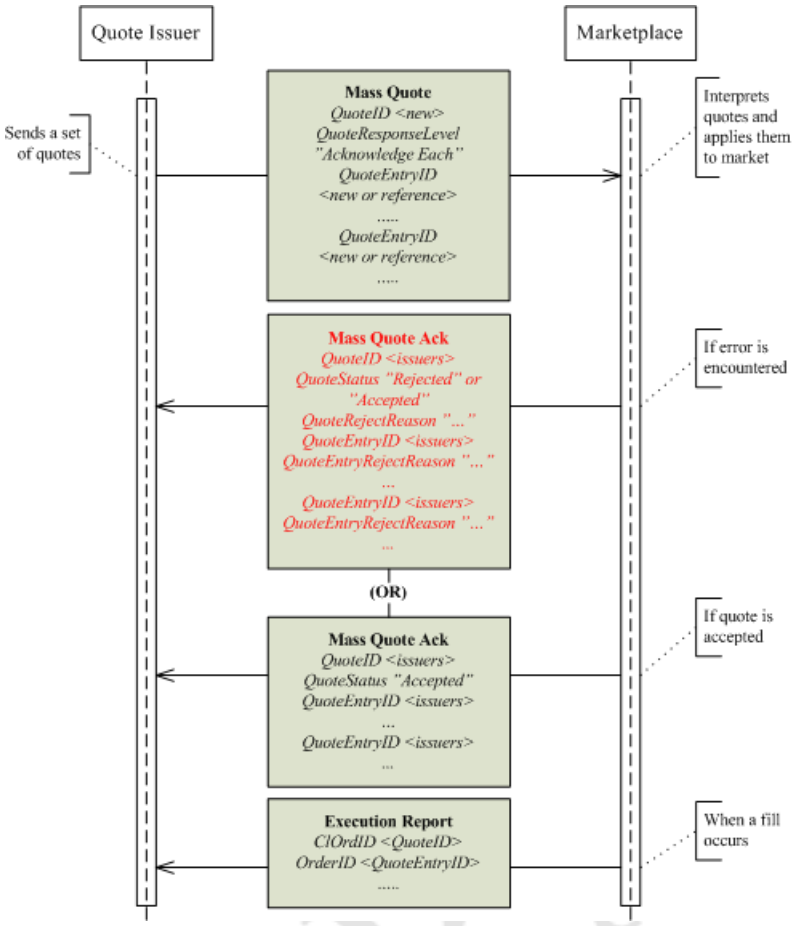


*Figure 12: Mass Quote with Ack All*

*Table 115: Mass Quote with Ack All*

| Quote Issuer | | Marketplace |
|---|---|---|
| Mass Quote message<br>Options:<br>• One or more sets of quotes<br>• Set QuoteResponseLevel is set to 2 | -> | Interprets quotes applies them to a market |
| Interprets Mass Quote Acknowledgement | <- | Mass Quote Acknowledgement |

#### 8.5.2.3.4 Cancel Mass Quote using Quote Cancel

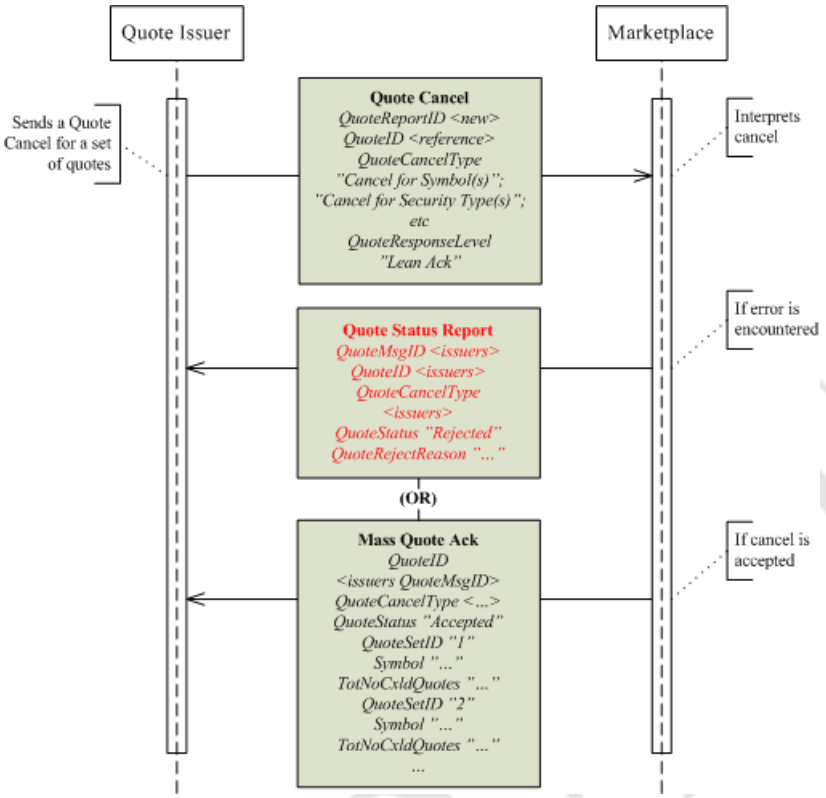


*Figure 13: Cancel Mass Quote using Quote Cancel*

*Table 116: Cancel Mass Quote using Quote Cancel*

| Quote Issuer | | Marketplace |
|---|---|---|
| Sends Quote Cancel message<br>Options:<br>• QuoteCancelType = 1, 2, 3 or 4 (identifying a set of quotes) | -> | Interprets Quote Cancel message and cancels quotes. |
| Interprets Mass Quote Acknowledgement | <- | Sends Mass Quote Acknowledgement |

#### 8.5.2.3.5 Unsolicited Restatement of Mass Quote Entry

Some marketplaces, when the quote size is exhausted, support the automatic replenishment with a pre-defined quantity (and moving the price). In such cases a restatement of the quote is appropriate.

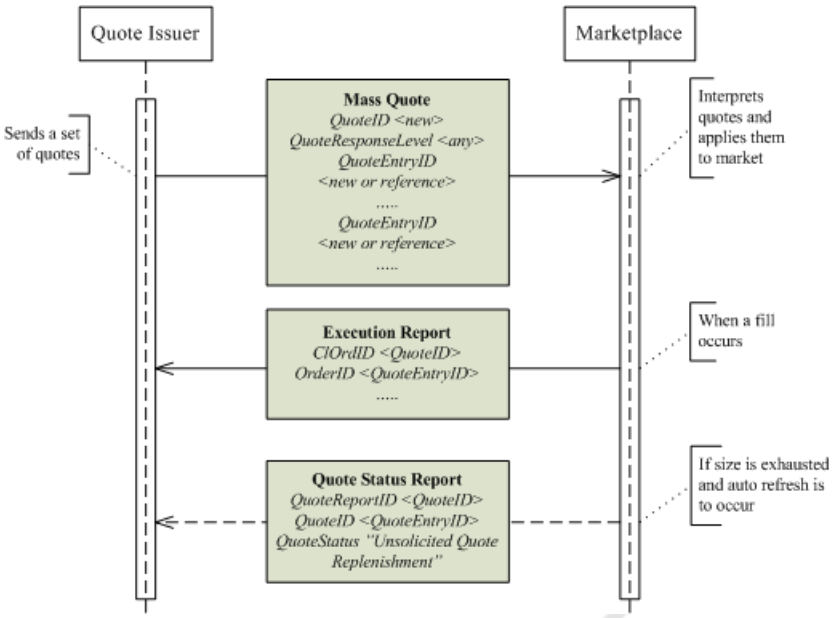


*Figure 14: Unsolicited Restatement of a (Single) Quote*

*Table 117: Unsolicited Restatement of a (Single) Quote*

| Quote Issuer | | Marketplace |
|---|---|---|
| Sends Mass Quote message | -> | Interprets Mass Quote and applies it to a market |
| Interprets Execution Report | <- | Execution Report on fill |
| Interprets Quote Status Report | <- | A Quote Status Report is issued when a side of a single quote is exhausted and subsequently restated |

#### 8.5.2.3.6 Query for Mass Quote

*This workflow is currently not supported!*

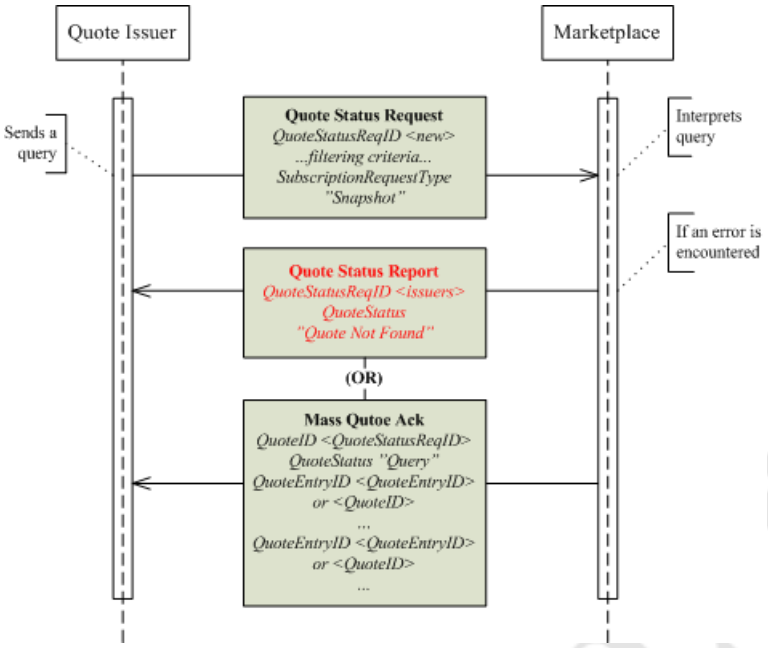


*Figure 15: Querying for Mass Quote Status*

*Table 118: Querying for Mass Quote Status*

| Quote Issuer | | Marketplace |
|---|---|---|
| Quote Status Request<br><br>Contains information on the securities for which the quote status request is being issued | -> | Accepts Quote Status Request |
| Accepts Mass Quote Acknowledgement and updates trading system | <- | Sends a Mass Quote Acknowledgement message with the QuoteStatus field set, bid and ask prices, and quantities for each quote belonging to the request issuer and meeting the criteria in the request.<br><br>If there is a current quote in the market – the Mass Quote Acknowledgement in response to a Quote Status Request should be sent with a QuoteStatus of “Query”.<br><br>The Mass Quote Acknowledgement message can also contain a QuoteStatus of “Quote Not Found” if no quote currently exists. |

# 9 Quote Negotiation

## 9.1 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 119: Business Messages*

| In/Out | Message Name | Comment |
|---|---|---|
| In | **Quote Request** | Used to request a public or private Quote |
| In | **Quote** | Used to submit a public or private Quote |
| In | **Quote Response** | Used to respond to a private Quote |
| Out | **Execution Report** | |
| Out | **Quote Request Reject** | Used to reject a Quote Request |
| Out | **Quote Status Report** | Used to reject Quote and Quote Response messages (and optionally to acknowledge them). Also used to report state changes in the quote, when such can not be derived from an Execution Report fill-notice. |

## 9.2 Quote Negotiation

### 9.2.1 Introduction

The Quote message is used by market makers and other actors with similar responsibilities to send quotes into a market. As described in this chapter, the same message type is also used in Quote Negotiation workflows. The Quote message, as described in this part of the document, is used to send solicited quotes in response to Quote Requests. For a description of the use of unsolicited quotes and general rules around the Quote message, please see Section 8 on page 111.

The Quote Negotiation model used is a Three-Party Matching Model where the marketplace determines that matching interest is available and automatically creates the trade. An alternative model is to finalize the negotiation by using the Privately Negotiated Trade workflows. Standard FIX also supports other negotiation models, those are normally two-party ones, may vary by asset class and do not involve a marketplace.

### 9.2.2 Main Workflow

#### 9.2.2.1 The Quote Request

The quote negotiation workflow starts with the user submitting a Quote Request message. There are two main types of requests:

- **Requests for Public Quote (Private Quote [1171] = N)**

The response to the Quote Request is a Quote in the market. I.e. the Quote is published in the same way as an unsolicited quote over market data and is available for all eligible market participants. Please see Section 8 on page 111 for further details.

- **Request for Private Quote (PrivateQuote [1171] = Y)**

  The response to the Quote Request is a Quote directed to the participant issuing the Quote Request. The Quote message follows the same rules as described in Section 8 on page 111 but is marked as a Private Quote and thereby only published to the initiator. This process continues below.

The request can also specify what actors should receive the Quote Request. The following options are available.

- **All Market Participants (RespondentType [1172] = 1)**

  The marketplace forwards the Quote Request to all eligible market participants.

- **Specified Market Participants only (RespondentType [1172] = 2)**

  The marketplace forwards the Quote Request to market participants specified in the <RootParties> component block (see below)

- **All Market Makers (RespondentType [1172] = 3)**

  The marketplace forwards the Quote Request to all registered market makers.

- **Primary Market Makers Only (RespondentType [1172] = 4)**

  The marketplace forwards the message to all registered primary market makers / specialists.

When the user chooses the option to send a Quote Request to specified market participants, he also must specify those in the <RootParties> component block of the message.

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

- RootPartyID (1117) = the identifier of the participant firm
- RootPartyIDSource (1118):
  - C = Generally accepted market participant identifier
  - D = Proprietary / Custom code
- RootPartyRole (1119) = see below

The following table specifies the relevant party roles:

*Table 120: Supported RootPartyRole (1119) values*

| | Business Role | Party Role | Comment |
|---|---|---|---|
| The respondent (Market Maker or other eligible quote party) | Firm | RootPartyRole (1119) = 17 (Contra Firm) | RootPartyIDSource = D. Relevant when the market-place issues an id-number for those actors. Applicable e.g. when the actor has a connection to the market-place.<br>RootPartyIDSource = C. Relevant when the market-place supports mnemonic symbolic names for those actors. Applicable e.g. when the actor has a connection to the marketplace. |

| | Business Role | Party Role | Comment |
|---|---|---|---|
| In cases where a specific unit or trader needs to be assigned, the RootSubParties component is also used | Unit | RootPartySubID (1121)<br>RootPartySubIDType (1122) = 25 (Location desk) | Currently not supported |
| | Trader | RootPartySubID (1121)<br>RootPartySubIDType (1122) = 2 (Person) | Currently not supported |

#### 9.2.2.2 The Quote

The respondent replies with a quote. In the case of a Public quote, the workflow continues in the Continuous Quoting workflow (see Section 8 on page 111). In the case of a Private Quote, the workflow continues below. An alternative when the terms of the Quote Request is sufficiently detailed is that the respondent issues a Trade Capture Report in which case the workflow continues as described in Section 12 on page 171.

Note that the Quote Issuer can cancel his quote as defined in the above mentioned workflow.

#### 9.2.2.3 The Quote Response

On receipt of the Quote message, the initiator has the following options:

- Hit the Quote through sending a Quote Response with QuoteRespType (694) = 1 (Hit/Lift). The response should also contain matching properties as Price and Quantity.
- Hit the Quote through issuing a Trade Capture Report in which case the workflow continues as described in Section 12 on page 171
- Counter the Quote through sending a Quote Response with QuoteRespType (694) = 2 (Counter). The response should contain counter properties as Price and Quantity.
- Pass the Quote through sending a Quote Response with QuoteRespType (694) = 6 (Pass)
- Do nothing and let the Quote time out.

#### 9.2.2.4 Fills

The marketplace matches Quotes and Quote Responses. If the QuoteRespType (694) of the Quote Response = 1 (Hit/Lift) and matches the Quote, a fill is created. Matching conditions include:

- Instrument
- Side
- Price
- Quantity - the hit quantity must not exceed the quoted quantity
- Expiry time - neither of the messages are allowed to pass their respective expiry time

When the quote negotiation results in a Fill, an Execution Report will be sentt containing details about the fill. In addition, a Trade Capture Report will be produced. The principal differences between the two are described in chapter Section 5.2.2 on page 51.

#### 9.2.2.5 Quote Request Rejects

There is a single message used to reject a Quote Request message:

*Table 121: Quote Reject Messages*

| Reject Message | Direction | Business Message | Comment |
|---|---|---|---|
| **Quote Request Reject** | Out | Quote Request | |

#### 9.2.2.6 Quote Response Rejects

There is a single message used to reject a Quote Response message:

*Table 122: Quote Response RejectMessages*

| Reject Message | Direction | Business Message | Comment |
|---|---|---|---|
| **Quote Status Report** | Out | Quote Response | |

### 9.2.3 Quote Request Features

#### 9.2.3.1 Quote Negotiation Identifiers

- **Quote Request ID (QuoteReqID, 131)**

  Every Quote Request must be associated with a unique message identifier per FIX session. The ID is echoed back on private quotes.

- **RFQ ID (RFQReqID, 644)**

  Every RFQ subscription request must be associated with a unique message identifier per FIX session. The ID is echoed back on Quote Requests.

- **Quote ID (QuoteID, 117)**

  Every quote must be associated with a unique identifier per FIX session. The ID is echoed back on Quote Responses, Execution Reports (trades) and Trade Capture Reports. In the latter two cases the OrderID field is used. The ID identifies the particular quote. If the quote is a Public Quote, the QuoteID is a static value (according to rules of Section 8 on page 111). If the quote is a Private Quote, users may choose assign a separate value per Quote Request.

- **Quote Message ID (QuoteMsgID, 1166)**

  Every quote submittal shall be associated with a unique message identifier per FIX session. The ID is echoed back on Quote Responses, Execution Reports (trades) and Trade Capture Reports. In the latter two cases the ClOrdID field is used.

- **Quote Response ID (QuoteRespID, 693)**

  Every Quote Response must be associated with a unique message identifier per FIX session. The ID is echoed back on private Quotes, Execution Reports (trades) and Trade Capture Reports. In the latter case the ClOrdID field is used.

#### 9.2.3.2 Minimum Quantity

When a user quotes a price to specific counterparties, he may choose to provide a better price under the condition that a certain size is filled. In a Quote Negotiation situation, parties may need to indicate a minimum execution quantity in order to solicit relevant prices and, in the case of the respondent, avoid getting hit on lower than expected quantity. This is done using the MinQty (110) field.

#### 9.2.3.3 Usage of Trade Capture Reports to finalize a Private Negotiation

As described in previous sections, the Trade Capture Report message (Section 12 on page 171) can be used to finalize the quote negotiation. When this is done the TrdSubType (829) = 97 - "OTC Quote" should be specified in order to enable supervisors to identify that the trade occurs as part of the Quote Negotiation workflow.

#### 9.2.3.4 Accounts, Pre-Allocations and Give-Ups

The initiator of a quote negotiation has the ability to submit pre-allocation details when responding to a quote (the Quote Response message). As in continuous quoting (refer to Section 8 on page 111), the respondent is expected to have such details pre-defined. Please refer to Section 4.5 on page 42 for further details.

## 9.3 Message Details

### 9.3.1 QuoteRequest

*Table 123: QuoteRequest*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | **StandardHeader** | Y | MsgType = R | |
| 131 | QuoteReqID | Y | | String |
| 644 | RFQReqID | N | For tradeable quote model - used to indicate to which RFQ Request this Quote Request is in response. | String |
| 1171 | PrivateQuote | Y (N) | Used to indicate whether a private negotiation is requested or if the response should be public. If field is not provided in message, the model must be bilaterally agreed.<br>**OMX Comment:** FIX 5.0 SP1 | Boolean |
| 1172 | RespondentType | Y (N) | **OMX Comment:** FIX 5.0 SP1 | int |
| 1091 | PreTradeAnonymity | N | **OMX Comment:** FIX 5.0 SP1 | Boolean |
| | **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..<br>**OMX Comment:** FIX 5.0 SP1 | |
| | **QuotReqGrp** | Y | Number of related symbols (instruments) in Request | |
| 58 | Text | N | | String |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | **StandardTrailer** | Y | | |

## 9.3.2 QuoteRequestReject

*Table 124: QuoteRequestReject*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | **StandardHeader** | Y | MsgType = AG | |
| 131 | QuoteReqID | Y | | String |
| 644 | RFQReqID | N | For tradeable quote model - used to indicate to which RFQ Request this Quote Request is in response. | String |
| 658 | QuoteRequestRejectReason | Y | Reason Quote was rejected | int |
| 1171 | PrivateQuote | Y (N) | Used to indicate whether a private negotiation is requested or if the response should be public.<br>**OMX Comment:** FIX 5.0 SP1 | Boolean |
| 1172 | RespondentType | Y (N) | **OMX Comment:** FIX 5.0 SP1 | int |
| 1091 | PreTradeAnonymity | N | **OMX Comment:** FIX 5.0 SP1 | Boolean |
| | **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..<br>**OMX Comment:** FIX 5.0 SP1 | |
| | **QuotReqRjctGrp** | Y | Number of related symbols (instruments) in Request | |
| 58 | Text | N | | String |
| | **StandardTrailer** | Y | | |

## 9.3.3 QuoteResponse

*Table 125: QuoteResponse*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | **StandardHeader** | Y | MsgType = AJ | |
| 693 | QuoteRespID | Y | Unique ID as assigned by the Initiator | String |
| 117 | QuoteID | N | Required only when responding to a Quote. | String |
| 1166 | QuoteMsgID | N | Optionally used when rersponding to a quote<br>**OMX Comment:** FIX 5.0 SP1 | String |
| 694 | QuoteRespType | Y | Type of response this Quote Response is. | int |
| 528 | OrderCapacity | N | | char |
| 529 | OrderRestrictions | N | **OMX Comment:** Currently not in FIX 5.0. OMX requests extension. | MultipleCharValue |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 1091 | PreTradeAnonymity | N | **OMX Comment:** FIX 5.0 SP1 | Boolean |
| | **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"<br>For multilegs supply minimally a value for Symbol (55).<br>**OMX Comment:** Messages shall use identifier fields only. | |
| 54 | Side | N | Required when countering a single instrument quote or "hit/lift" an IOI or Quote. | char |
| | **OrderQtyData** | N | Insert here the set of "OrderQtyData" fields defined in "Common Components of Application Messages"<br>Required when countering a single instrument quote or "hit/lift" an IOI or Quote. | |
| 110 | MinQty | N | **OMX Comment:** FIX 5.0 SP1 | Qty |
| 70 | AllocID | N | **OMX Comment:** Not in standard FIX. OMX requests an extension | String |
| | **PreAllocGrp** | N | **OMX Comment:** Not in standard FIX. OMX requests an extension | |
| 77 | PositionEffect | N | **OMX Comment:** Not in standard FIX. OMX requests an extension | char |
| 62 | ValidUntilTime | N | The time when the quote will expire.<br>Required for FI when the QuoteRespType is 2 (Counter quote) to indicate to the Respondent when the counter offer is valid until. | UTCTimestamp |
| 60 | TransactTime | N | | UTCTimestamp |
| 58 | Text | N | | String |
| 44 | Price | N | | Price |
| | **StandardTrailer** | Y | | |

## 9.4 Component Blocks (Quote Negotiation Specific)

For components that are not specific for Quote Negotiations, please see Section 19 on page 261.

### 9.4.1 Components

None.

## 9.4.2 Implicit Components

### 9.4.2.1 QuotReqGrp

*Table 126: QuotReqGrp*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 146 | NoRelatedSym | Y | Number of related symbols (instruments) in Request<br>**OMX Comment:** Always = 1 | NumInGroup |
| | **Instrument** | Y | Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages" | |
| >537 | QuoteType | N | Type of quote being requested from counterparty or market (e.g. Indicative, Firm, or Restricted Tradeable) Valid values used by FX in the request: 0 = Indicative, 1 = Tradeable; Absence implies a request for an indicative quote. | int |
| >54 | Side | N | If OrdType = "Forex - Swap", should be the side of the future portion of a F/X swap. The absence of a side implies that a two-sided quote is being requested.<br>For single instrument use. FX values, 1 = Buy, 2 = Sell; This is from the perspective of the Initiator. If absent then a two-sided quote is being requested for spot or forward. | char |
| | **OrderQtyData** | N | Required for single instrument quoting. Required for Fixed Income if QuoteType is Tradeable. | |
| >110 | MinQty | N | **OMX Comment:** FIX 5.0 SP1 | Qty |
| >62 | ValidUntilTime | N | Used by the quote initiator to indicate the period of time the resulting Quote must be valid until | UTCTimestamp |
| >126 | ExpireTime | N | The time when Quote Request will expire. | UTCTimestamp |
| >60 | TransactTime | N | Time transaction was entered | UTCTimestamp |
| >44 | Price | N | Quoted or target price | Price |
| | **Parties** | N | | |

### 9.4.2.2 QuotReqRjctGrp

*Table 127: QuotReqRjctGrp*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 146 | NoRelatedSym | Y | Number of related symbols (instruments) in Request<br>**OMX Comment:** Always = 1 | NumInGroup |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | **Instrument** | Y | Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages" | |
| >537 | QuoteType | N | Type of quote being requested from counterparty or market (e.g. Indicative, Firm, or Restricted Tradeable) | int |
| >54 | Side | N | If OrdType = "Forex - Swap", should be the side of the future portion of a F/X swap. The absence of a side implies that a two-sided quote is being requested.<br>Required if specified in Quote Request message. | char |
| | **OrderQtyData** | N | Insert here the set of "OrderQytData" fields defined in "Common Components of Application Messages"<br>Required if component is specified in Quote Request message. | |
| >126 | ExpireTime | N | The time when Quote Request will expire. | UTCTimestamp |
| >60 | TransactTime | N | Time transaction was entered | UTCTimestamp |
| >44 | Price | N | Quoted or target price | Price |
| | **Parties** | N | Insert here the set of "Parties" (firm identification) fields defined in "Common Components of Application Messages" | |

## 9.5 Workflows

### 9.5.1 Introduction

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

The standard FIX Protocol Specification includes workflows similar to the ones defined in this document.

## 9.5.2 Requesting Public Quotes

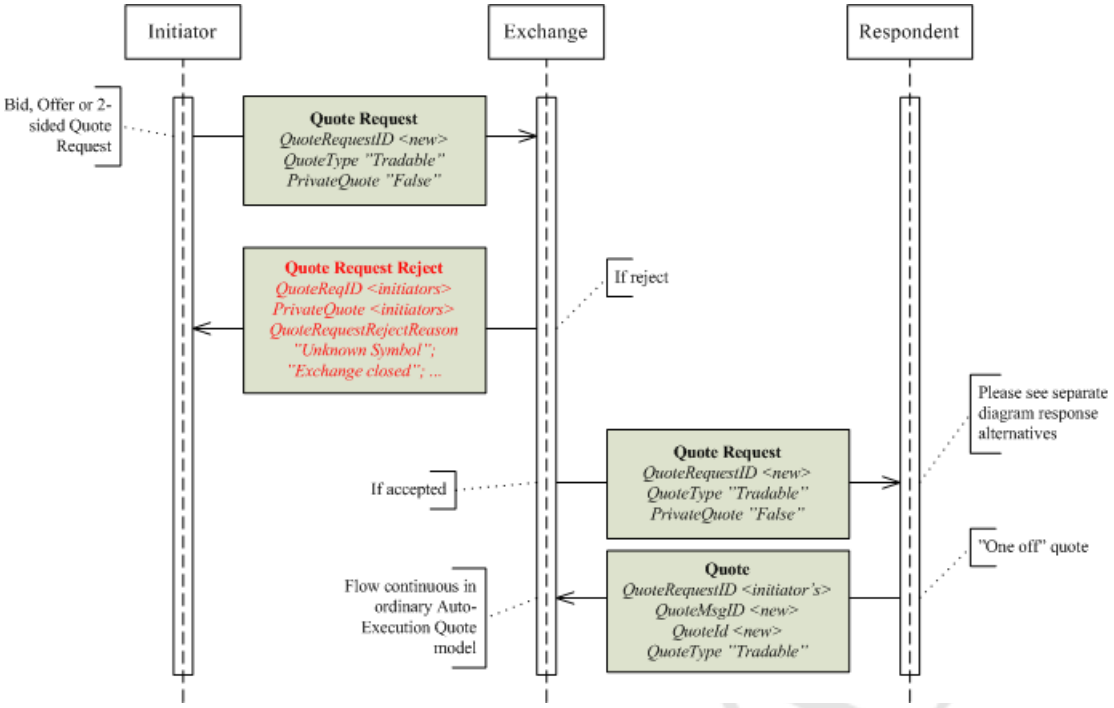


*Figure 16: Request for Public Quote*

The workflow continues according to the Continuous Quoting model, see Section 8.5 on page 126.

## 9.5.3 Requesting Private Quotes

### 9.5.3.1 Plain Vanilla Request for Private Quote

The following workflow shows the plain vanilla case.

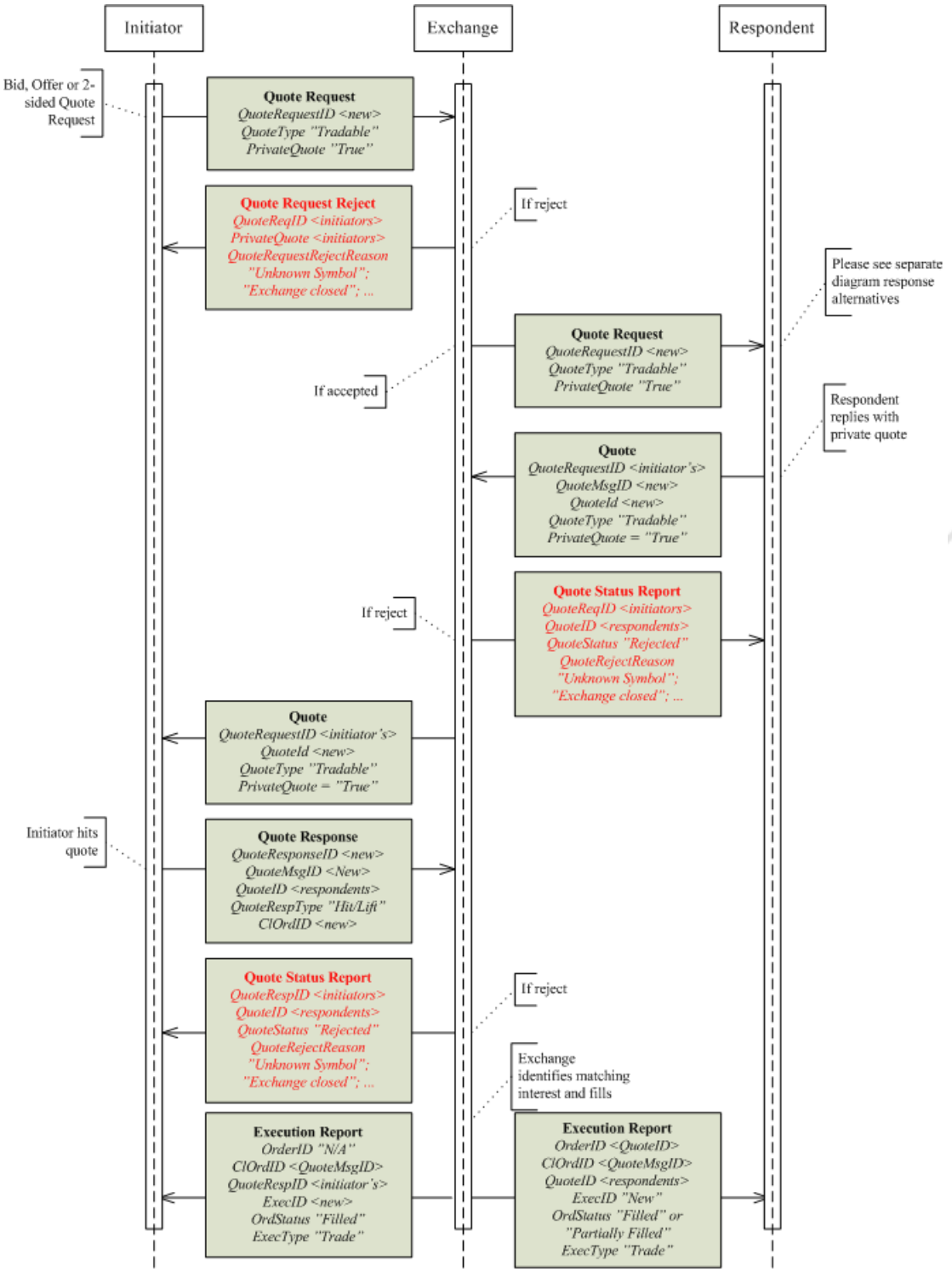


*Figure 17: Request for Private Quote*

Note that the Quote Status Report can be used to report the status of the Quote according to the same principles as described in Section 8 on page 111.

### 9.5.3.2 Respondent Alternatives

The workflow below shows the alternatives available to the respondent on receipt of a Quote Request. Note that Mass Quotes are not supported!

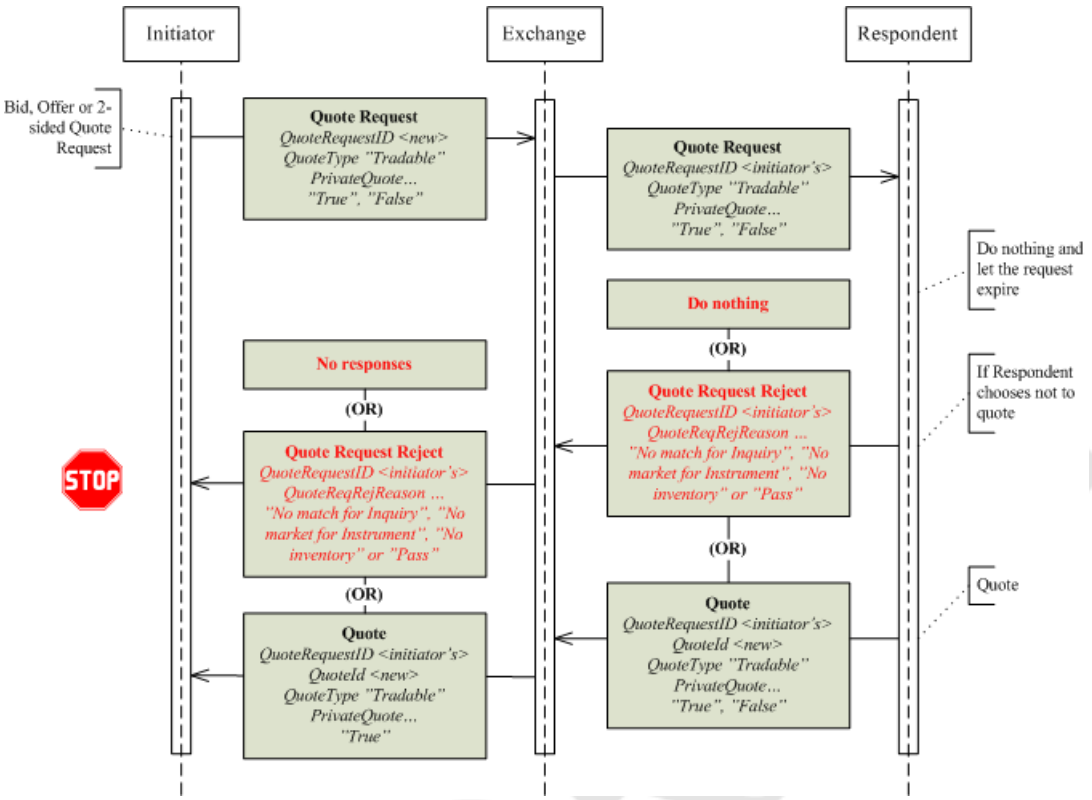


*Figure 18: Respondents alternatives*

### 9.5.3.3 Initiator Alternatives

The following diagram shows the alternatives available to the initiator on receipt of a private quote.

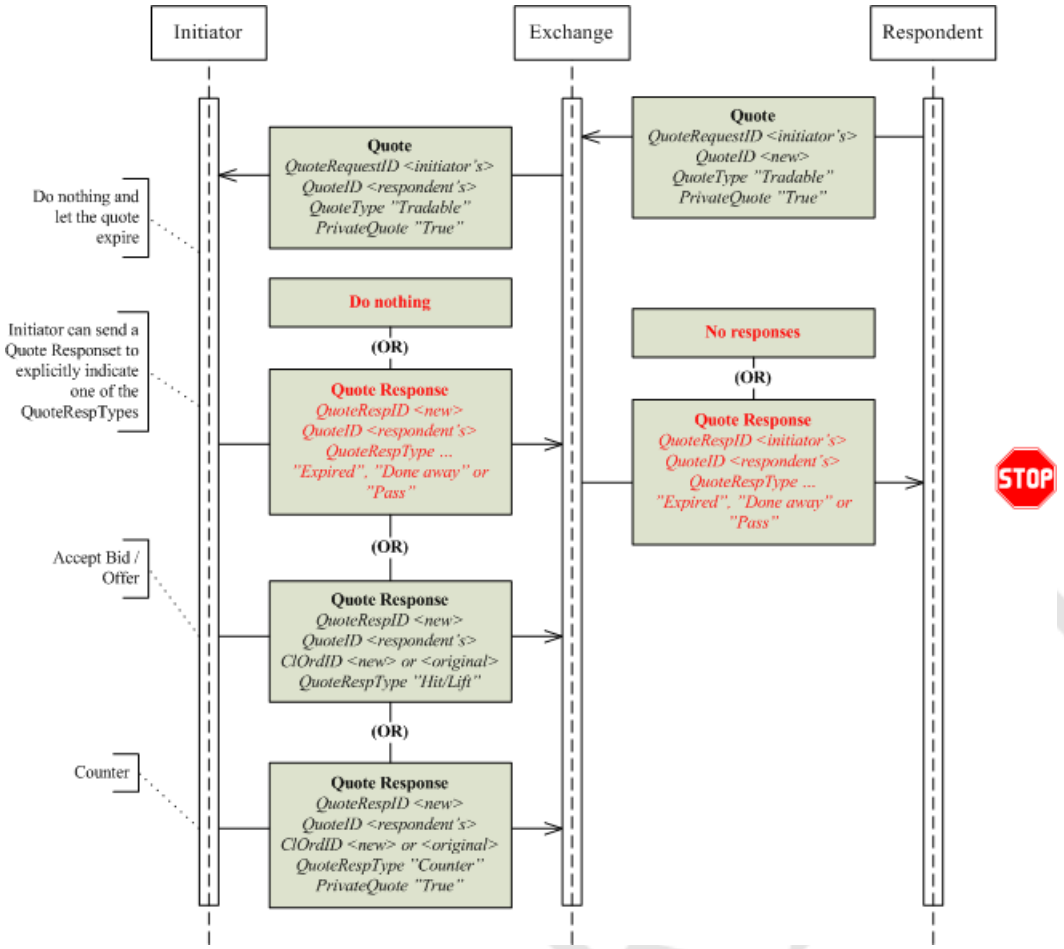


*Figure 19: Initiators alternatives on receipt of Quote*

### 9.5.3.4 Counter a Quote

The following diagram depicts the workflow when the initiator counters a private quote.

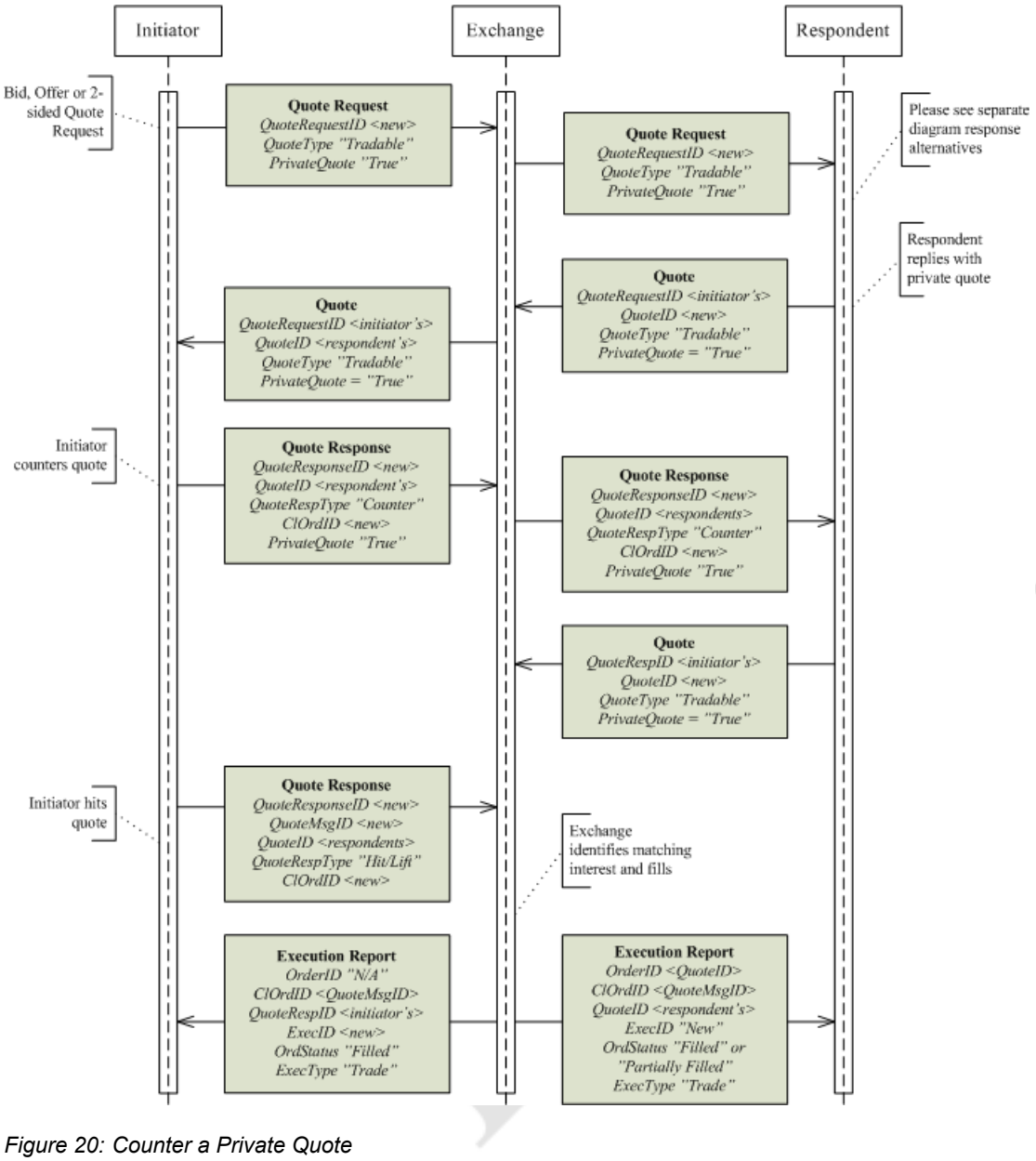


*Figure 20: Counter a Private Quote*

### 9.5.3.5 Respondent Alternatives on Counter Response

The following diagram depicts the respondent's alternatives on receipt of a counter response.

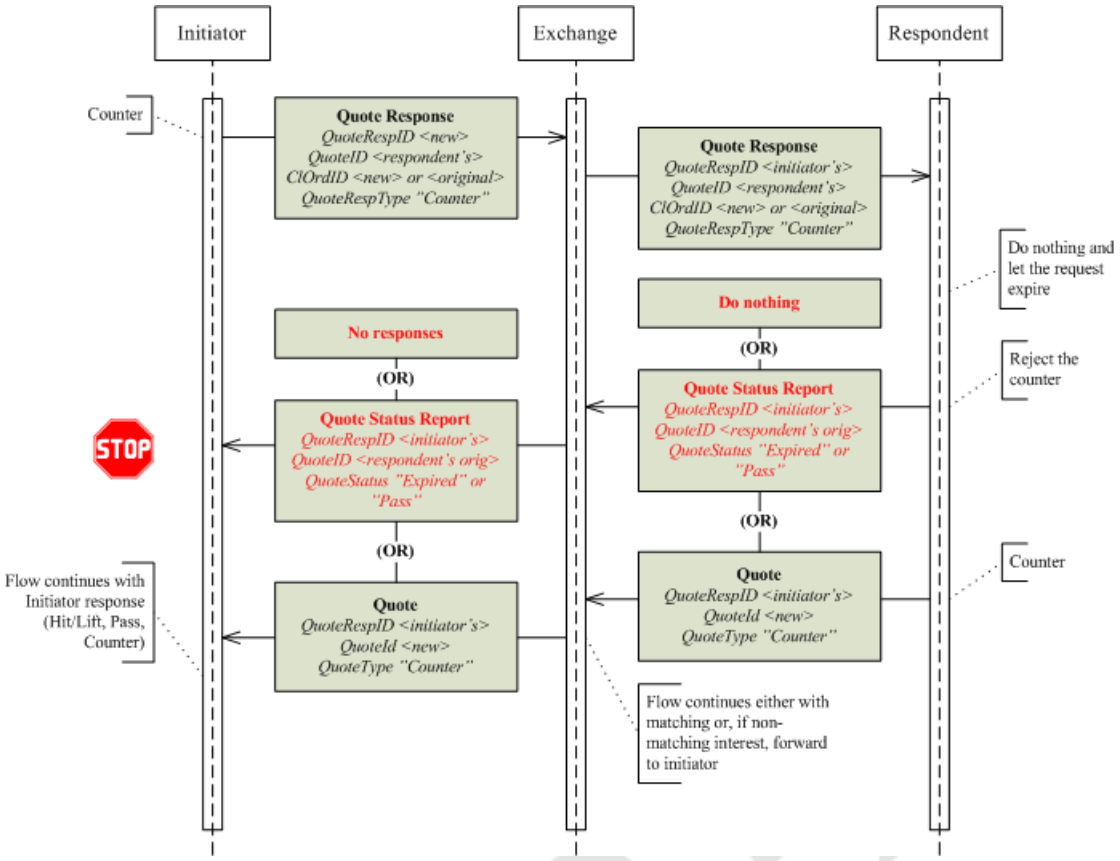


*Figure 21: Respondent's alternatives in Counter Situation*

### 9.5.3.6 Pass a Privete Quote

The following diagram depicts the workflow when the initiator declines a quote.

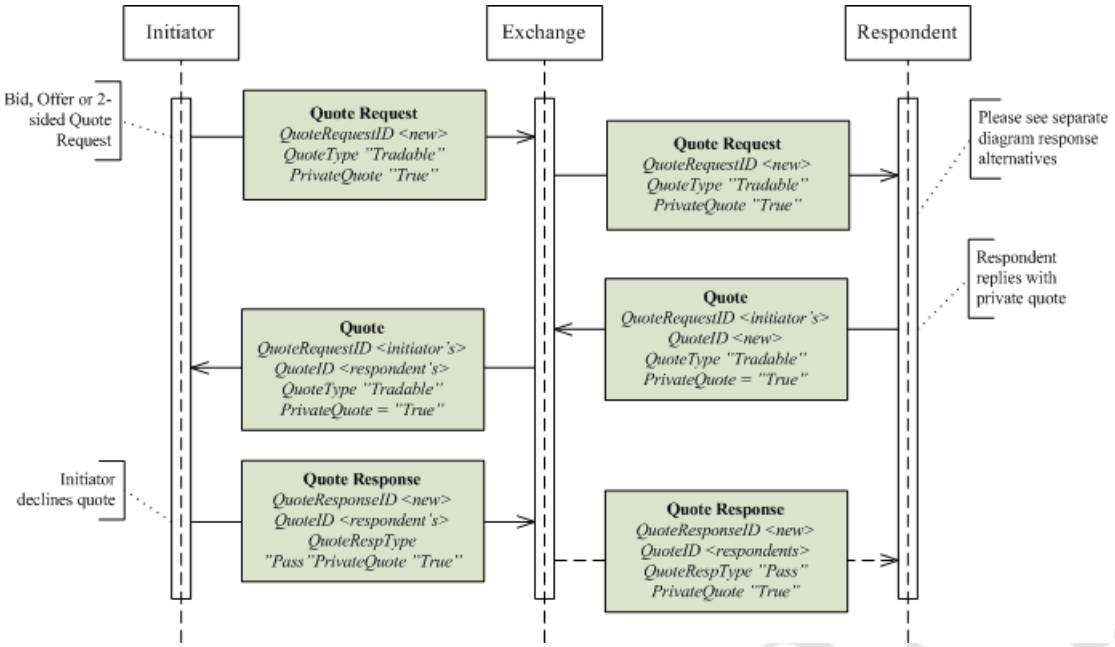


*Figure 22: Pass a Private Quote*

# 10 Indications of Interest (IOI)

**Note:**

The FIX standard does not explicitly mention the usage of IOI messages in a three party model (the exchange being in the middle). The FIX specification is instead based on the sell-side sending IOI's to the buy-side without an intermediary. Despite this, IOI hubs use FIX to route IOI messages between counterparties. OMX has discussed its service concept of "Bulletin Boards" for non-executable indications of interest and together with the FPL found that IOI-messages are the preferred model for implemention.

## 10.1 Business Message Types

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

*Table 128: Business Messages*

| In/Out | Message Name | Comment |
|---|---|---|
| In | **IOI** | Used to insert, modify or remove an Indication of Interest. |
| Out | **IOI** | Used to publish Indications of Interest to eligible market participants |

## 10.2 Indication of Interest

### 10.2.1 Introduction

Indication of interest (IOI) messages are used to market merchandise which the broker is buying or selling in either a proprietary or agency capacity. The indications can be time bound with a specific expiration value. Indications are distributed with the understanding that other firms may react to the message first and that the merchandise may no longer be available due to prior trade. From the point of view of a marketplace, IOIs are simply passed through to eligible receivers, an IOI is never matched or associated with other response processing.

Indication messages can be transmitted in various transaction types; New, Cancel, and Replace. All message types other than New modify the state of the message identified in IOIRefID (26).

**Note:**

IOI messages are different from many other FIX messages in that they are considered "one-way" only. They e.g. do not have Acks. A marketplace simply works like a hub in providing the IOIs to the receiving parties (generally the buy side).

## 10.2.2 Main Workflow

### 10.2.2.1 The Indication of Interest

The IOI message is used for all purposes:

- To submit a new IOI
- To update an IOI
- To cancel an IOI
- For the marketplace to publish an IOI to eligible market participants

You should note that IOIs are **not** included in the Market Data Snapshot Full Refresh or Market Incremental Refresh messages (see Section 14 on page 189 for details). However, this fact does not prevent the marketplace from publishing IOIs as part of a Market Data feed or as response to a Market Data subscription.

### 10.2.2.2 IOI Reject

There is no specific reject message associated with the IOI message. General market practice is that neither rejects nor accepts are issued. However, the Business Message Reject message could be used to reject IOI messages. See Section 4 on page 37 for a description of this message.

## 10.2.3 IOI Features

### 10.2.3.1 IOI Identification

#### 10.2.3.1.1 IOI ID

IOIs must be identified with a unique identification number (IOIID, 23). The marketplace does not assign its own identifier - but, to guarantee uniqueness, prepends the participant symbol (short name) to the IOIID (23) field when the message is forwarded to eligible receivers. Thus if the sender is "A" and has an IOIID = "123", the marketplace forwards it as "A-123".

#### 10.2.3.1.2 IOI ReferenceID

If a user wishes to update or cancel an IOI, the update is assigned a new IOIID (23) and the old IOIID is entered into the IOIRefID (26) field for reference.

### 10.2.3.2 Defining the Instrument

Instrument definition is done in the same manner as for orders, see Section 5 on page 49. However, in the case of IOIs the instrument definition may, subject to marketplace rules, be less formal. A user could e.g. indicate that he is interested to trade a treasury bill - but not the exact security.

### 10.2.3.3 Defining the Side

An IOI must specify the Side (54). The following options are available:

- Buy (1)
- Sell (2)

#### 10.2.3.4 Defining a Quantity

The following options are available for a definition of the quantity:

- IOIQty (27) must always be specified. The following values are available:
  - 0-1 000 000 000 (specifying the exact quantity)
  - U = Undisclosed Quantity

#### 10.2.3.5 Defining the Price

The price of an IOI is defined using the following options:

- Price (44) is used when the user wants to specify an exact limit price.

#### 10.2.3.6 IOI free text instructions

The user has an option to provide free text instructions. The free text instructions is provided in the Text (58) field.

## 10.3 Message Details

### 10.3.1 Indication of Interest

#### 10.3.1.1 IOI

*Table 129: IOI*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | **StandardHeader** | Y | MsgType = 6 | |
| 23 | IOIID | Y | | String |
| 28 | IOITransType | Y | | char |
| 26 | IOIRefID | N | Required for Cancel and Replace IOITransType messages | String |
| | **Instrument** | Y | Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages"<br>**OMX Comment:** Inbound messages shall use identifier fields only. | |
| | **Parties** | N | Insert here the set of "Parties" (firm identification) fields defined in "Common Components of Application Messages". | |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 54 | Side | Y | Side of Indication<br>Valid values:<br>1 = Buy<br>2 = Sell<br>7 = Undisclosed (for IOIs)<br>B = As Defined (for multilegs)<br>C = Opposite (for multilegs) | char |
| 27 | IOIQty | Y | The value zero is used if NoLegs repeating group is used | String |
| 44 | Price | N | | Price |
| 62 | ValidUntilTime | N | | UTCTimes-tamp |
| 58 | Text | N | | String |
| 60 | TransactTime | N | | UTCTimes-tamp |
| | **StandardTrailer** | Y | | |

## 10.4 Component Blocks (IOI Specific)

For components that are not specific for IOIs, please see Section 19 on page 261.

### 10.4.1 Components

None.

## 10.5 Not Supported FIX Functionality

OMX currently does not support the following functionality of the standard FIX Specification ([1] on page 21):

- Targeting / Routing as described in "Appendix 3-A" of volume 3
- Using IOIs to solicit Quote Responses as described in the "Product: Fixed Income (FI)" chapter of volume 7.

## 10.6 Workflows

### 10.6.1 Introduction

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

The FIX Protocol Specification does not include the workflows defined in this document.

## 10.6.2 IOI Workflows

### 10.6.2.1 Introduction

As IOI messages have no Acks, the workflows themselves are not very detailed. For the sake of completeness, they are included anyway.

### 10.6.2.2 Entering an IOI

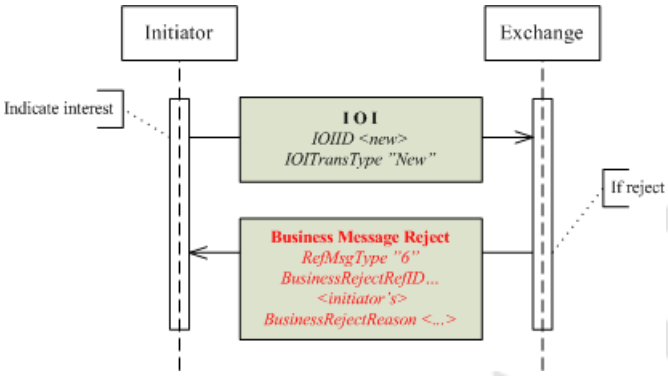


*Figure 23: Entering an IOI*

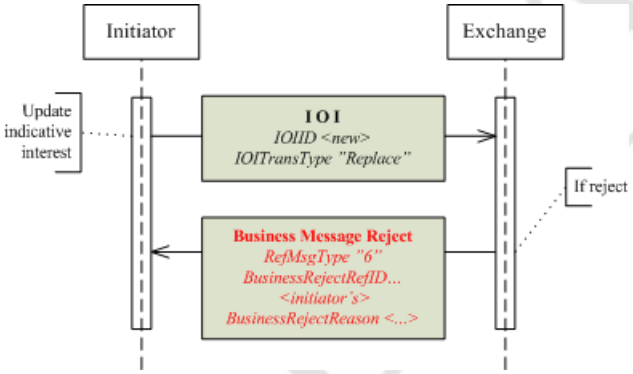


*Figure 24: Updating an IOI*

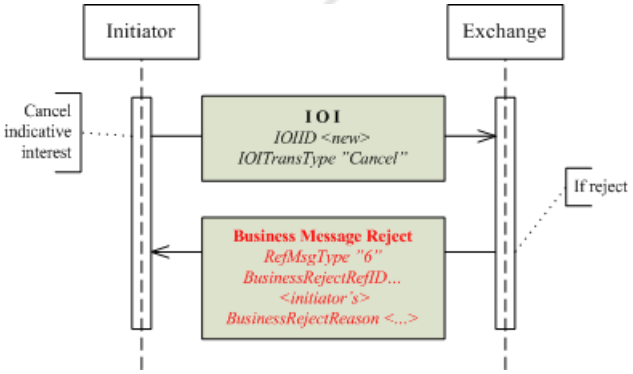


*Figure 25: Canceling an IOI*

# 11 Request for One-Sided Auction

**Note:**

Standard FIX does not support One-Sided Auctions. OMX User Defined messages, fields and workflows are used. There is currently no FPL discussion for an extension of the FIX standard.

## 11.1 Business Message Types

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

*Table 130: Business Messages*

| In/Out | Message Name | Comment |
|---|---|---|
| In | **One Sided Auction Request** | Used to request a One-Sided Auction |
| Out | **One Sided Auction Info** | Used to publish information about a One Sided Auction |
| Out | **One-Sided Auction Request Ack** | Used to acknowledge or reject a request |
| Out | **One-Sided Auction Report** | Used to report the result of a One-Sided Auction to the auctioneer.<br>Optionally used to publish the result of a One-Sided Auction. |

## 11.2 Requesting an Auction

### 11.2.1 Main Workflow

#### 11.2.1.1 Introduction

A One-Sided Auction is a call auction where one actor is alone on one side of the book. One-Sided Auctions are typically used to issue new fixed income instruments, but can also be used to issue other security or for on-tap issues. The functionality is also used to buy-back previously issued securities.

Normally the issuer of the security is the single actor on one side of the book. The issuer can use an agent to manage the auction. The manager of the auction is called the auctioneer.

#### 11.2.1.2 Initiating the Auction

The auction is started by sending a One-Sided Action Request message defining the terms of the auction. The marketplace validates the request, and if the request is approved, schedules the auction.

When the auction request is approved, an announcement is made to other actors using the One Sided Auction Info message. This message is published as part of the market data stream or subject to subscriptions using the One Sided Auction Info Request message.

#### 11.2.1.3 Bidding

Participants may enter bids (issuing auction) or offers (buy-back auction) for a defined time, possible during a period of days. Subject to marketplace rules and auction parameters, market data is distributed during the bidding period.

#### 11.2.1.4 Execution

At a certain time the bidding period is closed and the auction enters a state called "Issuer Position Modification" where the auctioneer is the only actor allowed interaction with the book. The auctioneer may now enter or modify his bid and thereby change the outcome of the auction. He may also, subject to marketplace rules, be allowed to remove orders on the other side of the book. During this period an indicative auction result is calculated and distributed to the auctioneer using the One-Sided Auction Report message.

Finally the auction is executed and the result published using the One-Sided Auction Report.

### 11.2.2 One-Sided Auction Features

#### 11.2.2.1 Market Data during Auction

The marketplace may have standard rules for how pre-trade market data (order book information) is distributed during the auction. The auctioneer may override those by specifying:

- BookTransparency (20246).
- MDBookType (1021). Refer to Section 20 on page 273 for allowed values.

## 11.3 Message Details

### 11.3.1 OneSidedAuctionInfo

*Table 131: OneSidedAuctionInfo*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | **StandardHeader** | Y | MsgType = UB | |
| 20091 | AuctionInfoID | Y | Unique message identifier | String |
| 20092 | AuctionInfoReqID | N | Provided for a response to a specific One Sided Auction Info Request message. | String |
| | **Parties** | N | Insert here the set of "Parties" (firm identification) fields defined in "Common Components of Application Messages" | |
| | **Instrument** | N | Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages" | |
| 20048 | AuctionType | N | | int |
| 20246 | BookTransparency | N | | int |
| 1021 | MDBookType | N | | int |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 20049 | AuctionTime | N | Time the auction is scheduled for execution | UTCTimes-tamp |
| 64 | SettlDate | N | Settlement date for trades | LocalMktDate |
| 60 | TransactTime | N | | UTCTimes-tamp |
| | **StandardTrailer** | Y | | |

## 11.3.2 OneSidedAuctionReport

*Table 132: OneSidedAuctionReport*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | **StandardHeader** | Y | MsgType = U4 | |
| 20054 | AuctionResultID | N | | String |
| 20047 | AuctionReqID | N | | String |
| | **Parties** | N | Insert here the set of "Parties" (firm identification) fields defined in "Common Components of Application Messages" | |
| | **Instrument** | N | Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages" | |
| 346 | NumberOfOrders | N | Number of orders at the investor side of the book | int |
| 20056 | NumberOfMatchOrders | N | Number of matching investor orders at the auction clearing price | int |
| 20057 | ContraQty | N | Aggregated quantity at the investor side of the book | Qty |
| 20058 | TradeQty | N | Total quantity that can be or have been executed | Qty |
| 332 | HighPx | N | The highest price at the investor side of the book | Price |
| 333 | LowPx | N | The lowest price at the investor side of the book | Price |
| 20060 | TradeVWAP | N | The volume weighted average price of executed orders. Relevant for Multiple Price Auctions. | Price |
| 20062 | PctMatchQty | N | Percentage of investor quantity at the auction clearing price that is or will be executed | Percentage |
| 60 | TransactTime | N | | UTCTimes-tamp |
| | **StandardTrailer** | Y | | |

## 11.3.3 OneSidedAuctionRequest

*Table 133: OneSidedAuctionRequest*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | **StandardHeader** | Y | MsgType = U2 | |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 20047 | AuctionReqID | Y | Unique message identifier | String |
| | **Parties** | N | Insert here the set of "Parties" (firm identification) fields defined in "Common Components of Application Messages" | |
| | **Instrument** | N | Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages" | |
| 20048 | AuctionType | N | | int |
| 20246 | BookTransparency | N | | int |
| 1021 | MDBookType | N | | int |
| 20049 | AuctionTime | N | Time the auction is scheduled for execution | UTCTimestamp |
| 64 | SettlDate | N | Settlement date for trades | LocalMktDate |
| 60 | TransactTime | N | | UTCTimestamp |
| | **StandardTrailer** | Y | | |

## 11.3.4 OneSidedAuctionRequestAck

*Table 134: OneSidedAuctionRequestAck*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | **StandardHeader** | Y | MsgType = U3 | |
| 20047 | AuctionReqID | Y | | String |
| 20053 | AuctionRejectReason | N | | int |
| | **Parties** | N | Insert here the set of "Parties" (firm identification) fields defined in "Common Components of Application Messages" | |
| | **Instrument** | N | Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages" | |
| 20048 | AuctionType | N | | int |
| 20246 | BookTransparency | N | | int |
| 1021 | MDBookType | N | | int |
| 20049 | AuctionTime | N | Time the auction is scheduled for execution | UTCTimestamp |
| 64 | SettlDate | N | Settlement date for trades | LocalMktDate |
| 60 | TransactTime | N | | UTCTimestamp |
| | **StandardTrailer** | Y | | |

## 11.4 Component Blocks (Request for Auction Specific)

For components that are not specific for Request for Auction, please see Section 19 on page 261.

### 11.4.1 Components

None.

### 11.4.2 Implicit Components

None.

## 11.5 Workflows

### 11.5.1 Introduction

The following workflow mainly defines the interaction between the auctioneer and the marketplace.

### 11.5.2 Request for Auction

The following workflow depicts the interaction between the auctioneer and the marketplace.

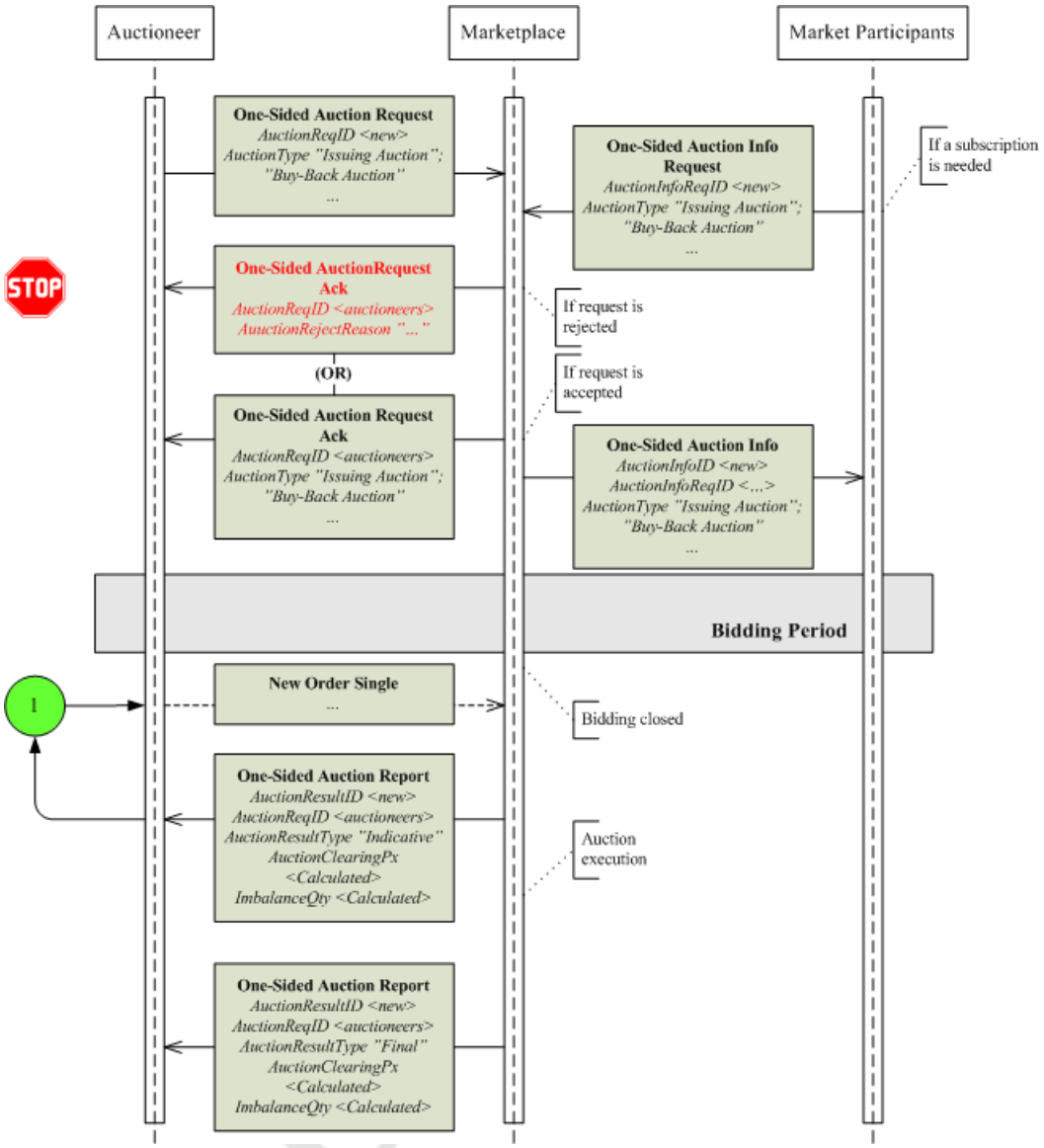


*Figure 26: One Sided Auction*

Note that actors eligible to participate in the auction receives a One-Sided Auction Info message describing the auction. Otherwise they interact using the same mechanisms as available in normal market situations, e.g.:

- They receive state changes as normally through the Trading Session Status and Security Status messages
- They receive market data messages containing price information
- They interact using normal order messages during the "bidding period"
- They receive Execution Reports and Trade Capture Reports as normally

# 12 Reporting Privately Negotiated Trades

## 12.1 Business Message Types

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

*Table 135: 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

### 12.2.1 Introduction

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

*Currently not supported*. 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 the standard FIX Specification [1] on page 21:

- 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.

## 12.2.2 Main Workflow

### 12.2.2.1 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 Section 13 on page 177 for further details)

When the user chooses the option to send a Trade Capture Report, he also must specify the counterparty in the <RootParties> component block of the message.

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

- RootPartyID (1117) = the identifier of the participant firm
- RootPartyIDSource (1118):
  - C = Generally accepted market participant identifier
  - D = Proprietary / Custom code
- RootPartyRole (1119) = see below

The following table specifies the relevant party roles:

*Table 136: Supported RootPartyRole (1119) values*

| | Business Role | Party Role | Comment |
|---|---|---|---|
| The respondent (Market Maker or other eligible quote party) | Firm | RootPartyRole (1119) = 17 (Contra Firm) | RootPartyIDSource = D. Relevant when the market-place issues an id-number for those actors. Applicable e.g. when the actor has a connection to the market-place.<br>RootPartyIDSource = C. Relevant when the market-place supports mnemonic symbolic names for those actors. Applicable e.g. when the actor has a connection to the marketplace. |
| In cases where a specific unit or trader needs to be assigned, the RootSubPar-ties component is also used | Unit | RootPartySubID (1121)<br>RootPartySubIDType (1122) = 25 (Location desk) | Currently not supported |
| | Trader | RootPartySubID (1121)<br>RootPartySubIDType (1122) = 2 (Person) | Currently not supported |

#### 12.2.2.2 Trade Capture Report Reject

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

*Table 137: Trade Capture Report Reject Messages*

| Reject Message | Direc-tion | Business Message | Comment |
|---|---|---|---|
| **Trade Capture Re-port Ack** | Out | Trade Capture Report | |

### 12.2.3 Trade Reporting Features

#### 12.2.3.1 Trade Report Identification

##### 12.2.3.1.1 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 (571) and be chained to the earlier version using the TradeReportRefID (572).

##### 12.2.3.1.2 Trade Report Reference ID

If a user wishes to update or cancel an Trade Capture Report, the update is assigned a new TradeReportID (571) and the old TradeReportID is entered into the TradeReportRefID (572) field for reference.

#### 12.2.3.2 Defining the Instrument

Instrument definition is done in the same manner as for orders, see Section 5 on page 49 .

#### 12.2.3.3 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 a single side only.

The reporting party specifies the counterparty in the <RootParties> component block as specified above.

#### 12.2.3.4 Accounts, Pre-Allocations and Give-Ups

Please refer to Section 4.5 on page 42.

#### 12.2.3.5 Trade Publication Indicators

Subject to regulation and market practices, trade publication can be governed by users. The PublishTrdIndicator (852) governs the default behavior of the trade publication and reporting services of the marketplace.

Example:

- Regulations may stipulate that trades over a certain size may receive deferred Market Data publication subject to user request, while all other reporting occurs in a timely manner. The user indicates that he wants deferred publication by:
  - PublishTrdIndicator (852) = 2 - Deferred publication

As trades are published to a variety of receiver categories (including market data), the wish to publish or not may vary by such receiver. The Trade Capture Report provides a repeating group (<TrdRepIndGrp>) of publication indicators per party role. This component is used to override the default behavior as indicated in the PublishTrdIndicator (852) field. Bilateral agreement governs in what contexts thise indicators are eligible. The allowed set of roles are:

- 10 - **Settlement Location** (CSD) - applies to trade confirmations
- 21 - **Clearing Organization** (Clearing House or Central CounterParty, CCP) - applies to trade confirmations
- 22 - **Exchange** - applies to market data publication only
- 23 - **Regulatory Body** - applies to trade confirmations

**Note:**

FIX does not currently support a repeating group of publication indicators. OMX is working to expand FIX with this.

Examples:

- The entity reporting a confirmed trade to the marketplace has a direct connection to the CSD and has the option to report certain trades there itself. Publication to other parties occur through the marketplace. The user indicates that he does not want publication to the CSD by setting the following fields:
  - TrdRepPartyRole (20088) = 10 - Settlement Location
  - TrdRepIndicator (20245) = 0 - Do not report trade

## 12.3 Message Details

See Section 13 on page 177 for the message specifications.

## 12.4 Component Blocks (Privately Negotiated Trades Specific)

See Section 13 on page 177 for the components specifications.

## 12.5 Workflows

### 12.5.1 Introduction

OMX has, together with the FPL Global Exchanges and Markets Committee and other exchanges, contributed to the more detailed and extended workflows of FIX 5.0. As the FIX standard specification captures the OMX workflows, the following chapter simply reference the standard.

### 12.5.2 One-Party Report for Pass Thru

Refer to volume 5 of the standard FIX specification ([1] on page 21) and the following chapter:

- CATEGORY: TRADE CAPTURE ("STREETSIDE") REPORTING (the main chapter)
  - One-Party Report for Pass-Through to Model (detailed workflow diagram)
- Appendix B - Trade Capture Report (TCR) Work Flow and Usage Tables
  - One-Party Report for Pass-Thru (detailed message scenario table)

### 12.5.3 One-Party Report for Matching

Refer to volume 5 of the standard FIX specification ([1] on page 21) and the following chapter:

- CATEGORY: TRADE CAPTURE ("STREETSIDE") REPORTING (the main chapter)
  - One-Party Report for Matching Model (detailed workflow diagram)
- Appendix B - Trade Capture Report (TCR) Work Flow and Usage Tables
  - One-Party Report for Matching (detailed message scenario table)

### 12.5.4 Two-Party Reporting

Refer to volume 5 of the standard FIX specification ([1] on page 21) and the following chapter:

- CATEGORY: TRADE CAPTURE ("STREETSIDE") REPORTING (the main chapter)
  - Two-Party Reporting (detailed workflow diagram)
- Appendix B - Trade Capture Report (TCR) Work Flow and Usage Tables
  - Two-Party Report (detailed message scenario table)

### 12.5.5 Confirmed Trade Reporting

*Currently not supported*.

Refer to volume 5 of the standard FIX specification ([1] on page 21) and the following chapter:

- CATEGORY: TRADE CAPTURE ("STREETSIDE") REPORTING (the main chapter)
  - Confirmed Trade Reporting Model (detailed workflow diagram)
- Appendix B - Trade Capture Report (TCR) Work Flow and Usage Tables
  - Reporting of Locked-In Trade to Marketplace (detailed message scenario table)

# 13 Trade Confirmation and Management

## 13.1 Business Message Types

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

*Table 138: Business Messages*

| In/Out | Message Name | Comment |
|---|---|---|
| In | **Trade Capture Report** | Used for the following purposes:<br>• Report confirmed trades from other marketplaces<br>• Request cancelation of a confirmed trade (a.k.a. Trade Break)<br>• 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 Introduction

A confirmed trade occurs when orders or quotes are executed and when privately negotiated trades are approved. The marketplace publishes confirmed trades to counterparties and other actors involved in the downstream processing of trades. Such actors can include:

- Broker back office
- Broker clearing firms
- Clearing houses, Central Counter Parties (CCP)
- Central Securities Depositories (CSD)
- Regulators

**Note:**

The Execution Report message is also used to report fills, but this message is primarily intended as responses to orders and quotes, i.e. for front-office use. The Trade Capture Report message (as described in this chapter) on the other hand is primarily intended for actors that process trades in the downstream part of the transaction chain - and thereby designed to contain complete trade information. Market Data messages are used to publish public trade information for so called trade tickers.

Subject to marketplace rules, users are also allowed to request amendments and cancelation of previously confirmed trades.

## 13.2.2 Main Workflow

There following main workflows are supported:

- Marketplace publication of confirmed trades to users involved in the downstream processing of trades
- User request to amend a trade
- User request to cancel a trade

For details about general workflows and functionality, please see the following chapters in volume 5 of the standard FIX Specification (refer to [1] on page 21):

- CATEGORY: TRADE CAPTURE ("STREETSIDE") REPORTING

Also refer to to detailed scenarios provided in Workflows on page 185.

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

## 13.2.3 Trade Confirmation Features

### 13.2.3.1 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 = 99). 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.

#### 13.2.3.2 Accounts, Pre-Allocation and Give-Ups

Please refer to Section 4.5 on page 42.

## 13.3 Message Details

### 13.3.1 TradeCaptureReport

*Table 139: 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 |
| 829 | TrdSubType | N | **OMX Comment:** Used to distinguish between On-Hours and Off-Hours trades | int |
| 855 | SecondaryTrdType | N | | 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 subsequent trade transaction such as a transfer | String |
| 572 | TradeReportRefID | N | The TradeReportID that is being referenced for some action, such as correction or cancelation<br>**OMX Comment:** Refers to a previous Trade Capture Report. Relevant for references also in workflows for Privately Negotiated Trades | String |
| 880 | TrdMatchID | N | | String |
| 17 | ExecID | N | Exchanged assigned Execution ID (Trade Identifier)<br>**OMX Comment:** If specified refers to the Execution Report issued for the fill. | String |
| 527 | SecondaryExecID | N | | String |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | **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"<br>**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.<br>**OMX Comment:** Used when non-standard settlement applies. | LocalMktDate |
| 574 | MatchType | N | | String |
| 1115 | OrderCategory | N | | char |
| | **TrdCapRptSideGrp** | Y | Number of sides | |
| 20228 | Volatility | N | **OMX Comment:** Not in FIX. OMX request addition | float |
| 20230 | DividendYield | N | **OMX Comment:** Not in FIX. OMX request addition | float |
| 20232 | RiskfreeRate | N | **OMX Comment:** Not in FIX. OMX request addition | float |
| 20233 | CurrencyRatio | N | **OMX Comment:** Not in FIX. OMX request addition | float |
| 797 | CopyMsgIndicator | N | Indicates drop copy. | Boolean |
| 852 | PublishTrdIndicator | N | | int |
| | **TrdRepIndGrp** | N | Icludes a list of Trade Publication indicators per party role<br>**OMX Comment:** Currently not in FIX 5.0. OMX requests extension. | |
| | **StandardTrailer** | Y | | |

## 13.3.2 TradeCaptureReportAck

*Table 140: 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 |
| 829 | TrdSubType | 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 |
| | **TrdRepIndGrp** | N | **OMX Comment:** Currently not in FIX 5.0. OMX requests extension. | |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 58 | Text | N | May be used by the executing market to record any execution Details that are particular to that market | String |
| | **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 Section 19 on page 261.

### 13.4.1 Components

#### 13.4.1.1 YieldData

*Table 141: YieldData*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 236 | Yield | N | | Percentage |

### 13.4.2 Implicit Components

#### 13.4.2.1 TrdRepIndGrp

*Table 142: TrdRepIndGrp*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 20087 | NoTrdRepIndicators | N | Number of trade reporting indicators following<br>**OMX Comment:** Currently not in FIX 5.0. OMX requests extension. | NumInGroup |
| >20088 | TrdRepPartyRole | N | Specifies the type of party for which the trade reporting indicator applies. Required if the NoTrdRepIndicators (20087) is specified.<br>**OMX Comment:** Currently not in FIX 5.0. OMX requests extension. | int |
| >20245 | TrdRepIndicator | N | Indicates whether parties of the specified role should receive a trade report or not. Required if the NoTrdRepIndicators (20087) is specified.<br>**OMX Comment:** Currently not in FIX 5.0. OMX requests extension. | Boolean |

### 13.4.2.2 TrdAllocGrp

*Table 143: TrdAllocGrp*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 78 | NoAllocs | N | Number of repeating groups for trade allocation<br>**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" | |

### 13.4.2.3 TrdCapRptSideGrp

*Table 144: 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:<br>- QuoteMsgID (1166) of a single Quote<br>- QuoteID (117) of a Mass Quote<br>**OMX Comment:** FIX 5.0 SP1 | String |
| >19 | ExecRefID | N | **OMX Comment:** Applicable for new trades, not only for trade cancels and corrects | String |
| >66 | ListID | N | **OMX Comment:** Identifies a set of Contingent Orders | String |
| | **Parties** | N | Insert here the set of "Parties" (firm identification) fields defined in "Common Components of Application Messages"<br>Range of values on report:<br>**OMX Comment:** Enter side-related clearing party info here | |
| >1093 | LotType | N | | char |
| >377 | SolicitedFlag | N | | Boolean |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| >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.<br>**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 |
| >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 |
| >70 | AllocID | N | Used to assign an ID to the block of preallocations | String |
| | **TrdAllocGrp** | N | | |
| >1057 | AggressorIndicator | N | | Boolean |

### 13.4.2.4 TrdCapRptAckSideGrp

*Table 145: 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" | |
| >1093 | LotType | N | | char |
| >377 | SolicitedFlag | N | | Boolean |
| >528 | OrderCapacity | N | | char |
| >529 | OrderRestrictions | N | | MultipleCharValue |
| >483 | TransBkdTime | N | | UTCTimestamp |
| >336 | TradingSessionID | N | | String |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| >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

OMX has, together with the FPL Global Exchanges and Markets Committee and other exchanges, contributed to the more detailed and extended workflows of FIX 5.0. As the FIX standard specification captures the OMX workflows, the following chapter simply reference the standard.

## 13.5.2 Reporting of Confirmed Trades

Refer to the volume 5 of the standard FIX specification ([1] on page 21) and the following chapter:

- CATEGORY: TRADE CAPTURE ("STREETSIDE") REPORTING (the main chapter)
  - Reporting Confirmed Trades to Miscellaneous Parties
  - Extension to Workflows resulting in Fills reported as Execution Reports (detailed workflow diagram)
- Appendix A – Trade Amendment and Trade Cancel Work Flow Diagrams
  - (detailed message scenario table)
- Appendix B - Trade Capture Report (TCR) Work Flow and Usage Tables
  - Confirmed Trade – Published by Marketplace (detailed message scenario table)

## 13.5.3 Requesting an Amendment of a Confirmed Trade

Refer to the volume 5 of the standard FIX specification ([1] on page 21) and the following chapter:

- CATEGORY: TRADE CAPTURE ("STREETSIDE") REPORTING (the main chapter)
  - Trade Amendment
- Appendix A – Trade Amendment and Trade Cancel Work Flow Diagrams
  - Trade Amendment for Trade Capture Report
  - Trade Amendment - One-Party Report for Pass-Thru Model (detailed message scenario table)
  - Trade Amendment - One-Party Matching Model (detailed message scenario table)

- Trade Amendment - Two-Party Report (detailed message scenario table)
- Trade Amendment - Confirmed Trade Reporting Model (detailed message scenario table)

- Appendix B - Trade Capture Report (TCR) Work Flow and Usage Tables
  - Trade Amendment - One-Party Report for Pass-Thru (detailed message scenario table)
  - Trade Amendment – One-Party Report for Matching (detailed message scenario table)
  - Trade Amendment - Two-Party Report (detailed message scenario table)
  - Trade Amendment - Locked-In Amendment (detailed message scenario table)

### 13.5.4 Requesting Cancelation of a Confirmed Trade

Refer to the volume 5 of the standard FIX specification ([1] on page 21) and the following chapter:

- CATEGORY: TRADE CAPTURE ("STREETSIDE") REPORTING (the main chapter)
  - Trade Break / Trade Cancel
- Appendix A – Trade Amendment and Trade Cancel Work Flow Diagrams
  - Trade Capture Report Trade Cancel
  - Trade Cancel - One-Party Pass-Thru Model (detailed message scenario table)
  - Trade Cancel - One-Party Matching Model (detailed message scenario table)
  - Trade Cancel - Two-Party Report (detailed message scenario table)
  - Trade Cancel - Confirmed Trade Reporting Model (detailed message scenario table)
- Appendix B - Trade Capture Report (TCR) Work Flow and Usage Tables
  - Trade Cancel - One-Party Report for Pass-Thru (detailed message scenario table)
  - Trade Cancel - One-Party Report for Matching (detailed message scenario table)
  - Trade Cancel - Two-Party Report (detailed message scenario table)
  - Trade Cancel - Locked-In Cancellation (detailed message scenario table)

### 13.5.5 Generic Sub-Workflows

Refer to the volume 5 of the standard FIX specification ([1] on page 21) and the following chapter:

- Appendix A – Trade Amendment and Trade Cancel Work Flow Diagrams
  - Generic Sub-Workflows
  - Canceling a Pre-confirmed Trade Capture Report (detailed message scenario table)
  - Updating (Replacing) a Trade Capture Report (detailed message scenario table)

## 13.5.6 Delayed Publication of Trades

The below diagram depicts the workflow to the counterparties when a confirmed trade is publicly relayed over market data without delay:

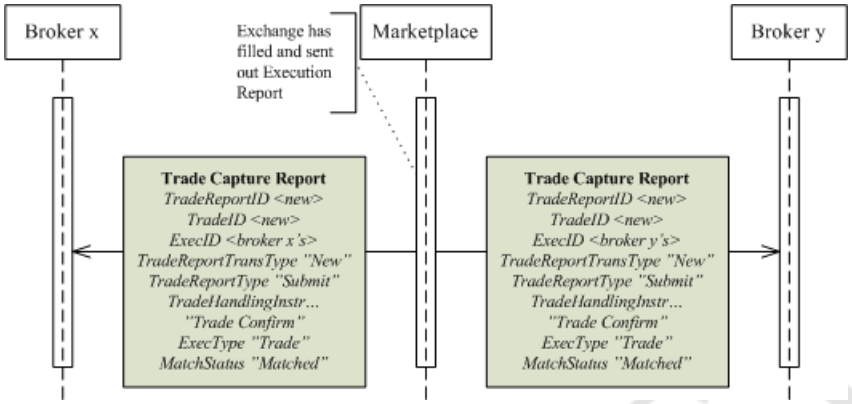


*Figure 27: Publication without delay*

In cases publication over market data is delayed (subject to user instruction), the below diagram depicts the workflow:

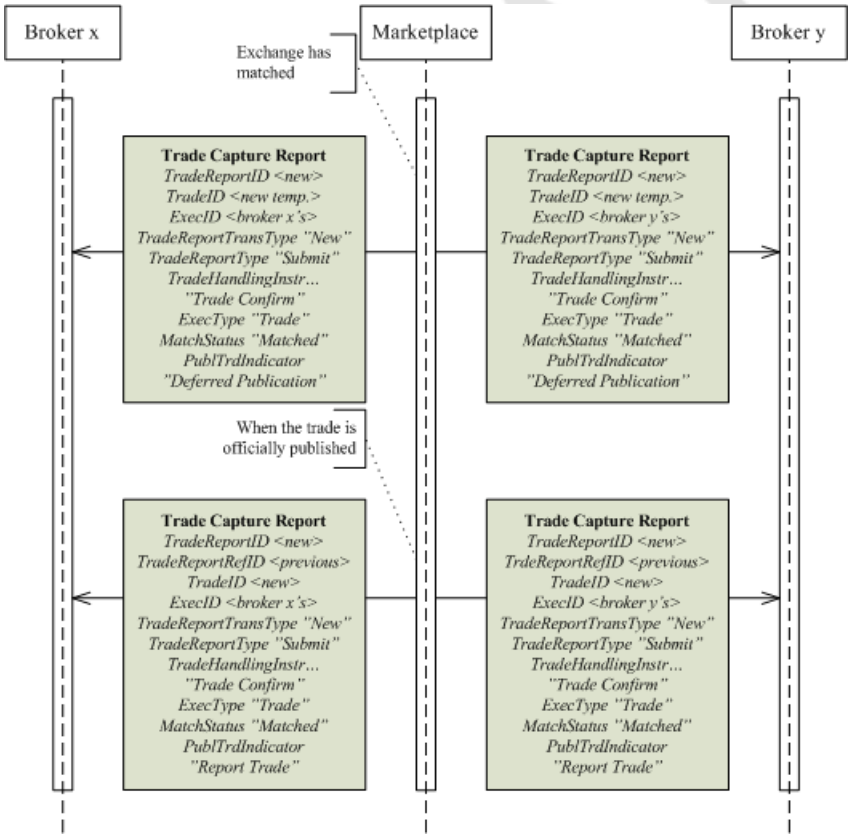


*Figure 28: Delayed Publication*

# 14 Market Data

**Note:**

Standard FIX does not include a separate Statistics message. There is currently no FPL discussion for an extension of the FIX standard. An OMX user defined message including user defined fields and workflows support more extensive statics.

## 14.1 Business Message Types

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

*Table 146: 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. OMX user defined message. |

## 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, hereafter 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 - see Section 15 on page 203 for details.

Please refer to [4] on page 21 published by the FPL Market Data Optimization Working Group.

## 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 unsolicited 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.

## 14.2.4 Market Data Features

### 14.2.4.1 Introduction

Please refer to [3] on page 21 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 MDBookType (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, it 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 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 separately.
- 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:

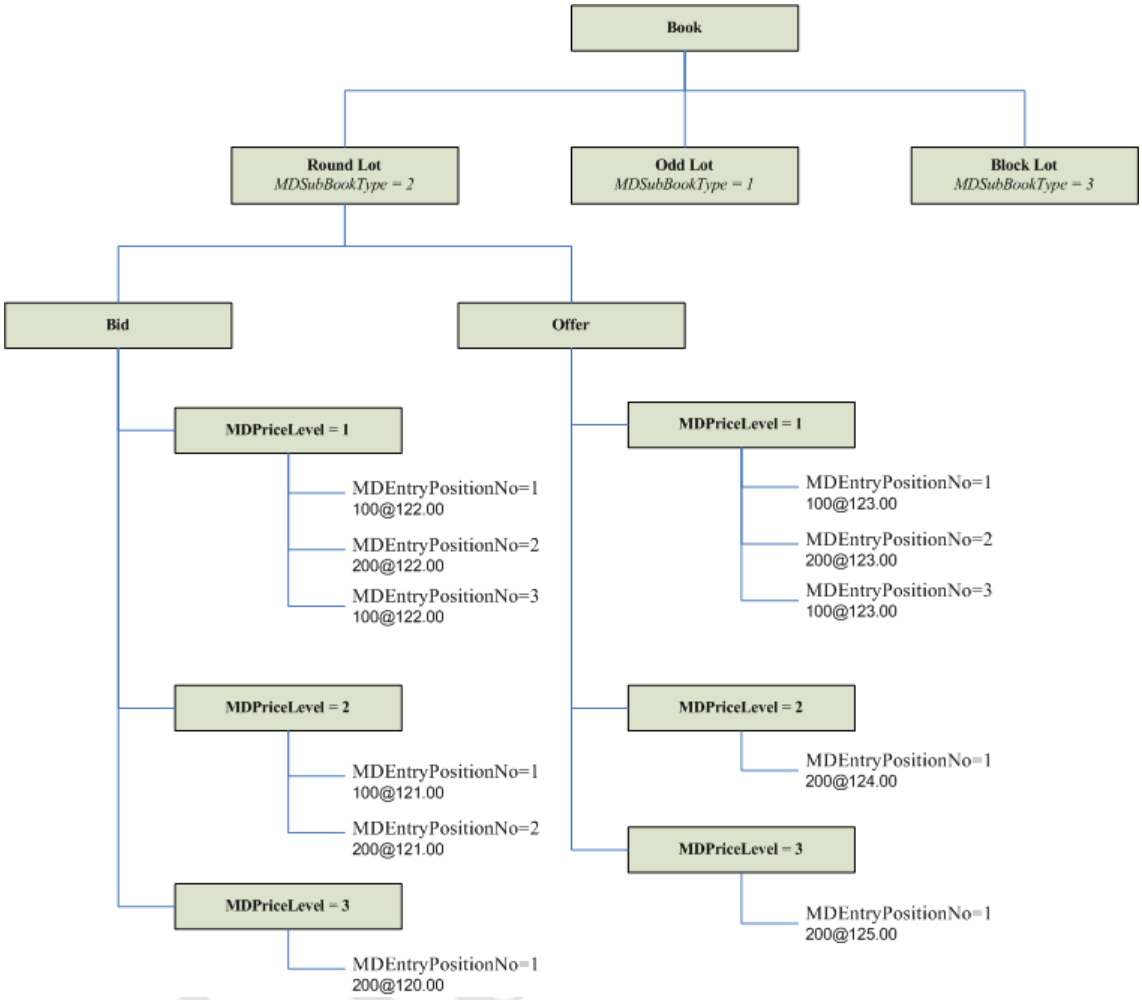


*Figure 29: Book Sorting and Ranking Example*

### 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 Implied Prices

Refer to Section 6.2.3.3 on page 90 for an explanation of Implied Prices.

Provided the marketplace is supporting Implied Prices, those are also published over market data. The marketplace can choose to indicate that a price is implied through QuoteCondition (276) = "K" - Implied Price. The indicator is relevant for Order Depth book updates only.

#### 14.2.4.5 Publication of Streaming FX Prices

Streaming indicative FX (Foreign eXchange or cross currency) prices may be provided as a separate feed or as part of other market data feeds. The representation of such prices follow the recommendations of the FPL Global FX Committee described in Volume 7 of the standard FIX Specification (refer to [1] on page 21). As the main purpose of this feed is to provide reference prices, not to support FX-trading, this specification deviates from some of the rules provided in the standard. The following example illustrates how an entry can be represented:

- MDEntryType (269) = "H" - Mid Price
- Instrument
  - SecurityID (48) = id of currency pair or interest rate
  - SecurityIDSource (22) = "99" - Marketplace-assigned identifier
- MDEntryPx (270) = price/conversion rate
- MDQuoteType (1070) = "0" - Indicative

  MDEntryDate (272)
- MDEntryTime (273)
- PartyID (448) = source of price
- PartyIDSource (447) = "D" - Proprietary / Custom code (exchange-assigned id-number)
- PartyRole (452) = "74" - Maket data entry originator

#### 14.2.4.6 Publication of Fixing Values

Fixing values, as e.g. FX or interest rate fixings, are published as separate Market Data entries. The following example illustrates how:

- MDEntryType (269) = "5" - Closing Price
- Instrument
  - SecurityID (48) = id of currency pair or interest rate
  - SecurityIDSource (22) = "99" - Marketplace-assigned identifier
- MDEntryPx (270) = fixing price/value
- MDEntryDate (272) = current date
- MDEntryTime (273) = current time

### 14.2.4.7 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.
- Separate Market Data Statistics message. Typically used to periodically publish statistics.

The following chapters discuss the respective alternatives.

#### 14.2.4.7.1 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 certain statistics. A market data entry marked with a certain statistics indicator is eligible for the specified type of statistic, but the user must compare 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)

#### 14.2.4.7.2 Separate Market Data Statistics Message

*The Market Data Statistics message is not standard FIX. OMX, through the FPL Global Exchanges and Markets Committee, is working to include the message in a future version of the standard.*

The Market Data Statistics message allows bundling of many pieces of statistics for multiple instruments. As the type of statistics is generalized, the message is less sensitive to extensions in the type of statistics published.

The Market Data Statistics message contains two levels of repeating groups:

- The instruments group (NoMDStatsInstruments) is used to identify the instrument
- The statistics group (NoMDStats) is used to relay the individual statistics entries for each instrument

Statistics are generalized so each entry is defined by its statistics type (MDStatsType, 20032), e.g.:

- 3 - Index Value
- 4 -Opening Price
- 5 - Closing Price
- 6 - Settlement Price

- 7 -High Price
- 8 -Low Price
- 9 -VWAP Price
- B -Trade Volume
- C -Open Interest

With the statistics entry comes a Price and/or a Quantity.

What type of statistics are relayed and in what frequency is bilaterally agreed.

#### 14.2.4.8 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 147: 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<br>**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<br>**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 148: 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 | | |

## 14.4 Component Blocks (Market Data Specific)

For components that are not specific for Market Data, please see Section 19 on page 261.

### 14.4.1 Components

None.

### 14.4.2 Implicit Components

#### 14.4.2.1 MDIncGrp

*Table 149: 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 |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| >269 | MDEntryType | N | Conditionally required if MDUpdateAction = New(0). Cannot be changed.<br>**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 = Change (1) and MDEntryRefID is not specified, or must be unique among currently active entries if MDUpdateAction = Change(1) and MDEntryRefID is specified.. | String |
| >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"<br>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"<br>**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 |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| >625 | TradingSessionSubID | N | | String |
| >326 | SecurityTradingStatus | N | **OMX Comment:** FIX 5.0 SP1 | int |
| >276 | QuoteCondition | N | Space-delimited list of conditions describing a quote. | MultipleString-Value |
| >277 | TradeCondition | N | Space-delimited list of conditions describing a trade<br>**OMX Comment:** Currently not supported | MultipleString-Value |
| >828 | TrdType | N | For optional use in reporting Trades<br>**OMX Comment:** FIX 5.0 SP1 | int |
| >574 | MatchType | N | For optional use in reporting Trades<br>**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<br>**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<br>**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.<br>**OMX Comment:** FIX 5.0 SP1 | UTCTimes-tamp |
| >60 | TransactTime | N | For optional use in reporting Trades. Used to specify the time of matching.<br>**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 150: 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 151: 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 StatsIndGrp

*Table 152: StatsIndGrp*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 1175 | NoStatsIndicators | N | Number of statistics indicators<br>**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. | Int |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | | | **OMX Comment:** FIX 5.0 SP1 | |

# 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 Data Workflows

### 14.5.2.1 Push Model Workflows

Market data is published as a stream of messages packaged as separately defined by the marketplace. The marketplace might e.g. choose to package pre-trade information per market segment. The message stream can contain any of the outbound messages defined as part of Market Data. The marketplace may also choose to package information based on participant privileges and thereby provide feeds per type of privilege set.

### 14.5.2.2 Pull Model Workflows

A participant subscribes to market data using the Market Data Request message. The Market Data Snapshot Full Refresh, the Market Data Incremental Refresh and the Market Data Statistics messages are provided as a result.

# 15 Trading Sessions and States

**Note:**

Standard FIX does not explicitly prescribe how market state and security trading status changes are relayed. OMX is currently engaged in discussions with the FPL Global Exchanges and Markets Committee to harmonize and extend the standard in this area. This specification is based on the extension proposal put forth by this group.

## 15.1 Business Message Types

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

*Table 153: Business Messages*

| In/Out | Message Name | Comment |
|---|---|---|
| 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 |

## 15.2 Trading Sessions and States

### 15.2.1 Introduction

**State information** is 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 section (see Section 17 on page 221 for details).

#### Security Status

Relays state changes for individual order books / securities.

**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.

## 15.2.2 Unsolicited Feeds versus Subscriptions

Trading Session and State information can either be sent on a subscription basis (using Trading Session Status Request / Security Status Request), or unsolicited per agreement with the exchange.

### 15.2.2.1 Subscriptions

Currently not supported.

### 15.2.2.2 Unsolicited Feeds

For unsolicited feeds the information is sent unsolicited, i.e. no subscription is required.

## 15.2.3 Main Workflow

### 15.2.3.1 Security State Changes

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

- Pre-Trading
- Opening (Auction)
- Continuous Trading
- Closing (Auction)
- Quiescent
- Post-Trading

The entire market segment or individual security may also be subject to non-scheduled states as Trading Halt, Fast Market, etc.

## 15.2.4 Trading Session and State Features

### 15.2.4.1 Trading Session Changes

Trading sessions represent the main trading state of a market, market segment, underlying or a security.

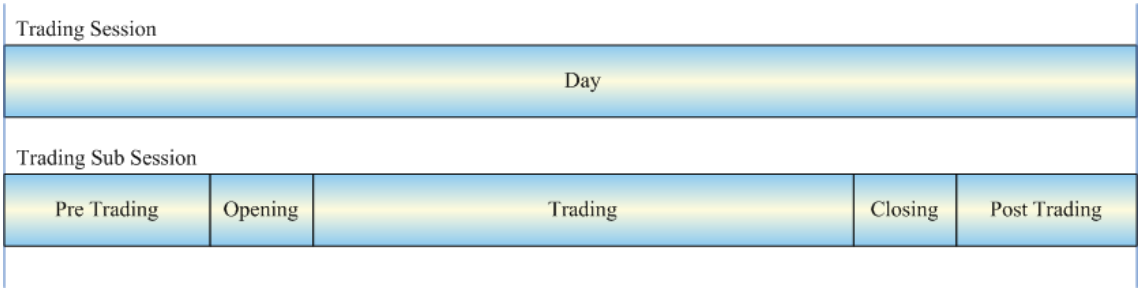


*Figure 30: Trading Sessions and Sub Sessions*

Associated with a trading session are the capabilities that the marketplace offer, typically what type of behavior is allowed.

Trading session changes can be relayed in a variety of ways subject to bilateral agreement:

- **Changes per Market / Market Segment level only**

  In this case session changes are relayed using the Trading Session Status message only. The TradingSessionID (336) and TradingSessionSubID (625) define the current main session and subsession respectively. The TradSesStatus (340) gives additional state information and the TradSesEvent (20089) field provides less formal complementary information. A state change should be interpreted as all Security of the market / market segment transition to the relevant state instantaneously. This model is typically used in markets with large numbers of instruments where opening / closing call auctions provide no business value (e.g. due to low liquidity).

- **Changes per Security only**

  In this case all state changes are relayed at the Security level only. This is mainly relevant in cases when a market hierarchy is not in place.

- **Changes per Security and Market / Market Segment**

  In this case session changes are relayed at both the Security and the Market / Market Segment level. Session changes at the security level indicate what state the individual security is in (and thereby what capabilities are available). This model is typically used when staggered transitions are used (e.g. when Security open/close one by one using call auctions).

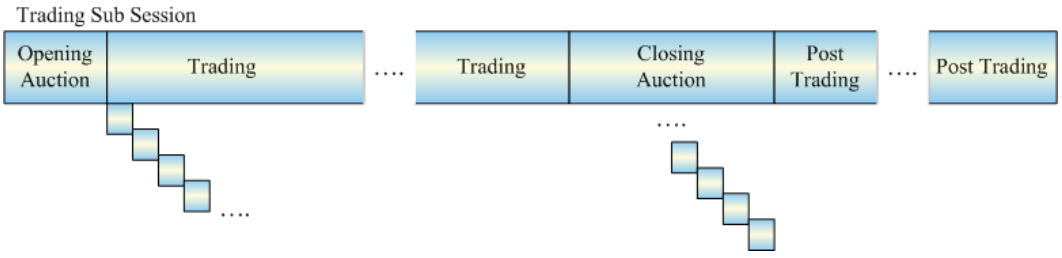


*Figure 31: Staggered State Transitions*

Session changes at the Market / Market Segment level can be relayed in three ways:

- Before the first Security transitions to the new state
- After the last Security transition to the new state
- Both before and after. In this case the TradSesEvent (20089) field will contain additional information:

- 101 - Initializing. Indicates that the state change is initiated, e.g. that the first security is starting to transition.
- 102 - Complete. Indicates that the the state change is completed, e.g. that the last security has transitioned.

What model is used for a particular Market / Market Segment is bilaterally agreed.

- **Changes per Underlying Security (optionally per Market / Market Segment too)**

  In this case, typically used for options trading, all options for a certain underlying or option class transition to a new state at the same time. A state change should be interpreted as all Security of the underlying transition to the relevant state instantaneously.

Trading halts and releases per individual Security are always relayed using the SecurityTradingStatus (326) = "2 - Trading Halt" and the HaltReason (327) fields in the Security Status message.

**Note:**

The concept of a "FIX Session" is a transport level concept and thus totally different from the business layer concept of a "Trading Session".

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:
  - 1 - Day
- TradingSessionSubID (625). Defines the main trading phases within a trading session, typically according to the bullet list shown above.
  - 1 - Pre-Trading
  - 2 - Opening or Opening Auction
  - 3 - (Continuous) Trading
  - 4 - Closing or Closing Auction
  - 5 - Post-Trading
  - 6 - Intraday Auction
  - 7 - Quiescent
  - (100+ - used for specific named Intraday Auctions in a "continuous call auction" market)
- TrdSesStatus (340). Defines the status of the trading phase as defined in the TradingSessionSubID (625):
  - 1 - Halted
  - 2 - Open
  - 3 - Closed. Only used when no other trading (sub) session follows.

A Security is closed for the remainder of the trading session (TradingSessionID) when the SecurityTradingStatus (326) = "18" - Not available for trading (end of session).

A Market Segment is closed for the remainder of the trading session (TradingSessionID) when the TradSesStatus (340) = "3" - Closed.

#### 15.2.4.2 Quiescent Phases

A quiescent phase (TradingSessionSubID [625] = 7) is used when the market is paused. An example of this is the closing of books in order to produce end of day statistics. The state may e.g. include market operations activity as cancelling trades.

#### 15.2.4.3 Halts and other Special States

The SecurityTradingStatus (326) field defines subordinate states for an individual book when such are applicable. The Field is typically used to relay states as:

- 2 - Trading Halt. A trade halt can be further qualified by HaltReason (327), e.g.:
  - D - News Dissemination
  - E - Order Influx
  - M - Additional Information
- 23 - Fast Market

The SecurityTradingStatus (326) field is also used to relay subordinate states used in call auctions, e.g.:

- 100 - Hidden Auction. Means there is no book information (orders, price levels) published
- 101 - Open Auction. Means book information is published
- 102 - Issuer Position Modification. For a description, please refer to Section 11 on page 165.

Please refer to Section 20 on page 273 for a complete list of applicable field values for the SecurityTradingStatus (326) and HaltReason (327) fields.

#### 15.2.4.4 Events

The Security Status and / or Trading Session Status messages are 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) in the Security Status message
- TradSesEvent (20089) in the Trading Session Status message.

## 15.3 Message Details

### 15.3.1 TradingSessionStatus

*Table 154: TradingSessionStatus*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | **StandardHeader** | Y | MsgType = h (lowercase) | |
| 20084 | MarketID | N | | Exchange |
| 20036 | MarketSegmentID | N | | String |
| 336 | TradingSessionID | Y | Identifier for Trading Session | String |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 625 | TradingSessionSubID | N | | String |
| 339 | TradSesMode | N | Trading Session Mode | int |
| 340 | TradSesStatus | Y | State of the trading session | int |
| 20089 | TradSesEvent | N | **OMX Comment:** Currently not in FIX 5.0. OMX requests extension. | int |
| 58 | Text | N | | String |
| | **StandardTrailer** | Y | | |

## 15.3.2 SecurityStatus

*Table 155: SecurityStatus*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | **StandardHeader** | Y | MsgType = f (lowercase) | |
| 20084 | MarketID | N | | Exchange |
| 20036 | MarketSegmentID | N | | String |
| | **Instrument** | Y | Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages" | |
| | **UndInstrmtGrp** | N | Number of underlyings | |
| 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<br>**OMX Comment:** FIX 5.0 SP1 | int |
| 292 | CorporateAction | N | | MultipleCharValue |
| 327 | HaltReason | N | Denotes the reason for the Opening Delay or Trading Halt. | char |
| 328 | InViewOfCommon | N | | Boolean |
| 329 | DueToRelated | N | | Boolean |
| 1021 | MDBookType | N | Used to relay changes in the book type<br>**OMX Comment:** FIX 5.0 SP1 | int |
| 264 | MarketDepth | N | Used to relay changes in Market Depth.<br>**OMX Comment:** FIX 5.0 SP1 | int |
| 60 | TransactTime | N | Trade Dissemination Time | UTCTimestamp |
| 58 | Text | N | Comment, instructions, or other identifying information. | String |
| | **StandardTrailer** | Y | | |

## 15.4 Component Blocks (Trading Session Specific)

For components that are not specific for Market Data, please see Section 19 on page 261.

### 15.4.1 Components

None.

### 15.4.2 Implicit Components

None

## 15.5 Workflows

### 15.5.1 A Trading Day

The following example workflows illustrate the principles used when relaying trading session and state information for a particular Market / Market Segment.

In the below example all security open and close without call auctions, individual transitions do not need to be relayed per security.

*Table 156: Trading Day with Simultaneous Security Transitions*

| Message Used | Trading Session ID | Trading Session Sub ID | Trad Ses Status / Security Trading Status | Comment |
|---|---|---|---|---|
| Trading Session Status | 1 - Day | 1 - Pre-Trading | 2 - Open | First state message of day |
| Trading Session Status | 1 - Day | 3 - (Continuous) Trading | 2 - Open | Normally sent before the first book is opened. |
| Trading Session Status | 1 - Day | 7 - Quiescent | 2 - Open | Sent when quiescent state starts. |
| Trading Session Status | 1 - Day | 5 - Post-Trading | 2 - Open | Sent when post-trading starts |
| Trading Session Status | 1 - Day | 5 - Post-Trading | 3 - Closed | Sent when the Day session is closed down |

In this example the market is opened and closed using call auctions were order prices are disseminated to the market during the auction. Staggered transitions are used, i.e. security open (and close) one by one.

*Table 157: Trading Day with Staggered Security Transitions*

| Message Used | Trading Session ID | Trading Session Sub ID | Trad Ses Status / Security Trading Status | Comment |
|---|---|---|---|---|
| Trading Session Status | 1 - Day | 1 - Pre-Trading | 2 - Open | First state message of the day for a specific market segment |
| Security Status (for each Security) | 1 - Day | 1 - Pre-Trading | N/A | Sent when the order book is in pre-trading |
| Trading Session Status | 1 - Day | 3 - (Continuous) Trading | 2 - Open | Normally sent before the first book is opened |
| Security Status (for each Security) | 1 - Day | 2 - Opening (Auction) | 100 - Hidden auction | Sent when the auction starts |
| Security Status (for each Security) | 1 - Day | 3 - (Continuous) Trading | N/A | Used in staggered opening and sent after the book is uncrossed |
| Security Status (for each Security) | 1 - Day | 4 - Closing (Auction) | 100 - Hidden auction | Sent when the auction starts |
| Security Status (for each Security) | 1 - Day | 7 - Quiescent | 2 - Open | Sent when the order book goes into quiescent state. |
| Trading Session Status | 1 - Day | 7 - Quiescent | 2 - Open | Normally sent when first orderbook enters quiescent state |
| Trading Session Status | 1 - Day | 5 - Post trading | 2 - Open | Normally sent when last order book enters post trading |
| Security Status (for each Security) | 1 - Day | 5 - Post-Trading | N/A | Sent when the order book enters the post trading state. |
| Security Status (for each Security) | 1 - Day | 5 - Post-Trading | 18 - not available for trading (end of session) | Sent when the order book is closed |
| Trading Session Status | 1 - Day | 5 - Post-Trading | 3 - Closed | Sent when the Day session is closed down |

Markets that use intraday call auctions followed by continuous trading are illustrated below.

*Table 158: A Scheduled Intraday Call Auction*

| Message Used | Trading Session ID | Trading Session Sub ID | Trad Ses Status / Security Trading Status | Comment |
|---|---|---|---|---|
| ... | ... | ... | ... | |
| Security Status (for each Security) | 1 - Day | 6 - Intraday Auction | 100 - Hidden Auction | Sent when the auction starts. Example shows a hidden auction. |

| Message Used | Trading Session ID | Trading Session Sub ID | Trad Ses Status / Security Trading Status | Comment |
|---|---|---|---|---|
| Security Status (for each Security) | 1 - Day | 3 - (Continuous) Trading | N/A | Used in staggered auctions and sent after the book is uncrossed |
| ... | ... | ... | ... | |

Naturally the TradingSessionSubID (625) following an intraday call auction could be something other than (Continuous) Trading, e.g. a closed state or another intraday auction.

## 15.5.2 Exception States

The following example illustrates an exceptional state in an order book.

*Table 159: Trading Halt in a Security*

| Message Used | Trading Session ID | Trading Session Sub ID | Security Trading Status/Security Trading Event | Comment |
|---|---|---|---|---|
| ... | ... | ... | ... | ... |
| Security Status (Security X) | any trading session | any state | 2 - Trading Halt | Security Trading Status indicating that a Trade Halt is instated |
| Security Status | any trading session | any state | 3 - Trading Resumes (after halt) | Security Trading Event indicating that the halt is lifted. |
| ... | ... | ... | ... | ... |

The TradingSessionID (336) / TradingSessionSubID (625) values always follow the normal schedule, so while a Trade Halt may still apply the sessions could change.

# 16 News Management

**Note:**

Standard FIX covers News publication. There is not an FPL roadmap for extensions in this area.

OMX user defined messages, fields and workflows are used to extend the functionality.

## 16.1 Business Message Types

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

*Table 160: Business Messages*

| In/Out | Message Name | Comment |
|---|---|---|
| In | **News Publication Request** | Used to submit News for publication. Currently not in FIX. OMX User Defined message |
| Out | **News** | Used to publish news |
| Out | **News Publication Request Reject** | Used too reject a News Publication Request. Currently not in FIX. OMX User Defined message |

## 16.2 Publishing News

### 16.2.1 Main Workflow

#### 16.2.1.1 News Availability

News are made available in three different models:

- Through a separate News feed. In this case users connect to the feed and fetch the News messages as they are published. News may be packaged in separate feeds similar to Market Data (Section 14 on page 189).
- Through subscriptions. In this case the user defines the filter for what type of News to receive. The News Data Request message is used to query and establish or terminate a subscription.
- Through unsolicited publication. In this case News are published together with Market Data (see Section 14 on page 189).

Different types of News can be published using different models, subject to marketplace set-up. Please refer to separate documentation.

#### 16.2.1.2 Requesting Publication

Certain parties are authorized to request the publication of News messages. Such requests are sent to the marketplace using the News Publication Request message. In the case of a reject the News Publication Request Reject is returned.

### 16.2.2 News Features

#### 16.2.2.1 Filtering Criteria

News messages can be associated with a set of criteria to enable filtering and sorting of News messages. The criteria include:

- News Category
- Security Exchange
- Market Segment
- Instrument, including Underlying and / or Legs in the case that is relevant

News may also be published in a specific language, or multiple languages (one News message per language).

#### 16.2.2.2 News Text

The News text includes four options that can be used one by one or in combinations:

- Lines of (ASCII) text through the Text (58) and/or EncodedText (355) fields
- URLLink (149), i.e. a link to a web page
- RawData (96) which can include bitmaps, PDF or Word documents, etc
- XMLData (213) which can include xml-formatted data

A News message can also be related to previously sent out News messages. Links to such messages are provided through the RefNewsID (20067) and RefNewsType (20068) fields.

## 16.3 Message Details

### 16.3.1 News

*Table 161: News*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | **StandardHeader** | Y | MsgType = B | |
| 20063 | NewsID | N | Uniquely identifies a news message<br>**OMX Comment:** Currently not in FIX 5.0. OMX requests extension. | String |
| 42 | OrigTime | N | | UTCTimes-tamp |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 61 | Urgency | N | | char |
| 20064 | NewsCategory | N | **OMX Comment:** Currently not in FIX 5.0. OMX requests extension. | int |
| 20084 | MarketID | N | **OMX Comment:** Currently not in FIX 5.0. OMX requests extension. | Exchange |
| 20036 | MarketSegmentID | N | **OMX Comment:** Currently not in FIX 5.0. OMX requests extension. | String |
| 20065 | LanguageCode | N | **OMX Comment:** Currently not in FIX 5.0. OMX requests extension. | String |
| 148 | Headline | Y | Specifies the headline text | String |
| | **InstrmtGrp** | N | Specifies the number of repeating symbols (instruments) specified | |
| | **InstrmtLegGrp** | N | Number of legs<br>Identifies a Multi-leg Execution if present and non-zero. | |
| | **UndInstrmtGrp** | N | Number of underlyings | |
| | **LinesOfTextGrp** | Y | Specifies the number of repeating lines of text specified | |
| 149 | URLLink | N | A URL (Uniform Resource Locator) link to additional information (i.e. http://www.XYZ.com/research.html) | String |
| | **RefNewsGrp** | N | **OMX Comment:** Currently not in FIX 5.0. OMX requests extension. | |
| | **StandardTrailer** | Y | | |

## 16.3.2 NewsPublicationRequest

*Table 162: NewsPublicationRequest*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | **StandardHeader** | N | MsgType = U7 | |
| 20072 | NewsPublReqID | N | Uniquely identifies a news publication request message | String |
| 20073 | OrigNewsPublReqID | N | Used when updating or canceling a publication request (before it is published) | String |
| 20074 | NewsUpdAction | N | Type of News update action. | int |
| | **Parties** | N | Insert here the set of "Parties" fields defined in "COMMON COMPONENTS OF APPLICATION MESSAGES" | |
| 42 | OrigTime | N | | UTCTimestamp |
| 20075 | PublTime | N | Requested Publication date and time | UTCTimestamp |
| 61 | Urgency | N | | char |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 20064 | NewsCategory | N | | int |
| 20084 | MarketID | N | | Exchange |
| 20036 | MarketSegmentID | N | | String |
| 20065 | LanguageCode | N | | String |
| 148 | Headline | N | Specifies the headline text | String |
| | **InstrmtGrp** | N | Specifies the number of repeating symbols (instruments) specified | |
| | **InstrmtLegGrp** | N | Number of legs<br>Identifies a Multi-leg Execution if present and non-zero. | |
| | **UndInstrmtGrp** | N | Number of underlyings | |
| | **LinesOfTextGrp** | N | Specifies the number of repeating lines of text specified | |
| 149 | URLLink | N | A URL (Uniform Resource Locator) link to additional information (i.e. http://www.XYZ.com/research.html) | String |
| | **RefNewsGrp** | N | | |
| | **StandardTrailer** | N | | |

## 16.3.3 NewsPublicationRequestReject

*Table 163: NewsPublicationRequestReject*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | **StandardHeader** | N | MsgType = U8 | |
| 20072 | NewsPublReqID | N | Uniquely identifies a news publication request message | String |
| 20073 | OrigNewsPublReqID | N | Used when updating or canceling a publication request (before it is published) | String |
| 20074 | NewsUpdAction | N | Type of News update action. | int |
| 20076 | NewsPublRejReason | N | | int |
| | **Parties** | N | Insert here the set of "Parties" fields defined in "COMMON COMPONENTS OF APPLICATION MESSAGES" | |
| 42 | OrigTime | N | | UTCTimestamp |
| 20075 | PublTime | N | Requested Publication date and time | UTCTimestamp |
| 61 | Urgency | N | | char |
| 20064 | NewsCategory | N | | int |
| 20084 | MarketID | N | | Exchange |
| 20036 | MarketSegmentID | N | | String |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 20065 | LanguageCode | N | | String |
| 148 | Headline | N | Specifies the headline text | String |
| | **InstrmtGrp** | N | Specifies the number of repeating symbols (instruments) specified | |
| | **InstrmtLegGrp** | N | Number of legs<br>Identifies a Multi-leg Execution if present and non-zero. | |
| | **UndInstrmtGrp** | N | Number of underlyings | |
| | **LinesOfTextGrp** | N | Specifies the number of repeating lines of text specified | |
| 149 | URLLink | N | A URL (Uniform Resource Locator) link to additional information (i.e. http://www.XYZ.com/research.html) | String |
| | **RefNewsGrp** | N | | |
| | **StandardTrailer** | N | | |

# 16.4 Component Blocks (News Specific)

For components that are not specific for News, please see Section 19 on page 261.

## 16.4.1 Components

None.

## 16.4.2 Implicit Components

### 16.4.2.1 InstrmtGrp

*Table 164: InstrmtGrp*

| 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" | |

### 16.4.2.2 LinesOfTextGrp

*Table 165: LinesOfTextGrp*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 33 | NoLinesOfText | Y | Specifies the number of repeating lines of text specified | NumInGroup |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| >58 | Text | Y | Repeating field, number of instances defined in LinesOfText | String |

#### 16.4.2.3 NewsInstrsGrp

*Table 166: NewsInstrsGrp*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 20071 | NoNewsInstrs | Y | Specifies the number of repeating instruments | NumInGroup |
| | **Instrument** | Y | Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages" | |
| | **UnderlyingInstrument** | N | | |

#### 16.4.2.4 RefNewsGrp

*Table 167: RefNewsGrp*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 20066 | NoRefNews | N | Specifies the number of repeating news references<br>**OMX Comment:** Currently not in FIX 5.0. OMX requests extension. | NumInGroup |
| >20067 | RefNewsID | N | References the NewsID (20063) of another News message<br>**OMX Comment:** Currently not in FIX 5.0. OMX requests extension. | String |
| >20068 | RefNewsType | N | The type of reference<br>**OMX Comment:** Currently not in FIX 5.0. OMX requests extension. | int |

## 16.5 Workflows

### 16.5.1 Requesting the Publication of News

Eligible participants can request that news are published. The below diagram depicts the workflow:


Originator
Marketplace
News Consumer
News Publication Request
NewsPublReqID <new>
NewsUpdAction "New"
If rejceted
News Publication Request Reject
NewsPublID <originators>
NewsUpdAction "New"
NewsPublRejReason "..."
News
NewsID <new>
Headline <...>
...

# 17 Reference Data

**Note:**

The FIX standard historically has included a limited set of reference data. OMX and other actors are currently engaged in discussions with the FPL Global Exchanges and Markets Committee and the Global Derivatives Committee (GDC) Product Reference Working Group to expand and clarify the scope of Reference Data. This chapter is based on the latest state of the following proposals from those groups:

- GDC Product Reference. The scope is a product reference model for the purpose of representing financial instruments and the relationship between these instruments for the listed derivatives industry. This product reference model provides support for derivatives trading that may include Futures, Options on Futures and Equity Options.
- EEWG Market Segmentation. The scope is a market structure model that is applicable across asset classes.

This chapter is provided on a best effort basis to give an indication of the final specification.

OMX user defined messages, fields and workflows are used to extend the functionality.

## 17.1 Business Message Types

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

*Table 168: Business Messages*

| In/Out | Message Name | Comment |
|---|---|---|
| Out | **Market Segment** | Used to relay the market structure in the form of markets (exchanges or marketplaces) and market segments. Currently not in FIX, the FPL Global Exchanges and Markets Committee proposes an extension |
| Out | **Market Segment Update Report** | Used to relay individual updates to the Market / Market Segments. Currently not in FIX, the FPL Global Exchanges and Markets Committee proposes an extension |
| Out | **Security List** | Used to relay a list of securities - e.g. grouped per Market / Market Segment |
| Out | **Security List Update Report** | Used to relay individual updates to a securities list |
| Out | **Derivative Security List** | Used to relay a list of derivatives where the underlying and the derivatives class is at the main level of the message and the individual series / strikes are listed per underlying |
| Out | **Derivative Security List Update Report** | Used to relay individual updates to the Derivative Security List. Currently not in FIX, the FPL Global Derivatives Committee proposes an extension. |
| Out | **Security Definition** | Used to either:<br>• acknowledge or reject a Security Definition Request or<br>• provide details about an individual security |
| Out | **Security Definition Update Report** | Used to relay individual updates to a security |

| In/Out | Message Name | Comment |
|---|---|---|
| Out | **Trading Session List** | Used to relay a list of Trading Sessions comprising the trading schedule and grouped per Market / Market Segment |
| Out | **Trading Session List Update Report** | Used to relay individual updates to the Trading Session List |
| Out | **Participant List** | Used to relay a list of Participants. Currently not in FIX, the FPL Global Exchanges and Markets Committee proposes an extension. |
| Out | **Participant List Update Report** | Used to relay individual updates to the Participants List. Currently not in FIX, the FPL Global Exchanges and Markets Committee proposes an extension. |

## 17.2 Publication of Reference Data

### 17.2.1 Introduction

Product Reference Data is an essential aspect of securities automation whether the context is electronic trading or clearing. A good product interface reflects the structure of the instruments offered by an organization and provides a thorough definition of each instrument.

Product Reference Data is provided as a base set of information that must be established at a customer site prior to conducting business with the entity that is offering those products; usually an exchange or clearing entity.

Users of Product Reference Data should be able to establish a local set of the market structure and instruments that precisely reflects the products being offered by an exchange or clearing entity. It is important to note that these instruments may also be referred to as “contracts” and carry a legal obligation to fulfill the specified terms of the contract. For Listed Derivatives, this involves a broad array of rules that govern where and how the instrument is listed, quoted, traded, cleared, settled and delivered.

The main messages used are:

- **Market Segment**

  Used to provide reference data about:

  - Markets, i.e. Exchanges or other marketplaces. Relevant in cases where a single FIX connection provides access to multiple marketplaces.
  - Market Segments. A marketplace can subdivide its entire market into separate segments (or venues).

  Reference data relating to markets and market segments can include default trading rules.

- **Trading Session List**

  Used to provide the trading day schedule for a specific Market / Market Segment. Can includes default trading rules that are specific for the respective sessions and sub sessions. Trading rules specified here override rules specified in the Market Segment message.

- **Security List**

  Used for a variety of purposes including:

  - Listing the security of a specified Market / Market Segment. In this case may include specific trading rules applicable for the specified Market / Market Segment or even Trading (Sub) Session.

Trading rules specified here overrides rules specified in the Market Segment and Trading Session List messages.

- Listing the security of publication list (e.g. the official "Turnover List")
- Listing the security for other grouping purposes

- **Derivative Security List**

  Used to define classes of derivatives (based on an underlying) and the individual series / strikes of those classes. The underlying is itself defined using the Security Definition. Trading rules specified here override rules specified in the Market Segment and Trading Session List messages

- **Security Definition**

  Used to define an individual security. Can be used to specify the trading rules for the security, especially relevant in cases where those are not relayed in a Security List.

  Can be used as an alternative to the Derivative Security List in defining the individual series / strikes of a derivatives class. Relevant e.g. for user-defined derivative security.

**Note:**

In cases when the marketplace trades the same (fungible) security in multiple market segments a set of Security Definition messages are issued:

- One representing the "order book" of each market segment
  - Either with the fungible properties repeated for every Security Definition
  - Or a separate message for the fungible properties. In this case the fungible Security is identified with a SecurityID (48) + SecurityIDSource (22) and referenced from the "order books" via SecurityAltID (455) + SecurityAltIDSource (456). Usually a Security(Alt)IDSource of an ISIN code, a CUSIP or similar will be used for this purpose.

The SecurityID (48) + SecurityIDSource (22) = "99 - Marketplace assigned Identifier" will be different for each "order book". As orders, quotes etc must be routed to the correct book, the identifier of the fungible security is not an allowed ID in those cases.

## 17.2.2 Main Workflow

### 17.2.2.1 “Start of Day” Download of Securities

Securities are downloaded the following way:

- Push. The marketplace publishes traded instruments using the Market Segment, Trading Session List, Security Definition, Security List and Derivative Securities List messages. The Security Type and Participant List messages may also be used. Reference data may, as Market Data (Section 14 on page 189), be packaged in separate feeds, e.g. per marketplace or market segment - but can also be provided outside the scope of a FIX session, e.g. through a file transfer.
- Pull. The user requests a list of securities through the Market Segment Request, Trading Session List Request, Security List Request (or the Derivative Security List Request) message. The marketplace responds with a Market Segment, Trading Session List, Security List (and 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.

An example sequence of messages is:

- A Security Definition message for each Instrument traded
- A Market Segment message for each Market
    - A Market Segment message for each segment per the Market
        - A Trading Session List message for each Market Segment, listing the applicable trading sessions and subsessions
        - A Security List message for each Market Segment, listing the applicable order books (securities)
        - A Derivative Security List for each derivatives class, containing the individual series / strikes that are traded at the Market Segment

The download typically occurs at start of day, week or other relevant period subject to bilateral agreement. In between downloads, real time updates can be provided through the associated Update Report messages.

#### 17.2.2.2 Continuous Updates of Securities

The marketplace may continuously relay updates to the "start of day" download, either subscription based or as a part of a separate reference data feed:

- Security Definition Update Report
- Derivative Security List Update Report
- Market Segment Update Report
- Trading Session List Update Report
- Security List Update Report
- Participant List Update Report

Updates are published in the following way:

- Push. The marketplace publishes Market Segment Update Report, Trading Session Update Report, Security Definition Update Report, Security List Update Report and Participant List Update Report messages when a updates occur. Updates may again be packaged into separate feeds.

#### 17.2.2.3 User-Defined Instruments

Subject to bilateral agreement, a user can request the registration of a user-defined security (including multileg 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) as described above.

#### 17.2.2.4 Security Status

Trading Session Status and Security Status messages are published over Market Data feeds when the trading session or state changes. See Section 15 on page 203 for details.

## 17.2.3 Reference Data Features

### 17.2.3.1 Extended Attributes

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

### 17.2.3.2 Note Codes

So called "Note Codes" are used in many front-end trading systems to indicate that a Security is subject to some event that may affect its price.

The source of those note codes is generally the Corporate Action (292) and EventType (865) fields in the Securities messages depending on whether the event is classified as a corporate action or not. Users should be aware that note codes can be modified with the Security Status message intraday.

### 17.2.3.3 Market Hierarchy based Reference Data

Relaying reference data, including security data, in a Market Hierarchy centric way is typically relevant in cases where participants are authorized to or use different trading applications for particular markets or market segments.

The market hierarchy is relayed using messages as depicted in the following diagram:

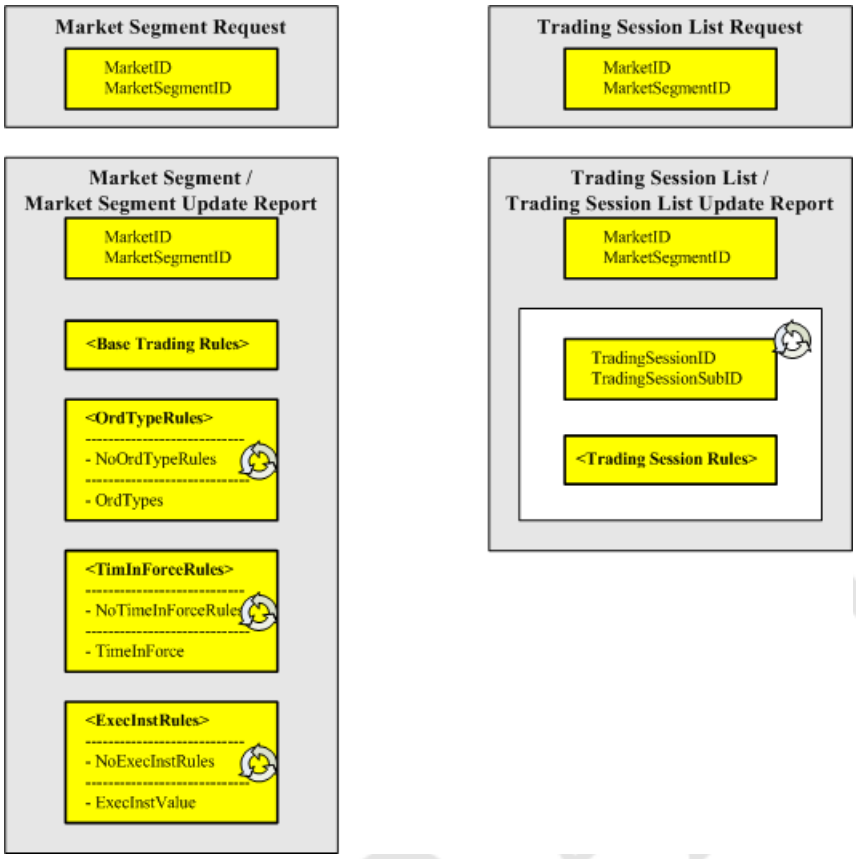


*Figure 33: Market Hierarchy Messages*

In addition the messages in the diagram, the Security List and Derivative Security List messages are used to list the individual security (or order books) of a specified Market / Market Segment. The individual securities can then have trading rules specified in which case those override the rules at higher levels.

### 17.2.3.4 Security Hierarchy based Reference Data

Relaying reference data in a Security centric way is mainly relevant when market segmentation is not in place or when all markets are accessed by a single trading application.

The Security messages have the ability to specify trading rules by Market Segment and Trading Session. Requests can be issued using MarketID and MarketSegmentID in which case the response messages will contain the specified Markets / Segments only.

Trading rules have been separated into two components; <Base Trading Rules> which contain the basic set of trading rules for a given Market Segment, and <Trading Session Rules Grp> which contains trading rules which are specific to a Trading Session. The message structures shown below include the Security Definition set, Derivative Security List set, and Security List set.

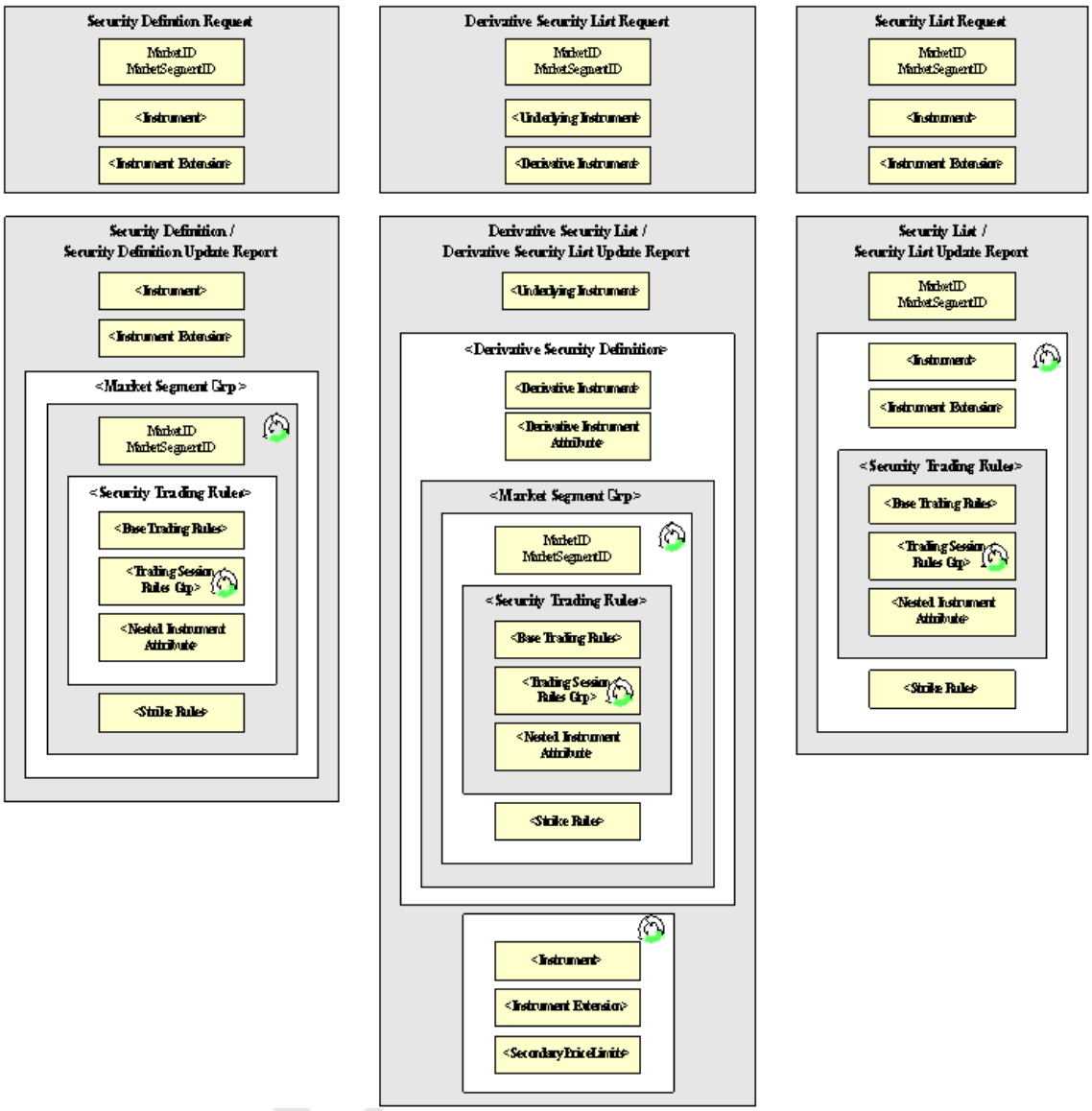


*Figure 34: Security Hierarchy Messages*

### 17.2.3.5 <Trading Rules> Component Blocks

The diagrams below illustrate the trading rules component blocks used in reference data messages. The <Base Trading Rules> component contains trading rules that are specified at the level of Market Segment. The <Trading Session Rules> component contains trading rules that are specified at the Trading Session level. Each component contains its own set of individual trading rules components.

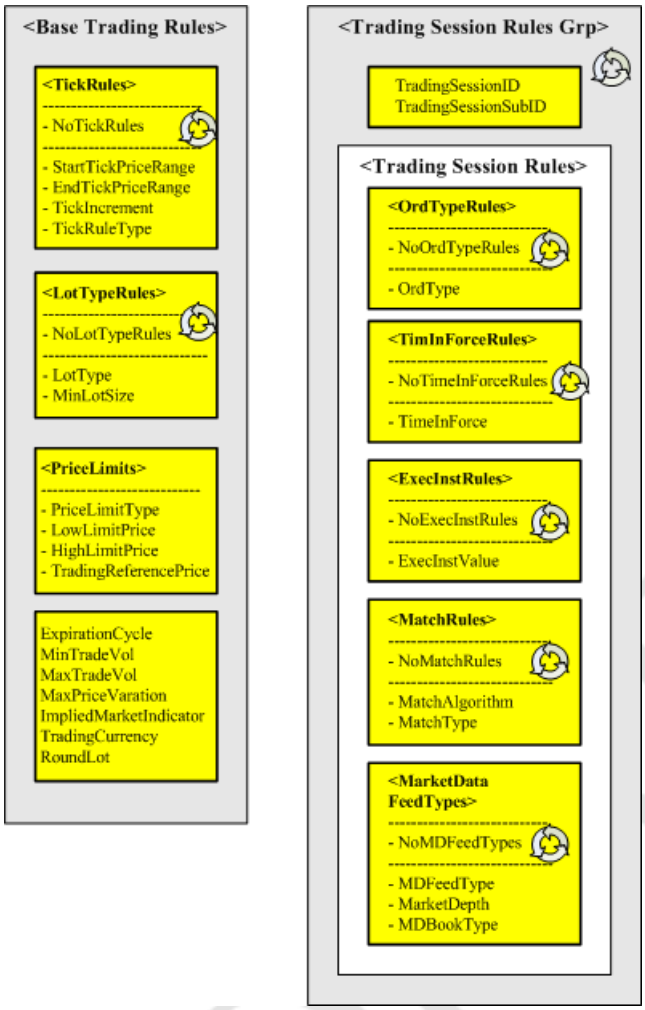


*Figure 35: Trading Rules Components*

### 17.2.3.6 Security Relationships

A security can have a relationship to another security. The below diagram exemplifies this:

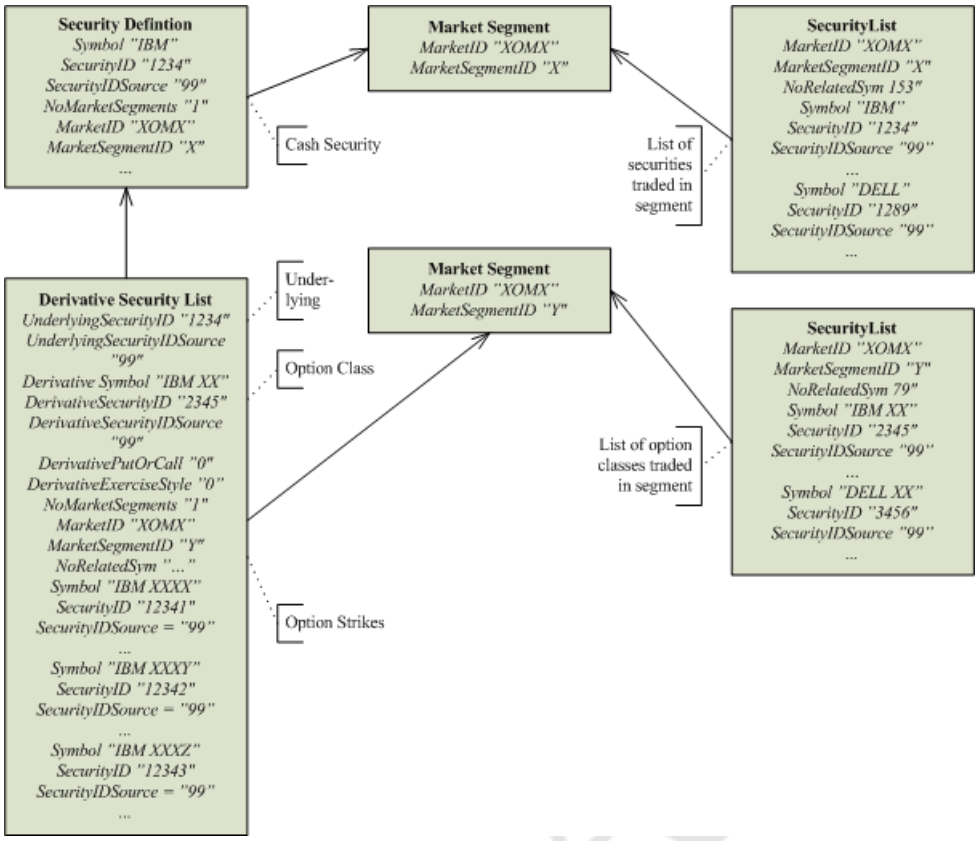


*Figure 36: Security Relationships*

The "IBM XX" option class is related to the underlying security, "IBM". While the underlying is represented with a Security Definition message, the option class is defined as a part of the Derivative Security List.

The individual strikes or series of the option class are only represented through the Derivatives Security List message. The "IBM XX" option class is the underlying in this message.

The following diagram represents a securitized strategy (a.k.a. "multileg" or "combination") for a so called Bull Call Spread:

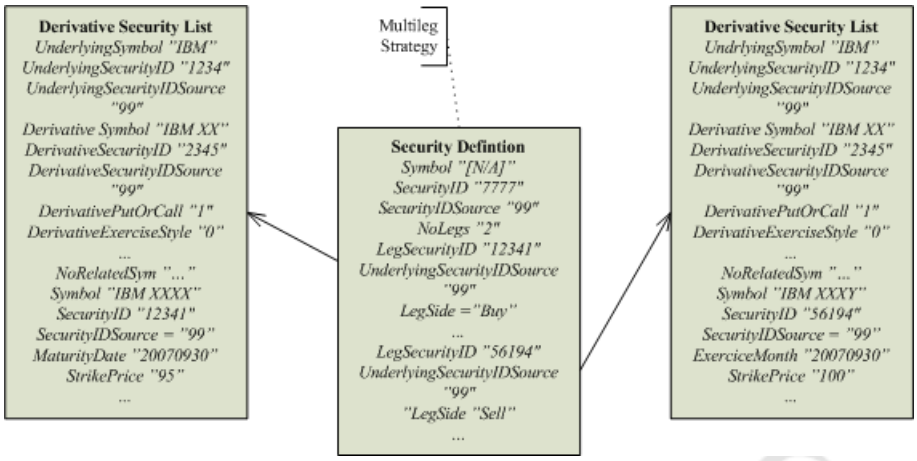


*Figure 37: A Security Strategy*

In this case the the strategy (SecurityID = 7777) itself has not been assigned a symbol. The legs refer to security defined either through Security Definition messages or, in the case of strikes, through the individual strikes of a Derivative Security List. (Establishing a bull call spread involves the purchase of a call option on a particular underlying stock, while simultaneously writing a call option on the same underlying stock with the same expiration month, at a higher strike price)

## 17.3 Message Details

### 17.3.1 DerivativeSecurityList

*Table 169: 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 |
| 20084 | MarketID | N | | Exchange |
| 20036 | MarketSegmentID | N | | String |
| | **UnderlyingInstrument** | N | Underlying security for which derivatives are being returned | |
| | **DerivativeSecurityDefinition** | N | Group block which contains all information for an option family. If provided DerivativeSecurityDefinition qualifies the strikes specified in the Instrument block.<br>**OMX Comment:** Not in FIX. The GDC request addition | |
| 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 |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 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 | | |

## 17.3.2 DerivativeSecurityListUpdateReport

*Table 170: DerivativeSecurityListUpdateReport*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | **StandardHeader** | Y | MsgType = UH | |
| 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 |
| 980 | SecurityUpdateAction | N | Specifies New (0), Cancel (1) or Replace (2). Updates can be applied to Underlying or option class. If Series information provided, then Series has explicitly changed | char |
| | **UnderlyingInstrument** | N | Underlying security which qualifies option family of DerivativeInstrumentGrp is specified. If DerivativeInstrumentGrp is not specified it directly qualifies strikes specified in Instrument block. | |
| | **DerivativeSecurityDefinition** | N | Group block which contains all information for an option family. If provided DerivativeSecurityDefinition qualifies the strikes specified in the Instrument block. | |
| 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 |
| | **DerSecLstUpdRelSymGrp** | N | | |
| | **StandardTrailer** | Y | | |

## 17.3.3 MarketSegment

*Table 171: MarketSegment*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | **StandardHeader** | Y | MsgType = U1 | |
| 20226 | MktSegmMsgID | Y | Unique identifier for each Market egment message | String |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 20085 | MktSegmReqID | N | | String |
| 20084 | MarketID | N | | Exchange |
| 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 |
| 15 | Currency | N | The default trading currency | Currency |
| | **BaseTradingRules** | N | | |
| | **OrdTypeRules** | N | | |
| | **TimeInForceRules** | N | | |
| | **ExecInstRules** | N | | |
| 60 | TransactTime | N | | UTCTimestamp |
| 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 | | |

## 17.3.4 MarketSegmentUpdateReport

*Table 172: MarketSegmentUpdateReport*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | **StandardHeader** | Y | MsgType = UD | |
| 20226 | MktSegmMsgID | Y | Unique identifier for each Market egment message | String |
| 20085 | MktSegmReqID | N | | String |
| 20227 | MktSegmUpdateAction | N | Specifies the action taken | char |
| 20084 | MarketID | N | | Exchange |
| 20036 | MarketSegmentID | N | | String |
| 20037 | MarketSegmentDesc | N | | String |
| 20038 | EncodedMktSegmDescLen | N | | Length |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 20039 | EncodedMktSegmDesc | N | | data |
| 20040 | ParentMktSegmID | N | | String |
| 15 | Currency | N | | Currency |
| | **BaseTradingRules** | N | | |
| | **OrdTypeRules** | N | | |
| | **TimeInForceRules** | N | | |
| | **ExecInstRules** | N | | |
| 60 | TransactTime | N | | UTCTimes-tamp |
| 58 | Text | N | | String |
| 354 | EncodedTextLen | N | | Length |
| 355 | EncodedText | N | | data |
| | **StandardTrailer** | Y | | |

## 17.3.5 ParticipantList

*Table 173: 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 | | |

## 17.3.6 ParticipantListUpdateReport

*Table 174: ParticipantListUpdateReport*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | **StandardHeader** | Y | MsgType = UJ | |
| 20079 | PartyReqID | N | | String |
| 20080 | PartyReportID | N | | String |
| 20081 | PartyRequestResult | N | | int |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 20251 | PartyListUpdateAction | N | | char |
| 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 | | |

## 17.3.7 SecurityDefinition

*Table 175: 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 |
| | **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 | |
| | **MarketSegmentGrp** | N | Insert here the set of "SecurityTradingDefinition" fields defined in "Common Components of Application Messages" | |
| | **StandardTrailer** | Y | | |

## 17.3.8 SecurityDefinitionUpdateReport

*Table 176: 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 |
| 980 | SecurityUpdateAction | N | Specifies New (0), Cancel (1) or Replace (2) | char |
| 292 | CorporateAction | N | Identifies the type of Corporate Action | MultipleCharValue |
| | **Instrument** | N | | |
| | **UnderlyingInstrument** | N | | |
| 15 | Currency | N | | 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 | | |
| | **MarketSegmentGrp** | N | Insert here the set of " SecurityTradingDefinition" fields defined in " COMMON COMPONENTS OF APPLICATION MESSAGES " | |
| | **StandardTrailer** | Y | | |

## 17.3.9 SecurityList

*Table 177: SecurityList*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | **StandardHeader** | Y | MsgType = y (lowercase Y) | |
| 20235 | SecListID | N | **OMX Comment:** Not in FIX. OMX request addition | String |
| 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 |
| 20236 | SecListDesc | N | **OMX Comment:** Not in FIX. OMX request addition | String |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 20237 | SecListType | N | **OMX Comment:** Not in FIX. OMX request addition | int |
| 20238 | PartOfSecListID | N | **OMX Comment:** Not in FIX. OMX request addition | String |
| 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 |
| 20084 | MarketID | N | | Exchange |
| 20036 | MarketSegmentID | N | | String |
| | **SecListGrp** | N | Specifies the number of repeating symbols (instruments) specified | |
| | **StandardTrailer** | Y | | |

## 17.3.10 SecurityListUpdateReport

*Table 178: SecurityListUpdateReport*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | **StandardHeader** | Y | MsgType = BK | |
| 20235 | SecListID | N | **OMX Comment:** Not in FIX. OMX request addition | String |
| 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 |
| 20084 | MarketID | N | | Exchange |
| 20036 | MarketSegmentID | N | | String |
| | **SecLstUpdRelSymGrp** | N | Specifies the number of repeating symbols (instruments) specified | |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 20236 | SecListDesc | N | **OMX Comment:** Not in FIX. OMX request addition | String |
| 20237 | SecListType | N | **OMX Comment:** Not in FIX. OMX request addition | int |
| 20238 | PartOfSecListID | N | **OMX Comment:** Not in FIX. OMX request addition | String |
| | **StandardTrailer** | Y | | |

### 17.3.11 TradingSessionList

*Table 179: 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 | | |

### 17.3.12 TradingSessionListUpdateReport

*Table 180: TradingSessionListUpdateReport*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | **StandardHeader** | Y | MsgType = UE | |
| 335 | TradSesReqID | N | Provided for a response to a specific Trading Session List Request message (snapshot). | String |
| 20225 | TradSesUpdateAction | N | Specifies the action taken | char |
| | **TrdSessLstGrp** | N | | |
| | **StandardTrailer** | Y | | |

## 17.4 Component Blocks (Reference Data Specific)

For components that are not specific for Reference Data, please see Section 19 on page 261.

## 17.4.1 Components

### 17.4.1.1 BaseTradingRules

*Table 181: BaseTradingRules*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | **TickRules** | N | This block specifies the rules for determining how a security ticks, i.e. the price increments at which it can be quoted and traded, depending on the current price of the security | |
| | **LotTypeRules** | N | Specifies the lot types that are valid for trading. The scope of the rule is determined by the context in which the component is used. | |
| | **PriceLimits** | N | Specifies the price limit rules that are valid for trading. The scope of the rule is determined by the context in which the component is used. | |
| 827 | ExpirationCycle | N | Manner in which the security trading eligibility will expire. New value allows previously specified last eligible trade date and time to be indicated | int |
| 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. For listed derivatives this indicates the minimum quantity necessary for an order or trade to qualify as a block trade | Qty |
| 1143 | MaxPriceVariation | N | The maximum price variation of an execution from one event to the next for a given security. Expressed in absolute price terms. | Float |
| 1144 | ImpliedMarketIndicator | N | Commonly used in listed derivatives. Indicates that an implied market should be created for either the legs of a multi-leg instrument (Implied-out) or for the multi-leg instrument based on the existence of the legs (Implied-out). Determination as to whether implied markets should be created is generally done at the level of the multi-leg instrument | int |
| 20203 | TradingCurrency | N | Used when the trading currency can differ from the price currency | Currency |
| 561 | RoundLot | N | Trading lot size of security | Qty |
| 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.<br>**OMX Comment:** Currently not in FIX 5.0. OMX requests extension. | int |

### 17.4.1.2 DerivativeInstrument

*Table 182: DerivativeInstrument*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 20117 | DerivativeSymbol | N | | String |
| 20118 | DerivativeSymbolSfx | N | | String |
| 20119 | DerivativeSecurityID | N | | String |
| 20120 | DerivativeSecurityIDSource | N | | String |
| | **DerSecAltIDGrp** | N | | |
| 20124 | DerivativeProduct | N | | int |
| 20199 | DerivativeProductComplex | N | | String |
| 20202 | DerivFlexProductEligibili-tyIndicator | N | | Boolean |
| 20125 | DerivativeSecurityGroup | N | | String |
| 20126 | DerivativeCFICode | N | | String |
| 20127 | DerivativeSecurityType | N | | String |
| 20128 | DerivativeSecuritySubType | N | | String |
| 20129 | DerivativeMaturityMon-thYear | N | | month-year |
| 20130 | DerivativeMaturityDate | N | | LocalMktDate |
| 20131 | DerivativeMaturityTime | N | | TZTimeOnly |
| 20132 | DerivativeSettleOnOpen-Flag | N | | String |
| 20133 | DerivativeInstrmtAssign-mentMethod | N | | char |
| 20134 | DerivativeSecurityStatus | N | | String |
| 20135 | DerivativeIssueDate | N | | LocalMktDate |
| 20136 | DerivativeInstrRegistry | N | | String |
| 20137 | DerivativeCountryOfIssue | N | | Country |
| 20138 | DerivativeStateOrProvince-OfIssue | N | | String |
| 20139 | DerivativeLocaleOfIssue | N | | String |
| 20141 | DerivativeStrikePrice | N | | Price |
| 20142 | DerivativeStrikeCurrency | N | | Currency |
| 20143 | DerivativeStrikeMultiplier | N | | float |
| 20144 | DerivativeStrikeValue | N | | float |
| 20145 | DerivativeOptAttribute | N | | char |
| 20146 | DerivativeContractMultiplier | N | | float |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 20147 | DerivativeMinPriceIncrement | N | | float |
| 20148 | DerivativeMinPriceIncrementAmount | N | | Amt |
| 20149 | DerivativeUnitofMeasure | N | | String |
| 20150 | DerivativeUnitofMeasureQty | N | | Qty |
| 20213 | DerivativePriceUnitOfMeasure | N | | String |
| 20214 | DerivativePriceUnitOfMeasureQty | N | | Qty |
| 20215 | DerivativeSettlMethod | N | | char |
| 20218 | DerivativePriceQuoteMethod | N | | String |
| 20219 | DerivaticeFuturesValueTypeCode | N | | String |
| 20220 | DerivativeListMethod | N | | int |
| 20221 | DerivativeCapPrice | N | | Price |
| 20222 | DerivativeFloorPrice | N | | Price |
| 20216 | DerivativePutOrCall | N | | int |
| 20151 | DerivativeExerciseStyle | N | | int |
| 20152 | DerivativeCashAmount | N | | Price |
| 20153 | DerivativeTimeUnit | N | | String |
| 20154 | DerivativeSecurityExchange | N | | Exchange |
| 20155 | DerivativePositionLimit | N | | int |
| 20156 | DerivativeNTPositionLimit | N | | int |
| 20157 | DerivativeIssuer | N | | String |
| 20158 | DerivativeEncodedIssuerLen | N | | Length |
| 20159 | DerivativeEncodedIssuer | N | | data |
| 20160 | DerivativeSecurityDesc | N | | String |
| 20161 | DerivativeEncodedSecurityDescLen | N | | Length |
| 20162 | DerivativeEncodedSecurityDesc | N | | data |
| 20163 | DerivativeSecurityXMLLen | N | | Length |
| 20164 | DerivativeSecurityXML | N | | data |
| 20165 | DerivativeSecurityXMLSchema | N | | data |
| 20166 | DerivativeContractSettlMonth | N | | month-year |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | **DerEvntGrp** | N | | |
| | **DerivativeInstrumentParties** | N | | |

### 17.4.1.3 DerivativeInstrumentParties

*Table 183: DerivativeInstrumentParties*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 20173 | NoDerivativeInstrumentParties | N | Repeating group below should contain unique combinations of DerivativeInstrumentPartyID, DerivativeInstrumentPartyIDSource, and DerivativeInstrumentPartyRole | NumINGroup |
| >20174 | DerivativeInstrumentPartyID | N | Used to identify party id related to instrument series | String |
| >20175 | DerivativeInstrumentPartyIDSource | N | Used to identify source of instrument series party id | char |
| >20176 | DerivativeInstrumentPartyRole | N | Used to identify the role of instrument series party id | int |

### 17.4.1.4 DerivativeSecurityDefinition

*Table 184: DerivativeSecurityDefinition*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | **DerivativeInstrument** | N | Optional block which can be used to to summarize common attributes shared across a set of option instruments which belong to the same series. | |
| | **DerivativeInstrumentAttribute** | N | Additional attribution for the instrument series | |
| | **MarketSegmentGrp** | N | Security trading and listing attributes for the series level | |

### 17.4.1.5 ExecInstRules

*Table 185: ExecInstRules*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 20182 | NoExecInstRules | N | Number of execution instructions | NumInGroup |
| >20183 | ExecInstValue | N | Indicates execution instructions that are valid for the specified market segment | char |

### 17.4.1.6 FinancingDetails

*Table 186: FinancingDetails*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 913 | AgreementDesc | N | The full name of the base standard agreement, annexes and amendments in place between the principals and applicable to this deal | String |
| 914 | AgreementID | N | A common reference to the applicable standing agreement between the principals | String |
| 915 | AgreementDate | N | A reference to the date the underlying agreement was executed. | LocalMktDate |
| 918 | AgreementCurrency | N | Currency of the underlying agreement. | Currency |
| 788 | TerminationType | N | For Repos the timing or method for terminating the agreement. | int |
| 916 | StartDate | N | Settlement date of the beginning of the deal | LocalMktDate |
| 917 | EndDate | N | Repayment / repurchase date | LocalMktDate |
| 919 | DeliveryType | N | Delivery or custody arrangement for the underlying securities | int |
| 898 | MarginRatio | N | Percentage of cash value that underlying security collateral must meet. | Percentage |

### 17.4.1.7 InstrumentExtension

*Table 187: InstrumentExtension*

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

### 17.4.1.8 LotTypeRules

*Table 188: LotTypeRules*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 20185 | NoLotTypeRules | N | The number of lot types | NumInGroup |
| >1093 | LotType | N | Defines the lot type assigned to the order. Use as an alternate to RoundLot/561. To be used with MinLotSize/Tbd. LotType + MinLotSize (max is next level minus 1) | char |
| >20197 | MinLotSize | N | Minimum lot size allowed based on lot type specified in LotType/1093 | Qty |

### 17.4.1.9 MarketDataFeedTypes

*Table 189: MarketDataFeedTypes*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 20187 | NoMarketDataFeedTypes | 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 |
| >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<br>**OMX Comment:** If the field is not provided, indicates "no transparency" | int |

### 17.4.1.10 MatchRules

*Table 190: MatchRules*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 20184 | NoMatchRules | N | Number of Match Rules | NumInGroup |
| >1142 | MatchAlgorithm | N | The type of algorithm used to match orders in a specific security on an electronic trading platform. | String |
| >574 | MatchType | N | The point in the matching process at which this trade was matched. | String |

### 17.4.1.11 MaturityRules

*Table 191: MaturityRules*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 20188 | NoMaturityRules | N | Number of maturity rule entries. This block specifies the rules for determining how new strikes should be listed within the stated price range of the underlying instrument | NumInGroup |
| >20196 | MaturityRuleID | N | Allows maturity rule to be referenced via an identifier so that rules do not need to be explicitly enumerated | String |
| >20193 | MMYFormat | N | Format used to generate the MMY for each option contract | int |
| >20192 | MMYIncrementUnitOfMeasure | N | Enumeration specifying the increment unit | int |
| >20189 | StartMMY | N | Starting maturity for the range to which the StrikeIncrement applies. Price refers to the price of the underlying | month-year |
| >20190 | EndMMY | N | Ending maturity monthy year to which the StrikeIncrement applies. Price refers to the price of the underlying | month-year |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| >20191 | MMYIncrement | N | Value by which maturity month year should be incremented within the specified price range. | int |

### 17.4.1.12 OrdTypeRules

*Table 192: OrdTypeRules*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 20180 | NoOrdTypeRules | N | Number of order types | NumInGroup |
| >40 | OrdType | N | Indicates order types that are valid for the specified market segment | char |

### 17.4.1.13 PriceLimits

*Table 193: PriceLimits*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 20208 | PriceLimitType | N | Describes the how the price limits are expressed | int |
| 1148 | LowLimitPrice | N | Allowable low limit price for the trading day. A key parameter in validating order price. Used as the lower band for validating order prices. Orders submitted with prices below the lower limit will be rejected | Price |
| 1149 | HighLimitPrice | N | Allowable high limit price for the trading day. A key parameter in validating order price. Used as the upper band for validating order prices. Orders submitted with prices above the upper limit will be rejected | Price |
| 1150 | TradingReferencePrice | N | Reference price for the current trading price range usually representing the mid price between the HighLimitPrice and LowLimitPrice. The value may be the settlement price or closing price of the prior trading day. | Price |

### 17.4.1.14 SecondaryPriceLimits

*Table 194: SecondaryPriceLimits*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 20209 | SecondaryPriceLimitType | N | Describes the how the price limits are expressed | int |
| 20205 | SecondaryLowLimitPrice | N | Allowable low limit price for the trading day. A key parameter in validating order price. Used as the lower band for validating order prices. Orders submitted with prices below the lower limit will be rejected | Price |
| 20206 | SecondaryHighLimitPrice | N | Allowable high limit price for the trading day. A key parameter in validating order price. Used as the upper band for validating order prices. Orders submitted with prices above the upper limit will be rejected | Price |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 20207 | SecondaryTradingReferen-cePrice | N | Reference price for the current trading price range usually representing the mid price between the HighLimitPrice and LowLimitPrice. The value may be the settlement price or closing price of the prior trading day. | Price |

### 17.4.1.15 SecurityTradingRules

*Table 195: SecurityTradingRules*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | **BaseTradingRules** | N | This block contains the base trading rules | |
| | **TradingSessionRulesGrp** | N | This block contains the trading rules specific to a trading session | |
| | **NestedInstrumentAttribute** | N | Used to describe instrument attributes that are specific to a given market segment. | |

### 17.4.1.16 StrikeRules

*Table 196: StrikeRules*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 20103 | NoStrikeRules | N | Number of strike rule entries. This block specifies the rules for determining how new strikes should be listed within the stated price range of the underlying instrument | NumInGroup |
| >20195 | StrikeRuleID | N | Allows strike rule to be referenced via an identifier so that rules do not need to be explicitly enumerated | String |
| >20104 | StartStrikePxRange | N | Starting price for the range to which the StrikeIncrement applies. Price refers to the price of the underlying | Price |
| >20105 | EndStrilePxRange | N | Ending price of the range to which the StrikeIncrement applies. Price refers to the price of the underlying | Price |
| >20106 | StrikeIncrement | N | Value by which strike price should be incremented within the specified price range. | float |
| >20194 | StrikeExerciseStyle | N | Enumeration that represents the exercise style for a class of options | int |
| | **MaturityRules** | N | Describes the maturity rules for a given set of strikes as defined by StrikeRules | |

### 17.4.1.17 TickRules

*Table 197: TickRules*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 20108 | NoTickRules | N | Number of tick rules. This block specifies the rules for determining how a security ticks, i.e. the | NumInGroup |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | | | price increments at which it can be quoted and traded, depending on the current price of the security | |
| >20109 | StartTickPriceRange | N | Starting price range for specified tick increment | Price |
| >20110 | EndTickPricerange | N | Ending price range for the specified tick increment | Price |
| >20111 | TickIncrement | N | Tick increment for stated price range. Specifies the valid price increments at which a security can be quoted and traded | Price |
| >20112 | TickRuleType | N | Specifies the type of tick rule which is being described | int |

### 17.4.1.18 TimeInForceRules

*Table 198: TimeInForceRules*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 20181 | NoTimeInForceRules | N | Number of time in force techniques | NumInGroup |
| >59 | TimeInForce | N | Indicates time in force techniques that are valid for the specified market segment | char |

### 17.4.1.19 TradingSessionRules

*Table 199: TradingSessionRules*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | **OrdTypeRules** | N | Specifies the order types that are valid for trading. The scope of the rule is determined by the context in which the component is used. In this case, the scope is trading session | |
| | **TimeInForceRules** | N | Specifies the time in force rules that are valid for trading. The scope of the rule is determined by the context in which the component is used. In this case, the scope is trading session | |
| | **ExecInstRules** | N | Specifies the execution instructions that are valid for trading. The scope of the rule is determined by the context in which the component is used. In this case, the scope is trading session | |
| | **MatchRules** | N | Specifies the matching rules that are valid for trading. The scope of the rule is determined by the context in which the component is used. In this case, the scope is trading session | |
| | **MarketDataFeedTypes** | N | Specifies the market data feed types that are valid for trading. The scope of the rule is determined by the context in which the component is used. In this case, the scope is trading session | |

## 17.4.2 Implicit Components

### 17.4.2.1 AttrbGrp

*Table 200: AttrbGrp*

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

### 17.4.2.2 DerivativeInstrumentAttribute

*Table 201: DerivativeInstrumentAttribute*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 20211 | NoDerivativeInstrAttr | N | | NumInGroup |
| >20115 | DerivativeInstrAttribType | N | Code to represent the type of instrument attribute | int |
| >20116 | DerivativeInstrAttribValue | N | Attribute value appropriate to the SeriesInstrAttribType field | String |

### 17.4.2.3 DerEvntGrp

*Table 202: DerEvntGrp*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 20167 | NoDerivativeEvents | N | | NumInGroup |
| >20168 | DerivativeEventType | N | | int |
| >20169 | DerivativeEventDate | N | | LocalMktDate |
| >20170 | DerivativeEventTime | N | | UTCTimestamp |
| >20171 | DerivativeEventPx | N | | Price |
| >20172 | DerivativeEventText | N | | String |

### 17.4.2.4 DerSecAltIDGrp

*Table 203: DerSecAltIDGrp*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 20121 | NoDerivativeSecurityAltID | N | | NumInGroup |
| >20122 | DerivativeSecurityAltID | N | | String |
| >20123 | DerivativeSecurityAltIDSource | N | | String |

### 17.4.2.5 DerSecLstUpdRelSymGrp

*Table 204: DerSecLstUpdRelSymGrp*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 146 | NoRelatedSym | N | Specifies the number of repeating symbols (instruments) specified | NumInGroup |
| >20223 | ListUpdateAction | N | Specifies New (0), Cancel (1) or Replace (2). If provided, then Instrument occurrence has explicitly changed | char |
| | **Instrument** | N | Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages"<br>of the requested Security | |
| | **InstrumentExtension** | N | Insert here the set of "InstrumentExtension" fields defined in "Common Components of Application Messages" | |
| | **SecondaryPriceLimits** | N | Secondary price limit rules | |
| >15 | Currency | N | | Currency |
| | **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 |

### 17.4.2.6 InstrmtLegSecListGrp

*Table 205: 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"<br>Required if NoLegs > 0 | |
| | **LegStipulations** | N | Insert here the set of "LegStipulations" (leg symbology) fields defined in "Common Components of Application Messages"<br>Required if NoLegs > 0<br>**OMX Comment:** Currently not supported | |
| | **LegBenchmarkCurveData** | N | Insert here the set of "LegBenchmarkCurveData" (leg symbology) fields defined in "Common Components of Application Messages"<br>Required if NoLegs > 0<br>**OMX Comment:** Currently not supported | |

### 17.4.2.7 MarketSegmentGrp

*Table 206: MarketSegmentGrp*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 20107 | NoMarketSegments | N | Number of Market Segments on which a security may trade. | NumInGroup |
| >20084 | MarketID | N | Identifies the market which lists and trades the instrument. Set to [N/A] if a specific Market identifier does not need to be specified.<br>String – market identifier<br>Required if NoMarketSegments > 0<br>Value can be set to [N/A] in order to indicate that trading and listing rules apply to all markets | Exchange |
| >20036 | MarketSegmentID | N | Identifies the segment of the market to which the specify trading rules and listing rules apply. The segment may indicate the venue, whether retail or wholesale, or even segregation by nationality. Set to [N/A] if a specific Market Segment identifier does not need to be specified. String – market segment identifier<br>Required if NoMarketSegments > 0<br>Value can be set to [N/A] in order to indicate that trading and listing rules apply to all market segments for the specified market id | String |
| | **SecurityTradingRules** | N | | |
| | **StrikeRules** | N | This block specifies the rules for determining how new strikes should be listed within the stated price range of the underlying instrument. StrikeRules are defined for each MarketSegment since the range of eligible strikes may vary from one segment or venue to the next | |

### 17.4.2.8 NestedInstrumentAttribute

*Table 207: NestedInstrumentAttribute*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 20212 | NoNestedInstrAttr | N | | NumInGroup |
| >20113 | NestedInstrAttribType | N | Code to represent the type of instrument attribute | int |
| >20114 | NestedInstrAttribValue | N | Attribute value appropriate to the NestedInstrAttribType field | String |

### 17.4.2.9 PartyAltIDGrp

*Table 208: PartyAltIDGrp*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 20248 | NoPartyAltID | N | Number of PartyAltID (20249) entries<br>**OMX Comment:** Not in standard FIX. OMX requests an extension | int |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| >20249 | PartyAltID | N | Alternate Party identifier value for this party of PartyAltIDSource (20250) type (e.g. BIC, etc). Requires PartyAltIDSource. Required if NoPartyAltIDs > 0.<br>**OMX Comment:** Not in standard FIX. OMX requests an extension | String |
| >20250 | PartyAltIDSource | N | Identifies class or source of the PartyAltID (20249) value. Required if PartyAltID is specified.<br>**OMX Comment:** Not in standard FIX. OMX requests an extension | char |

### 17.4.2.10 PartyLstGrp

*Table 209: 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.<br>**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 |
| >452 | PartyRole | N | Identifies the type of PartyID (e.g. Executing Broker). Required if NoPartyIDs > 0.<br>**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.<br>**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 |
| | **PartyAltIDGrp** | N | Alternate Party Identifiers | |

### 17.4.2.11 RelSymDerivSecGrp

*Table 210: RelSymDerivSecGrp*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 146 | NoRelatedSym | N | Specifies the number of repeating symbols (instruments) specified | NumInGroup |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | **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 |
| | **InstrumentExtension** | N | Insert here the set of "InstrumentExtension" fields defined in "Common Components of Application Messages"<br>**OMX Comment:** Currently not supported | |
| | **SecondaryPriceLimits** | N | Secondary price limit rules<br>**OMX Comment:** Not in FIX. The GDC request addition | |
| | **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 |

## 17.4.2.12 SecListGrp

*Table 211: 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"<br>**OMX Comment:** Currently not supported | |
| | **FinancingDetails** | N | Insert here the set of "FinancingDetails" fields defined in "Common Components of Application Messages" | |
| | **SecurityTradingRules** | N | Used to provide trading rules<br>**OMX Comment:** Not in FIX. The GDC request addition | |
| | **StrikeRules** | N | Used to provide listing rules<br>**OMX Comment:** Not in FIX. The GDC request addition | |
| | **UndInstrmtGrp** | N | | |
| >15 | Currency | N | | Currency |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| | **Stipulations** | N | Insert here the set of "Stipulations" fields defined in "Common Components of Application Messages"<br>**OMX Comment:** Currently not supported | |
| | **InstrmtLegSecListGrp** | 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 |

### 17.4.2.13 SecLstUpdRelSymGrp

*Table 212: 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 " | |
| | **SecurityTradingRules** | N | Used to provide trading rules<br>**OMX Comment:** Not in FIX. The GDC request addition | |
| | **StrikeRules** | N | Used to provide listing rules<br>**OMX Comment:** Not in FIX. The GDC request addition | |
| | **SecLstUpdRelSymsLegGrp** | 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 |

### 17.4.2.14 SecLstUpdRelSymsLegGrp

*Table 213: 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 | |

### 17.4.2.15 TradingSessionRulesGrp

*Table 214: TradingSessionRulesGrp*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 20204 | NoTradingSessions | N | Allows trading rules to be expressed by trading session | NumInGroup |
| >336 | TradingSessionID | N | Identifier for the trading session<br>Must be provided if NoTradingSessions > 0<br>Set to [N/A] if values are not specific to trading session | String |
| >625 | TradingSessionSubID | N | Identifier for the trading session<br>Set to [N/A] if values are not specific to trading session sub id | String |
| | **TradingSessionRules** | N | Contains trading rules specified at the trading session level | |

### 17.4.2.16 TrdSessLstGrp

*Table 215: TrdSessLstGrp*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 386 | NoTradingSessions | Y | | NumInGroup |
| >336 | TradingSessionID | Y | Identifier for Trading Session | String |
| >625 | TradingSessionSubID | N | | String |
| >20084 | MarketID | N | | Exchange |
| >20036 | MarketSegmentID | N | | String |
| >20224 | TradingSessionDesc | N | **OMX Comment:** Not in FIX. The EEWG request addition | String |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| >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 |
| | **TradingSessionRules** | N | Insert here the set of "TradingSessionRules" fields defined in "common components of application messages"<br>**OMX Comment:** Not in FIX. The EEWG request addition | |
| >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 |

## 17.5 Workflows

### 17.5.1 Introduction

Please refer to Section 17.2.2 on page 223.

### 17.5.2 "Start of Day" Download

The following diagram depicts how start of day (or other periodicity) market structure trading reference data can be relayed.

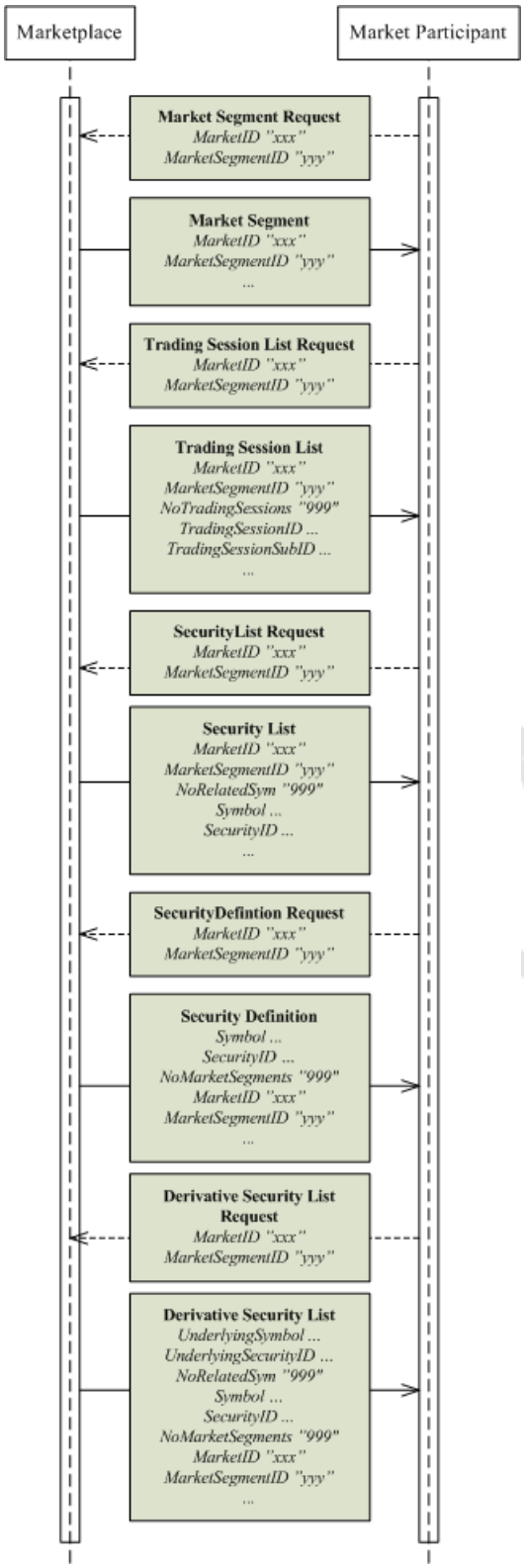


*Figure 38: Start Of Day Reference Data*

Note that whether a request is needed, what filtering parameters are applicable, what messages are returned and what fields are included are all bilaterally agreed. The marketplace may also choose to make the information available by other means as e.g. a down-loadable file.

The same message flow applies to a situation after the "Start of Day" where reference data has to be obtained anew.

### 17.5.3 Intraday Updates

The following diagram depicts how intra-day real time updates of the market structure trading reference data can be relayed:

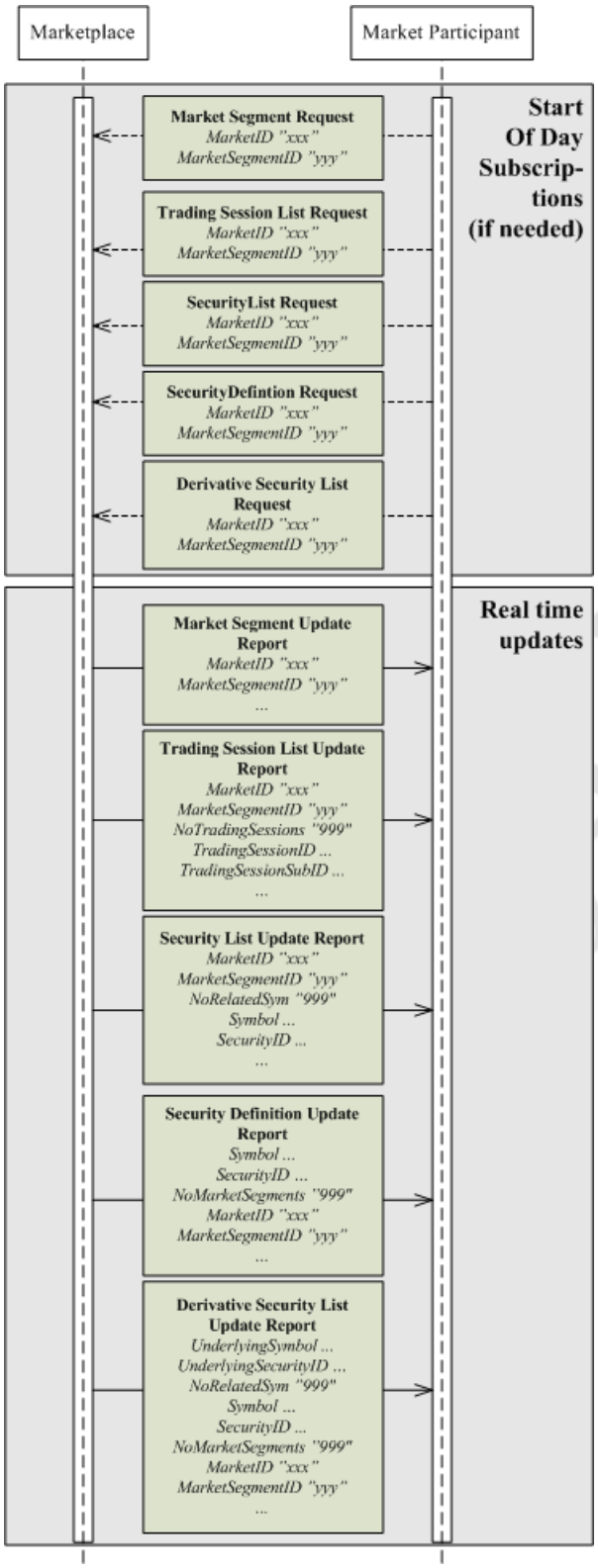


*Figure 39: Intraday Reference Data Updates*

# 18 General Messages

## 18.1 BusinessMessageReject

*Table 216: 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 | | |

# 19 General Component Blocks

## 19.1 Components

### 19.1.1 Instrument

*Table 217: 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)<br>Use "[N/A]" for products which do not have a symbol. | String |
| 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. | 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 |
| | **SecAltIDGrp** | N | Number of alternate Security Identifiers<br>**OMX Comment:** Used in outbound messages to provide an alternative identifier as e.g. an ISIN. | |
| 460 | Product | N | Indicates the type of product the security is associated with (high-level category) | int |
| 20198 | ProductComplex | N | Identifies an entire suite of products for a given market. In Futures this may be “interest rates”, “agricultural”, “equity indexes”, etc<br>**OMX Comment:** Not in FIX. The GDC request re-introduction | 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.<br>**OMX Comment:** Used in reference data messages only | String |
| 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. | String |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 167 | SecurityType | N | It is recommended that CFICode be used instead of SecurityType for non-Fixed Income instruments.<br>Required for Fixed Income. Refer to Volume 7 - Fixed Income<br>Futures and Options should be specified using the CFICode[461] field instead of SecurityType[167] (Refer to Volume 7 - Recommendations and Guidelines for Futures and Options Markets.) | String |
| 762 | SecuritySubType | N | Sub-type qualification/identification of the SecurityType (e.g. for SecurityType="MLEG"). If specified, SecurityType is required. | String |
| 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. | month-year |
| 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.<br>When using MaturityMonthYear, it is recommended that markets and sell sides report the MaturityDate on all outbound messages as a means of data enrichment.<br>**OMX Comment:** Options, Futures, Rights: Maps to Instrument Expiration Date<br>Debt:Maps to Security Maturity Date<br>Others: Not used | LocalMktDate |
| 1079 | MaturityTime | N | | TZTimeOnly |
| 965 | SecurityStatus | N | Gives the current state of the instrument | String |
| 224 | CouponPaymentDate | N | Date interest is to be paid. Used in identifying Corporate Bond issues.<br>**OMX Comment:** Debt: Maps to Security Coupons (Next Payment Date)<br>Others: Not used | LocalMktDate |
| 225 | IssueDate | N | Date instrument was issued. For Fixed Income IOIs for new issues, specifies the issue date.<br>**OMX Comment:** Equities, Debt: Maps to Security Issue Date<br>Others: Not used | LocalMktDate |
| 470 | CountryOfIssue | N | ISO Country code of instrument issue (e.g. the country portion typically used in ISIN). Can be used in conjunction with non-ISIN SecurityID (e.g. CUSIP for Municipal Bonds without ISIN) to provide uniqueness. | Country |
| 201 | PutOrCall | N | **OMX Comment:** Not in FIX. The GDC request re-introduction | int |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 202 | StrikePrice | N | Used for derivatives, such as options and covered warrants<br>**OMX Comment:** Options, Rights: Maps to Instrument Strike Price<br>Others: Not used | Price |
| 947 | StrikeCurrency | N | Used for derivatives<br>**OMX Comment:** Options, Rights: Maps to Instrument Strike Price Currency<br>Others: Not used | Currency |
| 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 |
| 231 | ContractMultiplier | N | For Fixed Income, Convertible Bonds, Derivatives, etc. Note: If used, quantities should be expressed in the "nominal" (e.g. contracts vs. shares) amount.<br>**OMX Comment:** Options, Futures:<br>Maps to Instrument Contract Size<br>Others: Not used | float |
| 969 | MinPriceIncrement | N | Minimum price increment for the instrument. Could also be used to represent tick value. | float |
| 996 | UnitofMeasure | N | Used to indicate the type of unit used to measure the underlying commodity on which the contract is based (e.g., lbs of lean cattle, barrels of crude oil, bushels of corn, etc.) | String |
| 1147 | UnitofMeasureQty | N | Used to indicate the size of the underlying commodity on which the contract is based, (e.g., 2500 lbs of lean cattle, 1000 barrels of crude oil, 1000 bushels of corn, etc.) | Qty |
| 20093 | PriceUnitOfMeasure | N | Used to express the UOM of the price if different from the contract. In futures, this can be different for cross-rate products in which the price is quoted in units differently from the contract<br>**OMX Comment:** Not in FIX. The GDC request addition | String |
| 20094 | PriceUnitOfMeasureQty | N | Used to express the UOM Quantity of the price if different from the contract. In futures, this can be different for physically delivered products in which price is quoted in a unit size different from the contract, i.e. a Cattle Future contract has a UOMQty of 40,000 and a PriceUOMQty of 100.<br>**OMX Comment:** Not in FIX. The GDC request addition | Qty |
| 20095 | SettlMethod | N | Settlement method for a contract. Can be used as an alternative to CFI Code value<br>**OMX Comment:** Not in FIX. The GDC request addition | char |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 20096 | ExerciseStyle | N | Type of exercise of a derivatives security<br>**OMX Comment:** Not in FIX. The GDC request addition | int |
| 20097 | CashAmount | N | Cash amount indicating the pay out associated with an option. For binary options this is a fixed amount<br>**OMX Comment:** Not in FIX. The GDC request addition | Price |
| 20098 | PriceQuoteMethod | N | Method for price quotation<br>**OMX Comment:** Not in FIX. The GDC request addition | String |
| 20099 | ValueTypeCode | N | For futures, indicates type of valuation method applied<br>**OMX Comment:** Not in FIX. The GDC request addition | String |
| 20100 | ListMethod | N | Indicates whether the instruments are pre-listed only or can also be defined via user request<br>**OMX Comment:** Not in FIX. The GDC request addition | int |
| 997 | TimeUnit | N | Used to indicate a time unit for the contract (e.g., days, weeks, months, etc.) | String |
| 20101 | CapPrice | N | Used to express the ceiling price of a capped call<br>**OMX Comment:** Not in FIX. The GDC request addition | Price |
| 20102 | FloorPrice | N | Used to express the floor price of a capped put<br>**OMX Comment:** Not in FIX. The GDC request addition | Price |
| 20200 | FlexibleIndicator | N | Used to indicate if a security has been defined as flexible according to “non-standard” means. Analog to CFICode Standard/Non-standard indicator<br>**OMX Comment:** Not in FIX. The GDC request addition | Boolean |
| 20201 | FlexProductEligibilityIndicator | N | Used to indicate if a product or group of product supports the creation of flexible securities<br>**OMX Comment:** Not in FIX. The GDC request addition | Boolean |
| 223 | CouponRate | N | For Fixed Income.<br>**OMX Comment:** Debt: Maps to Security Coupon Rate<br>Others: Not used | Percentage |
| 207 | SecurityExchange | N | Can be used to identify the security. | Exchange |
| 106 | Issuer | N | | String |
| 348 | EncodedIssuerLen | N | Must be set if EncodedIssuer field is specified and must immediately precede it. | Length |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 349 | EncodedIssuer | N | Encoded (non-ASCII characters) representation of the Issuer field in the encoded format specified via the MessageEncoding field. | data |
| 107 | SecurityDesc | N | | String |
| 350 | EncodedSecurityDescLen | N | Must be set if EncodedSecurityDesc field is specified and must immediately precede it. | Length |
| 351 | EncodedSecurityDesc | N | Encoded (non-ASCII characters) representation of the SecurityDesc field in the encoded format specified via the MessageEncoding field. | data |
| 691 | Pool | N | Identifies MBS / ABS pool<br>**OMX Comment:** Currently not used | String |
| 667 | ContractSettlMonth | N | Must be present for MBS/TBA<br>**OMX Comment:** Currently not used | month-year |
| 875 | CPProgram | N | The program under which a commercial paper is issued<br>**OMX Comment:** Currently not used | int |
| 876 | CPRegType | N | The registration type of a commercial paper issuance<br>**OMX Comment:** Currently not used | String |
| | **EvntGrp** | N | Number of repeating EventType group entries. | |
| 873 | DatedDate | N | If different from IssueDate<br>**OMX Comment:** Debt: Maps to Security Dated Date<br>Others: Not used | LocalMktDate |
| | **InstrumentParties** | N | Used to identify the parties listing a specific instrument<br>**OMX Comment:** Not in FIX. The GDC request addition | |

## 19.1.2 InstrumentLeg

*Table 218: InstrumentLeg*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 600 | LegSymbol | N | | String |
| 601 | LegSymbolSfx | N | | String |
| 602 | LegSecurityID | N | | String |
| 603 | LegSecurityIDSource | N | | String |
| | **LegSecAltIDGrp** | N | | |
| 608 | LegCFICode | N | | String |
| 315 | UnderlyingPutOrCall | N | **OMX Comment:** Not in FIX. The GDC request re-introduction | int |
| 616 | LegSecurityExchange | N | | Exchange |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 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 |

### 19.1.3 InstrumentParties

*Table 219: InstrumentParties*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 1018 | NoInstrumentParties | N | Repeating group below should contain unique combinations of InstrumentPartyID, Instrument-PartyIDSource, and InstrumentPartyRole | NumInGroup |
| >1019 | InstrumentPartyID | N | Used to identify party id related to instrument | String |
| >1050 | InstrumentPartyIDSource | N | Used to identify source of instrument party id | char |
| >1051 | InstrumentPartyRole | N | Used to identify the role of instrument party id | int |

### 19.1.4 NestedParties

*Table 220: 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.<br>**OMX Comment:** Valid values: 4; 14. | int |

### 19.1.5 NestedParties2

*Table 221: NestedParties2*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 756 | NoNested2PartyIDs | N | Repeating group below should contain unique combinations of Nested2PartyID, Nested2PartyIDSource, and Nested2PartyRole | NumInGroup |
| >757 | Nested2PartyID | N | Used to identify source of Nested2PartyID. Required if Nested2PartyIDSource is specified. Required 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 |
| >759 | Nested2PartyRole | N | Identifies the type of Nested2PartyID (e.g. Executing Broker). Required if NoNested2PartyIDs > 0. | int |

### 19.1.6 NestedParties3

*Table 222: NestedParties3*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 948 | NoNested3PartyIDs | N | Repeating group below should contain unique combinations of Nested3PartyID, Nested3PartyIDSource, and Nested3PartyRole | NumInGroup |
| >949 | Nested3PartyID | N | Used to identify source of Nested3PartyID. Required if Nested3PartyIDSource is specified. Required if NoNested3PartyIDs > 0. | String |
| >950 | Nested3PartyIDSource | N | Used to identify class source of Nested3PartyID value (e.g. BIC). Required if Nested3PartyID is specified. Required if NoNested3PartyIDs > 0. | char |
| >951 | Nested3PartyRole | N | Identifies the type of Nested3PartyID (e.g. Executing Broker). Required if NoNested3PartyIDs > 0. | int |

### 19.1.7 Parties

*Table 223: 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 |

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| >452 | PartyRole | N | Identifies the type of PartyID (e.g. Executing Broker). Required if NoPartyIDs > 0. | int |
| | **PtysSubGrp** | N | Repeating group of Party sub-identifiers. | |

## 19.1.8 RootParties

*Table 224: 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 |
| | **RootSubParties** | N | Repeating group of RootParty sub-identifiers. | |

## 19.1.9 UnderlyingInstrument

*Table 225: UnderlyingInstrument*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 311 | UnderlyingSymbol | N | | String |
| 312 | UnderlyingSymbolSfx | N | | String |
| 309 | UnderlyingSecurityID | N | | String |
| 305 | UnderlyingSecurityIDSource | N | | String |
| | **UndSecAltIDGrp** | N | | |
| 463 | UnderlyingCFICode | N | | String |
| 201 | PutOrCall | N | **OMX Comment:** Not in FIX. The GDC request re-introduction | int |
| 307 | UnderlyingSecurityDesc | N | | String |
| 364 | EncodedUnderlyingSecurity-DescLen | N | | Length |
| 365 | EncodedUnderlyingSecurity-Desc | N | | data |

# 19.2 Implicit Components

## 19.2.1 EvntGrp

*Table 226: EvntGrp*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 864 | NoEvents | N | | NumInGroup |
| >865 | EventType | N | | int |
| >866 | EventDate | N | | LocalMktDate |
| >1145 | EventTime | N | Specific time of event. To be used in combination with EventDate [866]<br>**OMX Comment:** Not in FIX. The GDC request addition | UTCTimes-tamp |

## 19.2.2 InstrmtLegGrp

*Table 227: InstrmtLegGrp*

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

## 19.2.3 LegSecAltIDGrp

*Table 228: LegSecAltIDGrp*

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

## 19.2.4 PtysSubGrp

*Table 229: PtysSubGrp*

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

### 19.2.5 RootSubParties

*Table 230: RootSubParties*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 1120 | NoRootPartySubIDs | N | Repeating group of RootParty sub-identifiers. | NumInGroup |
| >1121 | RootPartySubID | N | Sub-identifier (e.g. Clearing Acct for PartyID=Clearing Firm) if applicable. Required if NoRootPartySubIDs > 0. | String |
| >1122 | RootPartySubIDType | N | Type of Sub-identifier. Required if NoRootPartySubIDs > 0. | int |

### 19.2.6 SecAltIDGrp

*Table 231: SecAltIDGrp*

| Tag | FieldName | Req'd | Comments | Format |
|---|---|---|---|---|
| 454 | NoSecurityAltID | N | | NumInGroup |
| >455 | SecurityAltID | N | | String |
| >456 | SecurityAltIDSource | N | | String |

### 19.2.7 TrdgSesGrp

*Table 232: TrdgSesGrp*

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

### 19.2.8 UndInstrmtGrp

*Table 233: UndInstrmtGrp*

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

## 19.2.9 UndSecAltIDGrp

*Table 234: UndSecAltIDGrp*

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

# 20 Data Dictionary

## 20.1 Fields per Field Name

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| AffectedOrderID | 535 | String | | OrderID (37) of an order affected by a mass cancel request. | |
| AffectedSecondaryOrderID | 536 | String | | SecondaryOrderID (198) of an order affected by a mass cancel request. | |
| AggressorIndicator | 1057 | Boolean | | Used to identify whether the order initiator is an aggressor or not in the trade. | **Value** / **Description**<br>Y – Order initiator is aggressor<br>N – Order initiator is passive |
| AgreementCurrency | 918 | Currency | | Contractual currency forming the basis of a financing agreement and associated transactions. Usually, but not always, the same as the trade currency. | |
| AgreementDate | 915 | LocalMktDate | | A reference to the date the underlying agreement specified by AgreementID and AgreementDesc was executed. | |
| AgreementDesc | 913 | String | | The full name of the base standard agreement, annexes and amendments in place between the principals applicable to a financing transaction. | |
| AgreementID | 914 | String | | A common reference to the applicable standing agreement between the counterparties to a financing transaction. | |
| AllocAccount | 79 | String | | Sub-account mnemonic | |
| AllocAcctIDSource | 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.<br>(Prior to FIX 4.1 this field was of type int) | |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| AllocQty | 80 | Qty | | Quantity to be allocated to specific sub-account (Prior to FIX 4.2 this field was of type int) | |
| AllocSettlCurrency | 736 | Currency | | Currency code of settlement denomination for a specific AllocAccount (79). | |
| ApplQueueDepth | 813 | int | | Current number of application messages that were queued at the time that the message was created by the counterparty. | |
| ApplQueueResolution | 814 | int | | Resolution taken when ApplQueueDepth (813) exceeds ApplQueueMax (812) or system specified maximum queue size. | **Value** / **Description**<br>0 – No Action Taken<br>1 – Queue Flushed<br>2 – Overlay Last<br>3 – End Session |
| 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**<br>7 – FIX50 |
| AuctionInfoID | 20091 | String | | Uniquely identifies an Auction Info (UB) message | |
| AuctionInfoReqID | 20092 | String | | Uniquely identifies an Auction Info Request (UC) message | |
| AuctionRejectReason | 20053 | int | | Reason an Auction Request was rejected | **Value** / **Description**<br>1 – User not authorized<br>2 – Auction book not found<br>99 – Other |
| AuctionReqID | 20047 | String | | Uniquely identifies an Auction Request (U2) message | |
| AuctionResultID | 20054 | String | | Uniquely identifies an Auction result (U4) message | |
| AuctionTime | 20049 | UTCTimestamp | | Defines the time a call auction will be executed | |
| AuctionType | 20048 | int | | Defines the type of call auction | **Value** / **Description**<br>1 – Issuing Auction<br>2 – Buy-back Auction |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| 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)<br>Valid values:<br>FIXT.1.1 | |
| BidPx | 132 | Price | | Bid price/rate | |
| BidSize | 134 | Qty | | Quantity of bid<br>(Prior to FIX 4.2 this field was of type int) | |
| BidType | 394 | int | | Code to identify the type of Bid Request. | **Value** / **Description**<br>3 — No bidding process |
| BodyLength | 9 | Length | | Message length, in bytes, forward to the CheckSum field. ALWAYS SECOND FIELD IN MESSAGE. (Always unencrypted) | |
| BookTransparency | 20246 | int | | Defines the transparency of the book | **Value** / **Description**<br>0 — Default (as defined in reference data)<br>1 — No transparency<br>2 — As specified (in MDBookType [1021] and MarketDepth [264]) |
| BusinessRejectReason | 380 | int | | Code to identify reason for a Business Message Reject message. | **Value** / **Description**<br>0 — Other<br>1 — Unknown ID<br>2 — Unknown Security<br>3 — Unknown Message Type<br>4 — Application not available<br>5 — Conditionally required field missing |
| BusinessRejectRefID | 379 | String | | The value of the business-level "ID" field on the message being referenced. | |
| CapPrice | 20101 | Price | | Used to express the ceiling price of a capped call | |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| CashAmount | 20097 | Price | | Cash amount indicating the pay out associated with an option. For binary options this is a fixed amount<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| CFICode | 461 | String | | Indicates the type of security using ISO 10962 standard, Classification of Financial Instruments (CFI code) values. ISO 10962 is maintained by ANNA (Association of National Numbering Agencies) acting as Registration Authority. See "Appendix 6-B FIX Fields Based Upon Other Standards". See also the Product (460) and SecurityType (167) fields. It is recommended that CFICode be used instead of SecurityType (167) for non-Fixed Income instruments. A subset of possible values applicable to FIX usage are identified in "Appendix 6-D CFICode Usage - ISO 10962 Classification of Financial Instruments (CFI code)" | |
| 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 | |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | uniqueness across days, for example by embedding a date within the ClOrdID field. | |
| ContingencyType | 20077 | int | | Specifies the type of Contingent Order | Value: 4 — Description: One Updates the Other (OUO) - Proportional Quantity Reduction |
| ContractMultiplier | 231 | float | | Specifies the ratio or multiply factor to convert from "nominal" units (e.g. contracts) to total units (e.g. shares) (e.g. 1.0, 100, 1000, etc). Applicable For Fixed Income, Convertible Bonds, Derivatives, etc.<br>In general quantities for all calsses should be expressed in the basic unit of the instrument, e.g. shares for equities, norminal or par amount for bonds, currency for foreign exchange. When quantity is expressed in contracts, e.g. financing transactions and bond trade reporting, ContractMutliplier should contain the number of units in one contract and can be omitted if the multiplier is the default amount for the instrument, i.e. 1,000 par of bonds, 1,000,000 par for financing transactions. | |
| ContractSettlMonth | 667 | monthyear | | Specifies when the contract (i.e. MBS/TBA) will settle. | |
| ContraQty | 20057 | Qty | | Quantity at the contra side | |
| CopyMsgIndicator | 797 | Boolean | | Indicates whether or not this message is a drop copy of another message. | |
| CorporateAction | 292 | MultipleCharValue | | Identifies the type of Corporate Action. | Value: A — Description: Ex-Dividend<br>Value: B — Description: Ex-Distribution<br>Value: C — Description: Ex-Rights<br>Value: E — Description: Ex-Interest<br>Value: F — Description: Cash Dividend<br>Value: G — Description: Stock Dividend<br>Value: H — Description: Non-Integer Stock Split |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | **Value** / **Description**<br>I / Reverse Stock Split<br>J / Standard-Integer Stock Split<br>M / Merger Reorganization<br>N / Rights Offering<br>O / Shareholder Meeting<br>P / Spinoff<br>Q / Tender Offer<br>U / CUSIP / Name Change |
| CountryOfIssue | 470 | Country | | ISO Country code of instrument issue (e.g. the country portion typically used in ISIN). Can be used in conjunction with non-ISIN SecurityID (48) (e.g. CUSIP for Municipal Bonds without ISIN) to provide uniqueness. | |
| CouponPaymentDate | 224 | LocalMktDate | | Date interest is to be paid. Used in identifying Corporate Bond issues.<br>(Note tag # was reserved in FIX 4.1, added in FIX 4.3)<br>(prior to FIX 4.4 field was of type UTCDate) | |
| CouponRate | 223 | Percentage | | The rate of interest that, when multiplied by the principal, par value, or face value of a bond, provides the currency amount of the periodic interest payment. The coupon is always cited, along with maturity, in any quotation of a bond's price. | |
| CPProgram | 875 | int | | The program under which a commercial paper is issued<br>**OMX Comment:** Valid values are not yet defined | **Value** / **Description**<br>99 / Other |
| CPRegType | 876 | String | | The registration type of a commercial paper issuance | |
| CumQty | 14 | Qty | | Total quantity (e.g. number of shares) filled.<br>(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 | |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | 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. | |
| CurrencyRatio | 20233 | float | | Specifies the currency ratio between the currency used for a multileg price and the currency used by the outright book defined by the leg. Example: Multileg quoted in EUR, outright leg in USD and 1 EUR = 0,7 USD then LegCurrecyRatio = 0.7<br>**OMX Comment:** Not in FIX. OMX requires an extension | |
| CxlRejReason | 102 | int | | Code to identify reason for cancel rejection. | **Value** / **Description**<br>0 Too late to cancel<br>1 Unknown order<br>2 Broker / Exchange Option<br>6 Duplicate ClOrdID (11) received<br>18 Invalid price increment<br>99 Other |
| CxlRejResponseTo | 434 | char | | Identifies the type of request that a Cancel Reject is in response to. | **Value** / **Description**<br>1 Order cancel request<br>2 Order cancel/replace request |
| DatedDate | 873 | LocalMktDate | | The effective date of a new securities issue determined by its underwriters. Often but not always the same as the Issue Date and the Interest Accrual Date | |
| 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 | |
| DefBidSize | 293 | Qty | | Default Bid Size. | |
| DefOfferSize | 294 | Qty | | Default Offer Size. | |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| DeliverTo-CompID | 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. | |
| DeliverTo-SubID | 129 | String | | Assigned value used to identify specific message recipient (i.e. trader) if the message is delivered by a third party | |
| DeliveryType | 919 | int | | Identifies type of settlement | **Value** / **Description**<br>0 / "Versus Payment": Deliver (if sell) or Receive (if buy) vs. (against) Payment<br>1 / "Free": Deliver (if sell) or Receive (if buy) Free<br>2 / Tri-Party<br>3 / Hold In Custody |
| DerivaticeFuturesValue-TypeCode | 20219 | String | | For futures, indicates type of valuation method applied<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| Derivative-CapPrice | 20221 | Price | | Used to express the ceiling price of a capped call<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| Derivative-CashAmount | 20152 | Price | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeCFICode | 20126 | String | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| Derivative-ContractMultiplier | 20146 | float | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| Derivative-ContractSettlMonth | 20166 | month-year | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| Derivative-CountryOfIssue | 20137 | Country | | **OMX Comment:** Not in FIX. The GDC requires an extension | |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| DerivativeEncodedIssuer | 20159 | data | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeEncodedIssuerLen | 20158 | Length | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeEncodedSecurityDesc | 20162 | data | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeEncodedSecurityDescLen | 20161 | Length | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeEventDate | 20169 | LocalMktDate | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeEventPx | 20171 | Price | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeEventText | 20172 | String | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeEventTime | 20170 | UTCTimestamp | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeEventType | 20168 | int | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeExerciseStyle | 20151 | int | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeFloorPrice | 20222 | Price | | Used to express the floor price of a capped put<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeInstrAttribType | 20115 | int | | Code to represent the type of instrument attribute<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeInstrAttribValue | 20116 | String | | Attribute value appropriate to the SeriesInstrAttribType field<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeInstrmtAssignmentMethod | 20133 | char | | **OMX Comment:** Not in FIX. The GDC requires an extension | |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| DerivativeInstrRegistry | 20136 | String | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeInstrumentPartyID | 20174 | String | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeInstrumentPartyIDSource | 20175 | char | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeInstrumentPartyRole | 20176 | int | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeIssueDate | 20135 | LocalMktDate | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeIssuer | 20157 | String | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeListMethod | 20220 | int | | Indicates whether instruments are pre-listed only or can also be defined via user request<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeLocaleOfIssue | 20139 | String | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeMaturityDate | 20130 | LocalMktDate | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeMaturityMonthYear | 20129 | monthyear | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeMaturityTime | 20131 | TZTimeOnly | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeMinPriceIncrement | 20147 | float | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeMinPriceIncrementAmount | 20148 | Amt | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeNTPositionLimit | 20156 | int | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeOptAttribute | 20145 | char | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativePositionLimit | 20155 | int | | **OMX Comment:** Not in FIX. The GDC requires an extension | |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| DerivativePriceQuoteMethod | 20218 | String | | Method for price quotation<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativePriceUnitOfMeasure | 20213 | String | | Used to express the UOM of the price if different from the contract. In futures, this can be different for cross-rate products in which the price is quoted in units differently from the contract<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativePriceUnitOfMeasureQty | 20214 | Qty | | Used to express the UOM Quantity of the price if different from the contract. In futures, this can be different for physically delivered products in which price is quoted in a unit size different from the contract, i.e. a Cattle Future contract has a UOMQty of 40,000 and a PriceUOMQty of 100.<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeProduct | 20124 | int | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeProductComplex | 20199 | String | | Identifies an entire suite of products for a given market. In Futures this may be “interest rates”, “agricultural”, “equity indexes”, etc<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativePutOrCall | 20216 | int | | Used to express option right<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeSecurityAltID | 20122 | String | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeSecurityAltIDSource | 20123 | String | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeSecurityDesc | 20160 | String | | **OMX Comment:** Not in FIX. The GDC requires an extension | |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| DerivativeSecurityExchange | 20154 | Exchange | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeSecurityGroup | 20125 | String | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeSecurityID | 20119 | String | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeSecurityIDSource | 20120 | String | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeSecurityStatus | 20134 | String | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeSecuritySubType | 20128 | String | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeSecurityType | 20127 | String | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeSecurityXML | 20164 | data | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeSecurityXMLLen | 20163 | Length | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeSecurityXMLSchema | 20165 | data | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeSettleOnOpenFlag | 20132 | String | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeSettlMethod | 20215 | char | | Settlement method for a contract. Can be used as an alternative to CFI Code value<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeStateOrProvinceOfIssue | 20138 | String | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeStrikeCurrency | 20142 | Currency | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeStrikeMultiplier | 20143 | float | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeStrikePrice | 20141 | Price | | **OMX Comment:** Not in FIX. The GDC requires an extension | |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| DerivativeStrikeValue | 20144 | float | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeSymbol | 20117 | String | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeSymbolSfx | 20118 | String | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeTimeUnit | 20153 | String | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeUnitofMeasure | 20149 | String | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivativeUnitofMeasureQty | 20150 | Qty | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| DerivFlexProductEligibilityIndicator | 20202 | Boolean | | Used to indicate if a product or group of product supports the creation of flexible securities<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| DiscretionInst | 388 | char | | Code to identify the price a DiscretionOffsetValue (389) is related to and should be mathematically added to. | **Value** / **Description**<br>7 – 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** / **Description**<br>1 – Initial (use original DisplayQty)<br>2 – 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. | |
| DividendYield | 20230 | float | | Specifies the expected dividend of a security. Expressed as yield.<br>**OMX Comment:** Not in FIX. OMX requires an extension | |

| 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<br>Value: Y — Halt was due to related security being halted |
| EncodedIssuer | 349 | data | | Encoded (non-ASCII characters) representation of the Issuer field in the encoded format specified via the MessageEncoding (347) field. If used, the ASCII (English) representation should also be specified in the Issuer field. | |
| EncodedIssuerLen | 348 | Length | | Byte length of encoded (non-ASCII characters) EncodedIssuer (349) field. | |
| EncodedLegSecurityDesc | 622 | data | | Multileg instrument's individual security's EncodedSecurityDesc.<br>See EncodedSecurityDesc (35) field for description | |
| EncodedLegSecurityDescLen | 621 | Length | | Multileg instrument's individual security's EncodedSecurityDescLen.<br>See EncodedSecurityDescLen (350) field for description | |
| EncodedMktSegmDesc | 20039 | data | | Encoded (non-ASCII characters) representation of the MarketSegmDesc (20037) 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. | |
| EncodedSecurityDesc | 351 | data | | Encoded (non-ASCII characters) representation of the SecurityDesc (107) field in the encoded format specified via the MessageEncoding (347) field. If used, the ASCII (English) representation should also be specified in the SecurityDesc field. | |
| EncodedSecurityDescLen | 350 | Length | | Byte length of encoded (non-ASCII characters) EncodedSecurityDesc (351) field. | |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| 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) EncodedUnderlyingSecurityDesc (365) field. | |
| EncryptMethod | 98 | int | | Method of encryption. | **Value** / **Description**: 0 / None / Other |
| EndDate | 917 | LocalMktDate | | End date of a financing deal, i.e. the date the seller reimburses the buyer and takes back control of the collateral | |
| EndMMY | 20190 | monthyear | | Ending maturity month year for an option class<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| EndSeqNo | 16 | 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). | |
| EndStrilePxRange | 20105 | Price | | Ending price of the range to which the StrikeIncrement applies. Price refers to the price of the underlying<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| EndTick-Pricerange | 20110 | Price | | Ending price range for the specified tick increment<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| EventDate | 866 | LocalMkt-Date | | Date of event | |
| EventTime | 1145 | UTC-Timestamp | | Specific time of event. To be used in combination with EventDate [866] | |
| EventType | 865 | int | | Code to represent the type of event | **Value** / **Description**<br>5 Activation<br>6 Inactiviation<br>8 Swap Start Date<br>9 Swap End Date<br>10 Swap Next Start Date<br>11 Swap Roll Date<br>12 Swap Next Roll Date<br>13 First Delivery Date<br>14 Last Delivery Date<br>15 Initial Inventory Due Date<br>16 Final Inventory Due Date<br>17 First Intent Date<br>18 Last Intent Date<br>19 Position Removal Date<br>100 On the Surveillance List<br>101 Excluding Dividend<br>102 Excluding Participating in Rights<br>103 Excluding Participating in Split<br>104 Company subject to Public Offer<br>105 Under Drawing<br>106 Excluding combined Split and Issue Rights |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| 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 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.<br><br>(Prior to FIX 4.1 this field was of type int) | |
| 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**<br>G — All or none - AON<br>**OMX Comment:** Currently not supported - use MinQty (110) = QrderQty (38)<br>S — Suspend<br>**OMX Comment:** Used to report an order as suspended in Execution Reports. Also used to send in an order as suspended.<br>i — Imbalance Only<br>v — Release (from suspension)<br>**OMX Comment:** Not in FIX Standard. OMX requests addition<br>w — Execute as delta neutral using volatility provided<br>**OMX Comment:** Not in FIX Standard. OMX requests addition<br>x — Execute as duration neutral<br>**OMX Comment:** Not in FIX Standard. OMX requests addition<br>y — Execute as FX neutral<br>**OMX Comment:** Not in FIX Standard. |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | **Value** / **Description**<br>OMX requests addition<br>z / Suspend on Connection Loss<br>**OMX Comment:** Not in FIX Standard. OMX requests addition |
| ExecInstValue | 20183 | char | | Indicates execution instructions that are valid for the specified trading rule context<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 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**<br>0 / GT corporate action<br>1 / GT renewal / restatement (no corporate action)<br>2 / Verbal change<br>3 / Repricing of order<br>6 / Cancel on Trading Halt<br>7 / Cancel on System Failure<br>9 / Canceled, not best<br>11 / Peg Refresh<br>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**<br>0 / New<br>4 / Canceled<br>5 / Replaced<br>8 / Rejected<br>9 / Suspended<br>**OMX Comment:** Currently not used |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | **Value** / **Description**<br>C – Expired<br>D – Restated (Execution Report sent unsolicited by sellside, with ExecRestatementReason (378) set)<br>F – Trade (partial fill or fill) |
| ExerciseStyle | 20096 | int | | Type of exercise of a derivatives security<br>**OMX Comment:** Not in FIX. The GDC requires an extension | **Value** / **Description**<br>0 – European – exercise only on expiration date<br>1 – American – exercise up to and including expiration date<br>2 – Bermuda – exercise on specified dates only<br>3 – Binary – buyer of option receives a fixed amount option is in the money |
| ExpirationCycle | 827 | int | | Part of trading cycle when an instrument expires. Field is applicable for derivatives. | **Value** / **Description**<br>0 – Expire on trading session close (default)<br>1 – Expire on trading session open<br>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")<br>The meaning of expiration is specific to the context where the field is used. | |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | For orders, this is the expiration time of a Good Til Date TimeInForce.<br>For Quotes - this is the expiration of the quote.<br>Expiration time is provided across the quote message dialog to control the length of time of the overall quoting process.<br>For collateral requests, this is the time by which collateral must be assigned.<br>For collateral assignments, this is the time by which a response to the assignment is expected. | |
| FinancialStatus | 291 | MultipleCharValue | | Identifies a firm's or a security's financial status | **Value** / **Description**<br>1 Bankrupt<br>2 Pending delisting<br>3 Restricted |
| FlexibleIndicator | 20200 | Boolean | | Used to indicate if a security has been defined as flexible according to “non-standard” means. Analog to CFICode Standard/Non-standard indicator<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| FlexProductEligibilityIndicator | 20201 | Boolean | | Used to indicate if a product or group of product supports the creation of flexible securities<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| FloorPrice | 20102 | Price | | Used to express the floor price of a capped put<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| GapFillFlag | 123 | Boolean | | Indicates that the Sequence Reset message is replacing administrative or application messages which will not be resent. | **Value** / **Description**<br>N Sequence Reset, Ignore Msg Seq Num (N/A For FIXML - Not Used) |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | **Value** / **Description**<br>Y / Gap Fill Message, Msg Seq Num Field Valid |
| HaltReason | 327 | char | | Denotes the reason for the Opening Delay or Trading Halt. | **Value** / **Description**<br>D / News Dissemination<br>E / Order Influx<br>M / Additional Information |
| Headline | 148 | String | | The headline of a News message | |
| HeartBtInt | 108 | int | | Heartbeat interval (seconds) | |
| HighLimit-Price | 1149 | Price | | Allowable high limit price for the trading day. A key parameter in validating order price. Used as the upper band for validating order prices. Orders submitted with prices above the upper limit will be rejected | |
| HighPx | 332 | Price | | Represents an indication of the high end of the price range for a security prior to the open or reopen | |
| ImpliedMarketIndicator | 1144 | int | | Commonly used in listed derivatives. Indicates that an implied market should be created for either the legs of a multi-leg instrument (Implied-in) or for the multi-leg instrument based on the existence of the legs (Implied-out). Determination as to whether implied markets should be created is generally done at the level of the multi-leg instrument | **Value** / **Description**<br>0 / Not implied<br>1 / Implied-in – the existence of a multi-leg instrument is implied by the legs of that instrument<br>2 / Implied-out – The existence of the underlying legs are implied by the multi-leg instrument<br>3 / Both Implied-in and Implied-out |
| IndividualAllocID | 467 | String | | Unique identifier for a specific NoAllocs (78) repeating group instance (e.g. for an AllocAccount). | |
| InstrAttrib-Type | 871 | int | | Code to represent the type of instrument attribute | **Value** / **Description**<br>1 / Flat (securities pay interest on a current |

| FieldName | Tag | Type | OMXLen | Desc | Valid values | |
|---|---|---|---|---|---|---|
| | | | | | **Value** | **Description** |
| | | | | | | basis but are traded without interest) |
| | | | | | 2 | Zero coupon |
| | | | | | 3 | Interest bearing (for Euro commercial paper when not issued at discount) |
| | | | | | 4 | No periodic payments |
| | | | | | 5 | Variable rate |
| | | | | | 6 | Less fee for put |
| | | | | | 7 | Stepped coupon |
| | | | | | 8 | Coupon period (if not semi-annual). Supply redemption date in the InstrAttribValue (872) field. |
| | | | | | 9 | When [and if] issued |
| | | | | | 10 | Original issue discount |
| | | | | | 11 | Callable, puttable |
| | | | | | 12 | Escrowed to Maturity |
| | | | | | 13 | Escrowed to redemption date - callable. Supply redemption date in the InstrAttribValue (872) field |
| | | | | | 14 | Pre-refunded |
| | | | | | 15 | In default |
| | | | | | 16 | Unrated |
| | | | | | 17 | Taxable |
| | | | | | 18 | Indexed |
| | | | | | 19 | Subject To Alternative Minimum Tax |
| | | | | | 20 | Original issue discount price. Supply price in the InstrAttribValue (872) field |
| | | | | | 21 | Callable below maturity value |
| | | | | | 22 | Callable without notice by mail to holder unless registered |
| | | | | | 23 | Price tick rules for security. Tick rule val- |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | **Value** / **Description**<br>ues to be expressed using InstrAttribValue [872]<br>24 — Trade type eligibility details for security. Trade types to be expressed using InstrAttribValue [872]<br>25 — Instrument Denominator<br>26 — Instrument Numerator<br>27 — Instrument Price Precision<br>28 — Instrument Strike Price Precision<br>29 — Tradeable Indicator<br>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. | |
| InstrumentPartyID | 1019 | String | | PartyID value within an instrument party repeating group. Same values as PartyID (448) | |
| InstrumentPartyIDSource | 1050 | char | | PartyIDSource value within an instrument partyrepeating group.<br>Same values as PartyIDSource (447) | |
| InstrumentPartyRole | 1051 | int | | PartyRole value within an instrument partyepeating group.<br>Same values as PartyRole (452) | |
| InViewOfCommon | 328 | Boolean | | Indicates whether or not the halt was due to Common Stock trading being halted. | **Value** / **Description**<br>N — Halt was not related to a halt of the common stock<br>Y — Half was due to common stock being halted |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| IOIID | 23 | String | | Unique identifier of IOI message.<br>(Prior to FIX 4.1 this field was of type int) | |
| IOIQty | 27 | String | | Quantity (e.g. number of shares) in numeric form or relative size. | **Value** / **Description**<br>0 / 1000000000<br>S / Small<br>M / Medium<br>L / Large<br>U / Undisclosed Quantity |
| IOIRefID | 26 | String | | Reference identifier used with CANCEL and REPLACE, transaction types.<br>(Prior to FIX 4.1 this field was of type int) | |
| IOITransType | 28 | char | | Identifies IOI message transaction type | **Value** / **Description**<br>N / New<br>C / Cancel<br>R / Replace |
| IssueDate | 225 | LocalMktDate | | The date on which a bond or stock offering is issued. It may or may not be the same as the effective date ("Dated Date") or the date on which interest begins to accrue ("Interest Accrual Date")<br>(Note tag # was reserved in FIX 4.1, added in FIX 4.3)<br>(prior to FIX 4.4 field was of type UTCDate) | |
| Issuer | 106 | String | | Name of security issuer (e.g. International Business Machines, GNMA).<br>see also Volume 7: "PRODUCT: FIXED INCOME - Euro Issuer Values" | |
| LanguageCode | 20065 | String | | Language code according to ISO 639 2-alpha character 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** / **Description**<br>N / Not Last Message<br>Y / Last Message |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| 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) | |
| LegAllocAccount | 671 | String | | Allocation Account for the leg See AllocAccount (79) for description and valid values. | |
| LegAllocAcctIDSource | 674 | String | | The source of the LegAllocAccount (671) See AllocAcctIDSource (661) for description and valid values. | |
| LegAllocID | 20090 | String | | Unique identifier for allocation message. Also see AllocID (70)<br>**OMX Comment:** Not in FIX. The EEWG requires an extension | |
| LegAllocQty | 673 | Qty | | Leg allocation quantity. See AllocQty (80) for description and valid values. | |
| LegCFICode | 608 | String | | Multileg instrument's individual security's CFICode. See CFICode (461) field for description | |
| LegCurrencyRatio | 20234 | float | | Specifies the currency ratio between the currency used for a multileg price and the currency used by the outright book defined by the leg. Example: Multileg quoted in EUR, outright leg in USD and 1 EUR = 0,7 USD then LegCurrecyRatio = 0.7<br>**OMX Comment:** Not in FIX. OMX requires an extension | |
| LegDividendYield | 20231 | float | | Specifies the expected dividend for the Security of a leg. Expressed as yield.<br>**OMX Comment:** Not in FIX. OMX requires an extension | |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| LegExecInst | 20247 | MultipleCharValue | | Refer to ExecInst (18)<br>**OMX Comment:** Not in FIX. OMX requires an extension | |
| LegIndividualAllocID | 672 | String | | Reference for the individual allocation ticket<br>See IndividualAllocID (467) for description and valid values. | |
| LegOrderQty | 685 | Qty | | Quantity ordered of this leg.<br>See OrderQty (38) for description and valid values | |
| LegPositionEffect | 564 | char | | PositionEffect for leg of a multileg<br>See PositionEffect (77) field for description | |
| LegPrice | 566 | Price | | Price for leg of a multileg<br>See Price (44) 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.<br>See SecurityAltID (455) field for description | |
| LegSecurityAltIDSource | 606 | String | | Multileg instrument's individual security's SecurityAltIDSource.<br>See SecurityAltIDSource (456) field for description | |
| LegSecurityDesc | 620 | String | | Multileg instrument's individual security's SecurityDesc.<br>See SecurityDesc (07) field for description | |
| LegSecurityExchange | 616 | Exchange | | Multileg instrument's individual security's SecurityExchange.<br>See SecurityExchange (207) field for description | |
| LegSecurityID | 602 | String | | Multileg instrument's individual security's SecurityID.<br>See SecurityID (48) field for description | |
| LegSecurityIDSource | 603 | String | | Multileg instrument's individual security's SecurityIDSource.<br>See SecurityIDSource (22) field for description | |
| LegSettlCurrency | 675 | Currency | | Identifies settlement currency for the Leg.<br>See SettlCurrency (20) for description and valid values | |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| LegSide | 624 | char | | The side of this individual leg (multileg security).<br>See Side (54) field for description and values | |
| LegSymbol | 600 | String | | Multileg instrument's individual security's Symbol.<br>See Symbol (55) field for description | |
| LegSymbolSfx | 601 | String | | Multileg instrument's individual security's SymbolSfx.<br>See SymbolSfx (65) field for description | |
| LegVolatility | 20229 | float | | Annualized volatility for option model calculations<br>**OMX Comment:** Not in FIX. OMX requires an extension | |
| ListID | 66 | String | | Unique identifier for list as assigned by institution, used to associate multiple individual orders. Uniqueness must be guaranteed within a single trading day. Firms which generate multi-day orders should consider embedding a date within the ListID field to assure uniqueness across days. | |
| ListMethod | 20100 | int | | Indicates whether instruments are pre-listed only or can also be defined via user request<br>**OMX Comment:** Not in FIX. The GDC requires an extension | **Value** / **Description**<br>0 – Pre-listed only<br>1 – User requested<br>2 – Undefined (no product) |
| ListOrderStatus | 431 | int | | Code to represent the status of a list order. | **Value** / **Description**<br>3 – Executing<br>7 – Reject |
| ListRejectReason | 20244 | String | | Identifies the reason for rejection of a New Order List message. Note that OrdRejReason (103) is used if the rejection is based on properties of an individual order part of the List.<br>**OMX Comment:** Not in FIX. The EEWG requires an extension | **Value** / **Description**<br>0 – Broker / Exchange option<br>2 – Exchange closed<br>4 – Too late to enter<br>5 – Unknown order<br>6 – Duplicate Order (e.g. dupe ClOrdID)<br>11 – Unsupported order characteristic |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | **Value** / **Description**<br>99 / Other |
| ListSeqNo | 67 | int | | Sequence of individual order within list (i.e. ListSeqNo of TotNoOrders (68), 2 of 25, 3 of 25, . . . ) | |
| ListStatusType | 429 | int | | Code to represent the status type. | **Value** / **Description**<br>2 / Response |
| ListUpdateAction | 20223 | char | | Specifies New (0), Cancel (1) or Replace (2). If provided, then Instrument occurrence has explicitly changed<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| LotType | 1093 | char | | Defines the lot type assigned to the order. | **Value** / **Description**<br>1 / Odd Lot<br>2 / Round Lot<br>3 / Block Lot |
| LowLimitPrice | 1148 | Price | | Allowable low limit price for the trading day. A key parameter in validating order price. Used as the lower band for validating order prices. Orders submitted with prices below the lower limit will be rejected | |
| LowPx | 333 | Price | | Represents an indication of the low end of the price range for a security prior to the open or reopen | |
| MarginRatio | 898 | Percentage | | The fraction of the cash consideration that must be collateralized, expressed as a percent. A MarginRatio of 02% indicates that the value of the collateral (after deducting for "haircut") must exceed the cash consideration by 2%. | |
| MarketDepth | 264 | int | | Depth of market for Book Snapshot | **Value** / **Description**<br>0 / Full Book<br>**OMX Comment:** FIX 5.0 SP1<br>1 / Top of Book |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | **Value** / **Description**<br>**OMX Comment:** FIX 5.0 SP1 |
| MarketID | 20084 | Exchange | | Identifies a marketplace<br>Valid values:<br>- See "Appendix 6-C" | |
| MarketSegmentDesc | 20037 | String | | Description or name of Market Segment | |
| MarketSegmentID | 20036 | String | | Market Segment identifier value | |
| MassActionRejectReason | 20242 | int | | Reason Order Mass Suspend or Release Request was rejected<br>**OMX Comment:** Not in FIX. The EEWG requires an extension | **Value** / **Description**<br>0 Mass Suspend / Release Not Supported<br>1 Invalid or Unknown Security<br>2 Invalid or Unkown Underlying security<br>3 Invalid or Unknown Product<br>4 Invalid or Unknown CFICode<br>5 Invalid or Unknown SecurityType<br>6 Invalid or Unknown Trading Session<br>7 Invalid or unknown Market Segment<br>99 Other |
| MassActionReQuestScope | 20240 | int | | Specifies scope of Order Mass Action Request.<br>**OMX Comment:** Not in FIX. The EEWG requires an extension | **Value** / **Description**<br>1 All orders for a security<br>2 All orders for an underlying security<br>3 All orders for a Product<br>4 All orders for a CFICode<br>5 All orders for a SecurityType<br>6 All orders for a trading session |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | **Value** / **Description**<br>7 — All orders<br>8 — All orders for a Market Segment |
| MassAction-Response | 20241 | int | | Specifies the action taken by counterparty order handling system as a result of the Order Mass Suspend or Release Request<br>**OMX Comment:** Not in FIX. The EEWG requires an extension | **Value** / **Description**<br>0 — Action Request Rejected - See MassActionRejectReason (20242)<br>1 — All orders for a security<br>2 — All orders for an Underlying Security<br>3 — All orders for a Product<br>4 — All orders for a CFI-Code<br>5 — All orders for a SecurityType<br>6 — All orders for a trading session<br>7 — All Orders<br>8 — All orders for a Market Segment |
| MassAction-Type | 20239 | int | | Specifies the type of action requested<br>**OMX Comment:** Not in FIX. The EEWG requires an extension | **Value** / **Description**<br>1 — Suspend Orders<br>2 — Release Orders from Suspension |
| MassSus-pRelRepor-tID | 20243 | String | | Unique Identifier for the Order Mass Suspend or Release Report<br>**OMX Comment:** Not in FIX. The EEWG requires an extension | |
| MatchAlgo-rithm | 1142 | String | | The type of algorithm used to match orders in a specific security<br>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 | |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | 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. | |
| MatchStatus | 573 | char | | The status of this trade with respect to matching or comparison. | **Value** / **Description**<br>0 – Compared, matched or affirmed<br>1 – Uncompared, unmatched, or unaffired |
| MatchType | 574 | String | | The point in the matching process at which this trade was matched. | **Value** / **Description**<br>1 – One-Party Trade Report (privately negotiated trade)<br>4 – Auto-match<br>5 – Cross Auction<br>6 – Counter-Order Selection<br>7 – Call Auction<br>8 – Issuing/Buy-Back Auction<br>**OMX Comment:** FIX 5.0 SP1 |
| MaturityDate | 541 | LocalMktDate | | Date of maturity. | |
| MaturityMonthYear | 200 | monthyear | | Can be used with standardized derivatives vs. the MaturityDate (54) field. Month and Year of the maturity (used for standardized futures and options).<br>Format:<br>YYYYMM (i.e. 99903)<br>YYYYMMDD (20030323)<br>YYYYMMwN (200303w) for week<br>A specific date or can be appended to the MaturityMonthYear. For instance, if multiple standard products exist that mature in the same Year and Month, but actually mature at a different time, a value can be appended, such as | |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | "w" or "w2" to indicate week as opposed to week 2 expiration. Likewise, the date (0-3) can be appended to indicate a specific expiration (maturity date). | |
| MaturityRuleID | 20196 | String | | Allows maturity rule to be referenced via an identifier so that rules do not need to be explicitly enumerated<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 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**<br>2 / Price Depth<br>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. | |
| 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. | |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| MDEntry-Time | 273 | UTC-TimeOn-ly | | Time of Market Data Entry. | |
| MDEntry-Type | 269 | char | | Type Market Data entry. | **Value** / **Description**<br>0 Bid<br>1 Offer<br>2 Trade<br>3 Index Value<br>A Imbalance |
| MDFeed-Type | 1022 | String | | Describes a class of service for a given data feed, ie Regular and Market Maker, Bandwidth Intensive or Bandwidth Conservative | |
| MDPriceLev-el | 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 | |
| MDQuote-Type | 1070 | int | | Identifies market data quote type. | **Value** / **Description**<br>0 Indicative<br>1 Tradeable |
| MDReportID | 20034 | String | | Identifier for Market Data reports | |
| MDReqID | 262 | String | | Unique identifier for Market Data Request | |
| MD-StatScope | 20035 | String | | Defines the scope of the statistics in periods of time. | |
| MDStatType | 20032 | String | | Type of statistic. Additional values can be bilaterally agreed between parties. | **Value** / **Description**<br>3 Index Value<br>4 Opening Price<br>5 Closing Price<br>6 Settlement Price<br>7 High Price<br>8 Low Price<br>9 VWAP Price<br>B Trade Volume<br>C Open Interest |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | **Value** / **Description**<br>D – Composite Underlying Price<br>E – Simulated Buy Price<br>F – Simulated Sell Price<br>G – Margin Rate<br>H – Mid Price<br>K – Settle High Price<br>L – Settle Low Price<br>N – High Bid<br>O – Low Offer<br>ZO – Last Privately Negotiated Trade Volume<br>ZP – Last Privately Negotiated Trade Price<br>ZQ – Exchange Best<br>ZR – Consolidated Best<br>ZS – Exchange Last<br>ZT – Final Price of Session<br>ZU – Turnover Value<br>ZV – TWAP<br>ZW – Auction Price<br>ZX – First Price<br>ZY – Number of Trades<br>ZZ – Number of Deals |
| 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 MDEntryPositionNo in order to sort the order properly.<br>**OMX Comment:** FIX 5.0 SP1 | |
| MDUpdateAction | 279 | char | | Type of Market Data update action. | **Value** / **Description**<br>0 – New<br>1 – Change |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | Value: 2 — Delete<br>3 — Delete Thru<br>4 — Delete From<br>5 — Overlay |
| MinLotSize | 20197 | Qty | | Minimum lot size allowed based on lot type specified in LotType/1093<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| MinPriceIncrement | 969 | float | | Minimum price increase for a given exchange-traded Instrument | |
| MinQty | 110 | Qty | | Minimum quantity of an order to be executed.<br>(Prior to FIX 4.2 this field was of type int) | |
| MinTradeVol | 562 | Qty | | The minimum trading volume for a security | |
| MktSegmMsgID | 20226 | String | | Market Segment message identifier.<br>**OMX Comment:** Not in FIX. The EEWG requires an extension | |
| MktSegmReqID | 20085 | String | | Unique ID of a Market Segment Request message. | |
| MktSegmUpdateAction | 20227 | char | | Specifies the action taken for the specified MarketID / MarketSegmentID.<br>**OMX Comment:** Not in FIX. The EEWG requires an extension | |
| MMYFormat | 20193 | int | | Format used to generate the MMY for each option contract:<br>**OMX Comment:** Not in FIX. The GDC requires an extension | Value: 0 — YearMonth Only (default)<br>1 — YearMonthDay<br>2 — YearMonthWeek |
| MMYIncrement | 20191 | int | | Increment between successive maturities for an option class<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| MMYIncrementUnitOfMeasure | 20192 | int | | Unit of measure for the MMYIncrement<br>**OMX Comment:** Not in FIX. The GDC requires an extension | **Value** / **Description**<br>0 Months<br>1 Days<br>2 Weeks<br>3 Years |
| MsgSeqNum | 34 | SeqNum | | Integer message sequence number. | |
| MsgType | 35 | String | | Defines message type ALWAYS THIRD FIELD IN MESSAGE. (Always unencrypted)<br>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.<br>*** Note the use of lower case letters *** | **Value** / **Description**<br>0 Heartbeat<br>1 Test Request<br>2 Resend Request<br>3 Reject<br>4 Sequence Reset<br>5 Logout<br>8 Execution Report<br>9 Order Cancel Reject<br>A Logon<br>D New Order - Single<br>G Order Cancel/Replace Request (a.k.a. Order Modification Request)<br>Q Don't Know Trade (DK)<br>X Market Data - Incremental Refresh<br>d Security Definition<br>f Security Status<br>j Business Message Reject<br>y Security List<br>AA Derivative Security List<br>y Derivative Securiy List Update Report<br>**OMX Comment:** Not in FIX. The GDC requests addition<br>AE Trade Capture Report |

| FieldName | Tag | Type | OMXLen | Desc | Valid values | |
|---|---|---|---|---|---|---|
| | | | | | **Value** | **Description** |
| | | | | | AR | Trade Capture Report Ack |
| | | | | | BJ | Trading Session List |
| | | | | | UJ | Participant List Update Report<br>**OMX Comment:** Not in FIX Standard. OMX requests addition |
| | | | | | U3 | SyncHeartbeat |
| | | | | | U0 | Market Data Statistics<br>**OMX Comment:** Not in FIX Standard. User-defined message |
| | | | | | U1 | Market Segment<br>**OMX Comment:** Not in FIX Standard. OMX requests addition |
| | | | | | U2 | One Sided Auction Request<br>**OMX Comment:** Not in FIX Standard. User-defined message |
| | | | | | U3 | One Sided Auction Request Ack<br>**OMX Comment:** Not in FIX Standard. User-defined message |
| | | | | | U4 | One Sided Auction Result<br>**OMX Comment:** Not in FIX Standard. User-defined message |
| | | | | | U5 | NewsDataRequest<br>**OMX Comment:** Not in FIX Standard. User-defined message |
| | | | | | U6 | NewsDataRequestReject<br>**OMX Comment:** Not in FIX Standard. User-defined message |
| | | | | | U7 | NewsPublicationRequest |

| FieldName | Tag | Type | OMXLen | Desc | Valid values | |
|---|---|---|---|---|---|---|
| | | | | | **Value** | **Description** |
| | | | | | | **OMX Comment:** Not in FIX Standard. User-defined message |
| | | | | | U8 | NewsPublicationRequestReject |
| | | | | | | **OMX Comment:** Not in FIX Standard. User-defined message |
| | | | | | U9 | Participant List |
| | | | | | UA | Market Segment Request |
| | | | | | | **OMX Comment:** Not in FIX Standard. OMX requests addition |
| | | | | | UB | One Sided Auction Info |
| | | | | | | **OMX Comment:** Not in FIX Standard. User-defined message |
| | | | | | UC | One Sided Auction Info Request |
| | | | | | | **OMX Comment:** Not in FIX Standard. User-defined message |
| | | | | | UD | Market Segment Update Report |
| | | | | | | **OMX Comment:** Not in FIX Standard. OMX requests addition |
| | | | | | UE | Trading Session List Update Report |
| | | | | | | **OMX Comment:** Not in FIX Standard. OMX requests addition |
| | | | | | UF | Order Mass Suspend Or Release Request |
| | | | | | | **OMX Comment:** Not in FIX Standard. OMX requests addition |
| | | | | | UG | Order Mass Suspend Or Release Report |
| | | | | | | **OMX Comment:** Not in FIX Standard. OMX requests addition |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | **Value** / **Description**<br>UH — Derivative Security List Update Report<br>**OMX Comment:** Not in FIX Standard. OMX requests addition<br>UI — Participant List Request<br>**OMX Comment:** Not in FIX Standard. OMX requests addition |
| MultilegModel | 20016 | int | | Specifies the type of multileg model the user is targeting. | **Value** / **Description**<br>0 — Predefined Multileg Security<br>2 — Strategy Order |
| MultilegPriceMethod | 20004 | int | | Defines the type of combination price the multileg uses | **Value** / **Description**<br>1 — Net Price<br>2 — Reversed Net Price<br>3 — Yield Difference<br>4 — Individual<br>5 — Weighted Average Price<br>6 — Multiplied Price |
| Nested2PartyID | 757 | String | | PartyID value within a "second instance" Nested repeating group.<br>Same values as PartyID (448) | |
| Nested2PartyIDSource | 758 | char | | PartyIDSource value within a "second instance" Nested repeating group.<br>Same values as PartyIDSource (447) | |
| Nested2PartyRole | 759 | int | | PartyRole value within a "second instance" Nested repeating group.<br>Same values as PartyRole (452) | |
| Nested3PartyID | 949 | String | | PartyID value within a "third instance" Nested repeating group.<br>Same values as PartyID (448) | |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| Nested3PartyIDSource | 950 | char | | PartyIDSource value within a "third instance" Nested repeating group.<br>Same values as PartyIDSource (447) | |
| Nested3PartyRole | 951 | int | | PartyRole value within a "third instance" Nested repeating group.<br>Same values as PartyRole (452) | |
| NestedInstrAttribType | 20113 | int | | Code to represent the type of instrument attribute<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| NestedInstrAttribValue | 20114 | String | | Attribute value appropriate to the NestedInstrAttribType field<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| NestedPartyID | 524 | String | | PartyID value within a nested repeating group.<br>Same values as PartyID (448) | |
| NestedPartyIDSource | 525 | char | | PartyIDSource value within a nested repeating group.<br>Same values as PartyIDSource (447) | |
| NestedPartyRole | 538 | int | | PartyRole value within a nested repeating group.<br>Same values as PartyRole (452) | |
| NetChgPrevDay | 451 | PriceOffset | | Net change from previous day's closing price vs. last traded price. | |
| NewsCategory | 20064 | int | | Describes a category of news | **Value** / **Description**<br>1 Company news<br>2 Marketplace news<br>3 Financial market news<br>4 Technical news<br>99 Other |
| NewSeqNo | 36 | SeqNum | | New sequence number | |
| NewsID | 20063 | String | | Unique identifier for News messages | |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| NewsPublRejReason | 20076 | int | | Reject reason for a news puyblication request | **Value** / **Description**<br>1 Unknown symbol<br>2 Duplicate NewsPublReqID<br>3 News reference unfound<br>4 Insufficient Permissions<br>5 Unsupported UpdateAction<br>6 Unknown Security Exchange<br>7 Unknown Market Segment<br>8 Unsupported Language<br>99 Other |
| NewsPublReqID | 20072 | String | | Unique identifier for a News Publiation Request | |
| NewsUpdAction | 20074 | int | | Type of News update action. | **Value** / **Description**<br>1 New<br>2 Change<br>3 Delete |
| NextExpectedMsgSeqNum | 789 | SeqNum | | Next expected MsgSeqNum value to be received. | |
| NoAffectedOrders | 534 | NumInGroup | | Number of affected orders in the repeating group of order ids. | |
| NoAllocs | 78 | NumInGroup | | Number of repeating AllocAccount (79)/AllocPrice (366) entries. | |
| NoDerivativeEvents | 20167 | NumInGroup | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| NoDerivativeInstrAttr | 20211 | NumInGroup | | Number of derivative instrument attributes<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| NoDerivativeInstrumentParties | 20173 | NumINGroup | | **OMX Comment:** Not in FIX. The GDC requires an extension | |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| NoDerivativeSecurityAltID | 20121 | NumInGroup | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| NoEvents | 864 | NumInGroup | | Number of repeating EventType entries. | |
| NoExecInstRules | 20182 | NumInGroup | | Number of execution instructions<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| NoInstrAttrib | 870 | NumInGroup | | Number of repeating InstrAttribType entries. | |
| NoInstrumentParties | 1018 | NumInGroup | | Identifies the number of parties identified with an instrument | |
| NoLegAllocs | 670 | NumInGroup | | Number of Allocations for the leg | |
| NoLegs | 555 | NumInGroup | | Number of InstrumentLeg repeating group instances. | |
| NoLegSecurityAltID | 604 | String | | Multileg instrument's individual security's NoSecurityAltID. See NoSecurityAltID (454) field for description | |
| NoLinesOfText | 33 | NumInGroup | | Identifies number of lines of text body | |
| NoLotTypeRules | 20185 | NumInGroup | | Number of Lot Type Rules<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| NoMarketDataFeedTypes | 20187 | NumInGroup | | Number of Feed Types<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| NoMarketSegments | 20107 | NumInGroup | | No of Markets or Market Segments which a security may trade<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| NoMatchRules | 20184 | NumInGroup | | Number of Match Rules<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| NoMaturityRules | 20188 | NumInGroup | | Number of maturity rules in MarurityRules component block<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| NoMDEntries | 268 | NumIn-Group | | Number of entries in Market Data message. | |
| NoMDStatIn-struments | 20030 | NumIn-Group | | Number of Instrument entries in a statistics message | |
| NoMDStats | 20031 | NumIn-Group | | Number of statistics entries for an instrument | |
| NoNest-ed2PartyIDs | 756 | NumIn-Group | | Number of Nested2PartyID (757), Nested2PartyIDSource (758), and Nested2PartyRole (759) entries | |
| NoNest-ed3PartyIDs | 948 | NumIn-Group | | Number of Nested3PartyID (949), Nested3PartyIDSource (950), and Nested3PartyRole (95) entries | |
| NoNestedIn-strAttr | 20212 | NumIn-Group | | Number of nested instrument attributes<br>**OMX Comment:** Not in FIX. The GDC requires an exten-sion | |
| NoNestedPar-tyIDs | 539 | NumIn-Group | | Number of NestedPartyID (524), NestedPartyIDSource (525), and NestedPartyRole (538) entries | |
| NoOrders | 73 | NumIn-Group | | Indicates number of orders to be combined for average pricing and allocation. | |
| NoOrdTypeR-ules | 20180 | NumIn-Group | | Number of order types<br>**OMX Comment:** Not in FIX. The GDC requires an exten-sion | |
| NoPartyAltID | 20248 | int | | Number of PartyAltID (20249) and PartyAltIDSource (20250) entries<br>**OMX Comment:** Not in FIX. OMX requires an extension | |
| NoPartyIDs | 453 | NumIn-Group | | Number of PartyID (448), PartyIDSource (447), and PartyRole (452) entries | |
| NoParty-SubIDs | 802 | NumIn-Group | | Number of PartySubID (523)and PartySubIDType (803) entries | |
| NoQuoteEn-tries | 295 | NumIn-Group | | The number of quote entries for a QuoteSet. | |
| NoQuoteSets | 296 | NumIn-Group | | The number of sets of quotes in the message. | |
| NoRefNews | 20066 | NumIn-Group | | Specifies the number of re-peating news references | |
| NoRelat-edSym | 146 | NumIn-Group | | Specifies the number of re-peating symbols specified. | |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| NoRootPartyIDs | 1116 | NumInGroup | | Number of RootPartyID (1117), RootPartyIDSource (1118), and RootPartyRole (1119) entries | |
| NoRootPartySubIDs | 1120 | NumInGroup | | Number of RootPartySubID (1121) and RootPartySubIDType (1122) entries | |
| NoRpts | 82 | int | | Total number of reports within series. | |
| NoSecurityAltID | 454 | NumInGroup | | Number of SecurityAltID (455) entries. | |
| NoSides | 552 | NumInGroup | | Number of Side repeating group instances. | **Value** / **Description**<br>1 — One Side<br>2 — Both Sides |
| NoStatsIndicators | 1175 | NumInGroup | | Number of statistics indicator repeating group entries<br>**OMX Comment:** FIX 5.0 SP1 | |
| NoStrikeRules | 20103 | NumInGroup | | Number of strike rule entries. This block specifies the rules for determining how new strikes should be listed within the stated price range of the underlying instrument<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| NoTickRules | 20108 | NumInGroup | | Number of tick rules. This block specifies the rules for determining how a security ticks, i.e. the price increments at which it can be quoted and traded, depending on the current price of the security<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| NoTimeInForceRules | 20181 | NumInGroup | | Number of time in force techniques<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| NoTradingSessions | 386 | NumInGroup | | Number of TradingSessionIDs (336) in repeating group. | |
| NoTradingSessions | 20204 | NumInGroup | | Allows trading rules to be expressed by trading session<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| NoTrdRepIndicators | 20087 | NumInGroup | | Number of trade reporting indicators<br>**OMX Comment:** Not in FIX. OMX requires an extension | |
| NoUnderlyings | 711 | NumInGroup | | Number of underlying legs that make up the security. | |
| NoUnderlyingSecurityAltID | 457 | NumInGroup | | Number of UnderlyingSecurityAltID (458) entries. | |
| NumberOfMatchOrders | 20056 | int | | Number of matched orders | |
| NumberOfOrders | 346 | int | | Number of orders in the market. | |
| OfferPx | 133 | Price | | Offer price/rate | |
| OfferSize | 135 | Qty | | Quantity of offer<br>(Prior to FIX 4.2 this field was of type int) | |
| 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)<br>(see Volume : "Glossary" for value definitions) | **Value** / **Description**<br>A – Agency<br>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**<br>1 – Order<br>2 – Quote<br>3 – Privately Negotiated Trade<br>4 – Multileg order<br>5 – Linked order<br>6 – Quote Request<br>7 – Implied Order<br>8 – Cross Order |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| 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.<br>(Prior to FIX 4.2 this field was of type int) | |
| 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**<br>5 — Acting as Market Maker or Specialist in the security<br>Y — Issuer Holding<br>**OMX Comment:** Not in FIX Standard. OMX requests addition<br>Z — Issue Price Stabilization<br>**OMX Comment:** Not in FIX Standard. OMX requests addition |
| 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**<br>0 — Broker / Exchange option<br>1 — Unknown symbol<br>2 — Exchange closed<br>3 — Order exceeds limit<br>4 — Too late to enter<br>5 — Unknown order<br>6 — Duplicate Order (e.g. dupe ClOrdID)<br>7 — Duplicate of a verbally communicated order<br>8 — Stale order<br>9 — Trade along required |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | **Value** / **Description**<br>10 Invalid Investor ID<br>11 Unsupported order characteristic<br>12 Surveillence Option<br>13 Incorrect quantity<br>14 Incorrect allocated quantity<br>15 Unknown account(s)<br>18 Invalid price increment<br>99 Other |
| OrdStatus | 39 | char | | Identifies current status of order. *** SOME VALUES HAVE BEEN REPLACED - See "Replaced Features and Supported Approach" *** (see Volume : "Glossary" for value definitions) | **Value** / **Description**<br>0 New<br>1 Partially filled<br>2 Filled<br>4 Canceled<br>8 Rejected<br>9 Suspended<br>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**<br>1 Market<br>2 Limit<br>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. | |
| OrigNewsPubReqID | 20073 | String | | Reference to the identifier for a News Publiation Request | |
| OrigOrdModTime | 586 | UTCTimestamp | | The most recent (or current) modification TransactTime (tag 60) reported on an Execution Report for the order. The OrigOrdModTime is provided as an optional field on Order Cancel Request and Order Cancel Replace Re- | |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | quests to identify that the state of the order has not changed since the request was issued.<br>This is provided to support markets similar to Eurex and A/C/E. | |
| 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. | |
| OrigTime | 42 | UTCTimestamp | | Time of message origination (always expressed in UTC (Universal Time Coordinated, also known as "GMT")) | |
| 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 (20036) | |
| PartOfSecListID | 20238 | String | | Indicates a higher level Security List that this list is part of<br>**OMX Comment:** Not in FIX. OMX requires an extension | |
| PartyAltID | 20249 | String | | Alternate Party identifier/code. See PartyAltIDSource (20250). See "Appendix 6-G – Use of <Parties> Component Block"<br>**OMX Comment:** Not in FIX. OMX requires an extension | |
| PartyAltIDSource | 20250 | char | | Identifies class or source of the PartyAltID (20249) value. Required if PartyAltID is specified.<br>See "Appendix 6-G – Use of <Parties> Component Block"<br>**OMX Comment:** Not in FIX. OMX requires an extension | |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| PartyID | 448 | String | | Party identifier/code. See PartyIDSource (447) and PartyRole (452).<br>See "Appendix 6-G – Use of <Parties> Component Block" | |
| PartyID-Source | 447 | char | | Identifies class or source of the PartyID (448) value. Required if PartyID is specified. Note: applicable values depend upon PartyRole (452) specified.<br>See "Appendix 6-G – Use of <Parties> Component Block" | **Value** / **Description**<br>C – Generally accepted market participant identifier (e.g. NASD mnemonic)<br>D – Proprietary / Custom code<br>H – CSD participant/member code (e.g.. Euroclear, DTC, CREST or Kassenverein number)<br>**OMX Comment:** Includes Clearing house participant/member code |
| PartyListUpdateAction | 20251 | char | | Specifies the action taken for the specified PartyID.<br>**OMX Comment:** Not in FIX. OMX requires an extension | **Value** / **Description**<br>A – Add<br>D – Delete<br>M – Modify |
| 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**<br>0 – Valid request<br>1 – Invalid or unsupported request<br>2 – No data found that match selection criteria<br>3 – Not authorized to retrieve data |
| PartyRole | 452 | int | | Identifies the type or role of the PartyID (448) specified.<br>See "Appendix 6-G – Use of <Parties> Component Block"<br>(see Volume : "Glossary" for value definitions) | **Value** / **Description**<br>1 – Executing Firm (formerly FIX 4.2 ExecBroker)<br>**OMX Comment:** The role of the firm legally |

| FieldName | Tag | Type | OMXLen | Desc | Valid values | |
|---|---|---|---|---|---|---|
| | | | | | **Value** | **Description** |
| | | | | | | responsible for a business transaction sent to the marketplace |
| | | | | | 3 | Client ID (formerly FIX 4.2 ClientID)<br>**OMX Comment:** Used when specifying an allocation to a defined client |
| | | | | | 7 | Entering Firm<br>**OMX Comment:** The role of a firm performing data entry on behalf of the executing firm (includes marketplace operations) |
| | | | | | 10 | Settlement Location (formerly FIX 4.2 SettlLocation)<br>**OMX Comment:** The role of a CSD (or similar) in the context of trades |
| | | | | | 12 | Executing Trader (associated with Executing Firm - actually executes)<br>**OMX Comment:** The role of the individual responsible for a business transaction sent to the marketplace |
| | | | | | 17 | Contra Firm<br>**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<br>**OMX Comment:** The role of a Clearing House / CCP in the context of trades |
| | | | | | 22 | Exchange<br>**OMX Comment:** Used in Trade Capture Reports when a |

| FieldName | Tag | Type | OMXLen | Desc | Valid values | |
|---|---|---|---|---|---|---|
| | | | | | **Value** | **Description** |
| | | | | | | user request that a trade is not made public over market data. |
| | | | | | 34 | Regulatory body |
| | | | | | | **OMX Comment:** Used in Trade Capture Reports when a user requests that a trade is not reported to the regulator. Only allowed when the user reports trades directly to the regulator |
| | | | | | 36 | Entering trader |
| | | | | | | **OMX Comment:** The role of an individual performing data entry on behalf of the executing "trader" (includes marektplace officials) |
| | | | | | 37 | Contra trader |
| | | | | | | **OMX Comment:** Used in Trade Capture Reports for trade confirmations divulding a single side only. Also used in Quote Requests to specify receiving parties |
| | | | | | 58 | Entering Unit |
| | | | | | | **OMX Comment:** The rolle of an organizational unit the performs data entry on behalf of an executing firm |
| | | | | | 59 | Executing Unit |
| | | | | | | **OMX Comment:** The role of an organizational unit the is responsible for a business transaction sent to the marketplace |
| | | | | | 72 | Reporting intermediary (medium/vendor via which report has been published) |
| | | | | | 1000 | Access provider |
| | | | | | | **OMX Comment:** For use in Parties refer- |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | **Value** / **Description**<br>ence data mesages only<br>1001 / Vendor<br>**OMX Comment:** For use in Parties reference data mesages only<br>1002 / ISV<br>**OMX Comment:** For use in Parties reference data mesages only<br>1003 / Periferal application<br>**OMX Comment:** For use in Parties reference data mesages only<br>1004 / Fund Manager<br>**OMX Comment:** For use in Parties reference data mesages only<br>1005 / Other<br>**OMX Comment:** For use in Parties reference data mesages only |
| 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<br>4000+ = Reserved and available for bi-laterally agreed upon user defined values | **Value** / **Description**<br>1 / Firm<br>2 / Person<br>5 / Full legal name of firm<br>6 / Postal address<br>7 / Phone number<br>8 / Email address<br>9 / Contact name<br>16 / BIC |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | **Value** / **Description**<br>18 — Registered address<br>21 — Fax number<br>25 — Location desk |
| Password | 554 | String | | Password or passphrase. | |
| PctMatchQty | 20062 | Percentage | | % of quantity that matched | |
| Pool | 691 | String | | For Fixed Income, identifies MBS / ABS pool. | |
| PositionEffect | 77 | char | | Indicates whether the resulting position after a trade should be an opening position or closing position. Used for omnibus accounting - where accounts are held on a gross basis instead of being netted together. | **Value** / **Description**<br>C — Close<br>O — Open<br>N — Close but Notify on Open<br>**OMX Comment:** Not in FIX Standard. OMX requests addition<br>D — Default<br>**OMX Comment:** Not in FIX Standard. OMX requests addition |
| PossDupFlag | 43 | Boolean | | Indicates possible retransmission of message with this sequence number | **Value** / **Description**<br>N — Original transmission<br>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**<br>N — Not reported to counterparty<br>Y — Perviously reported to counterparty |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| Price | 44 | Price | | Price per unit of quantity (e.g. per share) | |
| PriceLimit-Type | 20208 | int | | Describes the how the price limits are expressed<br>**OMX Comment:** Not in FIX. The GDC requires an extension | **Value** / **Description**<br>0 – Price<br>1 – Ticks<br>2 – Percentage |
| Price-QuoteMethod | 20098 | String | | Method for price quotation<br>**OMX Comment:** Not in FIX. The GDC requires an extension | **Value** / **Description**<br>STD – Standard, money per unit of a physical<br>INDX – Index<br>INT – Interest rate index |
| PriceType | 423 | int | | Code to represent the price type.<br>(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".<br>See Volume : "Glossary" for further value definitions) | **Value** / **Description**<br>1 – Percentage (i.e. percent of par) (often called "dollar price" for fixed income)<br>**OMX Comment:** Only relevant for Fixed Income trading<br>2 – Per unit (i.e. per share or contract)<br>**OMX Comment:** Default value, should be specified (if applicable) for Fixed Income trading<br>3 – Fixed amount (absolute value)<br>**OMX Comment:** Only allowed for IOI's<br>4 – Discount - percentage points below par<br>**OMX Comment:** Only allowed for IOI's<br>5 – Premium - percentage points over par<br>**OMX Comment:** Only allowed for IOI's<br>6 – Spread (basis points spread)<br>**OMX Comment:** Only allowed for IOI's<br>7 – TED Price |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | **Value** / **Description**<br>**OMX Comment:** Only allowed for IOI's<br>8 / TED Yield<br>**OMX Comment:** Only allowed for IOI's<br>9 / Yield<br>**OMX Comment:** Only relevant for Fixed Income trading |
| PriceUnitOfMeasure | 20093 | String | | Used to express the UOM of the price if different from the contract. In futures, this can be different for cross-rate products in which the price is quoted in units differently from the contract<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| PriceUnitOfMeasureQty | 20094 | Qty | | Used to express the UOM Quantity of the price if different from the contract. In futures, this can be different for physically delivered products in which price is quoted in a unit size different from the contract, i.e. a Cattle Future contract has a UOMQty of 40,000 and a PriceUOMQty of 100.<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| PrivateQuote | 1171 | Boolean | | Specifies whether a quote is public, i.e. available to the market, or private, i.e. available to a specified counterparty only.<br>Valid Values:<br>TRUE = Private Quote<br>FALSE = Public Quote<br>**OMX Comment:** FIX 5.0 SP1 | |
| Product | 460 | int | | Indicates the type of product the security is associated with. See also the CFICode (461) and SecurityType (167) fields. | **Value** / **Description**<br>1 / AGENCY<br>2 / COMMODITY<br>3 / CORPORATE<br>4 / CURRENCY |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | **Value** / **Description**<br>5 EQUITY<br>6 GOVERNMENT<br>7 INDEX<br>8 LOAN<br>9 MONEYMARKET<br>10 MORTGAGE<br>11 MUNICIPAL<br>12 OTHER<br>13 FINANCING |
| ProductComplex | 20198 | String | | Identifies an entire suite of products for a given market. In Futures this may be “interest rates”, “agricultural”, “equity indexes”, etc<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| PublishTrdIndicator | 852 | int | | Indicates if a trade should be reported via a market reporting service. The indicator governs all reporting services of the receipient.<br>**OMX Comment:** Field is Boolean in FIX 5.0, OMX requests change | **Value** / **Description**<br>0 Do Not Report Trade<br>1 Report Trade<br>2 Deferred Publication |
| PublTime | 20075 | UTCTimestamp | | Publication date and time | |
| PutOrCall | 201 | int | | Indicates whether an Option is for a put or call<br>**OMX Comment:** The GDC requests re-introduction | **Value** / **Description**<br>0 Put<br>1 Call |
| QuoteCancelType | 298 | int | | Identifies the type of quote cancel. | **Value** / **Description**<br>1 Cancel for Symbol(s)<br>3 Cancel for Underlying Symbol<br>4 Cancel All Quotes<br>5 Cancel quote specified in QuoteID |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| QuoteCondition | 276 | MultipleStringValue | | Space-delimited list of conditions describing a quote. | **Value** / **Description**<br>Z — Order Imbalance<br>3 — Rest of Book VWAP<br>**OMX Comment:** FIX 5.0 SP1<br>4 — Better Prices in Conditional Orders<br>**OMX Comment:** FIX 5.0 SP1<br>9 — Median Price<br>**OMX Comment:** Not in FIX Standard. OMX requests addition |
| QuoteEntryID | 299 | String | | Uniquely identifies the quote as part of a QuoteSet.<br>**OMX Comment:** Unique identifier for a quote. The QuoteEntryID stays with the quote as a static identifier even if the quote is updated. | |
| QuoteEntryRejectReason | 368 | int | | Reason Quote Entry was rejected: | **Value** / **Description**<br>1 — Unknown symbol (security)<br>3 — Quote exceeds limit<br>5 — Unknown quote<br>6 — Duplicate quote<br>7 — Invalid bid/ask spread<br>8 — Invalid price<br>9 — Not authorized to quote security<br>99 — Other |
| QuoteEntryStatus | 1167 | int | | Identifies the status of an individual quote. See also QuoteStatus (297) which used for single Quotes.<br>**OMX Comment:** FIX 5.0 SP1 | **Value** / **Description**<br>0 — Accepted<br>**OMX Comment:** FIX 5.0 SP1<br>5 — Rejected<br>**OMX Comment:** FIX 5.0 SP1<br>6 — Removed from Market |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | **Value** / **Description**<br>**OMX Comment:** FIX 5.0 SP1<br>7 Expired<br>**OMX Comment:** FIX 5.0 SP1<br>16 Active<br>**OMX Comment:** FIX 5.0 SP1 |
| QuoteID | 117 | String | | Unique identifier for quote<br>**OMX Comment:** Unique identifier for a quote message | |
| QuoteMsgID | 1166 | String | | Unique identifier for a quote message<br>**OMX Comment:** FIX 5.0 SP1 | |
| QuoteRejectReason | 300 | int | | Reason Quote was rejected: | **Value** / **Description**<br>5 Unknown Quote<br>6 Duplicate Quote<br>10 Quote Locked - Unable to Update/Cancel<br>**OMX Comment:** FIX 5.0 SP1<br>99 Other |
| QuoteReqID | 131 | String | | Unique identifier for quote request | |
| QuoteRequestRejectReason | 658 | int | | Reason Quote was rejected: | **Value** / **Description**<br>1 Unknown Symbol (Security)<br>2 Exchange (Security) Closed<br>3 Quote Request Exceeds Limit<br>5 Invalid Price<br>6 Not Authorized To Request Quote<br>7 No Match For Inquiry<br>8 No Market For Instrument<br>99 Other |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| QuoteRespID | 693 | String | | Message reference for Quote Response | |
| QuoteResponseLevel | 301 | int | | Level of Response requested from receiver of quote messages. | **Value** / **Description**<br>0 — No Acknowledgement (default)<br>1 — Acknowledge only negative or erroneous quotes<br>**OMX Comment:** Default<br>2 — Acknowledge each quote messages<br>3 — Summary Acknowledgement<br>**OMX Comment:** FIX 5.0 SP1 |
| QuoteRespType | 694 | int | | Identifies the type of Quote Response. | **Value** / **Description**<br>1 — Hit/Lift<br>2 — Counter<br>6 — Pass |
| QuoteSetID | 302 | String | | Unique id for the Quote Set. | |
| QuoteStatus | 297 | int | | Identifies the status of the quote acknowledgement. | **Value** / **Description**<br>5 — Rejected<br>6 — Removed from Market<br>7 — Expired<br>8 — Query<br>9 — Quote Not Found<br>16 — Active<br>**OMX Comment:** FIX 5.0 SP1<br>17 — Canceled<br>**OMX Comment:** FIX 5.0 SP1<br>18 — Unsolicited Quote Replenishment<br>**OMX Comment:** FIX 5.0 SP1 |
| QuoteStatusReqID | 649 | String | | Unique identifier for Quote Status Request. | |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| QuoteType | 537 | int | | Identifies the type of quote. An indicative quote is used to inform a counterparty of a market. An indicative quote does not result directly in a trade.<br>A tradeable quote is submitted to a market and will result directly in a trade against other orders and quotes in a market.<br>A restricted tradeable quote is submitted to a market and within a certain restriction (possibly based upon price or quantity) will automatically trade against orders. Order that do not comply with restrictions are sent to the quote issuer who can choose to accept or decline the order.<br>A counter quote is used in the negotiation model. See Volume 7 – Product: Fixed Income for example usage. | Value: Description<br>0: Indicative<br>1: Tradeable |
| RefMsgType | 372 | String | | The MsgType (35) of the FIX message being referenced. | |
| RefNewsID | 20067 | String | | NewsID of referenced News message | |
| RefNewsType | 20068 | int | | Specifies the type of news reference. | Value: Description<br>1: Replacement<br>2: Other language<br>3: Complimentary |
| 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<br>0: SecondaryOrderID (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<br>N: No |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | **Value** / **Description**<br>Y – Yes, reset sequence numbers |
| RespondentType | 1172 | int | | Specifies the type of respondents requested.<br>**OMX Comment:** FIX 5.0 SP1 | **Value** / **Description**<br>1 – All market participants<br>2 – Specified market participants<br>3 – All market makers<br>4 – Primary market makers(s) |
| RFQReqID | 644 | String | | RFQ Request ID – used to identify an RFQ Request. | |
| RiskfreeRate | 20232 | float | | Specifies the expected risk-free interest rate<br>**OMX Comment:** Not in FIX. OMX requires an extension | |
| 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) | |
| RootPartySubID | 1121 | String | | PartySubID value within a root parties component. Same values as PartySubID (523) | |
| RootPartySubIDType | 1122 | int | | Type of RootPartySubID (1121) value. Same values as PartySubIDType (803) | |
| RoundLot | 561 | Qty | | The trading lot size of a security | |
| RptSeq | 83 | int | | Sequence number of message within report series. Used to carry reporting sequence number of the fill as represented on the Trade Report Side. | |
| SecListDesc | 20236 | String | | Name or description of a Security List<br>**OMX Comment:** Not in FIX. OMX requires an extension | |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| SecListID | 20235 | String | | Identifies a Security List message<br>**OMX Comment:** Not in FIX. OMX requires an extension | |
| SecListType | 20237 | int | | Type of Security List<br>**OMX Comment:** Not in FIX. OMX requires an extension | **Value** / **Description**<br>0 Traded Security<br>1 Turnover List<br>2 Index population |
| 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. | |
| 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.<br>**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. | |
| SecondaryHighLimitPrice | 20206 | Price | | Allowable high limit price for the trading day. A key parameter in validating order price. Used as the upper band for validating order prices. Orders submitted with prices above the upper limit will be rejected<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| SecondaryLowLimitPrice | 20205 | Price | | Allowable low limit price for the trading day. A key parameter in validating order price. Used as the lower band for validating order prices. Orders submitted with prices below the lower limit will be rejected<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 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. | |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| SecondaryPriceLimitType | 20209 | int | | Describes the how the price limits are expressed<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| SecondaryTradingReferencePrice | 20207 | Price | | Reference price for the current trading price range usually representing the mid price between the HighLimitPrice and LowLimitPrice. The value may be the settlement price or closing price of the prior trading day.<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| SecondaryTrdType | 855 | int | | Additional TrdType (see tag 828) assigned to a trade by trade match system. | |
| SecurityAltID | 455 | String | | Alternate Security identifier value for this security of SecurityAltIDSource (456) type (e.g. CUSIP, SEDOL, ISIN, etc). Requires SecurityAltIDSource. | |
| SecurityAltIDSource | 456 | String | | Identifies class or source of the SecurityAltID (455) value. Required if SecurityAltID is specified.<br>Valid values:<br>Same valid values as the SecurityIDSource (22) field | |
| SecurityDesc | 107 | String | | Security description. | |
| SecurityExchange | 207 | Exchange | | Market used to help identify a security.<br>Valid values:<br>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.<br>100+ are reserved for private security identifications | **Value** / **Description**<br>99 / Marketplace-assigned Identifier |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| 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**<br>0 Valid request<br>1 Invalid or unsupported request<br>2 No instruments found that match selection criteria<br>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**<br>1 Accept security proposal as-is<br>2 Accept security proposal with revisions as indicated in the message<br>5 Reject security proposal |
| SecurityStatus | 965 | String | | Used for derivatives. Denotes the current state of the Instrument. | **Value** / **Description**<br>1 Active<br>2 Inactive |
| SecuritySubType | 762 | String | | Sub-type qualification/identification of the SecurityType (e.g. for SecurityType="REPO"), or the CFICode if SecurityType is not specified. If specified, SecuirtyType or CFICode is required. Example Values:<br>General = General Collateral (for SecurityType=REPO)<br>For SecurityType="MLEG" markets can provide the name of the option or futures strategy, such as Calendar, Vertical, Butterfly, etc. | |

<table>
<tr><th>FieldName</th><th>Tag</th><th>Type</th><th>OMXLen</th><th>Desc</th><th>Valid values</th></tr>
<tr><td></td><td></td><td></td><td></td><td>NOTE: Additional values may be used by mutual agreement of the counterparties</td><td></td></tr>
<tr><td>Security-TradingEvent</td><td>1174</td><td>int</td><td></td><td>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.<br>OMX Comment: FIX 5.0 SP1</td><td><table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Order imbalance, auction is extended</td></tr><tr><td>2</td><td>Price Volatility Interruption</td></tr><tr><td>3</td><td>Trading resumes (after Halt)</td></tr><tr><td>4</td><td>Change of Trading Session</td></tr><tr><td>5</td><td>Change of Trading Subsession</td></tr><tr><td>6</td><td>Change of Security Status</td></tr><tr><td>7</td><td>Change of Book Type</td></tr><tr><td>8</td><td>Change of Market Depth</td></tr><tr><td>100</td><td>Session change alert</td></tr></table></td></tr>
<tr><td>SecurityTradingStatus</td><td>326</td><td>int</td><td></td><td>Identifies the trading status applicable to the transaction.</td><td><table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Opening delay</td></tr><tr><td>2</td><td>Trading halt</td></tr><tr><td>7</td><td>Market Imbalance Buy</td></tr><tr><td>8</td><td>Market Imbalance Sell</td></tr><tr><td>17</td><td>Ready to trade (start of session)</td></tr><tr><td>18</td><td>Not available for trading (end of session)</td></tr><tr><td>21</td><td>Pre-open</td></tr><tr><td>22</td><td>Opening Rotation</td></tr><tr><td>23</td><td>Fast Market</td></tr><tr><td>100</td><td>Hidden Auction</td></tr><tr><td>101</td><td>Open Auction</td></tr><tr><td>102</td><td>Issuer Position Modification</td></tr></table></td></tr>
</table>

<table>
<tr><th>FieldName</th><th>Tag</th><th>Type</th><th>OMXLen</th><th>Desc</th><th>Valid values</th></tr>
<tr><td>SecurityType</td><td>167</td><td>String</td><td></td><td>Indicates type of security. See also the Product (460) and CFICode (461) fields. It is recommended that CFICode be used instead of SecurityType for non-Fixed Income instruments.<br>Example values (grouped by Product field value) (Note: additional values may be used by mutual agreement of the counterparties):<br>* Identify the Issuer in the "Issuer" field(106)<br>*** REPLACED values - See "Replaced Features and Supported Approach" ***<br>NOTE: Additional values may be used by mutual agreement of the counterparties)<br>OMX Comment: Valid values are not yet defined!</td><td>
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>ABS</td><td>Asset-backed Securities</td></tr>
<tr><td>AN</td><td>Other Anticipation Notes (BAN, GAN, etc.)</td></tr>
<tr><td>BA</td><td>Bankers Acceptance</td></tr>
<tr><td>BRADY</td><td>Brady Bond</td></tr>
<tr><td>CORP</td><td>Corporate Bond</td></tr>
<tr><td>CS</td><td>Common Stock</td></tr>
<tr><td>EUSUPRA</td><td>Euro Supranational Coupons *</td></tr>
<tr><td>FOR</td><td>Foreign Exchange Contract</td></tr>
<tr><td>MF</td><td>Mutual Fund</td></tr>
<tr><td>REPO</td><td>Repurchase</td></tr>
<tr><td>TERM</td><td>Term Loan</td></tr>
<tr><td>BN</td><td>Bank Notes</td></tr>
<tr><td>CMBS</td><td>Corp. Mortgage-backed Securities</td></tr>
<tr><td>COFO</td><td>Certificate Of Obligation</td></tr>
<tr><td>CPP</td><td>Corporate Private Placement</td></tr>
<tr><td>EUSOV</td><td>Euro Sovereigns *</td></tr>
<tr><td>FAC</td><td>Federal Agency Coupon</td></tr>
<tr><td>FORWARD</td><td>Forward</td></tr>
<tr><td>FUT</td><td>Future</td></tr>
<tr><td>MLEG</td><td>Multileg Instrument</td></tr>
<tr><td>PS</td><td>Preferred Stock</td></tr>
<tr><td>RVLV</td><td>Revolver Loan</td></tr>
<tr><td>BOX</td><td>Bill Of Exchanges</td></tr>
<tr><td>BUYSELL</td><td>Buy Sellback</td></tr>
<tr><td>CB</td><td>Convertible Bond</td></tr>
<tr><td>CMO</td><td>Collateralized Mortgage Obligation</td></tr>
<tr><td>COFP</td><td>Certificate Of Participation</td></tr>
</table>
</td></tr>
</table>

| FieldName | Tag | Type | OMXLen | Desc | Valid values | |
|---|---|---|---|---|---|---|
| | | | | | **Value** | **Description** |
| | | | | | FADN | Federal Agency Discount Note |
| | | | | | NONE | No Security Type |
| | | | | | OPT | Option |
| | | | | | RVLVTRM | Revolver/Term Loan |
| | | | | | TBOND | US Treasury Bond |
| | | | | | BRIDGE | Bridge Loan |
| | | | | | CD | Certificate Of Deposit |
| | | | | | DUAL | Dual Currency |
| | | | | | GO | General Obligation Bonds |
| | | | | | IET | IOETTE Mortgage |
| | | | | | PEF | Private Export Funding * |
| | | | | | SECLOAN | Securities Loan |
| | | | | | TINT | Interest Strip From Any Bond Or Note |
| | | | | | UST | US Treasury Note (Deprecated Value Use TNOTE) |
| | | | | | OOF | Options on Futures |
| | | | | | CL | Call Loans |
| | | | | | EUCORP | Euro Corporate Bond |
| | | | | | LOFC | Letter Of Credit |
| | | | | | MBS | Mortgage-backed Securities |
| | | | | | MT | Mandatory Tender |
| | | | | | SECPLEDGE | Securities Pledge |
| | | | | | SUPRA | USD Supranational Coupons * |
| | | | | | TIPS | Treasury Inflation Protected Securities |
| | | | | | USTB | US Treasury Bill (Deprecated Value Use TBILL) |
| | | | | | OOP | Options on Physical |
| | | | | | CP | Commercial Paper |

| FieldName | Tag | Type | OMXLen | Desc | Valid values | |
|---|---|---|---|---|---|---|
| | | | | | **Value** | **Description** |
| | | | | | MIO | Mortgage Interest Only |
| | | | | | RAN | Revenue Anticipation Note |
| | | | | | SWING | Swing Line Facility |
| | | | | | TCAL | Principal Strip Of A Callable Bond Or Note |
| | | | | | WLD | Wildcard Entry (was "?" in 4.4, used on Security Definition Request message) |
| | | | | | XLINKD | Indexed Linked |
| | | | | | DINP | Debtor In Possession |
| | | | | | DN | Deposit Notes |
| | | | | | MPO | Mortgage Principal Only |
| | | | | | REV | Revenue Bonds |
| | | | | | STRUCT | Structured Notes |
| | | | | | TPRN | Principal Strip From A Non-Callable Bond Or Note |
| | | | | | CASH | Cash |
| | | | | | DEFLT-ED | Defaulted |
| | | | | | EUCD | Euro Certificate Of Deposit |
| | | | | | MPP | Mortgage Private Placement |
| | | | | | SPCLA | Special Assessment |
| | | | | | TNOTE | US Treasury Note |
| | | | | | YANK | Yankee Corporate Bond |
| | | | | | EUCP | Euro Commercial Paper |
| | | | | | MPT | Miscellaneous Pass-through |
| | | | | | SPCLO | Special Obligation |
| | | | | | TBILL | US Treasury Bill |
| | | | | | WITH-DRN | Withdrawn |
| | | | | | LQN | Liquidity Note |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | **Value** / **Description**<br>PFAND / Pfandbriefe *<br>RE-PLACD / Replaced<br>SPCLT / Special Tax<br>MA-TURED / Matured<br>MTN / Medium Term Notes<br>TAN / Tax Anticipation Note<br>TBA / To Be Announced<br>AMEND-ED / Amended & Restated<br>ONITE / Overnight<br>TAXA / Tax Allocation<br>PN / Promissory Note<br>RE-TIRED / Retired<br>TECP / Tax Exempt Commer-cial Paper<br>PZFJ / Plazos Fijos<br>TRAN / Tax Revenue Antici-pation Note<br>STN / Short Term Loan Note<br>VRDN / Variable Rate De-mand Note<br>TD / Time Deposit<br>WAR / Warrant<br>XCN / Extended Comm Note<br>YCD / Yankee Certificate Of Deposit |
| SecurityUp-dateAction | 980 | char | | | **Value** / **Description**<br>A / Add<br>D / Delete<br>M / Modify |
| SenderCom-pID | 49 | String | | Assigned value used to iden-tify firm sending message. | |

<table>
<tr><th>FieldName</th><th>Tag</th><th>Type</th><th>OMXLen</th><th>Desc</th><th>Valid values</th></tr>
<tr><td>SenderSubID</td><td>50</td><td>String</td><td></td><td>Assigned value used to identify specific message originator (desk, trader, etc.)</td><td></td></tr>
<tr><td>SendingTime</td><td>52</td><td>UTC-Timestamp</td><td></td><td>Time of message transmission (always expressed in UTC (Universal Time Coordinated, also known as "GMT")</td><td></td></tr>
<tr><td>SessionRejectReason</td><td>373</td><td>int</td><td></td><td>Code to identify reason for a session-level Reject message.</td><td><table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>Invalid Tag Number</td></tr><tr><td>1</td><td>Required Tag Missing</td></tr><tr><td>2</td><td>Tag not defined for this message type</td></tr><tr><td>3</td><td>Undefined tag</td></tr><tr><td>4</td><td>Tag specified without a value</td></tr><tr><td>5</td><td>Value is incorrect (out of range) for this tag</td></tr><tr><td>6</td><td>Incorrect data format for value</td></tr><tr><td>7</td><td>Decryption problem</td></tr><tr><td>8</td><td>Signature problem</td></tr><tr><td>9</td><td>CompID problem</td></tr><tr><td>10</td><td>SendingTime Accuracy Problem</td></tr><tr><td>11</td><td>Invalid MsgType</td></tr><tr><td>12</td><td>XML Validation Error</td></tr><tr><td>13</td><td>Tag appears more than once</td></tr><tr><td>14</td><td>Tag specified out of required order</td></tr><tr><td>15</td><td>Repeating group fields out of order</td></tr><tr><td>16</td><td>Incorrect NumInGroup count for repeating group</td></tr><tr><td>17</td><td>Non "Data" value includes field delimiter (<SOH> character)</td></tr></table></td></tr>
<tr><td>SettlCurrency</td><td>120</td><td>Currency</td><td></td><td>Currency code of settlement denomination.</td><td></td></tr>
</table>

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| SettlDate | 64 | LocalMktDate | | Specific date of trade settlement (SettlementDate) in YYYYMMDD format.<br>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)<br>(expressed in local time at place of settlement) | |
| SettlMethod | 20095 | char | | Settlement method for a contract. Can be used as an alternative to CFI Code value<br>**OMX Comment:** Not in FIX. The GDC requires an extension | **Value** / **Description**<br>C – Cash settlement required<br>P – Physical settlement required |
| Side | 54 | char | | Side of order (see Volume : "Glossary" for value definitions) | **Value** / **Description**<br>1 – Buy<br>2 – Sell<br>B – "As Defined" (for use with multileg instruments)<br>**OMX Comment:** Valid for multileg Orders only |
| SolicitedFlag | 377 | Boolean | | Indicates whether or not the order was solicited. | **Value** / **Description**<br>N – Was not solicited<br>Y – Was solicited |
| StartDate | 916 | LocalMktDate | | Start date of a financing deal, i.e. the date the buyer pays the seller cash and takes control of the collateral | |
| StartMMY | 20189 | month-year | | Starting maturity month year for an option class<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| StartStrikePxRange | 20104 | Price | | Starting price for the range to which the StrikeIncrement applies. Price refers to the price of the underlying<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| StartTick-PriceRange | 20109 | Price | | Starting price range for specified tick increment<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| StatsType | 1176 | Int | | Type of statistics the MDEntry is eligible to be included in<br>**OMX Comment:** FIX 5.0 SP1 | **Value** / **Description**<br>1 Exchange Last<br>2 High / Low Price<br>3 Average Price (VWAP, TWAP...)<br>4 Turnover (Px * Qty) |
| StrikeCurrency | 947 | Currency | | Currency in which the StrikePrice is denominated. | |
| StrikeExerciseStyle | 20194 | int | | Expiration Style for an option class:<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| StrikeIncrement | 20106 | float | | Value by which strike price should be incremented within the specified price range.<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| StrikeMultiplier | 967 | float | | Used for derivatives. Multiplier applied to the strike price for the purpose of calculating the settlement value. | |
| StrikePrice | 202 | Price | | Strike Price for an Option. | |
| StrikeRuleID | 20195 | String | | Allows strike rule to be referenced via an identifier so that rules do not need to be explicitly enumerated<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| StrikeValue | 968 | float | | Used for derivatives. The number of shares/units for the financial instrument involved in the option trade. | |
| Suspended | 20083 | Boolean | | Indicates whether a suspension applies or not. | **Value** / **Description**<br>0 Not suspended<br>1 Suspended |
| Symbol | 55 | String | | Ticker symbol. Common, "human understood" represen- | |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | tation 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. | |
| SymbolSfx | 65 | String | | Additional information about the security (e.g. preferred, warrants, etc.). Note also see SecurityType (167). As defined in the NYSE Stock and bond Symbol Directory and in the AMEX Fitch Directory. | **Value** / **Description**<br>WI — "When Issued" for a security to be reissued under an old CUSIP or ISIN |
| 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. | |
| TerminationType | 788 | int | | Type of financing termination. | **Value** / **Description**<br>1 — Overnight<br>2 — Term<br>3 — Flexible<br>4 — Open |
| 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) | |
| TickDirection | 274 | char | | Direction of the "tick". | **Value** / **Description**<br>0 — Plus Tick<br>1 — Zero-Plus Tick<br>2 — Minus Tick<br>3 — Zero-Minus Tick |
| TickIncrement | 20111 | Price | | Tick increment for stated price range. Specifies the valid price increments at which a security can be quoted and traded<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| TickRule-Type | 20112 | int | | Specifies the type of tick rule which is being described<br>**OMX Comment:** Not in FIX. The GDC requires an extension | **Value** / **Description**<br>0 Regular<br>1 Variable<br>2 Fixed<br>3 Traded as spread leg<br>4 Settled as spread leg |
| 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**<br>0 Day (or session)<br>1 Good Till Cancel (GTC)<br>2 At the Opening (OPG)<br>3 Immediate Or Cancel (IOC)<br>4 Fill Or Kill (FOK)<br>5 Good Till Crossing (GTX)<br>6 Good Till Date (GTD)<br>7 At the Close<br>y Good Through Crossing<br>**OMX Comment:** Not in FIX Standard. OMX requests addition<br>z At Crossing<br>**OMX Comment:** Not in FIX Standard. OMX requests addition |
| TimeUnit | 997 | String | | Unit of time associated with the contract.<br>NOTE: Additional values may be used by mutual agreement of the counterparties | **Value** / **Description**<br>H Hour<br>Min Minute<br>S Second<br>D Day<br>Wk Week<br>Mo Month<br>Yr Year |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| ToNoRelParties | 20082 | int | | Total number of participants | |
| TotalAffectedOrders | 533 | int | | Total number of orders affected by mass cancel request. | |
| TotNoAccQuotes | 1169 | int | | Specifies the number of accepted quotes<br>**OMX Comment:** FIX 5.0 SP1 | |
| TotNoCxldQuotes | 1168 | int | | Specifies the number of canceled quotes<br>**OMX Comment:** FIX 5.0 SP1 | |
| TotNoQuoteEntries | 304 | int | | Total number of quotes for the quote set across all messages. Should be the sum of all NoQuoteEntries (295) in each message that has repeating quotes that are part of the same quote set. (Prior to FIX 4.4 this field was named TotQuoteEntries) | |
| TotNoRejQuotes | 1170 | int | | Specifies the number of rejected quotes<br>**OMX Comment:** FIX 5.0 SP1 | |
| 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**<br>P — Imbalance More Buyers (cannot be used in combination with Q)<br>Q — Imbalance More Sellers (cannot be used in combination with P)<br>X — Crossed<br>AV — Outside Spread<br>**OMX Comment:** FIX 5.0 SP1<br>ZX — Marketplace Entered Trade<br>**OMX Comment:** Not in FIX Standard. OMX requests addition<br>ZY — Multileg to Multileg Trade<br>**OMX Comment:** Not in FIX Standard. |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | **Value** / **Description**<br>OMX requests addition<br>ZZ / Cross Asset Multileg Trade<br>**OMX Comment:** Not in FIX Standard. OMX requests addition |
| 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** / **Description**<br>0 / Trade Confirmation<br>2 / 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. | |
| TradeOriginationDate | 229 | LocalMktDate | | Used with Fixed Income for Muncipal New Issue Market. Agreement in principal between counter-parties prior to actual trade date. (Note tag # was reserved in FIX 4.1, added in FIX 4.3)<br>(prior to FIX 4.4 field was of type UTCDate) | |
| TradeQty | 20058 | Qty | | Fill quantity | |
| 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** / **Description**<br>0 / Successful (default)<br>1 / Invalid party onformation<br>2 / Unknown instrument<br>3 / Unauthorized to report trades |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | **Value** / **Description**<br>4 — Invalid trade type<br>99 — Other |
| TradeReport-TransType | 487 | int | | Identifies Trade Report message transaction type (Prior to FIX 4.4 this field was of type char) | **Value** / **Description**<br>0 — New<br>1 — Cancel<br>2 — Replace |
| TradeReport-Type | 856 | int | | Type of Trade Report | **Value** / **Description**<br>0 — Submit<br>1 — Alleged<br>2 — Accept<br>3 — Decline<br>4 — Addendum<br>5 — No/Was<br>7 — (Locked-In) Trade Break<br>11 — Alleged New<br>12 — Alleged Addendum<br>14 — Alleged Trade Report Cancel<br>15 — Alleged (Locked-In) Trade Break |
| TradeVWAP | 20060 | Price | | Volume Weigthed Average Price for fills | |
| TradingCurrency | 20203 | Currency | | Used when the trading currency can differ from the price currency<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| TradingReferencePrice | 1150 | Price | | Reference price for the current trading price range usually representing the mid price between the HighLimitPrice and LowLimitPrice. The value may be the settlement price or closing price of the prior trading day. | |
| TradingSessionDesc | 20224 | String | | Trading Session description | |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | **OMX Comment:** Not in FIX. The EEWG requires an extension | |
| 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).<br>To specify good for session where session spans more than one calendar day, use TimeInForce = Day in conjunction with TradingSessionID.<br>Values should be bi-laterally agreed to between counterparties.<br>Firms may register Trading Session values on the FIX website (presently a document maintained within "ECN and Exchanges" working group section). | **Value** / **Description**<br>1 — Day<br>**OMX Comment:** Not in FIX Standard. OMX requests addition<br>2 — Half Day<br>**OMX Comment:** Not in FIX Standard. OMX requests addition<br>3 — Morning<br>**OMX Comment:** Not in FIX Standard. OMX requests addition<br>4 — Afternoon<br>**OMX Comment:** Not in FIX Standard. OMX requests addition<br>5 — Evening<br>**OMX Comment:** Not in FIX Standard. OMX requests addition<br>6 — After hours<br>**OMX Comment:** Not in FIX Standard. OMX requests addition |
| 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. | **Value** / **Description**<br>1 — Pre-Trading<br>**OMX Comment:** Not in FIX Standard. OMX requests addition<br>2 — Opening or Opening Auction<br>**OMX Comment:** Not in FIX Standard. OMX requests addition<br>3 — (Continuous) Trading |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | **Value** / **Description**<br>**OMX Comment:** Not in FIX Standard. OMX requests addition<br>4 — Closing or Closing Auction<br>**OMX Comment:** Not in FIX Standard. OMX requests addition<br>5 — Post-Trading<br>**OMX Comment:** Not in FIX Standard. OMX requests addition<br>6 — Intraday Auction<br>**OMX Comment:** Not in FIX Standard. OMX requests addition<br>7 — Quiescent<br>**OMX Comment:** Not in FIX Standard. OMX requests addition |
| TradSesCloseTime | 344 | UTCTimestamp | | Closing time of the trading session | |
| TradSesEndTime | 345 | UTCTimestamp | | End time of the trading session | |
| TradSesEvent | 20089 | int | | Identifies an event related to a Trading Session. An event occurs and is gone, it is not a state that applies for a period of time.<br>**OMX Comment:** Not in FIX. The EEWG requires an extension | **Value** / **Description**<br>1 — Trading Resumes (after Halt)<br>2 — Change of Trading Session<br>3 — Change of Trading Subsession<br>4 — Change of Trading Status<br>101 — Initializing (transition to specified trading phase)<br>102 — Completed (transition to specified trading phase) |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| TradSes-Mode | 339 | int | | Trading Session Mode | **Value** / **Description**<br>1 Testing<br>3 Production |
| Trad-SesOpen-Time | 342 | UTC-Times-tamp | | Time of the opening of the trading session | |
| TradSesPre-CloseTime | 343 | UTC-Times-tamp | | Time of the pre-closed of the trading session | |
| TradSesRe-qID | 335 | String | | Unique ID of a Trading Session Status message. | |
| TradSesStart-Time | 341 | UTC-Times-tamp | | Starting time of the trading session | |
| TradSesSta-tus | 340 | int | | State of the trading session. | **Value** / **Description**<br>1 Halted<br>2 Open<br>3 Closed |
| TradSesSta-tusRejRea-son | 567 | int | | Indicates the reason a Trading Session Status Request was rejected. | **Value** / **Description**<br>1 Unknown or invalid TradingSessionID<br>99 Other |
| TradSesUp-dateAction | 20225 | char | | Specifies the action taken for the specified trading sessions.<br>**OMX Comment:** Not in FIX. The EEWG requires an extension | |
| Transact-Time | 60 | UTC-Times-tamp | | Time of execution/order creation (expressed in UTC (Universal Time Coordinated, also known as "GMT") | |
| TransBkd-Time | 483 | UTC-Times-tamp | | For CIV A date and time stamp to indicate the time a CIV order was booked by the fund manager.<br>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. | |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| TrdMatchID | 880 | String | | Identifier assigned to a trade by a matching system.<br>**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. | |
| TrdRepIndicator | 20245 | Boolean | | Specifies whether the trade should be reported (or not) to parties of the provided TrdRepPartyRole (20088). Used to override standart reporting behavior by the receiver of the trade report and thereby complements the PublishTrdIndicator (852).<br>**OMX Comment:** Not in FIX. OMX requires an extension | **Value** / **Description**<br>0 — Do Not Report Trade<br>1 — Report Trade |
| TrdRepPartyRole | 20088 | int | | Identifies the type of party for trade reporting. Same values as PartyRole (452).<br>**OMX Comment:** Not in FIX. OMX requires an extension | |
| TrdRptStatus | 939 | int | | Trade Report Status | **Value** / **Description**<br>0 — Accepted<br>1 — Rejected |
| TrdSubType | 829 | int | | Further qualification to the trade type | **Value** / **Description**<br>97 — OTC Quote<br>**OMX Comment:** Not in FIX Standard. OMX requests addition<br>98 — On-Hours<br>**OMX Comment:** Not in FIX Standard. OMX requests addition<br>99 — Off-hours<br>**OMX Comment:** Not in FIX Standard. OMX requests addition |
| TrdType | 828 | int | | Type of Trade: | **Value** / **Description**<br>0 — Regular Trade<br>4 — Late Trade |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | **Value** / **Description**<br>6 — Weighted Average Price Trade<br>38 — Block trade (same as large trade)<br>47 — Financing transaction (includes repo and stock lending)<br>48 — Non-Standard Settlment<br>**OMX Comment:** FIX 5.0 SP1<br>49 — Derivative Related Transaction<br>**OMX Comment:** FIX 5.0 SP1<br>50 — Portfolio Trade<br>**OMX Comment:** FIX 5.0 SP1<br>51 — Volume Weighted Average Trade<br>**OMX Comment:** FIX 5.0 SP1<br>52 — Exchange Granted Trade<br>**OMX Comment:** FIX 5.0 SP1<br>53 — Repurchase Agreement<br>**OMX Comment:** FIX 5.0 SP1<br>54 — OTC<br>**OMX Comment:** FIX 5.0 SP1 |
| UnderlyingCFICode | 463 | String | | Underlying security's CFICode.<br>Valid values: see CFICode (461) field | |
| UnderlyingPutOrCall | 315 | int | | Underlying security's PutOrCall. See PutOrCall field for description<br>**OMX Comment:** The GDC requests re-introduction | |
| UnderlyingSecurityAltID | 458 | String | | Alternate Security identifier value for this underlying security of UnderlyingSecurityAltIDSource (459) type (e.g. | |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | CUSIP, SEDOL, ISIN, etc). Requires UnderlyingSecurityAltIDSource. | |
| UnderlyingSecurityAltID-Source | 459 | String | | Identifies class or source of the UnderlyingSecurityAltID (458) value. Required if UnderlyingSecurityAltID is specified.<br>Valid values:<br>Same valid values as the SecurityIDSource (22) field | |
| UnderlyingSecurityDesc | 307 | String | | Underlying security's SecurityDesc.<br>See SecurityDesc (07) field for description | |
| UnderlyingSecurityID | 309 | String | | Underlying security's SecurityID.<br>See SecurityID (48) field for description | |
| UnderlyingSecurityID-Source | 305 | String | | Underlying security's SecurityIDSource.<br>Valid values: see SecurityID-Source (22) field | |
| UnderlyingSymbol | 311 | String | | Underlying security's Symbol.<br>See Symbol (55) field for description | |
| UnderlyingSymbolSfx | 312 | String | | Underlying security's SymbolSfx.<br>See SymbolSfx (65) field for description | |
| UnitofMeasure | 996 | String | | Physical unit of measure for Derivative products.<br>NOTE: Additional values may be used by mutual agreement of the counterparties<br>(http://www.unc.edu/~rowlett/units/index.html is a good source for units) | **Value** / **Description**<br>Bbl – Barrels<br>Bcf – Billion cubic feet<br>Bu – Bushels<br>lbs – pounds<br>Gal – Gallons<br>MMbbl – Million Barrels<br>MMBtu – One Million BTU<br>MWh – Megawatt hours<br>oz_tr – Troy Ounces<br>t – Metric Tons (aka Tonne)<br>tn – Tons (US)<br>USD – US Dollars<br>BTU – BTU |

| FieldName | Tag | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| UnitofMeasureQty | 1147 | Qty | | Used to indicate the size of the underlying commodity on which the contract is based, (e.g., 2500 lbs of lean cattle, 1000 barrels of crude oil, 1000 bushels of corn, etc.) | |
| Urgency | 61 | char | | Urgency flag | **Value** / **Description**<br>0 Normal<br>1 Flash<br>2 Background |
| URLLink | 149 | String | | A URI (Uniform Resource Identifier) or URL (Uniform Resource Locator) link to additional information (i.e. http://www.XYZ.com/research.html)<br>See "Appendix 6-B FIX Fields Based Upon Other Standards" | |
| ValidUntilTime | 62 | UTCTimestamp | | Indicates expiration time of indication message (always expressed in UTC (Universal Time Coordinated, also known as "GMT") | |
| ValueTypeCode | 20099 | String | | For futures, indicates type of valuation method applied<br>**OMX Comment:** Not in FIX. The GDC requires an extension | **Value** / **Description**<br>EQTY Premium style<br>FUT Futures style mark-to-market<br>FUTDA Futures style with an attached cash adjustment |
| Volatility | 20228 | float | | Annualized volatility for option model calculations<br>**OMX Comment:** Not in FIX. OMX requires an extension | |
| 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**<br>N Order has been accepted but not yet in a working state<br>Y Order is currently being worked |
| Yield | 236 | Percentage | | Yield percentage.<br>(Note tag # was reserved in FIX 4.1, added in FIX 4.3) | |

## 20.2 Fields per Tag Number

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| 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)<br>Valid values:<br>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.<br>(Prior to FIX 4.2 this field was of type int) | |
| 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 in- | |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | formation 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.<br>(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**<br>G — All or none - AON<br>**OMX Comment:** Currently not supported - use MinQty (110) = QrderQty (38)<br>S — Suspend<br>**OMX Comment:** Used to report an order as suspended in Execution Reports. Also used to send in an order as suspended.<br>i — Imbalance Only<br>v — Release (from suspension)<br>**OMX Comment:** Not in FIX Standard. OMX requests addition<br>w — Execute as delta neutral using volatility provided<br>**OMX Comment:** Not in FIX Standard. OMX requests addition |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | **Value** / **Description**<br>x — Execute as duration neutral<br>**OMX Comment:** Not in FIX Standard. OMX requests addition<br>y — Execute as FX neutral<br>**OMX Comment:** Not in FIX Standard. OMX requests addition<br>z — Suspend on Connection Loss<br>**OMX Comment:** Not in FIX Standard. OMX requests addition |
| 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**<br>99 — Marketplace-assigned Identifier |
| 23 | IOIID | String | | Unique identifier of IOI message. (Prior to FIX 4.1 this field was of type int) | |
| 26 | IOIRefID | String | | Reference identifier used with CANCEL and REPLACE, transaction types. (Prior to FIX 4.1 this field was of type int) | |
| 27 | IOIQty | String | | Quantity (e.g. number of shares) in numeric form or relative size. | **Value** / **Description**<br>0 — 1000000000<br>S — Small<br>M — Medium<br>L — Large<br>U — Undisclosed Quantity |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| 28 | IOITransType | char | | Identifies IOI message transaction type | Value: N — New; C — Cancel; R — Replace |
| 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) | |
| 33 | NoLinesOfText | NumInGroup | | Identifies number of lines of text body | |
| 34 | MsgSeqNum | SeqNum | | Integer message sequence number. | |
| 35 | MsgType | String | | Defines message type ALWAYS THIRD FIELD IN MESSAGE. (Always unencrypted)<br>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.<br>*** Note the use of lower case letters *** | Value: 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 |

| Tag | FieldName | Type | OMXLen | Desc | Valid values | |
|---|---|---|---|---|---|---|
| | | | | | **Value** | **Description** |
| | | | | | y | Derivative Securiy List Update Report |
| | | | | | | **OMX Comment:** Not in FIX. The GDC requests addition |
| | | | | | AE | Trade Capture Report |
| | | | | | AR | Trade Capture Report Ack |
| | | | | | BJ | Trading Session List |
| | | | | | UJ | Participant List Update Report |
| | | | | | | **OMX Comment:** Not in FIX Standard. OMX requests addition |
| | | | | | U3 | SyncHeartbeat |
| | | | | | U0 | Market Data Statistics |
| | | | | | | **OMX Comment:** Not in FIX Standard. User-defined message |
| | | | | | U1 | Market Segment |
| | | | | | | **OMX Comment:** Not in FIX Standard. OMX requests addition |
| | | | | | U2 | One Sided Auction Request |
| | | | | | | **OMX Comment:** Not in FIX Standard. User-defined message |
| | | | | | U3 | One Sided Auction Request Ack |
| | | | | | | **OMX Comment:** Not in FIX Standard. User-defined message |
| | | | | | U4 | One Sided Auction Result |
| | | | | | | **OMX Comment:** Not in FIX Standard. User-defined message |
| | | | | | U5 | NewsDataRequest |
| | | | | | | **OMX Comment:** Not in FIX Standard. User-defined message |
| | | | | | U6 | NewsDataRequestReject |

| Tag | FieldName | Type | OMXLen | Desc | Valid values | |
|---|---|---|---|---|---|---|
| | | | | | **Value** | **Description** |
| | | | | | | **OMX Comment:** Not in FIX Standard. User-defined message |
| | | | | | U7 | NewsPublicationRequest |
| | | | | | | **OMX Comment:** Not in FIX Standard. User-defined message |
| | | | | | U8 | NewsPublicationRequestReject |
| | | | | | | **OMX Comment:** Not in FIX Standard. User-defined message |
| | | | | | U9 | Participant List |
| | | | | | UA | Market Segment Request |
| | | | | | | **OMX Comment:** Not in FIX Standard. OMX requests addition |
| | | | | | UB | One Sided Auction Info |
| | | | | | | **OMX Comment:** Not in FIX Standard. User-defined message |
| | | | | | UC | One Sided Auction Info Request |
| | | | | | | **OMX Comment:** Not in FIX Standard. User-defined message |
| | | | | | UD | Market Segment Update Report |
| | | | | | | **OMX Comment:** Not in FIX Standard. OMX requests addition |
| | | | | | UE | Trading Session List Update Report |
| | | | | | | **OMX Comment:** Not in FIX Standard. OMX requests addition |
| | | | | | UF | Order Mass Suspend Or Release Request |
| | | | | | | **OMX Comment:** Not in FIX Standard. OMX requests addition |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | **Value** / **Description**<br>UG — Order Mass Suspend Or Release Report<br>**OMX Comment:** Not in FIX Standard. OMX requests addition<br>UH — Derivative Security List Update Report<br>**OMX Comment:** Not in FIX Standard. OMX requests addition<br>UI — Participant List Request<br>**OMX Comment:** Not in FIX Standard. OMX requests addition |
| 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.<br>(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** / **Description**<br>0 — New<br>1 — Partially filled<br>2 — Filled<br>4 — Canceled<br>8 — Rejected<br>9 — Suspended<br>C — Expired |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| 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**<br>1 Market<br>2 Limit<br>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. | |
| 42 | OrigTime | UTCTimestamp | | Time of message origination (always expressed in UTC (Universal Time Coordinated, also known as "GMT")) | |
| 43 | PossDupFlag | Boolean | | Indicates possible retransmission of message with this sequence number | **Value** / **Description**<br>N Original transmission<br>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**<br>1 Buy<br>2 Sell<br>B "As Defined" (for use with multileg instruments)<br>**OMX Comment:** Valid for multileg Orders only |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| 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**<br>0 — Day (or session)<br>1 — Good Till Cancel (GTC)<br>2 — At the Opening (OPG)<br>3 — Immediate Or Cancel (IOC)<br>4 — Fill Or Kill (FOK)<br>5 — Good Till Crossing (GTX)<br>6 — Good Till Date (GTD)<br>7 — At the Close<br>y — Good Through Crossing<br>**OMX Comment:** Not in FIX Standard. OMX requests addition<br>z — At Crossing<br>**OMX Comment:** Not in FIX Standard. OMX requests addition |
| 60 | TransactTime | UTCTimestamp | | Time of execution/order creation (expressed in UTC (Universal Time Coordinated, also known as "GMT") | |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| 61 | Urgency | char | | Urgency flag | Value: 0 — Description: Normal<br>Value: 1 — Description: Flash<br>Value: 2 — Description: Background |
| 62 | ValidUntil-Time | UTC-Times-tamp | | Indicates expiration time of indication message (always expressed in UTC (Universal Time Coordinated, also known as "GMT") | |
| 64 | SettlDate | Lo-calMkt-Date | | Specific date of trade settle-ment (SettlementDate) in YYYYMMDD format.<br>If present, this field overrides SettlType (63). This field is required if the value of Settl-Type (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)<br>(expressed in local time at place of settlement) | |
| 65 | SymbolSfx | String | | Additional information about the security (e.g. preferred, warrants, etc.). Note also see SecurityType (167).<br>As defined in the NYSE Stock and bond Symbol Directory and in the AMEX Fitch Direc-tory. | Value: WI — Description: "When Issued" for a security to be reis-sued under an old CUSIP or ISIN |
| 66 | ListID | String | | Unique identifier for list as assigned by institution, used to associate multiple individu-al orders. Uniqueness must be guaranteed within a single trading day. Firms which generate multi-day orders should consider embedding a date within the ListID field to assure uniqueness across days. | |
| 67 | ListSeqNo | int | | Sequence of individual order within list (i.e. ListSeqNo of TotNoOrders (68), 2 of 25, 3 of 25, . . . ) | |
| 70 | AllocID | String | | Unique identifier for allocation message.<br>(Prior to FIX 4.1 this field was of type int) | |
| 73 | NoOrders | NumIn-Group | | Indicates number of orders to be combined for average pricing and allocation. | |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| 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). | |
| 77 | PositionEffect | char | | Indicates whether the resulting position after a trade should be an opening position or closing position. Used for omnibus accounting - where accounts are held on a gross basis instead of being netted together. | **Value** / **Description**<br>C – Close<br>O – Open<br>N – Close but Notify on Open<br>**OMX Comment:** Not in FIX Standard. OMX requests addition<br>D – Default<br>**OMX Comment:** Not in FIX Standard. OMX requests addition |
| 78 | NoAllocs | NumInGroup | | Number of repeating AllocAccount (79)/AllocPrice (366) entries. | |
| 79 | AllocAccount | String | | Sub-account mnemonic | |
| 80 | AllocQty | Qty | | Quantity to be allocated to specific sub-account (Prior to FIX 4.2 this field was of type int) | |
| 82 | NoRpts | int | | Total number of reports within series. | |
| 83 | RptSeq | int | | Sequence number of message within report series. Used to carry reporting sequence number of the fill as represented on the Trade Report Side. | |
| 98 | EncryptMethod | int | | Method of encryption. | **Value** / **Description**<br>0 – None / Other |
| 102 | CxlRejReason | int | | Code to identify reason for cancel rejection. | **Value** / **Description**<br>0 – Too late to cancel<br>1 – Unknown order<br>2 – Broker / Exchange Option<br>6 – Duplicate ClOrdID (11) received |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | **Value** / **Description**<br>18 — Invalid price increment<br>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 pre-allocation information errors. | **Value** / **Description**<br>0 — Broker / Exchange option<br>1 — Unknown symbol<br>2 — Exchange closed<br>3 — Order exceeds limit<br>4 — Too late to enter<br>5 — Unknown order<br>6 — Duplicate Order (e.g. dupe ClOrdID)<br>7 — Duplicate of a verbally communicated order<br>8 — Stale order<br>9 — Trade along required<br>10 — Invalid Investor ID<br>11 — Unsupported order characteristic<br>12 — Surveillence Option<br>13 — Incorrect quantity<br>14 — Incorrect allocated quantity<br>15 — Unknown account(s)<br>18 — Invalid price increment<br>99 — Other |
| 106 | Issuer | String | | Name of security issuer (e.g. International Business Machines, GNMA).<br>see also Volume 7: "PRODUCT: FIXED INCOME - Euro Issuer Values" | |
| 107 | SecurityDesc | String | | Security description. | |
| 108 | HeartBtInt | int | | Heartbeat interval (seconds) | |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| 110 | MinQty | Qty | | Minimum quantity of an order to be executed.<br>(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 | |
| 117 | QuoteID | String | | Unique identifier for quote<br>**OMX Comment:** Unique identifier for a quote message | |
| 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**<br>N – Sequence Reset, Ignore Msg Seq Num (N/A For FIXML - Not Used)<br>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")<br>The meaning of expiration is specific to the context where the field is used.<br>For orders, this is the expiration time of a Good Til Date TimeInForce.<br>For Quotes - this is the expiration of the quote. | |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | Expiration time is provided across the quote message dialog to control the length of time of the overall quoting process.<br>For collateral requests, this is the time by which collateral must be assigned.<br>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 (i.e. trader) if the message is delivered by a third party | |
| 131 | QuoteReqID | String | | Unique identifier for quote request | |
| 132 | BidPx | Price | | Bid price/rate | |
| 133 | OfferPx | Price | | Offer price/rate | |
| 134 | BidSize | Qty | | Quantity of bid<br>(Prior to FIX 4.2 this field was of type int) | |
| 135 | OfferSize | Qty | | Quantity of offer<br>(Prior to FIX 4.2 this field was of type int) | |
| 141 | ResetSeqNumFlag | Boolean | | Indicates that the both sides of the FIX session should reset sequence numbers. | **Value** / **Description**<br>N / No<br>Y / Yes, reset sequence numbers |
| 146 | NoRelatedSym | NumInGroup | | Specifies the number of repeating symbols specified. | |
| 148 | Headline | String | | The headline of a News message | |
| 149 | URLLink | String | | A URI (Uniform Resource Identifier) or URL (Uniform Resource Locator) link to additional information (i.e. http://www.XYZ.com/research.html) | |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | See "Appendix 6-B FIX Fields Based Upon Other Standards" | |
| 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**<br>0 New<br>4 Canceled<br>5 Replaced<br>8 Rejected<br>9 Suspended<br>**OMX Comment:** Currently not used<br>C Expired<br>D Restated (Execution Report sent unsolicited by sellside, with ExecRestatementReason (378) set)<br>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).<br>(Prior to FIX 4.2 this field was of type int) | |
| 167 | SecurityType | String | | Indicates type of security. See also the Product (460) and CFICode (461) fields. It is recommended that CFICode be used instead of SecurityType for non-Fixed Income instruments.<br>Example values (grouped by Product field value) (Note: additional values may be used by mutual agreement of the counterparties):<br>* Identify the Issuer in the "Issuer" field(106)<br>*** REPLACED values - See "Replaced Features and Supported Approach" ***<br>NOTE: Additional values may be used by mutual agreement of the counterparties) | **Value** / **Description**<br>ABS Asset-backed Securities<br>AN Other Anticipation Notes (BAN, GAN, etc.)<br>BA Bankers Acceptance<br>BRADY Brady Bond<br>CORP Corporate Bond<br>CS Common Stock<br>EUSUPRA Euro Supranational Coupons *<br>FOR Foreign Exchange Contract<br>MF Mutual Fund |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | **OMX Comment:** Valid values are not yet defined! | |

| Value | Description |
|---|---|
| REPO | Repurchase |
| TERM | Term Loan |
| BN | Bank Notes |
| CMBS | Corp. Mortgage-backed Securities |
| COFO | Certificate Of Obligation |
| CPP | Corporate Private Placement |
| EUSOV | Euro Sovereigns * |
| FAC | Federal Agency Coupon |
| FOR-WARD | Forward |
| FUT | Future |
| MLEG | Multileg Instrument |
| PS | Preferred Stock |
| RVLV | Revolver Loan |
| BOX | Bill Of Exchanges |
| BUY-SELL | Buy Sellback |
| CB | Convertible Bond |
| CMO | Collateralized Mortgage Obligation |
| COFP | Certificate Of Participation |
| FADN | Federal Agency Discount Note |
| NONE | No Security Type |
| OPT | Option |
| RVLVTRM | Revolver/Term Loan |
| TBOND | US Treasury Bond |
| BRIDGE | Bridge Loan |
| CD | Certificate Of Deposit |
| DUAL | Dual Currency |
| GO | General Obligation Bonds |
| IET | IOETTE Mortgage |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | **Value** / **Description** |
| | | | | | PEF — Private Export Funding * |
| | | | | | SECLOAN — Securities Loan |
| | | | | | TINT — Interest Strip From Any Bond Or Note |
| | | | | | UST — US Treasury Note (Deprecated Value Use TNOTE) |
| | | | | | OOF — Options on Futures |
| | | | | | CL — Call Loans |
| | | | | | EUCORP — Euro Corporate Bond |
| | | | | | LOFC — Letter Of Credit |
| | | | | | MBS — Mortgage-backed Securities |
| | | | | | MT — Mandatory Tender |
| | | | | | SECPLEDGE — Securities Pledge |
| | | | | | SUPRA — USD Supranational Coupons * |
| | | | | | TIPS — Treasury Inflation Protected Securities |
| | | | | | USTB — US Treasury Bill (Deprecated Value Use TBILL) |
| | | | | | OOP — Options on Physical |
| | | | | | CP — Commercial Paper |
| | | | | | MIO — Mortgage Interest Only |
| | | | | | RAN — Revenue Anticipation Note |
| | | | | | SWING — Swing Line Facility |
| | | | | | TCAL — Principal Strip Of A Callable Bond Or Note |
| | | | | | WLD — Wildcard Entry (was "?" in 4.4, used on Security Definition Request message) |
| | | | | | XLINKD — Indexed Linked |
| | | | | | DINP — Debtor In Possession |
| | | | | | DN — Deposit Notes |

| Tag | FieldName | Type | OMXLen | Desc | Valid values | |
|---|---|---|---|---|---|---|
| | | | | | **Value** | **Description** |
| | | | | | MPO | Mortgage Principal Only |
| | | | | | REV | Revenue Bonds |
| | | | | | STRUCT | Structured Notes |
| | | | | | TPRN | Principal Strip From A Non-Callable Bond Or Note |
| | | | | | CASH | Cash |
| | | | | | DEFLT-ED | Defaulted |
| | | | | | EUCD | Euro Certificate Of Deposit |
| | | | | | MPP | Mortgage Private Placement |
| | | | | | SPCLA | Special Assessment |
| | | | | | TNOTE | US Treasury Note |
| | | | | | YANK | Yankee Corporate Bond |
| | | | | | EUCP | Euro Commercial Pa-per |
| | | | | | MPT | Miscellaneous Pass-through |
| | | | | | SPCLO | Special Obligation |
| | | | | | TBILL | US Treasury Bill |
| | | | | | WITH-DRN | Withdrawn |
| | | | | | LQN | Liquidity Note |
| | | | | | PFAND | Pfandbriefe * |
| | | | | | RE-PLACD | Replaced |
| | | | | | SPCLT | Special Tax |
| | | | | | MA-TURED | Matured |
| | | | | | MTN | Medium Term Notes |
| | | | | | TAN | Tax Anticipation Note |
| | | | | | TBA | To Be Announced |
| | | | | | AMEND-ED | Amended & Restated |
| | | | | | ONITE | Overnight |
| | | | | | TAXA | Tax Allocation |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | **Value** / **Description**<br>PN / Promissory Note<br>RE-TIRED / Retired<br>TECP / Tax Exempt Commercial Paper<br>PZFJ / Plazos Fijos<br>TRAN / Tax Revenue Anticipation Note<br>STN / Short Term Loan Note<br>VRDN / Variable Rate Demand Note<br>TD / Time Deposit<br>WAR / Warrant<br>XCN / Extended Comm Note<br>YCD / Yankee Certificate Of Deposit |
| 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. | |
| 200 | MaturityMonthYear | month-year | | Can be used with standardized derivatives vs. the MaturityDate (54) field. Month and Year of the maturity (used for standardized futures and options).<br>Format:<br>YYYYMM (i.e. 99903)<br>YYYYMMDD (20030323)<br>YYYYMMwN (200303w) for week<br>A specific date or can be appended to the MaturityMonthYear. For instance, if multiple standard products exist that mature in the same Year and Month, but actually mature at a different time, a value can be appended, such as "w" or "w2" to indicate week as opposed to week 2 expiration. Likewise, the date (0-3) can be appended to indicate a specific expiration (maturity date). | |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| 201 | PutOrCall | int | | Indicates whether an Option is for a put or call<br>**OMX Comment:** The GDC requests re-introduction | **Value** / **Description**<br>0 Put<br>1 Call |
| 202 | StrikePrice | Price | | Strike Price for an Option. | |
| 207 | SecurityExchange | Exchange | | Market used to help identify a security.<br>Valid values:<br>See "Appendix 6-C" | |
| 223 | CouponRate | Percentage | | The rate of interest that, when multiplied by the principal, par value, or face value of a bond, provides the currency amount of the periodic interest payment. The coupon is always cited, along with maturity, in any quotation of a bond's price. | |
| 224 | CouponPaymentDate | LocalMktDate | | Date interest is to be paid. Used in identifying Corporate Bond issues.<br>(Note tag # was reserved in FIX 4.1, added in FIX 4.3)<br>(prior to FIX 4.4 field was of type UTCDate) | |
| 225 | IssueDate | LocalMktDate | | The date on which a bond or stock offering is issued. It may or may not be the same as the effective date ("Dated Date") or the date on which interest begins to accrue ("Interest Accrual Date")<br>(Note tag # was reserved in FIX 4.1, added in FIX 4.3)<br>(prior to FIX 4.4 field was of type UTCDate) | |
| 229 | TradeOriginationDate | LocalMktDate | | Used with Fixed Income for Muncipal New Issue Market. Agreement in principal between counter-parties prior to actual trade date.<br>(Note tag # was reserved in FIX 4.1, added in FIX 4.3)<br>(prior to FIX 4.4 field was of type UTCDate) | |
| 231 | ContractMultiplier | float | | Specifies the ratio or multiply factor to convert from "nominal" units (e.g. contracts) to total units (e.g. shares) (e.g. 1.0, 100, 1000, etc). Applicable For Fixed Income, Convertible Bonds, Derivatives, etc. | |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | In general quantities for all calsses should be expressed in the basic unit of the instrument, e.g. shares for equities, norminal or par amount for bonds, currency for foreign exchange. When quantity is expressed in contracts, e.g. financing transactions and bond trade reporting, ContractMutliplier should contain the number of units in one contract and can be omitted if the multiplier is the default amount for the instrument, i.e. 1,000 par of bonds, 1,000,000 par for financing transactions. | |
| 236 | Yield | Percentage | | Yield percentage. (Note tag # was reserved in FIX 4.1, added in FIX 4.3) | |
| 262 | MDReqID | String | | Unique identifier for Market Data Request | |
| 264 | MarketDepth | int | | Depth of market for Book Snapshot | **Value** / **Description**<br>0 — Full Book<br>**OMX Comment:** FIX 5.0 SP1<br>1 — Top of Book<br>**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**<br>0 — Bid<br>1 — Offer<br>2 — Trade<br>3 — Index Value<br>A — Imbalance |
| 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) | |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| 273 | MDEntry-Time | UTC-TimeOn-ly | | Time of Market Data Entry. | |
| 274 | TickDirection | char | | Direction of the "tick". | **Value** / **Description**<br>0 / Plus Tick<br>1 / Zero-Plus Tick<br>2 / Minus Tick<br>3 / Zero-Minus Tick |
| 276 | QuoteCondi-tion | Multi-pleString-Value | | Space-delimited list of condi-tions describing a quote. | **Value** / **Description**<br>Z / Order Imbalance<br>3 / Rest of Book VWAP<br>**OMX Comment:** FIX 5.0 SP1<br>4 / Better Prices in Con-ditional Orders<br>**OMX Comment:** FIX 5.0 SP1<br>9 / Median Price<br>**OMX Comment:** Not in FIX Standard. OMX requests addi-tion |
| 277 | TradeCondi-tion | Multi-pleString-Value | | Space-delimited list of condi-tions describing a trade | **Value** / **Description**<br>P / Imbalance More Buy-ers (cannot be used in combination with Q)<br>Q / Imbalance More Sell-ers (cannot be used in combination with P)<br>X / Crossed<br>AV / Outside Spread<br>**OMX Comment:** FIX 5.0 SP1<br>ZX / Marketplace Entered Trade<br>**OMX Comment:** Not in FIX Standard. OMX requests addi-tion<br>ZY / Multileg to Multileg Trade |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | **Value** / **Description**<br>**OMX Comment:** Not in FIX Standard. OMX requests addition<br>ZZ — Cross Asset Multileg Trade<br>**OMX Comment:** Not in FIX Standard. OMX requests addition |
| 278 | MDEntryID | String | | Unique Market Data Entry identifier. | |
| 279 | MDUpdateAction | char | | Type of Market Data update action. | **Value** / **Description**<br>0 — New<br>1 — Change<br>2 — Delete<br>3 — Delete Thru<br>4 — Delete From<br>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 . | |
| 291 | FinancialStatus | MultipleCharValue | | Identifies a firm's or a security's financial status | **Value** / **Description**<br>1 — Bankrupt<br>2 — Pending delisting<br>3 — Restricted |
| 292 | CorporateAction | MultipleCharValue | | Identifies the type of Corporate Action. | **Value** / **Description**<br>A — Ex-Dividend<br>B — Ex-Distribution<br>C — Ex-Rights<br>E — Ex-Interest |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | **Value** / **Description**<br>F – Cash Dividend<br>G – Stock Dividend<br>H – Non-Integer Stock Split<br>I – Reverse Stock Split<br>J – Standard-Integer Stock Split<br>M – Merger Reorganization<br>N – Rights Offering<br>O – Shareholder Meeting<br>P – Spinoff<br>Q – Tender Offer<br>U – CUSIP / Name Change |
| 293 | DefBidSize | Qty | | Default Bid Size. | |
| 294 | DefOfferSize | Qty | | Default Offer Size. | |
| 295 | NoQuoteEntries | NumInGroup | | The number of quote entries for a QuoteSet. | |
| 296 | NoQuoteSets | NumInGroup | | The number of sets of quotes in the message. | |
| 297 | QuoteStatus | int | | Identifies the status of the quote acknowledgement. | **Value** / **Description**<br>5 – Rejected<br>6 – Removed from Market<br>7 – Expired<br>8 – Query<br>9 – Quote Not Found<br>16 – Active<br>**OMX Comment:** FIX 5.0 SP1<br>17 – Canceled<br>**OMX Comment:** FIX 5.0 SP1<br>18 – Unsolicited Quote Replenishment<br>**OMX Comment:** FIX 5.0 SP1 |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| 298 | QuoteCancel-Type | int | | Identifies the type of quote cancel. | **Value** / **Description**<br>1 Cancel for Symbol(s)<br>3 Cancel for Underlying Symbol<br>4 Cancel All Quotes<br>5 Cancel quote specified in QuoteID |
| 299 | QuoteEntryID | String | | Uniquely identifies the quote as part of a QuoteSet.<br>**OMX Comment:** Unique identifier for a quote. The QuoteEntryID stays with the quote as a static identifier even if the quote is updated. | |
| 300 | QuoteRejectReason | int | | Reason Quote was rejected: | **Value** / **Description**<br>5 Unknown Quote<br>6 Duplicate Quote<br>10 Quote Locked - Unable to Update/Cancel<br>**OMX Comment:** FIX 5.0 SP1<br>99 Other |
| 301 | QuoteResponseLevel | int | | Level of Response requested from receiver of quote messages. | **Value** / **Description**<br>0 No Acknowledgement (default)<br>1 Acknowledge only negative or erroneous quotes<br>**OMX Comment:** Default<br>2 Acknowledge each quote messages<br>3 Summary Acknowledgement<br>**OMX Comment:** FIX 5.0 SP1 |
| 302 | QuoteSetID | String | | Unique id for the Quote Set. | |
| 304 | TotNoQuoteEntries | int | | Total number of quotes for the quote set across all messages. Should be the sum of all NoQuoteEntries (295) in each message that has re- | |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | peating quotes that are part of the same quote set.<br>(Prior to FIX 4.4 this field was named TotQuoteEntries) | |
| 305 | UnderlyingSecurityID-Source | String | | Underlying security's SecurityIDSource.<br>Valid values: see SecurityID-Source (22) field | |
| 307 | UnderlyingSecurityDesc | String | | Underlying security's SecurityDesc.<br>See SecurityDesc (07) field for description | |
| 309 | UnderlyingSecurityID | String | | Underlying security's SecurityID.<br>See SecurityID (48) field for description | |
| 311 | Underly-ingSymbol | String | | Underlying security's Symbol.<br>See Symbol (55) field for description | |
| 312 | Underly-ingSymbolSfx | String | | Underlying security's SymbolSfx.<br>See SymbolSfx (65) field for description | |
| 315 | Underlying-PutOrCall | int | | Underlying security's PutOrCall. See PutOrCall field for description<br>**OMX Comment:** The GDC requests re-introduction | |
| 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** / **Description**<br>1 – Accept security proposal as-is<br>2 – Accept security proposal with revisions as indicated in the message<br>5 – Reject security proposal |
| 326 | SecurityTradingStatus | int | | Identifies the trading status applicable to the transaction. | **Value** / **Description**<br>1 – Opening delay<br>2 – Trading halt<br>7 – Market Imbalance Buy<br>8 – Market Imbalance Sell |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | **Value** / **Description**<br>17 Ready to trade (start of session)<br>18 Not available for trading (end of session)<br>21 Pre-open<br>22 Opening Rotation<br>23 Fast Market<br>100 Hidden Auction<br>101 Open Auction<br>102 Issuer Position Modification |
| 327 | HaltReason | char | | Denotes the reason for the Opening Delay or Trading Halt. | **Value** / **Description**<br>D News Dissemination<br>E Order Influx<br>M Additional Information |
| 328 | InViewOfCommon | Boolean | | Indicates whether or not the halt was due to Common Stock trading being halted. | **Value** / **Description**<br>N Halt was not related to a halt of the common stock<br>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**<br>N Halt was not related to a halt of the related security<br>Y Halt was due to related security being halted |
| 332 | HighPx | Price | | Represents an indication of the high end of the price range for a security prior to the open or reopen | |
| 333 | LowPx | Price | | Represents an indication of the low end of the price range for a security prior to the open or reopen | |
| 335 | TradSesReqID | String | | Unique ID of a Trading Session Status message. | |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| 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).<br>To specify good for session where session spans more than one calendar day, use TimeInForce = Day in conjunction with TradingSessionID.<br>Values should be bi-laterally agreed to between counterparties.<br>Firms may register Trading Session values on the FIX website (presently a document maintained within "ECN and Exchanges" working group section). | **Value** / **Description**<br>1 — Day<br>**OMX Comment:** Not in FIX Standard. OMX requests addition<br>2 — Half Day<br>**OMX Comment:** Not in FIX Standard. OMX requests addition<br>3 — Morning<br>**OMX Comment:** Not in FIX Standard. OMX requests addition<br>4 — Afternoon<br>**OMX Comment:** Not in FIX Standard. OMX requests addition<br>5 — Evening<br>**OMX Comment:** Not in FIX Standard. OMX requests addition<br>6 — After hours<br>**OMX Comment:** Not in FIX Standard. OMX requests addition |
| 339 | TradSesMode | int | | Trading Session Mode | **Value** / **Description**<br>1 — Testing<br>3 — Production |
| 340 | TradSesStatus | int | | State of the trading session. | **Value** / **Description**<br>1 — Halted<br>2 — Open<br>3 — Closed |
| 341 | TradSesStartTime | UTCTimestamp | | Starting time of the trading session | |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| 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. | |
| 348 | EncodedIssuerLen | Length | | Byte length of encoded (non-ASCII characters) EncodedIssuer (349) field. | |
| 349 | EncodedIssuer | data | | Encoded (non-ASCII characters) representation of the Issuer field in the encoded format specified via the MessageEncoding (347) field. If used, the ASCII (English) representation should also be specified in the Issuer field. | |
| 350 | EncodedSecurityDescLen | Length | | Byte length of encoded (non-ASCII characters) EncodedSecurityDesc (351) field. | |
| 351 | EncodedSecurityDesc | data | | Encoded (non-ASCII characters) representation of the SecurityDesc (107) field in the encoded format specified via the MessageEncoding (347) field. If used, the ASCII (English) representation should also be specified in the SecurityDesc field. | |
| 354 | EncodedTextLen | Length | | Byte length of encoded (non-ASCII characters) EncodedText (355) field. | |
| 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. | |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| 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. | |
| 368 | QuoteEntryRejectReason | int | | Reason Quote Entry was rejected: | **Value** / **Description**<br>1 Unknown symbol (security)<br>3 Quote exceeds limit<br>5 Unknown quote<br>6 Duplicate quote<br>7 Invalid bid/ask spread<br>8 Invalid price<br>9 Not authorized to quote security<br>99 Other |
| 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**<br>0 Invalid Tag Number<br>1 Required Tag Missing<br>2 Tag not defined for this message type<br>3 Undefined tag<br>4 Tag specified without a value<br>5 Value is incorrect (out of range) for this tag<br>6 Incorrect data format for value<br>7 Decryption problem<br>8 Signature problem<br>9 CompID problem<br>10 SendingTime Accuracy Problem<br>11 Invalid MsgType |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | **Value** / **Description**<br>12 — XML Validation Error<br>13 — Tag appears more than once<br>14 — Tag specified out of required order<br>15 — Repeating group fields out of order<br>16 — Incorrect NumInGroup count for repeating group<br>17 — Non "Data" value includes field delimiter (<SOH> character) |
| 377 | SolicitedFlag | Boolean | | Indicates whether or not the order was solicited. | **Value** / **Description**<br>N — Was not solicited<br>Y — Was solicited |
| 378 | ExecRestatementReason | int | | Code to identify reason for an ExecutionRpt message sent with ExecType=Restated or used when communicating an unsolicited cancel. | **Value** / **Description**<br>0 — GT corporate action<br>1 — GT renewal / restatement (no corporate action)<br>2 — Verbal change<br>3 — Repricing of order<br>6 — Cancel on Trading Halt<br>7 — Cancel on System Failure<br>9 — Canceled, not best<br>11 — Peg Refresh<br>99 — Other |
| 379 | BusinessRejectRefID | String | | The value of the business-level "ID" field on the message being referenced. | |
| 380 | BusinessRejectReason | int | | Code to identify reason for a Business Message Reject message. | **Value** / **Description**<br>0 — Other<br>1 — Unknown ID<br>2 — Unknown Security |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | **Value** / **Description**<br>3 – Unknown Message Type<br>4 – Application not available<br>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**<br>7 – Average Price Guarantee |
| 393 | TotNoRelatedSym | int | | Total number of securities. (Prior to FIX 4.4 this field was named TotalNumSecurities) | |
| 394 | BidType | int | | Code to identify the type of Bid Request. | **Value** / **Description**<br>3 – No bidding process |
| 423 | PriceType | int | | Code to represent the price type.<br>(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".<br>See Volume : "Glossary" for further value definitions) | **Value** / **Description**<br>1 – Percentage (i.e. percent of par) (often called "dollar price" for fixed income)<br>**OMX Comment:** Only relevant for Fixed Income trading<br>2 – Per unit (i.e. per share or contract)<br>**OMX Comment:** Default value, should be specified (if applicable) for Fixed Income trading<br>3 – Fixed amount (absolute value)<br>**OMX Comment:** Only allowed for IOI's<br>4 – Discount - percentage points below par<br>**OMX Comment:** Only allowed for IOI's<br>5 – Premium - percentage points over par<br>**OMX Comment:** Only allowed for IOI's |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | **Value** / **Description**<br>6 — Spread (basis points spread)<br>**OMX Comment:** Only allowed for IOI's<br>7 — TED Price<br>**OMX Comment:** Only allowed for IOI's<br>8 — TED Yield<br>**OMX Comment:** Only allowed for IOI's<br>9 — Yield<br>**OMX Comment:** Only relevant for Fixed Income trading |
| 429 | ListStatusType | int | | Code to represent the status type. | **Value** / **Description**<br>2 — Response |
| 431 | ListOrderStatus | int | | Code to represent the status of a list order. | **Value** / **Description**<br>3 — Executing<br>7 — Reject |
| 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**<br>1 — Order cancel request<br>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.<br>See "Appendix 6-G – Use of <Parties> Component Block" | **Value** / **Description**<br>C — Generally accepted market participant identifier (e.g. NASD mnemonic)<br>D — Proprietary / Custom code<br>H — CSD participant/member code (e.g.. Euro- |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | **Value** / **Description**<br>clear, DTC, CREST or Kassenverein number)<br>**OMX Comment:** Includes Clearing house participant/member code |
| 448 | PartyID | String | | Party identifier/code. See PartyIDSource (447) and PartyRole (452).<br>See "Appendix 6-G – Use of <Parties> Component Block" | |
| 451 | NetChgPrevDay | PriceOffset | | Net change from previous day's closing price vs. last traded price. | |
| 452 | PartyRole | int | | Identifies the type or role of the PartyID (448) specified.<br>See "Appendix 6-G – Use of <Parties> Component Block"<br>(see Volume : "Glossary" for value definitions) | **Value** / **Description**<br>1 — Executing Firm (formerly FIX 4.2 ExecBroker)<br>**OMX Comment:** The role of the firm legally responsible for a business transaction sent to the marketplace<br>3 — Client ID (formerly FIX 4.2 ClientID)<br>**OMX Comment:** Used when specifying an allocation to a defined client<br>7 — Entering Firm<br>**OMX Comment:** The role of a firm performing data entry on behalf of the executing firm (includes marketplace operations)<br>10 — Settlement Location (formerly FIX 4.2 SettlLocation)<br>**OMX Comment:** The role of a CSD (or similar) in the context of trades<br>12 — Executing Trader (associated with Executing Firm - actually executes) |

| Tag | FieldName | Type | OMXLen | Desc | Valid values | |
|---|---|---|---|---|---|---|
| | | | | | **Value** | **Description** |
| | | | | | | **OMX Comment:** The role of the individual responsible for a business transaction sent to the market-place |
| | | | | | 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 |
| | | | | | | **OMX Comment:** The role of a Clearing House / CCP in the context of trades |
| | | | | | 22 | Exchange |
| | | | | | | **OMX Comment:** Used in Trade Capture Reports when a user request that a trade is not made public over market data. |
| | | | | | 34 | Regulatory body |
| | | | | | | **OMX Comment:** Used in Trade Capture Reports when a user requests that a trade is not reported to the regulator. Only allowed when the user reports trades directly to the regulator |
| | | | | | 36 | Entering trader |
| | | | | | | **OMX Comment:** The role of an individual performing data entry on behalf of the executing "trader" (includes marektplace officials) |
| | | | | | 37 | Contra trader |
| | | | | | | **OMX Comment:** Used in Trade Capture Reports for trade confirmations divulding a single side only. |

| Tag | FieldName | Type | OMXLen | Desc | Valid values | |
|---|---|---|---|---|---|---|
| | | | | | **Value** | **Description** |
| | | | | | | Also used in Quote Requests to specify receiving parties |
| | | | | | 58 | Entering Unit<br>**OMX Comment:** The rolle of an organizational unit the performs data entry on behalf of an executing firm |
| | | | | | 59 | Executing Unit<br>**OMX Comment:** The role of an organizational unit the is responsible for a business transaction sent to the marketplace |
| | | | | | 72 | Reporting intermediary (medium/vendor via which report has been published) |
| | | | | | 1000 | Access provider<br>**OMX Comment:** For use in Parties reference data mesages only |
| | | | | | 1001 | Vendor<br>**OMX Comment:** For use in Parties reference data mesages only |
| | | | | | 1002 | ISV<br>**OMX Comment:** For use in Parties reference data mesages only |
| | | | | | 1003 | Periferal application<br>**OMX Comment:** For use in Parties reference data mesages only |
| | | | | | 1004 | Fund Manager<br>**OMX Comment:** For use in Parties reference data mesages only |
| | | | | | 1005 | Other<br>**OMX Comment:** For use in Parties refer- |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | Value: Description: ence data mesages only |
| 453 | NoPartyIDs | NumInGroup | | Number of PartyID (448), PartyIDSource (447), and PartyRole (452) entries | |
| 454 | NoSecurityAltID | NumInGroup | | Number of SecurityAltID (455) entries. | |
| 455 | SecurityAltID | String | | Alternate Security identifier value for this security of SecurityAltIDSource (456) type (e.g. CUSIP, SEDOL, ISIN, etc). Requires SecurityAltIDSource. | |
| 456 | SecurityAltIDSource | String | | Identifies class or source of the SecurityAltID (455) value. Required if SecurityAltID is specified.<br>Valid values:<br>Same valid values as the SecurityIDSource (22) field | |
| 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.<br>Valid values:<br>Same valid values as the SecurityIDSource (22) field | |
| 460 | Product | int | | Indicates the type of product the security is associated with. See also the CFICode (461) and SecurityType (167) fields. | Value: Description:<br>1: AGENCY<br>2: COMMODITY<br>3: CORPORATE<br>4: CURRENCY<br>5: EQUITY<br>6: GOVERNMENT<br>7: INDEX |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | Value: 8 LOAN<br>9 MONEYMARKET<br>10 MORTGAGE<br>11 MUNICIPAL<br>12 OTHER<br>13 FINANCING |
| 461 | CFICode | String | | Indicates the type of security using ISO 10962 standard, Classification of Financial Instruments (CFI code) values. ISO 10962 is maintained by ANNA (Association of National Numbering Agencies) acting as Registration Authority. See "Appendix 6-B FIX Fields Based Upon Other Standards". See also the Product (460) and SecurityType (167) fields. It is recommended that CFICode be used instead of SecurityType (167) for non-Fixed Income instruments. A subset of possible values applicable to FIX usage are identified in "Appendix 6-D CFICode Usage - ISO 10962 Classification of Financial Instruments (CFI code)" | |
| 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). | |
| 470 | CountryOfIssue | Country | | ISO Country code of instrument issue (e.g. the country portion typically used in ISIN). Can be used in conjunction with non-ISIN SecurityID (48) (e.g. CUSIP for Municipal Bonds without ISIN) to provide uniqueness. | |
| 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**<br>0 New<br>1 Cancel<br>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.<br>**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)<br>(see Volume : "Glossary" for value definitions) | **Value** / **Description**<br>A Agency<br>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**<br>5 — Acting as Market Maker or Specialist in the security<br>Y — Issuer Holding<br>**OMX Comment:** Not in FIX Standard. OMX requests addition<br>Z — Issue Price Stabilization<br>**OMX Comment:** Not in FIX Standard. OMX requests addition |
| 533 | TotalAffectedOrders | int | | Total number of orders affected by mass cancel request. | |
| 534 | NoAffectedOrders | NumInGroup | | Number of affected orders in the repeating group of order ids. | |
| 535 | AffectedOrderID | String | | OrderID (37) of an order affected by a mass cancel request. | |
| 536 | AffectedSecondaryOrderID | String | | SecondaryOrderID (198) of an order affected by a mass cancel request. | |
| 537 | QuoteType | int | | Identifies the type of quote. An indicative quote is used to inform a counterparty of a market. An indicative quote does not result directly in a trade.<br>A tradeable quote is submitted to a market and will result directly in a trade against other orders and quotes in a market.<br>A restricted tradeable quote is submitted to a market and within a certain restriction (possibly based upon price or quantity) will automatically trade against orders. Order that do not comply with restrictions are sent to the quote issuer who can choose to accept or decline the order.<br>A counter quote is used in the negotiation model. See Volume 7 – Product: Fixed Income for example usage. | **Value** / **Description**<br>0 — Indicative<br>1 — Tradeable |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| 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 | |
| 541 | MaturityDate | LocalMktDate | | Date of maturity. | |
| 552 | NoSides | NumInGroup | | Number of Side repeating group instances. | **Value** / **Description**<br>1 — One Side<br>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**<br>0 — Valid request<br>1 — Invalid or unsupported request<br>2 — No instruments found that match selection criteria<br>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 | |
| 564 | LegPositionEffect | char | | PositionEffect for leg of a multileg<br>See PositionEffect (77) field for description | |
| 566 | LegPrice | Price | | Price for leg of a multileg<br>See Price (44) field for description | |
| 567 | TradSesStatusRejReason | int | | Indicates the reason a Trading Session Status Request was rejected. | **Value** / **Description**<br>1 — Unknown or invalid TradingSessionID<br>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**<br>N – Not reported to counterparty<br>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**<br>0 – Compared, matched or affirmed<br>1 – Uncompared, unmatched, or unaffired |
| 574 | MatchType | String | | The point in the matching process at which this trade was matched. | **Value** / **Description**<br>1 – One-Party Trade Report (privately negotiated trade)<br>4 – Auto-match<br>5 – Cross Auction<br>6 – Counter-Order Selection<br>7 – Call Auction<br>8 – Issuing/Buy-Back Auction<br>**OMX Comment:** FIX 5.0 SP1 |
| 586 | OrigOrdModTime | UTCTimestamp | | The most recent (or current) modification TransactTime (tag 60) reported on an Execution Report for the order. The OrigOrdModTime is provided as an optional field on Order Cancel Request and Order Cancel Replace Requests to identify that the state of the order has not changed since the request was issued.<br>This is provided to support markets similar to Eurex and A/C/E. | |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| 600 | LegSymbol | String | | Multileg instrument's individual security's Symbol.<br>See Symbol (55) field for description | |
| 601 | LegSymbolSfx | String | | Multileg instrument's individual security's SymbolSfx.<br>See SymbolSfx (65) field for description | |
| 602 | LegSecurityID | String | | Multileg instrument's individual security's SecurityID.<br>See SecurityID (48) field for description | |
| 603 | LegSecurityIDSource | String | | Multileg instrument's individual security's SecurityIDSource.<br>See SecurityIDSource (22) field for description | |
| 604 | NoLegSecurityAltID | String | | Multileg instrument's individual security's NoSecurityAltID.<br>See NoSecurityAltID (454) field for description | |
| 605 | LegSecurityAltID | String | | Multileg instrument's individual security's SecurityAltID.<br>See SecurityAltID (455) field for description | |
| 606 | LegSecurityAltIDSource | String | | Multileg instrument's individual security's SecurityAltIDSource.<br>See SecurityAltIDSource (456) field for description | |
| 608 | LegCFICode | String | | Multileg instrument's individual security's CFICode.<br>See CFICode (461) field for description | |
| 616 | LegSecurityExchange | Exchange | | Multileg instrument's individual security's SecurityExchange.<br>See SecurityExchange (207) field for description | |
| 620 | LegSecurityDesc | String | | Multileg instrument's individual security's SecurityDesc.<br>See SecurityDesc (07) field for description | |
| 621 | EncodedLegSecurityDescLen | Length | | Multileg instrument's individual security's EncodedSecurityDescLen.<br>See EncodedSecurityDescLen (350) field for description | |
| 622 | EncodedLegSecurityDesc | data | | Multileg instrument's individual security's EncodedSecurityDesc.<br>See EncodedSecurityDesc (35) field for description | |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| 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).<br>See Side (54) field for description and values | |
| 625 | TradingSessionSubID | 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. | **Value** / **Description**<br>1 — Pre-Trading<br>**OMX Comment:** Not in FIX Standard. OMX requests addition<br>2 — Opening or Opening Auction<br>**OMX Comment:** Not in FIX Standard. OMX requests addition<br>3 — (Continuous) Trading<br>**OMX Comment:** Not in FIX Standard. OMX requests addition<br>4 — Closing or Closing Auction<br>**OMX Comment:** Not in FIX Standard. OMX requests addition<br>5 — Post-Trading<br>**OMX Comment:** Not in FIX Standard. OMX requests addition<br>6 — Intraday Auction<br>**OMX Comment:** Not in FIX Standard. OMX requests addition<br>7 — Quiescent<br>**OMX Comment:** Not in FIX Standard. OMX requests addition |
| 636 | WorkingIndicator | Boolean | | Indicates if the order is currently being worked. Applicable only for OrdStatus = "New". For open outcry markets this indicates that the | **Value** / **Description**<br>N — Order has been accepted but not yet in a working state |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | 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**<br>Y / Order is currently being worked |
| 644 | RFQReqID | String | | RFQ Request ID – used to identify an RFQ Request. | |
| 649 | QuoteStatusReqID | String | | Unique identifier for Quote Status Request. | |
| 658 | QuoteRequestRejectReason | int | | Reason Quote was rejected: | **Value** / **Description**<br>1 / Unknown Symbol (Security)<br>2 / Exchange (Security) Closed<br>3 / Quote Request Exceeds Limit<br>5 / Invalid Price<br>6 / Not Authorized To Request Quote<br>7 / No Match For Inquiry<br>8 / No Market For Instrument<br>99 / Other |
| 661 | AllocAcctIDSource | int | | Used to identify the source of the AllocAccount (79) code. See AcctIDSource (660) for valid values. | |
| 667 | ContractSettlMonth | monthyear | | Specifies when the contract (i.e. MBS/TBA) will settle. | |
| 670 | NoLegAllocs | NumInGroup | | Number of Allocations for the leg | |
| 671 | LegAllocAccount | String | | Allocation Account for the leg See AllocAccount (79) for description and valid values. | |
| 672 | LegIndividualAllocID | String | | Reference for the individual allocation ticket See IndividualAllocID (467) for description and valid values. | |
| 673 | LegAllocQty | Qty | | Leg allocation quantity. See AllocQty (80) for description and valid values. | |
| 674 | LegAllocAcctIDSource | String | | The source of the LegAllocAccount (671) See AllocAcctIDSource (661) for description and valid values. | |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| 675 | LegSettlCurrency | Currency | | Identifies settlement currency for the Leg.<br>See SettlCurrency (20) for description and valid values | |
| 685 | LegOrderQty | Qty | | Quantity ordered of this leg.<br>See OrderQty (38) for description and valid values | |
| 691 | Pool | String | | For Fixed Income, identifies MBS / ABS pool. | |
| 693 | QuoteRespID | String | | Message reference for Quote Response | |
| 694 | QuoteRespType | int | | Identifies the type of Quote Response. | **Value** / **Description**<br>1 Hit/Lift<br>2 Counter<br>6 Pass |
| 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. | |
| 736 | AllocSettlCurrency | Currency | | Currency code of settlement denomination for a specific AllocAccount (79). | |
| 751 | TradeReportRejectReason | int | | Reason Trade Capture Request was rejected.<br>4000+ Reserved and available for bi-laterally agreed upon user-defined values | **Value** / **Description**<br>0 Successful (default)<br>1 Invalid party onformation<br>2 Unknown instrument<br>3 Unauthorized to report trades<br>4 Invalid trade type<br>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.<br>Same values as PartyID (448) | |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| 758 | Nested2PartyIDSource | char | | PartyIDSource value within a "second instance" Nested repeating group.<br>Same values as PartyIDSource (447) | |
| 759 | Nested2PartyRole | int | | PartyRole value within a "second instance" Nested repeating group.<br>Same values as PartyRole (452) | |
| 762 | SecuritySubType | String | | Sub-type qualification/identification of the SecurityType (e.g. for SecurityType="REPO"), or the CFICode if SecurityType is not specified. If specified, SecuirtyType or CFICode is required.<br>Example Values:<br>General = General Collateral (for SecurityType=REPO)<br>For SecurityType="MLEG" markets can provide the name of the option or futures strategy, such as Calendar, Vertical, Butterfly, etc.<br>NOTE: Additional values may be used by mutual agreement of the counterparties | |
| 788 | TerminationType | int | | Type of financing termination. | **Value** / **Description**<br>1 Overnight<br>2 Term<br>3 Flexible<br>4 Open |
| 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<br>4000+ = Reserved and available for bi-laterally agreed upon user defined values | **Value** / **Description**<br>1 Firm<br>2 Person<br>5 Full legal name of firm |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | **Value** / **Description**<br>6 Postal address<br>7 Phone number<br>8 Email address<br>9 Contact name<br>16 BIC<br>18 Registered address<br>21 Fax number<br>25 Location desk |
| 813 | ApplQueueDepth | int | | Current number of application messages that were queued at the time that the message was created by the counterparty. | |
| 814 | ApplQueueResolution | int | | Resolution taken when ApplQueueDepth (813) exceeds ApplQueueMax (812) or system specified maximum queue size. | **Value** / **Description**<br>0 No Action Taken<br>1 Queue Flushed<br>2 Overlay Last<br>3 End Session |
| 827 | ExpirationCycle | int | | Part of trading cycle when an instrument expires. Field is applicable for derivatives. | **Value** / **Description**<br>0 Expire on trading session close (default)<br>1 Expire on trading session open<br>2 Expires at specified expiration as the eligibility |
| 828 | TrdType | int | | Type of Trade: | **Value** / **Description**<br>0 Regular Trade<br>4 Late Trade<br>6 Weighted Average Price Trade<br>38 Block trade (same as large trade)<br>47 Financing transaction (includes repo and stock lending) |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | **Value** / **Description**<br>48 — Non-Standard Settlment<br>**OMX Comment:** FIX 5.0 SP1<br>49 — Derivative Related Transaction<br>**OMX Comment:** FIX 5.0 SP1<br>50 — Portfolio Trade<br>**OMX Comment:** FIX 5.0 SP1<br>51 — Volume Weighted Average Trade<br>**OMX Comment:** FIX 5.0 SP1<br>52 — Exchange Granted Trade<br>**OMX Comment:** FIX 5.0 SP1<br>53 — Repurchase Agreement<br>**OMX Comment:** FIX 5.0 SP1<br>54 — OTC<br>**OMX Comment:** FIX 5.0 SP1 |
| 829 | TrdSubType | int | | Further qualification to the trade type | **Value** / **Description**<br>97 — OTC Quote<br>**OMX Comment:** Not in FIX Standard. OMX requests addition<br>98 — On-Hours<br>**OMX Comment:** Not in FIX Standard. OMX requests addition<br>99 — Off-hours<br>**OMX Comment:** Not in FIX Standard. OMX requests addition |
| 845 | Discretion-Price | Price | | The current discretionary price of the order | |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| 852 | PublishTrdIndicator | int | | Indicates if a trade should be reported via a market reporting service. The indicator governs all reporting services of the receipient.<br>**OMX Comment:** Field is Boolean in FIX 5.0, OMX requests change | **Value** / **Description**<br>0 Do Not Report Trade<br>1 Report Trade<br>2 Deferred Publication |
| 855 | SecondaryTrdType | int | | Additional TrdType (see tag 828) assigned to a trade by trade match system. | |
| 856 | TradeReportType | int | | Type of Trade Report | **Value** / **Description**<br>0 Submit<br>1 Alleged<br>2 Accept<br>3 Decline<br>4 Addendum<br>5 No/Was<br>7 (Locked-In) Trade Break<br>11 Alleged New<br>12 Alleged Addendum<br>14 Alleged Trade Report Cancel<br>15 Alleged (Locked-In) Trade Break |
| 864 | NoEvents | NumInGroup | | Number of repeating EventType entries. | |
| 865 | EventType | int | | Code to represent the type of event | **Value** / **Description**<br>5 Activation<br>6 Inactiviation<br>8 Swap Start Date<br>9 Swap End Date<br>10 Swap Next Start Date<br>11 Swap Roll Date<br>12 Swap Next Roll Date<br>13 First Delivery Date<br>14 Last Delivery Date<br>15 Initial Inventory Due Date |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | **Value** / **Description**<br>16 Final Inventory Due Date<br>17 First Intent Date<br>18 Last Intent Date<br>19 Position Removal Date<br>100 On the Surveillance List<br>101 Excluding Dividend<br>102 Excluding Participating in Rights<br>103 Excluding Participating in Split<br>104 Company subject to Public Offer<br>105 Under Drawing<br>106 Excluding combined Split and Issue Rights |
| 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**<br>1 Flat (securities pay interest on a current basis but are traded without interest)<br>2 Zero coupon<br>3 Interest bearing (for Euro commercial paper when not issued at discount)<br>4 No periodic payments<br>5 Variable rate<br>6 Less fee for put<br>7 Stepped coupon<br>8 Coupon period (if not semi-annual). Supply redemption date in the InstrAttribValue (872) field. |

| Tag | FieldName | Type | OMXLen | Desc | Valid values | |
|---|---|---|---|---|---|---|
| | | | | | **Value** | **Description** |
| | | | | | 9 | When [and if] issued |
| | | | | | 10 | Original issue discount |
| | | | | | 11 | Callable, puttable |
| | | | | | 12 | Escrowed to Maturity |
| | | | | | 13 | Escrowed to redemption date - callable. Supply redemption date in the InstrAttribValue (872) field |
| | | | | | 14 | Pre-refunded |
| | | | | | 15 | In default |
| | | | | | 16 | Unrated |
| | | | | | 17 | Taxable |
| | | | | | 18 | Indexed |
| | | | | | 19 | Subject To Alternative Minimum Tax |
| | | | | | 20 | Original issue discount price. Supply price in the InstrAttribValue (872) field |
| | | | | | 21 | Callable below maturity value |
| | | | | | 22 | Callable without notice by mail to holder unless registered |
| | | | | | 23 | Price tick rules for security. Tick rule values to be expressed using InstrAttribValue [872] |
| | | | | | 24 | Trade type eligibility details for security. Trade types to be expressed using InstrAttribValue [872] |
| | | | | | 25 | Instrument Denominator |
| | | | | | 26 | Instrument Numerator |
| | | | | | 27 | Instrument Price Precision |
| | | | | | 28 | Instrument Strike Price Precision |
| | | | | | 29 | Tradeable Indicator |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | **Value** / **Description**<br>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. | |
| 873 | DatedDate | LocalMktDate | | The effective date of a new securities issue determined by its underwriters. Often but not always the same as the Issue Date and the Interest Accrual Date | |
| 875 | CPProgram | int | | The program under which a commercial paper is issued<br>**OMX Comment:** Valid values are not yet defined | **Value** / **Description**<br>99 / Other |
| 876 | CPRegType | String | | The registration type of a commercial paper issuance | |
| 880 | TrdMatchID | String | | Identifier assigned to a trade by a matching system.<br>**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**<br>N / Not Last Message<br>Y / Last Message |
| 898 | MarginRatio | Percentage | | The fraction of the cash consideration that must be collateralized, expressed as a percent. A MarginRatio of 02% indicates that the value of the collateral (after deducting for "haircut") must exceed the cash consideration by 2%. | |
| 913 | AgreementDesc | String | | The full name of the base standard agreement, annexes and amendments in place between the principals applicable to a financing transaction. | |
| 914 | AgreementID | String | | A common reference to the applicable standing agree- | |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | ment between the counterparties to a financing transaction. | |
| 915 | AgreementDate | LocalMktDate | | A reference to the date the underlying agreement specified by AgreementID and AgreementDesc was executed. | |
| 916 | StartDate | LocalMktDate | | Start date of a financing deal, i.e. the date the buyer pays the seller cash and takes control of the collateral | |
| 917 | EndDate | LocalMktDate | | End date of a financing deal, i.e. the date the seller reimburses the buyer and takes back control of the collateral | |
| 918 | AgreementCurrency | Currency | | Contractual currency forming the basis of a financing agreement and associated transactions. Usually, but not always, the same as the trade currency. | |
| 919 | DeliveryType | int | | Identifies type of settlement | **Value** / **Description**<br>0 / "Versus Payment": Deliver (if sell) or Receive (if buy) vs. (against) Payment<br>1 / "Free": Deliver (if sell) or Receive (if buy) Free<br>2 / Tri-Party<br>3 / Hold In Custody |
| 939 | TrdRptStatus | int | | Trade Report Status | **Value** / **Description**<br>0 / Accepted<br>1 / Rejected |
| 947 | StrikeCurrency | Currency | | Currency in which the StrikePrice is denominated. | |
| 948 | NoNested3PartyIDs | NumInGroup | | Number of Nested3PartyID (949), Nested3PartyIDSource (950), and Nested3PartyRole (95) entries | |
| 949 | Nested3PartyID | String | | PartyID value within a "third instance" Nested repeating group.<br>Same values as PartyID (448) | |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| 950 | Nested3PartyIDSource | char | | PartyIDSource value within a "third instance" Nested repeating group.<br>Same values as PartyIDSource (447) | |
| 951 | Nested3PartyRole | int | | PartyRole value within a "third instance" Nested repeating group.<br>Same values as PartyRole (452) | |
| 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**<br>1 Active<br>2 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** / **Description**<br>A Add<br>D Delete<br>M Modify |
| 996 | UnitofMeasure | String | | Physical unit of measure for Derivative products.<br>NOTE: Additional values may be used by mutual agreement of the counterparties<br>(http://www.unc.edu/~rowlett/units/index.html is a good source for units) | **Value** / **Description**<br>Bbl Barrels<br>Bcf Billion cubic feet<br>Bu Bushels<br>lbs pounds<br>Gal Gallons<br>MMbbl Million Barrels<br>MMBtu One Million BTU<br>MWh Megawatt hours<br>oz_tr Troy Ounces |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | Value: t — Description: Metric Tons (aka Tonne)<br>Value: tn — Description: Tons (US)<br>Value: USD — Description: US Dollars<br>Value: BTU — Description: BTU |
| 997 | TimeUnit | String | | Unit of time associated with the contract.<br>NOTE: Additional values may be used by mutual agreement of the counterparties | Value: H — Description: Hour<br>Value: Min — Description: Minute<br>Value: S — Description: Second<br>Value: D — Description: Day<br>Value: Wk — Description: Week<br>Value: Mo — Description: Month<br>Value: Yr — Description: Year |
| 1003 | TradeID | String | | The unique ID assigned to the trade entity once it is received or matched by the exchange or central counterparty. | |
| 1018 | NoInstrument-Parties | NumIn-Group | | Identifies the number of parties identified with an instrument | |
| 1019 | Instrument-PartyID | String | | PartyID value within an instrument party repeating group. Same values as PartyID (448) | |
| 1021 | MDBook-Type | 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<br>Value: 3 — Description: Order Depth |
| 1022 | MDFeed-Type | String | | Describes a class of service for a given data feed, ie Regular and Market Maker, Bandwidth Intensive or Bandwidth Conservative | |
| 1023 | MDPriceLev-el | 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 | |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| 1050 | InstrumentPartyIDSource | char | | PartyIDSource value within an instrument partyrepeating group.<br>Same values as PartyIDSource (447) | |
| 1051 | InstrumentPartyRole | int | | PartyRole value within an instrument partyepeating group.<br>Same values as PartyRole (452) | |
| 1057 | AggressorIndicator | Boolean | | Used to identify whether the order initiator is an aggressor or not in the trade. | **Value** / **Description**<br>Y – Order initiator is aggressor<br>N – Order initiator is passive |
| 1070 | MDQuoteType | int | | Identifies market data quote type. | **Value** / **Description**<br>0 – Indicative<br>1 – Tradeable |
| 1079 | MaturityTime | TZTimeOnly | | 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 | RefOrderIDSource | char | | Used to specify what identifier, provided in order depth market data, to use when hitting (taking) a specific order. | **Value** / **Description**<br>0 – SecondaryOrderID (198) |
| 1084 | DisplayMethod | char | | Defines what value to use in DisplayQty (1138). If not specified the default DisplayMethod is "1" | **Value** / **Description**<br>1 – Initial (use original DisplayQty)<br>2 – New (use RefreshQty) |
| 1088 | RefreshQty | Qty | | Defines the quantity used to refresh DisplayQty. | |
| 1089 | MatchIncrement | 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. | |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| 1091 | PreTradeAnonymity | 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**<br>1 Odd Lot<br>2 Round Lot<br>3 Block Lot |
| 1115 | OrderCategory | char | | Defines the type of interest behind a trade (fill or partial fill). | **Value** / **Description**<br>1 Order<br>2 Quote<br>3 Privately Negotiated Trade<br>4 Multileg order<br>5 Linked order<br>6 Quote Request<br>7 Implied Order<br>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) | |
| 1120 | NoRootPartySubIDs | NumInGroup | | Number of RootPartySubID (1121) and RootPartySubIDType (1122) entries | |
| 1121 | RootPartySubID | String | | PartySubID value within a root parties component. Same values as PartySubID (523) | |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| 1122 | RootParty-SubIDType | int | | Type of RootPartySubID (1121) value. Same values as PartySubIDType (803) | |
| 1123 | TradeHandlin-gInstr | char | | Specified how the Trade Capture Report should be handled by the Respondent. | **Value** / **Description**<br>0 / Trade Confirmation<br>2 / One-Party Report for Matching |
| 1126 | OrigTradeID | String | | Used to preserve original trade id when original trade is being referenced in a sub-sequent trade transaction such as a transfer | |
| 1127 | OrigSec-ondary-TradeID | String | | Used to preserve original secondary trade id when original trade is being refer-enced 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**<br>7 / FIX50 |
| 1137 | DefaultAp-plVerID | String | | Specifies the service pack release being applied, by de-fault, to message at the ses-sion 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 dis-played quantity. | |
| 1140 | MaxTradeVol | Qty | | The maximum order quantity that can be submitted for a security | |
| 1142 | MatchAlgo-rithm | String | | The type of algorithm used to match orders in a specific security<br>Possible values are FIFO, Allocation, Pro-rata, Lead Market Maker, Currency Cal-endar | |
| 1143 | MaxPriceVari-ation | Float | | The maximum price variation of an execution from one event to the next for a given security | |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| 1144 | ImpliedMarketIndicator | int | | Commonly used in listed derivatives. Indicates that an implied market should be created for either the legs of a multi-leg instrument (Implied-in) or for the multi-leg instrument based on the existence of the legs (Implied-out). Determination as to whether implied markets should be created is generally done at the level of the multi-leg instrument | **Value** / **Description**<br>0 – Not implied<br>1 – Implied-in – the existence of a multi-leg instrument is implied by the legs of that instrument<br>2 – Implied-out – The existence of the underlying legs are implied by the multi-leg instrument<br>3 – Both Implied-in and Implied-out |
| 1145 | EventTime | UTCTimestamp | | Specific time of event. To be used in combination with EventDate [866] | |
| 1147 | UnitofMeasureQty | Qty | | Used to indicate the size of the underlying commodity on which the contract is based, (e.g., 2500 lbs of lean cattle, 1000 barrels of crude oil, 1000 bushels of corn, etc.) | |
| 1148 | LowLimitPrice | Price | | Allowable low limit price for the trading day. A key parameter in validating order price. Used as the lower band for validating order prices. Orders submitted with prices below the lower limit will be rejected | |
| 1149 | HighLimitPrice | Price | | Allowable high limit price for the trading day. A key parameter in validating order price. Used as the upper band for validating order prices. Orders submitted with prices above the upper limit will be rejected | |
| 1150 | TradingReferencePrice | Price | | Reference price for the current trading price range usually representing the mid price between the HighLimitPrice and LowLimitPrice. The value may be the settlement price or closing price of the prior trading day. | |
| 1151 | SecurityGroup | String | | An exchange specific name assigned to a group of related securities which may be concurrently affected by market events and actions. | |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| 1166 | QuoteMsgID | String | | Unique identifier for a quote message<br>**OMX Comment:** FIX 5.0 SP1 | |
| 1167 | QuoteEntryStatus | int | | Identifies the status of an individual quote. See also QuoteStatus (297) which used for single Quotes.<br>**OMX Comment:** FIX 5.0 SP1 | **Value** / **Description**<br>0 – Accepted<br>**OMX Comment:** FIX 5.0 SP1<br>5 – Rejected<br>**OMX Comment:** FIX 5.0 SP1<br>6 – Removed from Market<br>**OMX Comment:** FIX 5.0 SP1<br>7 – Expired<br>**OMX Comment:** FIX 5.0 SP1<br>16 – Active<br>**OMX Comment:** FIX 5.0 SP1 |
| 1168 | TotNoCxldQuotes | int | | Specifies the number of canceled quotes<br>**OMX Comment:** FIX 5.0 SP1 | |
| 1169 | TotNoAccQuotes | int | | Specifies the number of accepted quotes<br>**OMX Comment:** FIX 5.0 SP1 | |
| 1170 | TotNoRejQuotes | int | | Specifies the number of rejected quotes<br>**OMX Comment:** FIX 5.0 SP1 | |
| 1171 | PrivateQuote | Boolean | | Specifies whether a quote is public, i.e. available to the market, or private, i.e. available to a specified counterparty only.<br>Valid Values:<br>TRUE = Private Quote<br>FALSE = Public Quote<br>**OMX Comment:** FIX 5.0 SP1 | |
| 1172 | RespondentType | int | | Specifies the type of respondents requested.<br>**OMX Comment:** FIX 5.0 SP1 | **Value** / **Description**<br>1 – All market participants<br>2 – Specified market participants |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | **Value** / **Description**<br>3 – All market makers<br>4 – Primary market maker(s) |
| 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.<br>**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.<br>**OMX Comment:** FIX 5.0 SP1 | **Value** / **Description**<br>1 – Order imbalance, auction is extended<br>2 – Price Volatility Interruption<br>3 – Trading resumes (after Halt)<br>4 – Change of Trading Session<br>5 – Change of Trading Subsession<br>6 – Change of Security Status<br>7 – Change of Book Type<br>8 – Change of Market Depth<br>100 – Session change alert |
| 1175 | NoStatsIndicators | NumInGroup | | Number of statistics indicator repeating group entries<br>**OMX Comment:** FIX 5.0 SP1 | |
| 1176 | StatsType | Int | | Type of statistics the MDEntry is eligible to be included in<br>**OMX Comment:** FIX 5.0 SP1 | **Value** / **Description**<br>1 – Exchange Last<br>2 – High / Low Price<br>3 – Average Price (VWAP, TWAP...)<br>4 – Turnover (Px * Qty) |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| 20004 | Multileg-PriceMethod | int | | Defines the type of combination price the multileg uses | **Value** / **Description**<br>1 Net Price<br>2 Reversed Net Price<br>3 Yield Difference<br>4 Individual<br>5 Weighted Average Price<br>6 Multiplied Price |
| 20016 | MultilegModel | int | | Specifies the type of multileg model the user is targeting. | **Value** / **Description**<br>0 Predefined Multileg Security<br>2 Strategy Order |
| 20030 | NoMDStatInstruments | NumInGroup | | Number of Instrument entries in a statistics message | |
| 20031 | NoMDStats | NumInGroup | | Number of statistics entries for an instrument | |
| 20032 | MDStatType | String | | Type of statistic. Additional values can be bilaterally agreed between parties. | **Value** / **Description**<br>3 Index Value<br>4 Opening Price<br>5 Closing Price<br>6 Settlement Price<br>7 High Price<br>8 Low Price<br>9 VWAP Price<br>B Trade Volume<br>C Open Interest<br>D Composite Underlying Price<br>E Simulated Buy Price<br>F Simulated Sell Price<br>G Margin Rate<br>H Mid Price<br>K Settle High Price<br>L Settle Low Price<br>N High Bid<br>O Low Offer |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | Value / Description<br>ZO Last Privately Negotiated Trade Volume<br>ZP Last Privately Negotiated Trade Price<br>ZQ Exchange Best<br>ZR Consolidated Best<br>ZS Exchange Last<br>ZT Final Price of Session<br>ZU Turnover Value<br>ZV TWAP<br>ZW Auction Price<br>ZX First Price<br>ZY Number of Trades<br>ZZ Number of Deals |
| 20034 | MDReportID | String | | Identifier for Market Data reports | |
| 20035 | MDStatScope | String | | Defines the scope of the statistics in periods of time. | |
| 20036 | MarketSegmentID | String | | Market Segment identifier value | |
| 20037 | MarketSegmentDesc | String | | Description or name of Market Segment | |
| 20038 | EncodedMktSegmDescLen | Length | | Byte length of encoded (non-ASCII characters) EncodedSecurityDesc (351) field. | |
| 20039 | EncodedMktSegmDesc | data | | Encoded (non-ASCII characters) representation of the MarketSegmDesc (20037) 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 (20036) | |
| 20047 | AuctionReqID | String | | Uniquely identifies an Auction Request (U2) message | |
| 20048 | AuctionType | int | | Defines the type of call auction | Value / Description<br>1 Issuing Auction<br>2 Buy-back Auction |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| 20049 | AuctionTime | UTC-Times-tamp | | Defines the time a call auc-tion will be executed | |
| 20053 | AuctionRejec-tReason | int | | Reason an Auction Request was rejected | **Value** / **Description**: 1 User not authorized; 2 Auction book not found; 99 Other |
| 20054 | AuctionResul-tID | String | | Uniquely identifies an Auction result (U4) message | |
| 20056 | NumberOf-MatchOrders | int | | Number of matched orders | |
| 20057 | ContraQty | Qty | | Quantity at the contra side | |
| 20058 | TradeQty | Qty | | Fill quantity | |
| 20060 | TradeVWAP | Price | | Volume Weigthed Average Price for fills | |
| 20062 | PctMatchQty | Percent-age | | % of quantity that matched | |
| 20063 | NewsID | String | | Unique identifier for News messages | |
| 20064 | NewsCatego-ry | int | | Describes a category of news | **Value** / **Description**: 1 Company news; 2 Marketplace news; 3 Financial market news; 4 Technical news; 99 Other |
| 20065 | Language-Code | String | | Language code according to ISO 639 2-alpha character values. | |
| 20066 | NoRefNews | NumIn-Group | | Specifies the number of re-peating news references | |
| 20067 | RefNewsID | String | | NewsID of referenced News message | |
| 20068 | RefNew-sType | int | | Specifies the type of news reference. | **Value** / **Description**: 1 Replacement; 2 Other language; 3 Complimentary |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| 20072 | NewsPublReqID | String | | Unique identifier for a News Publiation Request | |
| 20073 | OrigNewsPubReqID | String | | Reference to the identifier for a News Publiation Request | |
| 20074 | NewsUpdAction | int | | Type of News update action. | **Value** / **Description**: 1 New; 2 Change; 3 Delete |
| 20075 | PublTime | UTCTimestamp | | Publication date and time | |
| 20076 | NewsPublRejReason | int | | Reject reason for a news puyblication request | **Value** / **Description**: 1 Unknown symbol; 2 Duplicate NewsPublReqID; 3 News reference unfound; 4 Insufficient Permissions; 5 Unsupported UpdateAction; 6 Unknown Security Exchange; 7 Unknown Market Segment; 8 Unsupported Language; 99 Other |
| 20077 | ContingencyType | int | | Specifies the type of Contingent Order | **Value** / **Description**: 4 One Updates the Other (OUO) - Proportional Quantity Reduction |
| 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 |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | **Value** / **Description**<br>1 — Invalid or unsupported request<br>2 — No data found that match selection criteria<br>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**<br>0 — Not suspended<br>1 — Suspended |
| 20084 | MarketID | Exchange | | Identifies a marketplace<br>Valid values:<br>- See "Appendix 6-C" | |
| 20085 | MktSegmReqID | String | | Unique ID of a Market Segment Request message. | |
| 20087 | NoTrdRepIndicators | NumInGroup | | Number of trade reporting indicators<br>**OMX Comment:** Not in FIX. OMX requires an extension | |
| 20088 | TrdRepPartyRole | int | | Identifies the type of party for trade reporting. Same values as PartyRole (452).<br>**OMX Comment:** Not in FIX. OMX requires an extension | |
| 20089 | TradSesEvent | int | | Identifies an event related to a Trading Session. An event occurs and is gone, it is not a state that applies for a period of time.<br>**OMX Comment:** Not in FIX. The EEWG requires an extension | **Value** / **Description**<br>1 — Trading Resumes (after Halt)<br>2 — Change of Trading Session<br>3 — Change of Trading Subsession<br>4 — Change of Trading Status<br>101 — Initializing (transition to specified trading phase)<br>102 — Completed (transition to specified trading phase) |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| 20090 | LegAllocID | String | | Unique identifier for allocation message. Also see AllocID (70)<br>**OMX Comment:** Not in FIX. The EEWG requires an extension | |
| 20091 | AuctionInfoID | String | | Uniquely identifies an Auction Info (UB) message | |
| 20092 | AuctionInfoReqID | String | | Uniquely identifies an Auction Info Request (UC) message | |
| 20093 | PriceUnitOfMeasure | String | | Used to express the UOM of the price if different from the contract. In futures, this can be different for cross-rate products in which the price is quoted in units differently from the contract<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20094 | PriceUnitOfMeasureQty | Qty | | Used to express the UOM Quantity of the price if different from the contract. In futures, this can be different for physically delivered products in which price is quoted in a unit size different from the contract, i.e. a Cattle Future contract has a UOMQty of 40,000 and a PriceUOMQty of 100.<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20095 | SettlMethod | char | | Settlement method for a contract. Can be used as an alternative to CFI Code value<br>**OMX Comment:** Not in FIX. The GDC requires an extension | **Value** – **Description**<br>C – Cash settlement required<br>P – Physical settlement required |
| 20096 | ExerciseStyle | int | | Type of exercise of a derivatives security<br>**OMX Comment:** Not in FIX. The GDC requires an extension | **Value** – **Description**<br>0 – European – exercise only on expiration date<br>1 – American – exercise up to and including expiration date<br>2 – Bermuda – exercise on specified dates only |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | **Value** / **Description**<br>3 / Binary – buyer of option receives a fixed amount option is in the money |
| 20097 | CashAmount | Price | | Cash amount indicating the pay out associated with an option. For binary options this is a fixed amount<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20098 | PriceQuoteMethod | String | | Method for price quotation<br>**OMX Comment:** Not in FIX. The GDC requires an extension | **Value** / **Description**<br>STD / Standard, money per unit of a physical<br>INDX / Index<br>INT / Interest rate index |
| 20099 | ValueTypeCode | String | | For futures, indicates type of valuation method applied<br>**OMX Comment:** Not in FIX. The GDC requires an extension | **Value** / **Description**<br>EQTY / Premium style<br>FUT / Futures style mark-to-market<br>FUTDA / Futures style with an attached cash adjustment |
| 20100 | ListMethod | int | | Indicates whether instruments are pre-listed only or can also be defined via user request<br>**OMX Comment:** Not in FIX. The GDC requires an extension | **Value** / **Description**<br>0 / Pre-listed only<br>1 / User requested<br>2 / Undefined (no product) |
| 20101 | CapPrice | Price | | Used to express the ceiling price of a capped call<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20102 | FloorPrice | Price | | Used to express the floor price of a capped put<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20103 | NoStrikeRules | NumInGroup | | Number of strike rule entries. This block specifies the rules for determining how new | |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | strikes should be listed within the stated price range of the underlying instrument<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20104 | StartStrikePxRange | Price | | Starting price for the range to which the StrikeIncrement applies. Price refers to the price of the underlying<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20105 | EndStrilePxRange | Price | | Ending price of the range to which the StrikeIncrement applies. Price refers to the price of the underlying<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20106 | StrikeIncrement | float | | Value by which strike price should be incremented within the specified price range.<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20107 | NoMarketSegments | NumInGroup | | No of Markets or Market Segments which a security may trade<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20108 | NoTickRules | NumInGroup | | Number of tick rules. This block specifies the rules for determining how a security ticks, i.e. the price increments at which it can be quoted and traded, depending on the current price of the security<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20109 | StartTickPriceRange | Price | | Starting price range for specified tick increment<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20110 | EndTickPricerange | Price | | Ending price range for the specified tick increment<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| 20111 | TickIncrement | Price | | Tick increment for stated price range. Specifies the valid price increments at which a security can be quoted and traded<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20112 | TickRuleType | int | | Specifies the type of tick rule which is being described<br>**OMX Comment:** Not in FIX. The GDC requires an extension | **Value** / **Description**<br>0 Regular<br>1 Variable<br>2 Fixed<br>3 Traded as spread leg<br>4 Settled as spread leg |
| 20113 | NestedInstrAttribType | int | | Code to represent the type of instrument attribute<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20114 | NestedInstrAttribValue | String | | Attribute value appropriate to the NestedInstrAttribType field<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20115 | DerivativeInstrAttribType | int | | Code to represent the type of instrument attribute<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20116 | DerivativeInstrAttribValue | String | | Attribute value appropriate to the SeriesInstrAttribType field<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20117 | DerivativeSymbol | String | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20118 | DerivativeSymbolSfx | String | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20119 | DerivativeSecurityID | String | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20120 | DerivativeSecurityIDSource | String | | **OMX Comment:** Not in FIX. The GDC requires an extension | |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| 20121 | NoDerivativeSecurityAltID | NumInGroup | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20122 | DerivativeSecurityAltID | String | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20123 | DerivativeSecurityAltIDSource | String | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20124 | DerivativeProduct | int | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20125 | DerivativeSecurityGroup | String | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20126 | DerivativeCFICode | String | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20127 | DerivativeSecurityType | String | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20128 | DerivativeSecuritySubType | String | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20129 | DerivativeMaturityMonthYear | monthyear | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20130 | DerivativeMaturityDate | LocalMktDate | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20131 | DerivativeMaturityTime | TZTimeOnly | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20132 | DerivativeSettleOnOpenFlag | String | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20133 | DerivativeInstrmtAssignmentMethod | char | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20134 | DerivativeSecurityStatus | String | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20135 | DerivativeIssueDate | LocalMktDate | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20136 | DerivativeInstrRegistry | String | | **OMX Comment:** Not in FIX. The GDC requires an extension | |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| 20137 | DerivativeCountryOfIssue | Country | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20138 | DerivativeStateOrProvinceOfIssue | String | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20139 | DerivativeLocaleOfIssue | String | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20141 | DerivativeStrikePrice | Price | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20142 | DerivativeStrikeCurrency | Currency | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20143 | DerivativeStrikeMultiplier | float | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20144 | DerivativeStrikeValue | float | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20145 | DerivativeOptAttribute | char | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20146 | DerivativeContractMultiplier | float | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20147 | DerivativeMinPriceIncrement | float | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20148 | DerivativeMinPriceIncrementAmount | Amt | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20149 | DerivativeUnitofMeasure | String | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20150 | DerivativeUnitofMeasureQty | Qty | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20151 | DerivativeExerciseStyle | int | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20152 | DerivativeCashAmount | Price | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20153 | DerivativeTimeUnit | String | | **OMX Comment:** Not in FIX. The GDC requires an extension | |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| 20154 | DerivativeSecurityExchange | Exchange | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20155 | DerivativePositionLimit | int | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20156 | DerivativeNTPositionLimit | int | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20157 | DerivativeIssuer | String | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20158 | DerivativeEncodedIssuerLen | Length | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20159 | DerivativeEncodedIssuer | data | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20160 | DerivativeSecurityDesc | String | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20161 | DerivativeEncodedSecurityDescLen | Length | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20162 | DerivativeEncodedSecurityDesc | data | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20163 | DerivativeSecurityXMLLen | Length | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20164 | DerivativeSecurityXML | data | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20165 | DerivativeSecurityXMLSchema | data | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20166 | DerivativeContractSettlMonth | monthyear | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20167 | NoDerivativeEvents | NumInGroup | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20168 | DerivativeEventType | int | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20169 | DerivativeEventDate | LocalMktDate | | **OMX Comment:** Not in FIX. The GDC requires an extension | |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| 20170 | DerivativeEventTime | UTCTimestamp | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20171 | DerivativeEventPx | Price | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20172 | DerivativeEventText | String | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20173 | NoDerivativeInstrumentParties | NumINGroup | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20174 | DerivativeInstrumentPartyID | String | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20175 | DerivativeInstrumentPartyIDSource | char | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20176 | DerivativeInstrumentPartyRole | int | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20180 | NoOrdTypeRules | NumInGroup | | Number of order types<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20181 | NoTimeInForceRules | NumInGroup | | Number of time in force techniques<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20182 | NoExecInstRules | NumInGroup | | Number of execution instructions<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20183 | ExecInstValue | char | | Indicates execution instructions that are valid for the specified trading rule context<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20184 | NoMatchRules | NumInGroup | | Number of Match Rules<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20185 | NoLotTypeRules | NumInGroup | | Number of Lot Type Rules<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| 20187 | NoMarket-DataFeed-Types | NumIn-Group | | Number of Feed Types<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20188 | NoMaturi-tyRules | NumIn-Group | | Number of maturity rules in MarurityRules component block<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20189 | StartMMY | month-year | | Starting maturity month year for an option class<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20190 | EndMMY | month-year | | Ending maturity month year for an option class<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20191 | MMYIncre-ment | int | | Increment between successive maturities for an option class<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20192 | MMYIncre-mentU-nitOfMeasure | int | | Unit of measure for the MMYIncrement<br>**OMX Comment:** Not in FIX. The GDC requires an extension | **Value** / **Description**<br>0 Months<br>1 Days<br>2 Weeks<br>3 Years |
| 20193 | MMYFormat | int | | Format used to generate the MMY for each option contract:<br>**OMX Comment:** Not in FIX. The GDC requires an extension | **Value** / **Description**<br>0 YearMonth Only (default)<br>1 YearMonthDay<br>2 YearMonthWeek |
| 20194 | StrikeExercis-eStyle | int | | Expiration Style for an option class:<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20195 | StrikeRuleID | String | | Allows strike rule to be referenced via an identifier so that rules do not need to be explicitly enumerated | |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20196 | MaturityRuleID | String | | Allows maturity rule to be referenced via an identifier so that rules do not need to be explicitly enumerated<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20197 | MinLotSize | Qty | | Minimum lot size allowed based on lot type specified in LotType/1093<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20198 | ProductComplex | String | | Identifies an entire suite of products for a given market. In Futures this may be “interest rates”, “agricultural”, “equity indexes”, etc<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20199 | DerivativeProductComplex | String | | Identifies an entire suite of products for a given market. In Futures this may be “interest rates”, “agricultural”, “equity indexes”, etc<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20200 | FlexibleIndicator | Boolean | | Used to indicate if a security has been defined as flexible according to “non-standard” means. Analog to CFICode Standard/Non-standard indicator<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20201 | FlexProductEligibilityIndicator | Boolean | | Used to indicate if a product or group of product supports the creation of flexible securities<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20202 | DerivFlexProductEligibilityIndicator | Boolean | | Used to indicate if a product or group of product supports the creation of flexible securities | |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | **OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20203 | TradingCurrency | Currency | | Used when the trading currency can differ from the price currency<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20204 | NoTradingSessions | NumInGroup | | Allows trading rules to be expressed by trading session<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20205 | SecondaryLowLimitPrice | Price | | Allowable low limit price for the trading day. A key parameter in validating order price. Used as the lower band for validating order prices. Orders submitted with prices below the lower limit will be rejected<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20206 | SecondaryHighLimitPrice | Price | | Allowable high limit price for the trading day. A key parameter in validating order price. Used as the upper band for validating order prices. Orders submitted with prices above the upper limit will be rejected<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20207 | SecondaryTradingReferencePrice | Price | | Reference price for the current trading price range usually representing the mid price between the HighLimitPrice and LowLimitPrice. The value may be the settlement price or closing price of the prior trading day.<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20208 | PriceLimitType | int | | Describes the how the price limits are expressed<br>**OMX Comment:** Not in FIX. The GDC requires an extension | **Value** / **Description**<br>0 Price<br>1 Ticks<br>2 Percentage |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| 20209 | SecondaryPriceLimitType | int | | Describes the how the price limits are expressed<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20211 | NoDerivativeInstrAttr | NumInGroup | | Number of derivative instrument attributes<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20212 | NoNestedInstrAttr | NumInGroup | | Number of nested instrument attributes<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20213 | DerivativePriceUnitOfMeasure | String | | Used to express the UOM of the price if different from the contract. In futures, this can be different for cross-rate products in which the price is quoted in units differently from the contract<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20214 | DerivativePriceUnitOfMeasureQty | Qty | | Used to express the UOM Quantity of the price if different from the contract. In futures, this can be different for physically delivered products in which price is quoted in a unit size different from the contract, i.e. a Cattle Future contract has a UOMQty of 40,000 and a PriceUOMQty of 100.<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20215 | DerivativeSettlMethod | char | | Settlement method for a contract. Can be used as an alternative to CFI Code value<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20216 | DerivativePutOrCall | int | | Used to express option right<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20218 | DerivativePriceQuoteMethod | String | | Method for price quotation<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| 20219 | DerivaticeFuturesValueTypeCode | String | | For futures, indicates type of valuation method applied<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20220 | DerivativeListMethod | int | | Indicates whether instruments are pre-listed only or can also be defined via user request<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20221 | DerivativeCapPrice | Price | | Used to express the ceiling price of a capped call<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20222 | DerivativeFloorPrice | Price | | Used to express the floor price of a capped put<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20223 | ListUpdateAction | char | | Specifies New (0), Cancel (1) or Replace (2). If provided, then Instrument occurrence has explicitly changed<br>**OMX Comment:** Not in FIX. The GDC requires an extension | |
| 20224 | TradingSessionDesc | String | | Trading Session description<br>**OMX Comment:** Not in FIX. The EEWG requires an extension | |
| 20225 | TradSesUpdateAction | char | | Specifies the action taken for the specified trading sessions.<br>**OMX Comment:** Not in FIX. The EEWG requires an extension | |
| 20226 | MktSegmMsgID | String | | Market Segment message identifier.<br>**OMX Comment:** Not in FIX. The EEWG requires an extension | |
| 20227 | MktSegmUpdateAction | char | | Specifies the action taken for the specified MarketID / MarketSegmentID.<br>**OMX Comment:** Not in FIX. The EEWG requires an extension | |
| 20228 | Volatility | float | | Annualized volatility for option model calculations | |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | **OMX Comment:** Not in FIX. OMX requires an extension | |
| 20229 | LegVolatility | float | | Annualized volatility for option model calculations<br>**OMX Comment:** Not in FIX. OMX requires an extension | |
| 20230 | DividendYield | float | | Specifies the expected dividend of a security. Expressed as yield.<br>**OMX Comment:** Not in FIX. OMX requires an extension | |
| 20231 | LegDividendYield | float | | Specifies the expected dividend for the Security of a leg. Expressed as yield.<br>**OMX Comment:** Not in FIX. OMX requires an extension | |
| 20232 | RiskfreeRate | float | | Specifies the expected risk-free interest rate<br>**OMX Comment:** Not in FIX. OMX requires an extension | |
| 20233 | CurrencyRatio | float | | Specifies the currency ratio between the currency used for a multileg price and the currency used by the outright book defined by the leg. Example: Multileg quoted in EUR, outright leg in USD and 1 EUR = 0,7 USD then LegCurrecyRatio = 0.7<br>**OMX Comment:** Not in FIX. OMX requires an extension | |
| 20234 | LegCurrencyRatio | float | | Specifies the currency ratio between the currency used for a multileg price and the currency used by the outright book defined by the leg. Example: Multileg quoted in EUR, outright leg in USD and 1 EUR = 0,7 USD then LegCurrecyRatio = 0.7<br>**OMX Comment:** Not in FIX. OMX requires an extension | |
| 20235 | SecListID | String | | Identifies a Security List message<br>**OMX Comment:** Not in FIX. OMX requires an extension | |
| 20236 | SecListDesc | String | | Name or description of a Security List<br>**OMX Comment:** Not in FIX. OMX requires an extension | |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| 20237 | SecListType | int | | Type of Security List<br>**OMX Comment:** Not in FIX. OMX requires an extension | **Value** / **Description**<br>0 Traded Security<br>1 Turnover List<br>2 Index population |
| 20238 | PartOfSecListID | String | | Indicates a higher level Security List that this list is part of<br>**OMX Comment:** Not in FIX. OMX requires an extension | |
| 20239 | MassActionType | int | | Specifies the type of action requested<br>**OMX Comment:** Not in FIX. The EEWG requires an extension | **Value** / **Description**<br>1 Suspend Orders<br>2 Release Orders from Suspension |
| 20240 | MassActionRequestScope | int | | Specifies scope of Order Mass Action Request.<br>**OMX Comment:** Not in FIX. The EEWG requires an extension | **Value** / **Description**<br>1 All orders for a security<br>2 All orders for an underlying security<br>3 All orders for a Product<br>4 All orders for a CFICode<br>5 All orders for a SecurityType<br>6 All orders for a trading session<br>7 All orders<br>8 All orders for a Market Segment |
| 20241 | MassActionResponse | int | | Specifies the action taken by counterparty order handling system as a result of the Order Mass Suspend or Release Request<br>**OMX Comment:** Not in FIX. The EEWG requires an extension | **Value** / **Description**<br>0 Action Request Rejected - See MassActionRejectReason (20242)<br>1 All orders for a security<br>2 All orders for an Underlying Security<br>3 All orders for a Product |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | **Value** / **Description**<br>4 – All orders for a CFICode<br>5 – All orders for a SecurityType<br>6 – All orders for a trading session<br>7 – All Orders<br>8 – All orders for a Market Segment |
| 20242 | MassActionRejectReason | int | | Reason Order Mass Suspend or Release Request was rejected<br>**OMX Comment:** Not in FIX. The EEWG requires an extension | **Value** / **Description**<br>0 – Mass Suspend / Release Not Supported<br>1 – Invalid or Unknown Security<br>2 – Invalid or Unkown Underlying security<br>3 – Invalid or Unknown Product<br>4 – Invalid or Unknown CFICode<br>5 – Invalid or Unknown SecurityType<br>6 – Invalid or Unknown Trading Session<br>7 – Invalid or unknown Market Segment<br>99 – Other |
| 20243 | MassSuspRelReportID | String | | Unique Identifier for the Order Mass Suspend or Release Report<br>**OMX Comment:** Not in FIX. The EEWG requires an extension | |
| 20244 | ListRejectReason | String | | Identifies the reason for rejection of a New Order List message. Note that OrdRejReason (103) is used if the rejection is based on properties of an individual order part of the List.<br>**OMX Comment:** Not in FIX. The EEWG requires an extension | **Value** / **Description**<br>0 – Broker / Exchange option<br>2 – Exchange closed<br>4 – Too late to enter<br>5 – Unknown order<br>6 – Duplicate Order (e.g. dupe ClOrdID) |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | | **Value** / **Description**<br>11 / Unsupported order characteristic<br>99 / Other |
| 20245 | TrdRepIndicator | Boolean | | Specifies whether the trade should be reported (or not) to parties of the provided TrdRepPartyRole (20088). Used to override standart reporting behavior by the receiver of the trade report and thereby complements the PublishTrdIndicator (852).<br>**OMX Comment:** Not in FIX. OMX requires an extension | **Value** / **Description**<br>0 / Do Not Report Trade<br>1 / Report Trade |
| 20246 | BookTransparency | int | | Defines the transparency of the book | **Value** / **Description**<br>0 / Default (as defined in reference data)<br>1 / No transparency<br>2 / As specified (in MDBookType [1021] and MarketDepth [264]) |
| 20247 | LegExecInst | MultipleCharValue | | Refer to ExecInst (18)<br>**OMX Comment:** Not in FIX. OMX requires an extension | |
| 20248 | NoPartyAltID | int | | Number of PartyAltID (20249) and PartyAltIDSource (20250) entries<br>**OMX Comment:** Not in FIX. OMX requires an extension | |
| 20249 | PartyAltID | String | | Alternate Party identifier/code. See PartyAltIDSource (20250). See "Appendix 6-G – Use of <Parties> Component Block"<br>**OMX Comment:** Not in FIX. OMX requires an extension | |
| 20250 | PartyAltIDSource | char | | Identifies class or source of the PartyAltID (20249) value. Required if PartyAltID is specified.<br>See "Appendix 6-G – Use of <Parties> Component Block"<br>**OMX Comment:** Not in FIX. OMX requires an extension | |
| 20251 | PartyListUpdateAction | char | | Specifies the action taken for the specified PartyID. | **Value** / **Description**<br>A / Add |

| Tag | FieldName | Type | OMXLen | Desc | Valid values |
|---|---|---|---|---|---|
| | | | | **OMX Comment:** Not in FIX. OMX requires an extension | **Value** / **Description**<br>D / Delete<br>M / Modify |