BINARY ENTRYPOINT

Message Reference – version 8.0.0.1

Last modified: November 20th, 2023

1 CHANGE LOG

Date	Version	Description	Author
November 29th, 2019	1.0	- Initial version2.1 e 2.11	AYSF, EP, RC
December 20th, 2019	2.0	- Added New Application-Level Messages.	RC, AYSF
January 31st, 2020	3.0	- Added Session Level Messages.	RC, AYSF
February 28th, 2020	4.0	- Offset, length, SimpleNewOrderSingle	RC, AYSF
April 3rd, 2020	4.1	- *SimpleNewOrderSingle* changed or message name to *SimpleNewOrder* - *SimpleReplaceOrder* changed or message name to *SimpleModifyOrder* - Changed size for tag *SecurityIDSource* and *SecurityIDExchange* - Added new scenarios in Message Instructions - Added tag 372 (*RefMsgType*) in *Execution Report - Reject* message - Changed tag 44 (*Price*) for size *uint32* - Changed tag 35502 (*EnteringTrader*) for size 5	RC, AYSF
May 5th, 2020	5.0	- Added brief descriptions for each message	RC, SAA, AYSF
October 2nd, 2020	5.2	- Added Fractional value on Composite Datatypes for UTCTimestampNanos, Percentage, PriceOptional, Price - Added Simple Open Framing Header (SOFH)	AYSF, RC, RDCF

Date	Version	Description	Author
		- Altered tag 35526 (nextSeqNo) in messages Establish, EstablishmentAck, Retransmission, Sequence and tag 35527 (lastIncomingSeqNo) in message Retransmission for size uint32 - Altered tag 35528 (lastSeqNo) in message FinishedSending for size uint32 - Altered tag 35529 (fromSeqNo) in messages NotApllied and RetransmitRequest for size uint32 - Altered tag 45 (refSeqNum) in message BusinessMessageReject for size uint32 - Excluded tag 17 (execID) in message ExecutionReport_New - Added tag 198 (secondaryOrderID) in messages ExecutionReport_Modify and Execution Report – Trade - Excluded tag 60 (transactTime) in message OrderMassActionRequest - Added tag 712 (posMaintAction) in messages PositionMaintenanceRequest and PositionMaintenanceReport - Added tag 35505 (ordTagID) in message SimpleModifyOrder	
December 15th, 2020	5.3	- Added tag 35539 (selfTradePreventionInstruction) in messages NewOrderSingle, ExecutionReportNew and OrderCancelReplaceRequest - Update description enum ExecRestatementReason – value 103 to Cancel Aggressor due to Self-Match Prevention, add description CANCEL_RESTING_DUE_TO_SELF_MATCH_PREVENTION value 107 and delete value 202. - Modify Message Length to uint16 - Simple Open Framing Header (SOFH)	AYSF, RC, RDCF
December 23rd, 2020	5.3.1	- Removed tags 35504 and 35507 for PositionMaintenanceCancelRequest, PositionMaintenanceRequest and AllocationInstruction	AYSF, RDCF
October 21st ,2022	5.8	- Changing tag 1115 (OrderCategory) possible values. - Tag 5149 (Memo) added to NewOrderSingle, OrderCancelReplaceRequest, OrderCancelRequest, and NewOrderCross. - Including tag 1300 (MarketSegmentID) to AllocationInstruction. - Tag 97 (PossResend) included for outgoing messages (ExecutionReports, PositionMaintenanceReport, AllocationReport, etc.) - MetricType enum and NoMetrics group removed - Tag 35543 (MarketSegmentReceivedTime) added to ExecutionReport_Reject message	RNKS, AYSF

Date	Version	Description	Author
		- Updating OrderMassActionRequest/OrderMassActionReport for new filter parameters for mass cancel - Adding new domains for tags 1115(OrderCategory), (35487) RoutingInstruction and (378) ExecRestatementReason to support various QDF modes and Mass Cancel - Adding side and account fields to SimpleModifyOrder message - Removing QuantityU64 and QuantityOptionalU64 type definition - Removing NewOrderSingle_U64, OrderCancelReplaceRequest_U64 and ExecutionReport_Trade_U64 messages - Changing Quantiy and QuantityOptional fields to uint64 type - Moving tag 35503 (SenderLocation) field from Negotiate to incoming messages - Including tag 434 (CxlRejResponseTo) field in the ExecutionReport_Reject - Removing tag 35502 (EnteringTraderm) field from PositionMaintenanceReport and AllocationReport - Includingtag (35551) ExternalRFQIndicator for cross/trade execution report to support RFQ System - Adjusting message layout to not have any fields to tranverse cache line (64 bytes including SBE header) - Changing tag 88(AllocRejCode3) field type from uint64 to uint32 - Changing tag 834 (ThresholdAmount) type to PriceOffsetOptional type (uint32 to uint64) - Domains of NegotiationRejectCode, EstablishRejectCode, TerminationCode, RetransmitRejectCode changed - Including SimpleOpenFramingHeader composite type for reference (and non-functional message to allow stub generation) - Minor fixes regarding message names	
October 31st, 2022	5.9	- Changing aligment in some messages and types of some enumeration types from uint8 to int8.	AEF
January 17th, 2023	6.0	- Changing aligment of the fields on all business messages, including repeating groups and variable length fields. - Removing the following values on enums: MassActionScope (20) and MassActionType (1 and 2). - Adding explicit msgSeqNum in all business messages.	AEF

Date	Version	Description	Author
		- Adding new TerminationRejectCode: INVALID_MSGSEQNUM. - Creation of InboundBusinessHeader, OutboundBusinessHeader and BidirectionalBusinessHeader, which contain common fields for - business messages. The fields marketSegmentId and possResend were moved to these headers. - Adding SendingTime in all business headers. - Attribute "presence" of several fields were either changed or made explicit. Some types had Optional versions created. - Attribute "nullValue" of several optional fields were made explicit. - Creation of SimpleTimeInForce and SimpleOrdType to limit which operations can be done with SimpleNewOrder and SimpleModifyOrder. - Side changed from uint8 to char. - Minimum changes in description of several fields. - Created field "currentSessionVerID" in NegotiateReject to allow reconnection after a catastrophic failure on client side. - Added mmProtectionReset and selfTradePreventionInstruction on SimpleNewOrder and SimpleModifyOrder. - Added origClOrdId in SimpleModifyOrder. - Changing enteringTrader and executingTrader to fixed string on all messages. - Added several new fields on all types of ExecutionReport.	
February 14th, 2023	6.1	- Several enum types changed from numeric to char to preserve FIX compatibility and assure 0 as nullValue. - Several enum types changed from uint8 to uint8EnumEncoding to assure 0 as nullValue. - Constant AccountType added to SimpleModifyOrder to avoid misunderstandings. - Removed secondaryOrderID from SimpleModifyOrder and OrderCancelReplaceRequest. - Removed BooleanOptional type: all boolean fields now are mandatory. - Re-ordered fields on SimpleNewOrder, SimpleModifyOrder, NewOrderSingle, OrderCancelReplaceRequest and OrderCancelRequest. - Added secondaryOrderId on OrderCancelRequest. - Added expected credentials format in description. - Adding crossedIndicator to NewOrderCross, ExecutionReport_New, ExecutionReport_Trade and ExecutionReport_Reject.	AEF

Date	Version	Description	Author
		- SecurityIDSource changed from constant to optional on SecurityDefinitionRequest. - Changed SecurityIDSource constant definition. - Adding missing description on some fields.	
February 23rd, 2023	6.2	- Template version changed back to 0. - SelfTradePreventionInstruction changed from char to uint8. - CxlRejResponseTo null value changed to 0. - Created type CrossIDOptional. - Padding from all business headers moved into the composite. - Re-alignment of fields of several messages. - When present, marketSegmentReceivedTime changed to optional on Execution Reports. - Removed origClOrdId from ExecutionReport_Cancel and ExecutionReport_Forward. - ExecutionReport_New: Added execID and crossID. - ExecutionReport_Reject: Added origClOrdId; orderID and cxlRejResponseTo changed to optional. - Added deskID to NewOrderCross, NewOrderSingle, OrderCancelReplaceRequest, OrderCancelRequest and Quote. - origClOrdID changed to optional on OrderCancelReplaceRequest, SimpleModifyOrder and OrderCancelRequest. - Added orderID on OrderCancelReplaceRequest, OrderCancelRequest and SimpleModifyOrder. - Removed secondaryOrderID from OrderCancelRequest. - enteringTrader changed to required on OrderCancelRequest. - legSecurityID and securityIDSource removed from SecurityDefinitionRequest. - Added memo to SimpleNewOrder and SimpleModifyOrder. - Added deskID to Quote.	AEF, RNKH
March 3rd, 2023	6.3	- SessionID included in all business headers. - NextSeqNo included in the NegotiateReject message. - FinishedSending message removed. - FinishedReceiving message removed. - Condition fields informed in the request removed in the OrderMassActionReport message. - Converting clOrdID field to required in ExecutionReport_Modify and ExecutionReport_Cancel messages.	LMG, RNKH

Date	Version	Description	Author
		- Including more details about Message Header and SBE Header in the documentation.	
March 22nd, 2023	6.3.1	- All UTF-8 fields changed to ASCII.	AEF, RNKH
March 27th, 2023	6.4	- Removed *nextSeqNo* field from *NegotiateReject* message. - Added *lastIncomingSeqNo* field to *EstablishReject* message. - Added precision warning to the description of fields that use UTCTimestampNanos type.	AEF
April 12th, 2023	6.4.1	- Added *execRestatementReason* field (of *ExecRestatementReasonValidForSingleCancel* type) in the *OrderCancelRequest* message. - Added new domain: NEW_ORDER_SINGLE to *CxlRejResponseTo* field. - Created new variations of *ExecRestatementReason* for different purposes. - "NEW" (0) value included (and label changed for other values) in the domain of *CxlRejResponseTo* enum if the reject is related to *NewOrderSingle/SimpleNewOrder/NewOrderCross* inbound messages. - Removed group noPositions in PositionMaintenanceRequest message and all related nested fields moved to body. - Removed group noAllocs in AllocationInstruction message and all related nested fields moved to body. - Optional presence attribute reviewed in the messages.	RNKH
April 25th, 2023	6.4.2	- Several order entry fields echoed for all types of *ExecutionReport*.	RNKH

Date	Version	Description	Author
June 13th, 2023	7.0.0	- Support for mass cancel on behalf: affected messages: *ExecutionReport_Cancel* and *OrderMassActionRequest/OrderMassActionReport*. - *side* field included in *OrderCancelRequest* message. - *origClOrdID* added to *ExecutionReport_Cancel* and *ExecutionReport_Modify*. - All filter fields from *OrderMassActionRequest* echo back to *OrderMassActionReport*. - *Presence of legside field became optional.* - *Presence of RatioQty type became required.* - *NullValue of PriceOffsetOptional type is not zero anymore.* - UTCTimestampNanosOptional type created for marketSegmentReceivedTime field. - Added new domain: DUPLICATE_SESSION_CONNECTION to NegotiationRejectCode. - Added new domain: AUTHENTICATION_IN_PROGRESS to NegotiationRejectCode. - Added new domain: DUPLICATE_SESSION_CONNECTION to EstablishRejectCode. - Added new domain: AUTHENTICATION_IN_PROGRESS to EstablishRejectCode. - Added new domain: INVALID_SESSIONID to TerminationCode. - Added new domain: INVALID_SESSIONVERID to TerminationCode. - Added new domain: INVALID_TIMESTAMP to TerminationCode. - Added new domain: INVALID_NEXTSEQNO to TerminationCode. - Added new domain: UNRECOGNIZED_MESSAGE to TerminationCode. - Added new domain: INVALID_SOFH to TerminationCode. - Added new domain: DECODING_ERROR to TerminationCode. - Added new domain: THROTTLE_REJECT to RetransmitRejectCode.	RNKH
June 13th, 2023	7.0.1	- QuoteRequestReject: orderQty type changed to QuantityOptional.	AEF
August 14th, 2023	7.1.0	- *routingInstruction* field included in the *SimpleNewOrder* and *SimpleModifyOrder* messages. - *investorID* composite field added at the end of root block in the *SimpleNewOrder*, *SimpleModifyOrder*, *OrderCancelReplaceRequest*, *NewOrderSingle* to replace deprecated variable length data investorID field (renamed to deprecatedInvestorID). - Encoded Length column changed to Block Length column in each message definition for easier understanding. - Two examples of *Binary Entrypoint* messages in hex dump for clarification purposes.	RNKH

Date	Version	Description	Author
September 18th, 2023	8.0.0	**BEWARE, THIS NEW SCHEMA IS IMCOMPATIBLE WITH THE OLDER ONES. PLEASE MAKE THE NECESSARY CHANGES IN YOUR SYSTEM BEFORE USING IN PRODUCTION ENVIRONMENT.** Message types impacted by this update: ***SimpleNewOrder***, ***SimpleModifyOrder***, ***NewOrderSingle***, ***OrderCancelReplaceRequest***, ***OrderMassActionRequest***, ***OrderMassActionReport*** and ***ExecutionReport_Cancel.*** - The *routingInstruction* field is relocated in the *SimpleNewOrder*, *SimpleModifyOrder*, *NewOrderSingle*, *OrderCancelReplaceRequest* messages, so that the first 18 fields share the same order/position (76 bytes) between these messages. - *accountType* field was reallocated to the same position as the previous position of the *routingInstruction* field in the *OrderCancelReplaceRequest* message for the reason described in the sentence above. - *deprecatedInvestorID* variable-length field removed: the correct one to inform investor's identification is *investorID* composite field. - *SelfTradePreventionInstruction* enum includes a new value: **NONE** (**0**) to not enable self-trade prevention mechanism even when used for mass cancel on behalf mechanism. - Added value PROTOCOL_VERSION_NOT_SUPPORTED to *NegotiationRejectCode, EstablishRejectCode and TerminationCode* enums. - Add value SYSTEM_BUSY to *RetransmitRejectCode* enum. - Therefore, the presence of *selfTradePreventionInstruction* field becomes required (*mass cancel on behalf mechanism will be available in the future*). - Several unused types/enum values/fields removed from *OrderMassActionRequest* and *OrderMassActionReport* messages. *investorID* composite field replaced those fields at the end of *OrderMassActionRequest* and *OrderMassActionReport* messages for mass cancel on behalf purposes (*mass cancel on behalf mechanism will be available in the future*).	RNKH
November 20th, 2023	8.0.0.1	- Range of count field in *RetransmitRequest* redefined between 1 to 1000 (inclusive) in section 7.9. - Definition of prefix redefined in the *InvestorID* composite type in section 8.24. - Corrected *CrossedIndicator* null value declaration to zero (error was only in the specification document) in section 9. - Presence of *credentials* field (tag 35512) changed to “required” in *Negotiate* and *Establish* messages.	RNKH

2 SIMPLE BINARY ENCODING

EntryPoint Simple Binary Encoding (SBE) is based on the version 1.0 of the FIX SBE in Little Endian byte order. FIX SBE targets high performance trading systems. It is optimized for low latency of encoding and decoding while keeping bandwidth utilization reasonably small. For compatibility, it is intended to represent all FIX semantics. The encoding standard is complimentary to other FIX standards for session protocol and application-level behavior.

Each message in the packet starts with a Message Header that consists of the Framing Header and the SBE Message Header. The message header is in little-endian format (the least significant values come first). The total size is 12 bytes.

Framing Header is based on **Simple Open Framing Header (SOFH)**, but *MessageLength* field occupy only 2 bytes (support maximum message size of 16384 bytes): https://www.fixtrading.org/standards/fix-sofh/

The encoding type is SBE 1.0 little-endian = 0xEB50 (Because the header is in little endianness: 0x50, 0xEB in the wire).

Message Header has the following fields:

Name	Type	Size (bytes)	Description
(Framing Header)			
MessageLength	uint16	2	Overall message length including headers to support framing.
EncodingType	uint16	2	Identifier of the encoding used in the message payload (always “50 EB – SBE 1.0 Little-Endian)
(SBE Message Header)			
BlockLength	uint16	2	The total space reserved for the root level of the message not counting any repeating groups or variable-length fields.
TemplateID	uint16	2	Identifier of the message template.
SchemaID	uint16	2	Identifier of the message schema that contains the template.
SchemaVersion	uint16	2	The version of the message schema in which the message is defined.

2.1 Example – *Establish* message with credentials

Here is an example of a HEX Dump of a complete *Establish* message with credentials in one TCP packet for better understanding on how to work with fixed root block and one variable-length data (access key changed for security reasons):

```
          +-------------------------------------------------+
          |  0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f |
+---------+-------------------------------------------------+----------------+
|00000000| 8c 00 50 eb 2a 00 04 00 01 00 02 00 01 e1 f5 05 |..P.*...........|
|00000010| 66 70 f3 1c 89 01 00 00 40 ce 48 9a 01 6e 6e 17 |fp......@.H..nn.|
|00000020| 60 ea 00 00 00 00 00 00 01 00 00 00 03 00 f4 01 |`...............|
|00000030| 00 00 00 00 00 00 55 7b 20 20 20 22 61 75 74 68 |......U{   "auth|
|00000040| 5f 74 79 70 65 22 3a 20 22 62 61 73 69 63 22 2c |_type": "basic",|
|00000050| 20 20 20 22 75 73 65 72 6e 61 6d 65 22 3a 20 22 |   "username": "|
|00000060| 31 30 30 30 30 30 30 30 31 22 2c 20 20 20 22 61 |100000001",   "a|
|00000070| 63 63 65 73 73 5f 6b 65 79 22 3a 20 22 31 32 33 |ccess_key": "123|
|00000080| 34 35 36 37 38 39 41 42 43 22 20 7d              |456789ABC" }....|
+--------+-------------------------------------------------+----------------+
```

Offset	Length	Field	Hex bytes	Decoded value
0000	2	messageLength	8c 00	0x008c = 140
0002	2	encodingType	50 eb	0xeb50 = SBE 1.0 Little-Endian
0004	2	blockLength	2a 00	0x002a = 42
0006	2	templateID	04 00	0x0004 = 4 (Establish)
0008	2	schemaID	01 00	0x0001 = 1 (Schema ID)
000A	2	schemaVersion	02 00	0x0002 = 2 (Schema Version)
000C … 0035	42	SBE Message Root Block	01 e1 f5 05 …	
0036	1	CredentialsEncoding.length	55	0x55 = 85 (credentials length)
0037 ... 008B	85	CredentialsEncoding.varData	7b 20 20 20 ...	credentials

2.2 Example – *SimpleNewOrder* message

Here is an example of a HEX Dump of a complete *SimpleNewOrder* message in one TCP packet for better understanding on how to work with fixed root block (with *investorID* field filled) and one variable-length data filled (memo field):

```
          +-------------------------------------------------+
          |  0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f |
+--------+-------------------------------------------------+----------------+
|00000000| 75 00 50 eb 54 00 64 00 01 00 02 00 01 e1 f5 05 |..P.U.d.........|
|00000010| 05 00 00 00 80 11 49 0a 04 6e 6e 17 50 00 01 00 |......I..nn.P...|
|00000020| 6b 70 f3 1c 89 01 00 00 0f 00 00 00 54 41 44 41 |kp..........TADA|
|00000030| 00 00 00 00 00 00 54 41 44 41 00 00 55 4f f0 90 |......TADA..UO..|
|00000040| 2e 00 00 00 31 32 30 00 64 00 00 00 00 00 00 00 |....120.d.......|
|00000050| 08 43 0f 00 00 00 00 00 2C 01 00 00 40 E2 01 00 |.C......,...@...|
|00000060| 14 53 49 4d 50 4c 45 4e 45 57 4f 52 44 45 52 20 |.SIMPLENEWORDER |
|00000070| 42 55 59 20 35                                  |BUY 5...........|
+--------+-------------------------------------------------+----------------+
```

Offset	Length	Field	Hex bytes	Decoded value
0000	2	messageLength	76 00	0x0075 = 117
0002	2	encodingType	50 eb	0xeb50 = SBE 1.0 Little-Endian
0004	2	blockLength	54 00	0x0054 = 84
0006	2	templateID	64 00	0x0064 = 100 (SimpleNewOrder)
0008	2	schemaID	01 00	0x0001 = 1 (Schema ID)
000A	2	schemaVersion	02 00	0x0002 = 2 (Schema Version)
000C … 0060	84	SBE Message Root Block	01 e1 f5 05 …	
0062	1	MemoEncoding.length	14	0x14 = 20 (memo length)
0063 ... 0076	20	MemoEncoding.varData	53 49 4d 50 ...	memo

3 SCHEMA

Name	Description	Semantic version	Schema Version	Package	Byte Order
b3-entrypoint-messages.xml	SBE Template for Entrypoint Order Entry Messages	8.0.0	2	b3.entrypoint.fixp.sbe	Little Endian

4 TYPES

Null values are present in all types but should be used only for optional fields.

Name	Data Type	Size	Version	Null Value	Description
Account	uint32	4	0	0xFFFFFFFF	Account mnemonic.
AccountOptional	uint32	4	0	0	Optional account mnemonic.
AllocID	uint64	8	0	0xFFFFFFFFFFFFFFFF	Unique identifier for this allocation instruction message.
AllocReportID	uint64	8	0	0xFFFFFFFFFFFFFFFF	Unique identifier for this allocation report message.
AssetOptional	char	6	0	0	Asset associated with the security, such as DOL, BGI, OZ1, WDL, CNI, etc.

Name	Data Type	Size	Version	Null Value	Description
blockLength	uint16	2	0	0xFFFF	Root block length.
BusinessRejectRefID	uint64	8	0	0	Value of business-level identification field on the message being referenced.
ClOrdID	uint64	8	0	0xFFFFFFFFFFFFFFFF	Unique identifier of the order as assigned by the market participant.
ClOrdIDOptional	uint64	8	0	0	Optional unique identifier of the order as assigned by the market participant.
CrossID	uint64	8	0	0xFFFFFFFFFFFFFFFF	Identifier for a cross order. Must be unique during a given trading day.
CrossIDOptional	uint64	8	0	0	Identifier for a cross order.
custodian	uint32	4	0	0	Identifies the custodian.
custodyAccount	uint32	4	0	0	Identifies the custody account.
custodyAllocationType	uint32	4	0	0	Custody allocation type.

Name	Data Type	Size	Version	Null Value	Description
DaysToSettlement	uint16	2	0	0xFFFF	Deadline for completing the forward deal
DaysToSettlementOptional	uint16	2	0	0xFFFF	Optional deadline for completing the forward deal
encodingType	uint16	2	0	0xFFFF	Encoding type of payload.
ExecID	uint64	8	0	0xFFFFFFFFFFFFFFFF	Unique identifier of execution message as assigned by exchange.
ExecIDOptional	uint64	8	0	0	Optional unique identifier of execution message as assigned by exchange.
exponent	int8	0	0	N/A	Exponent (for fixed-point decimal numbers).
Firm	uint32	4	0	0xFFFFFFFF	Identification of the broker firm.
FirmOptional	uint32	4	0	0	Optional identification of the broker firm.
length	uint8	1	0	0xFF	Length of free format text string generated by exchange.

Name	Data Type	Size	Version	Null Value	Description
LocalMktDate	uint16	2	0	0xFFFF	Local calendar date: days since Unix epoch (January 1st, 1970).
LocalMktDateOptional	uint16	2	0	0	Local calendar date: days since Unix epoch (January 1st, 1970).
mantissa	int64	8	0	0	Mantissa (for fixed-point decimal numbers).
marketSegmentID	uint8	1	0	0xFF	Identifies the market segment.
MassActionReportID	uint64	8	0	0xFFFFFFFFFFFFFFFF	Unique ID of Order Mass Action Report as assigned by the matching engine.
MassActionReportIDOptional	uint64	8	0	0xFFFFFFFFFFFFFFFF	Optional unique ID of Order Mass Action Report as assigned by the matching engine.
MessageCounter	uint32	4	0	0xFFFFFFFF	Counter of related messages.
messageLength	uint16	2	0	0xFFFF	Message length (including Framing and SBE headers).
msgSeqNum	uint32	4	0	0xFFFFFFFF	Sequence number of a given SessionID/SessionVerID.

Name	Data Type	Size	Version	Null Value	Description
numInGroup	uint8	1	0	0xFF	Counter representing the number of entries in a repeating group.
OrderID	uint64	8	0	0xFFFFFFFFFFFFFFFF	Exchange-generated order identifier.
OrderIDOptional	uint64	8	0	0	Optional exchange-generated order identifier.
OrdTagID	uint8	1	0	0	Identifies the order tag identification.
padding	char	1	0	0	Padding for alignment purpose.
PosMaintRptID	uint64	8	0	0xFFFFFFFFFFFFFFFF	Unique identifier for this position maintenance report message.
PosMaintRptIDOptional	uint64	8	0	0	Optional unique identifier for this position maintenance report message.
PosReqID	uint64	8	0	0xFFFFFFFFFFFFFFFF	Unique identifier for the position maintenance request.
PosReqIDOptional	uint64	8	0	0	Optional unique identifier for the position maintenance request.

Name	Data Type	Size	Version	Null Value	Description
Quantity	uint64	8	0	0xFFFFFFFFFFFFFFFF	Quantity in order/trade.
QuantityOptional	uint64	8	0	0	Optional quantity in order/trade.
QuoteID	uint64	8	0	0xFFFFFFFFFFFFFFFF	Unique identifier for quote.
QuoteIDOptional	uint64	8	0	0	Optional unique identifier for quote.
RejReason	uint32	4	0	0xFFFFFFFF	Code to identify reason for order rejection. Please refer to the error codes document for domain information.
RejReasonOptional	uint32	4	0	0	Optional code to identify reason for order rejection. Please refer to the error codes document for domain information.
SecurityExchange	char	0	0	N/A	Market Identifier Code: 4-character used to identify stock markets and other trading exchanges.
SecurityGroup	char	3	0	N/A	Indicates the group this instrument belongs to.

Name	Data Type	Size	Version	Null Value	Description
SecurityID	uint64	8	0	0xFFFFFFFFFFFFFFFF	Security identification as defined by exchange. For the SecurityID list, see the Security List/Definition message in Market Data feed.
SecurityIDOptional	uint64	8	0	0	Optional security identification as defined by exchange. For the SecurityID list, see the Security List/Definition message in Market Data feed.
SecurityReqRespID	uint64	8	0	0xFFFFFFFFFFFFFFFF	Unique ID of a Security Definition Request/Response.
SecurityStrategyType	char	3	0	N/A	Indicates the type of Strategy created.
SenderLocation	char	10	0	N/A	Identifies the original location for routing orders.
SeqNum	uint32	4	0	0xFFFFFFFF	Sequence number of a given SessionID/SessionVerID.
SeqNumOptional	uint32	4	0	0	Optional sequence number of a given SessionID/SessionVerID.
SessionID	uint32	4	0	0xFFFFFFFF	Client connection identification on the gateway assigned by B3.
SessionIDOptional	uint32	4	0	0	Optional client connection identification on the gateway assigned by B3.

Name	Data Type	Size	Version	Null Value	Description
SessionVerID	uint64	8	0	0xFFFFFFFFFFFFFFFF	Session version identification: unique identification of a sequence of messages to be transmitted to exchange gateway associated with given SessionId.
SessionVerIDOptional	uint64	8	0	0	Optional session version identification: unique identification of a sequence of messages to be transmitted to exchange gateway associated with given SessionId.
Symbol	char	20	0	N/A	B3 requires that this field is properly set. It contains the human readable form of the SecurityID tag ans is also available in the Security List message in Market Data feed.
time	uint64	8	0	0	UTC timestamp with nanosecond precision (Unix Epoch).
TotNoRelatedSym	uint8	1	0	0	Number of leg fill notice messages sent with spread summary.
TradeID	uint32	4	0	0xFFFFFFFF	The unique identification assigned to the trade entity once it is received or matched by the exchange or central counterparty.

Name	Data Type	Size	Version	Null Value	Description
TradeIDOptional	uint32	4	0	0	Optional unique identification assigned to the trade entity once it is received or matched by the exchange or central counterparty.
Trader	char	5	0	N/A	Identification of the trader.
TraderOptional	char	5	0	N/A	Optional identification of the trader.
unit	uint8	0	0	N/A	time unit (nanoseconds).
varData	char	N	0	N/A	Free ASCII format text string generated by exchange.

5 SESSION-LEVEL MESSAGES

Binary EntryPoint uses the FIX Performance (FIXP) protocol to establish and manage bi-directional sessions. Per the FIXP protocol, a FIX session is defined as a bi-directional stream of ordered messages between two parties within a continuous sequence-number series.

Binary EntryPoint uses the FIX Performance (FIXP) protocol to establish and manage bi-directional sessions. FIXP session messages are summarized as follows:

Stage	Msg Sent (Client to B3)	Msg Received (B3 to Client)	Purpose
Initialization	Negotiate	-	Initiates Connection
	-	NegotiateResponse	Accepts Connection
	-	NegotiateReject	Rejects Connection
Binding	Establish	-	Binds Connection
	-	EstablishAck	Accepts Binding
	-	EstablishReject	Rejects Binding
Transferring	Sequence	-	Initiates a Sequenced Flow
	-	Sequence	(Keep-Alive)
	RetransmitRequest	-	Requests Replay
	-	Retransmission	Accepts Replay
	-	RetransmitReject	Rejects Replay
	-	NotApplied	Negative Acknowledgement of Missed Messages
Unbinding	Terminate	Terminate	Kill Connection Ungracefully or gracefully

6 APPLICATION-LEVEL MESSAGES

MessageType	TemplateID	Application Message	From B3 to Client	From Client to B3
14	206	BusinessMessageReject	✓	
15	100	SimpleNewOrder		✓
16	101	SimpleModifyOrder		✓
17	102	NewOrderSingle		✓
18	104	OrderCancelReplaceRequest		✓
19	105	OrderCancelRequest		✓
20	106	NewOrderCross		✓
21	200	ExecutionReport - New Order/Restated	✓	
22	201	ExecutionReport - Modify	✓	
23	202	ExecutionReport - Cancel	✓	
25	204	ExecutionReport - Reject	✓	
24	203	ExecutionReport - Trade	✓	
26	205	ExecutionReport - Forward	✓	
27	300	SecurityDefinitionRequest		✓
28	301	SecurityDefinition	✓	
31	401	QuoteRequest	✓	✓
32	402	QuoteStatusReport	✓	
33	403	Quote	✓	✓
34	404	QuoteCancel	✓	✓

35	405	QuoteRequestReject	✓	✓
36	501	PositionMaintenanceCancelRequest		✓
37	502	PositionMaintenanceRequest		✓
38	503	PositionMaintenanceReport	✓	
39	601	AllocationInstruction		✓
40	602	AllocationReport	✓	
29	701	OrderMassActionRequest		✓
30	702	OrderMassActionReport	✓	

7 MESSAGE LAYOUTS

Null values are present in all types but should be used only for optional fields. Block Length is the size of the root block of the message.

7.1 Message: Negotiate

Name	Template ID	Version	Block Length	Description
Negotiate	1	2	28	The client sends the Negotiate message to B3 to initiate a connection. Negotiate is the first message that the client must sent to start the communication between client and gateway through a TCP socket connection.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (uint8)		Message type = Negotiate. Constant: 0
35518	sessionID	R	SessionID (uint32)	0 (4)	Client connection identification on the gateway assigned by B3.
35519	sessionVerID	R	SessionVerID (uint64)	4 (8)	Session version identification: unique identification of a sequence of messages to be transmitted to/from B3´s gateways associated with given SessionId. Need to be incremented each time Negotiate message is sent to gateway.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35520	timestamp	R	UTCTimestampNanos	12 (8)	Time of request. Sent in number of nanoseconds since Unix epoch.
35516	clientFlow	C	FlowType Enum (uint8)		Type of flow from client to server. It is a constant value = idempotent. Constant: 3
35501	enteringFirm	R	Firm (uint32)	20 (4)	Identifies the broker firm that will enter orders.
35517	onbehalfFirm	O	FirmOptional (uint32)	24 (4)	Identifies the broker firm who executes their orders on behalf.
35512	credentials	R	CredentialsEncoding		Credentials to identify/authenticate the client. The format is a JSON with the following data: { "auth_type": "basic", "username": "session_id", "access_key": "somepassword" }.
35513	clientIP	O	ClientAppEncoding		Source IP from client system.
35514	clientAppName	O	ClientAppEncoding		Application name as informed during certification process.
35515	clientAppVersion	O	ClientAppEncoding		Application version as informed during certification process.

7.2 Message: NegotiateResponse

Name	Template ID	Version	Block Length	Description
NegotiateResponse	2	2	24	The NegotiationResponse message is sent when a Negotiate message from the client is accepted by B3.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (uint8)		Message type = NegotiateResponse. Constant: 1
35518	sessionID	R	SessionID (uint32)	0 (4)	Client connection identification on the gateway assigned by B3.
35519	sessionVerID	R	SessionVerID (uint64)	4 (8)	Session version identification: unique identification of a sequence of messages to be transmitted to/from B3´s gateways associated with given SessionId. Need to be incremented each time Negotiate message is sent to gateway.
35521	requestTimestamp	R	UTCTimestampNanos	12 (8)	Matches Negotiate timestamp.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35524	serverFlow	C	FlowType Enum (uint8)		Type of flow from client to server. It is a constant value = recoverable. Constant: 1
35516	clientFlow	C	FlowType Enum (uint8)		Type of flow from client to server. It is a constant value = idempotent. Constant: 3
35501	enteringFirm	R	Firm (uint32)	20 (4)	Identifies the broker firm that will enter orders.

7.3 Message: NegotiateReject

Name	Template ID	Version	Block Length	Description
NegotiateReject	3	2	36	NegotiateReject message is sent when B3 rejects a Negotiate message sent by the client.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (uint8)		Message type = NegotiateReject. Constant: 2
35518	sessionID	R	SessionID (uint32)	0 (4)	Client connection identification on the gateway assigned by B3.
35519	sessionVerID	R	SessionVerID (uint64)	4 (8)	Session version identification: unique identification of a sequence of messages to be transmitted to/from B3´s gateways associated with given SessionId. Need to be incremented each time Negotiate message is sent to gateway.
35521	requestTimestamp	R	UTCTimestampNanos	12 (8)	Matches Negotiate timestamp.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35516	clientFlow	C	FlowType Enum (uint8)		Type of flow from client to server. It is a constant value = idempotent. Constant: 3
35501	enteringFirm	O	FirmOptional (uint32)	20 (4)	Identifies the broker firm that will enter orders.
35522	negotiationRejectCode	R	NegotiationRejectCode Enum (uint8)	24 (1)	Identifies the code of reject negotiation.
	<padding>			25 (3)	*
35523	currentSessionVerID	O	SessionVerIDOptional (uint64)	28 (8)	The current sessionVerID informed at the first Negotiate message for that specific session.

7.4 Message: Establish

Name	Template ID	Version	Block Length	Description
Establish	4	2	42	After negotiation level is complete, the client must send an Establish message to start assigning sequence numbers and also to keep the connection active. Once the connection is established, client can submit orders and quotes.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (uint8)		Message type = Establish. Constant: 3
35518	sessionID	R	SessionID (uint32)	0 (4)	Client connection identification on the gateway assigned by B3.
35519	sessionVerID	R	SessionVerID (uint64)	4 (8)	Session version identification: unique identification of a sequence of messages to be transmitted to/from B3´s gateways associated with given SessionId. Need to be incremented each time Negotiate message is sent to gateway.
35520	timestamp	R	UTCTimestampNanos	12 (8)	Time of request. Sent in number of nanoseconds since Unix epoch.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35525	keepAliveInterval	R	DeltaInMillis	20 (8)	Longest time in milliseconds the initiator should remain silent before sending Sequence message. It should be in the range of 1000 to 60000 (inclusive).
35526	nextSeqNo	R	SeqNum (uint32)	28 (4)	The next application sequence number to be produced by the client.
35002	cancelOnDisconnectType	R	CancelOnDisconnectType Enum (uint8)	32 (1)	Criteria used to initiate cancel on disconnect mechanism by the gateway.
	<padding>			33 (1)	*
35003	codTimeoutWindow	R	DeltaInMillis	34 (8)	Gateway will not trigger CoD if the customer reconnects within the timeout window (milliseconds) which starts when the triggering event is detected. Range is 0 (as soon as possible) to 60000.
35512	credentials	R	CredentialsEncoding		Credentials to identify/authenticate the client. The format is a JSON with the following data: { "auth_type": "basic", "username": "session_id", "access_key": "somepassword" }.

7.5 Message: EstablishAck

Name	Template ID	Version	Block Length	Description
EstablishAck	5	2	36	The EstablishmentAck message is sent when an Establish message is accepted by B3. EstablishmentAck message contains next sequence number. At the start of a session, default value is 1 (one).

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (uint8)		Message type = EstablishAck. Constant: 4
35518	sessionID	R	SessionID (uint32)	0 (4)	Client connection identification on the gateway assigned by B3.
35519	sessionVerID	R	SessionVerID (uint64)	4 (8)	Session version identification: unique identification of a sequence of messages to be transmitted to/from B3´s gateways associated with given SessionId. Need to be incremented each time Negotiate message is sent to gateway.
35521	requestTimestamp	R	UTCTimestampNanos	12 (8)	Matches Negotiate timestamp.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35525	keepAliveInterval	R	DeltaInMillis	20 (8)	Longest time in milliseconds the initiator should remain silent before sending Sequence message.
35526	nextSeqNo	R	SeqNum (uint32)	28 (4)	The next application sequence number to be produced by the gateway.
35527	lastIncomingSeqNo	R	SeqNum (uint32)	32 (4)	Indicates the application sequence number of the last application message received by the server from the client. At the start of a session, default value is 0 (zero).

7.6 Message: EstablishReject

Name	Template ID	Version	Block Length	Description
EstablishReject	6	2	26	EstablishmentReject message is sent when an Establish message is rejected by B3 informing the reason of rejection.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (uint8)		Message type = EstablishReject. Constant: 5
35518	sessionID	R	SessionID (uint32)	0 (4)	Client connection identification on the gateway assigned by B3.
35519	sessionVerID	R	SessionVerID (uint64)	4 (8)	Session version identification: unique identification of a sequence of messages to be transmitted to/from B3´s gateways associated with given SessionId. Need to be incremented each time Negotiate message is sent to gateway.
35521	requestTimestamp	R	UTCTimestampNanos	12 (8)	Matches Negotiate timestamp.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35532	establishmentRejectCode	R	EstablishRejectCode Enum (uint8)	20 (1)	Identifies the code of reject establishment.
	<padding>			21 (1)	*
35527	lastIncomingSeqNo	O	SeqNumOptional (uint32)	22 (4)	If establishmentRejectCode = EstablishRejectCode.**INVALID_NEXTSEQNO**, indicates the application sequence number of the last application message received by the server from the client. At the start of a session, default value is 0 (zero).

7.7 Message: Sequence

Name	Template ID	Version	Block Length	Description
Sequence	9	2	4	Sequence message specifies the sequence number of the next business message both: Recoverable (B3 to client) and Idempotent (client to B3) flows. It is also used as heartbeat.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (uint8)		Message type = Sequence. Constant: 13
35526	nextSeqNo	R	SeqNum (uint32)	0 (4)	The next application sequence number to be produced by the client.

7.8 Message: NotApplied

Name	Template ID	Version	Block Length	Description
NotApplied	8	2	8	NotApplied message is sent when B3 detects messages that already been sent (concept of idempotence) or an invalid message format from the client.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (uint8)		Message type = NotApplied. Constant: 9
35529	fromSeqNo	R	SeqNum (uint32)	0 (4)	The first not applied sequence number.
35530	count	R	MessageCounter (uint32)	4 (4)	How many messages haven´t been applied?

7.9 Message: RetransmitRequest

Name	Template ID	Version	Block Length	Description
RetransmitRequest	12	2	20	RetransmitRequest message is used for client to recover missed messages.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (uint8)		Message type = RetransmitRequest. Constant: 10
35518	sessionID	R	SessionID (uint32)	0 (4)	Client connection identification on the gateway assigned by B3.
35520	timestamp	R	UTCTimestampNanos	4 (8)	Time of request. Sent in number of nanoseconds since Unix epoch.
35529	fromSeqNo	R	SeqNum (uint32)	12 (4)	The first applied sequence number.
35530	count	R	MessageCounter (uint32)	16 (4)	Maximum number of messages to be retransmitted. Range is 1 to 1000 (inclusive).

7.10 Message: Retransmission

Name	Template ID	Version	Block Length	Description
Retransmission	13	2	20	Retransmission message is sent when a RetransmitRequest message from the client is accepted by B3.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (uint8)		Message type = Retransmission. Constant: 11
35518	sessionID	R	SessionID (uint32)	0 (4)	Client connection identification on the gateway assigned by B3.
35521	requestTimestamp	R	UTCTimestampNanos	4 (8)	Matches Negotiate timestamp.
35526	nextSeqNo	R	SeqNum (uint32)	12 (4)	The sequence number of the first retransmitted message.
35530	count	R	MessageCounter (uint32)	16 (4)	Number of messages to be retransmitted.

7.11 Message: RetransmitReject

Name	Template ID	Version	Block Length	Description
RetransmitReject	14	2	13	RetransmitReject message is sent when a RetransmitRequest message is rejected by B3. More details are described in the Message Specification Guidelines document.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (uint8)		Message type = RetransmitReject. Constant: 12
35518	sessionID	R	SessionID (uint32)	0 (4)	Client connection identification on the gateway assigned by B3.
35521	requestTimestamp	R	UTCTimestampNanos	4 (8)	Matches Negotiate timestamp.
35534	retransmitRejectCode	R	RetransmitRejectCode Enum (uint8)	12 (1)	Identifies the code of reject retransmission.

7.12 Message: Terminate

Name	Template ID	Version	Block Length	Description
Terminate	7	2	13	Terminate message is sent to indicate that the sender is going to disconnect the TCP socket connection.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (uint8)		Message type = Terminate. Constant: 6
35518	sessionID	R	SessionID (uint32)	0 (4)	Client connection identification on the gateway assigned by B3.
35519	sessionVerID	R	SessionVerID (uint64)	4 (8)	Session version identification: unique identification of a sequence of messages to be transmitted to/from B3´s gateways associated with given SessionId. Need to be incremented each time Negotiate message is sent to gateway.
35533	terminationCode	R	TerminationCode Enum (uint8)	12 (1)	Identifies the code of termination.

7.13 Message: SimpleNewOrder

Name	Template ID	Version	Block Length	Description
SimpleNewOrder	100	2	84	SimpleNewOrder message submits a simple new order focused on sent only main parameters with low complexity. Used by client to enter a simple order in the system.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (uint8)		Message type = SimpleNewOrder. Constant: 15
35524	businessHeader	R	InboundBusinessHeader	0 (18)	Common header to all inbound business messages.
35505	ordTagID	O	OrdTagID (uint8)	18 (1)	Identifies the order tag identification.
9773	mmProtectionReset	R	Boolean Enum (uint8)	19 (1)	Resets Market Protections. When Market Protections are triggered, the Exchange will not accept new orders for that product group without tag MMProtectionReset: True = Reset Market Maker Protection; False = Do nothing related to Market Maker Protection.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
11	clOrdID	R	ClOrdID (uint64)	20 (8)	Unique identifier of the order as assigned by the market participant.
1	account	O	AccountOptional (uint32)	28 (4)	Account mnemonic of the order.
35503	senderLocation	R	SenderLocation (char)	32 (10)	Identifies the original location for routing orders.
35502	enteringTrader	R	Trader (char)	42 (5)	Identifies the trader who is inserting an order.
35539	selfTradePreventionInstruction	R	SelfTradePrevention Instruction Enum (uint8)	47 (1)	Indicates which order should be canceled due to Self-Trade Prevention.
48	securityID	R	SecurityID (uint64)	48 (8)	Security identification as defined by exchange.
22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID (Exchange Symbol). Constant: "8"

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
207	securityExchange	C	SecurityExchange (char)		Market to which the symbol belongs. Constant: "BVMF"
54	side	R	Side Enum (char)	56 (1)	Side of order.
40	ordType	R	SimpleOrdType Enum (char)	57 (1)	Order type.
59	timeInForce	R	SimpleTimeInForce Enum (char)	58 (1)	Specifies how long the order remains in effect.
35487	routingInstruction	O	RoutingInstruction Enum (uint8)	59 (1)	Indicates additional order instruction.
38	orderQty	R	Quantity (uint64)	60 (8)	Quantity ordered.
44	price	O	PriceOptional	68 (8)	Price per share or contract. Conditionally required if the order type requires a price (not market orders and RLP).
35508	investorID	O	InvestorID	76 (8)	Unique identifier of investor for self-trade prevention/mass cancel on behalf purposes.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
5149	memo	O	MemoEncoding		Free UTF-8 format text field. This field may be used to convey client's relevant info. It's always echoed in the reports.

7.14 Message: SimpleModifyOrder

Name	Template ID	Version	Block Length	Description
SimpleModifyOrder	101	2	100	The SimpleModifyOrder submits a simple modify request for basic parameters like price and quantity. The client sends the SimpleModifyOrder message to B3 to modify some order values only.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (uint8)		Message type = SimpleModifyOrder. Constant: 16
35524	businessHeader	R	InboundBusinessHeader	0 (18)	Common header to all inbound business messages.
35505	ordTagID	O	OrdTagID (uint8)	18 (1)	Identifies the order tag identification.
9773	mmProtectionReset	R	Boolean Enum (uint8)	19 (1)	Resets Market Protections. When Market Protections are triggered, the Exchange will not accept new orders for that product group without tag MMProtectionReset: True = Reset Market Maker Protection; False = Do nothing related to Market Maker Protection.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
11	clOrdID	R	ClOrdID (uint64)	20 (8)	Unique identifier of the order as assigned by the market participant.
1	account	O	AccountOptional (uint32)	28 (4)	Account mnemonic of the order.
35503	senderLocation	R	SenderLocation (char)	32 (10)	Identifies the original location for routing orders.
35502	enteringTrader	R	Trader (char)	42 (5)	Identifies the trader who is inserting an order.
35539	selfTradePreventionInstruction	R	SelfTradePrevention Instruction Enum (uint8)	47 (1)	Indicates which order should be canceled due to Self-Trade Prevention.
48	securityID	R	SecurityID (uint64)	48 (8)	Security identification as defined by exchange.
22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID (Exchange Symbol). Constant: "8"

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
207	securityExchange	C	SecurityExchange (char)		Market to which the symbol belongs. Constant: "BVMF"
54	side	R	Side Enum (char)	56 (1)	Side of order.
40	ordType	R	SimpleOrdType Enum (char)	57 (1)	Order type.
59	timeInForce	R	SimpleTimeInForce Enum (char)	58 (1)	Specifies how long the order remains in effect.
35487	routingInstruction	O	RoutingInstruction Enum (uint8)	59 (1)	Indicates additional order instruction.
38	orderQty	R	Quantity (uint64)	60 (8)	Quantity ordered.
44	price	O	PriceOptional	68 (8)	Price per share or contract. Conditionally required if the order type requires a price (not market orders and RLP).

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
37	orderID	O	OrderIDOptional (uint64)	76 (8)	Unique identifier for order as assigned by the exchange.
41	origClOrdID	O	ClOrdIDOptional (uint64)	84 (8)	ClOrdID which should be replaced.
35508	investorID	O	InvestorID	92 (8)	Unique identifier of investor for self-trade prevention/mass cancel on behalf purposes.
581	accountType	C	AccountType Enum (uint8)		Type of account associated with an order. Constant: 39
5149	memo	O	MemoEncoding		Free UTF-8 format text field. This field may be used to convey client's relevant info. It's always echoed in the reports.

7.15 Message: NewOrderSingle

Name	Template ID	Version	Block Length	Description
NewOrderSingle	102	2	127	NewOrderSingle message is used to enter an order in the system; the behavior of an order can be affected by many parameters such as order type and order type qualifier.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (uint8)		Message type = NewOrderSingle. Constant: 17
35524	businessHeader	R	InboundBusinessHeader	0 (18)	Common header to all inbound business messages.
35505	ordTagID	O	OrdTagID (uint8)	18 (1)	Identifies the order tag identification.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
9773	mmProtectionReset	R	Boolean Enum (uint8)	19 (1)	Resets Market Protections. When Market Protections are triggered, the Exchange will not accept new orders for that product group without tag MMProtectionReset: True = Reset Market Maker Protection; False = Do nothing related to Market Maker Protection.
11	clOrdID	R	ClOrdID (uint64)	20 (8)	Unique identifier of the order as assigned by the market participant.
1	account	O	AccountOptional (uint32)	28 (4)	Account mnemonic of the order.
35503	senderLocation	R	SenderLocation (char)	32 (10)	Identifies the original location for routing orders.
35502	enteringTrader	R	Trader (char)	42 (5)	Identifies the trader who is inserting an order.
35539	selfTradePreventionInstruction	R	SelfTradePrevention Instruction Enum (uint8)	47 (1)	Indicates which order should be canceled due to Self-Trade Prevention.
48	securityID	R	SecurityID (uint64)	48 (8)	Security identification as defined by exchange.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID (Exchange Symbol). Constant: "8"
207	securityExchange	C	SecurityExchange (char)		Market to which the symbol belongs. Constant: "BVMF"
54	side	R	Side Enum (char)	56 (1)	Side of order.
40	ordType	R	OrdType Enum (char)	57 (1)	Order type.
59	timeInForce	R	TimeInForce Enum (char)	58 (1)	Specifies how long the order remains in effect.
35487	routingInstruction	O	RoutingInstruction Enum (uint8)	59 (1)	Indicates additional order instruction.
38	orderQty	R	Quantity (uint64)	60 (8)	Quantity ordered.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
44	price	O	PriceOptional	68 (8)	Price per share or contract. Conditionally required if the order type requires a price (not market orders and RLP).
99	stopPx	O	PriceOptional	76 (8)	The stop price of a stop limit order (Conditionally required if OrdType = 4).
110	minQty	O	QuantityOptional (uint64)	84 (8)	Minimum quantity of an order to be executed.
111	maxFloor	O	QuantityOptional (uint64)	92 (8)	Maximum number of shares or contracts within an order to be shown on the match engine at any given time.
35506	executingTrader	O	TraderOptional (char)	100 (5)	Identifies the trader who is executing an order.
432	expireDate	O	LocalMktDateOptional (uint16)	105 (2)	Date of order expiration (last day the order can trade), always expressed in terms of the local market date.
35507	custodianInfo	O	CustodianInfo	107 (12)	Identifies the custodian.
35508	investorID	O	InvestorID	119 (8)	Unique identifier of investor for self-trade prevention/mass cancel on behalf purposes.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35510	deskID	O	DeskIDEncoding		Identifies the trading desk.
5149	memo	O	MemoEncoding		Free UTF-8 format text field. This field may be used to convey client's relevant info. It's always echoed in the reports.

7.16 Message: OrderCancelReplaceRequest

Name	Template ID	Version	Block Length	Description
OrderCancelReplaceRequest	104	2	144	Sent by client system to replace an existing order.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (uint8)		Message type = OrderCancelReplaceRequest. Constant: 18
35524	businessHeader	R	InboundBusinessHeader	0 (18)	Common header to all inbound business messages.
35505	ordTagID	O	OrdTagID (uint8)	18 (1)	Identifies the order tag identification.
9773	mmProtectionReset	R	Boolean Enum (uint8)	19 (1)	Resets Market Protections. When Market Protections are triggered, the Exchange will not accept new orders for that product group without tag MMProtectionReset: True = Reset Market Maker Protection; False = Do nothing related to Market Maker Protection.
11	clOrdID	R	ClOrdID (uint64)	20 (8)	Unique identifier of the order as assigned by the market participant.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
1	account	O	AccountOptional (uint32)	28 (4)	Account mnemonic of the order.
35503	senderLocation	R	SenderLocation (char)	32 (10)	Identifies the original location for routing orders.
35502	enteringTrader	R	Trader (char)	42 (5)	Identifies the trader who is inserting an order.
35539	selfTradePreventionInstruction	R	SelfTradePreventionInstruction Enum (uint8)	47 (1)	Indicates which order should be canceled due to Self-Trade Prevention.
48	securityID	R	SecurityID (uint64)	48 (8)	Security identification as defined by exchange.
22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID (Exchange Symbol). Constant: "8"
207	securityExchange	C	SecurityExchange (char)		Market to which the symbol belongs. Constant: "BVMF"
54	side	R	Side Enum (char)	56 (1)	Side of order.
40	ordType	R	OrdType Enum (char)	57 (1)	Order type.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
59	timeInForce	O	TimeInForce Enum (char)	58 (1)	Specifies how long the order remains in effect.
35487	routingInstruction	O	RoutingInstruction Enum (uint8)	59 (1)	Indicates additional order instruction.
38	orderQty	R	Quantity (uint64)	60 (8)	Quantity ordered.
44	price	O	PriceOptional	68 (8)	Price per share or contract. Conditionally required if the order type requires a price (not market orders and RLP).
37	orderID	O	OrderIDOptional (uint64)	76 (8)	Unique identifier for order as assigned by the exchange.
41	origClOrdID	O	ClOrdIDOptional (uint64)	84 (8)	ClOrdID which should be replaced.
99	stopPx	O	PriceOptional	92 (8)	The stop price of a stop limit order (Conditionally required if OrdType = 4).
110	minQty	O	QuantityOptional (uint64)	100 (8)	Minimum quantity of an order to be executed.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
111	maxFloor	O	QuantityOptional (uint64)	108 (8)	Maximum number of shares or contracts within an order to be shown on the match engine at any given time.
35506	executingTrader	O	TraderOptional (char)	116 (5)	Identifies the trader who is executing an order.
581	accountType	O	AccountType Enum (uint8)	121 (1)	Type of account associated with an order.
432	expireDate	O	LocalMktDateOptional (uint16)	122 (2)	Date of order expiration (last day the order can trade), always expressed in terms of the local market date.
35507	custodianInfo	O	CustodianInfo	124 (12)	Identifies the custodian.
35508	investorID	O	InvestorID	136 (8)	Unique identifier of investor for self-trade prevention/mass cancel on behalf purposes.
35510	deskID	O	DeskIDEncoding		Identifies the trading desk.
5149	memo	O	MemoEncoding		Free UTF-8 format text field. This field may be used to convey client's relevant info. It's always echoed in the reports.

7.17 Message: OrderCancelRequest

Name	Template ID	Version	Block Length	Description
OrderCancelRequest	105	2	76	OrderCancelRequest message submits a deletion of an existing order by referencing the original client order id.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (uint8)		Message type = OrderCancelRequest. Constant: 19
35524	businessHeader	R	InboundBusinessHeader	0 (18)	Common header to all inbound business messages.
	<padding>			18 (2)	*
11	clOrdID	R	ClOrdID (uint64)	20 (8)	Unique identifier of the order as assigned by the market participant.
48	securityID	R	SecurityID (uint64)	28 (8)	Security identification as defined by exchange.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID (Exchange Symbol). Constant: "8"
207	securityExchange	C	SecurityExchange (char)		Market to which the symbol belongs. Constant: "BVMF"
37	orderID	O	OrderIDOptional (uint64)	36 (8)	Unique identifier for order as assigned by the exchange.
41	origClOrdID	O	ClOrdIDOptional (uint64)	44 (8)	ClOrdID which should be canceled.
54	side	R	Side	52 (1)	Side of order.
378	execRestatementReason	O	ExecRestatementReasonValidForSingleCancelEnum (uint8)	53 (1)	Used to communicate a reason for a solicited cancel.
	<padding>			54 (2)	*
35503	senderLocation	R	SenderLocation (char)	56 (10)	Identifies the original location for routing orders.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35502	enteringTrader	R	Trader (char)	66 (5)	Identifies the trader who is inserting an order.
35506	executingTrader	O	TraderOptional (char)	71 (5)	Identifies the trader who is executing an order.
35510	deskID	O	DeskIDEncoding		Identifies the trading desk.
5149	memo	O	MemoEncoding		Free UTF-8 format text field. This field may be used to convey client's relevant info. It's always echoed in the reports.

7.18 Message: NewOrderCross

Name	Template ID	Version	Block Length	Description
NewOrderCross	106	2	74	The NewOrderCross message submits a Cross on Order Entry gateway, a two-sided order submitted by a single party/broker at the same price and quantity.

	Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
	35	messageType	C	MessageType Enum (uint8)		Message type = NewOrderCross. Constant: 20
	35524	businessHeader	R	InboundBusinessHeader	0 (18)	Common header to all inbound business messages.
		<padding>			18 (2)	*
	548	crossID	R	CrossID (uint64)	20 (8)	ID of electronically submitted cross order by the institution (if in response to a cross order).
	35503	senderLocation	R	SenderLocation (char)	28 (10)	Identifies the original location for routing orders.

	Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
	35502	enteringTrader	R	Trader (char)	38 (5)	Identifies the trader who is inserting an order.
	35506	executingTrader	O	TraderOptional (char)	43 (5)	Identifies the trader who is executing an order.
	48	securityID	R	SecurityID (uint64)	48 (8)	Security identification as defined by exchange.
	22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID (Exchange Symbol). Constant: "8"
	207	securityExchange	C	SecurityExchange (char)		Market to which the symbol belongs. Constant: "BVMF"
	38	orderQty	R	Quantity (uint64)	56 (8)	Quantity ordered.
	44	price	R	Price	64 (8)	Price per share or contract. Conditionally required if the order type requires a price (not market orders and RLP).
	2523	crossedIndicator	O	CrossedIndicator Enum (uint16)	72 (2)	Indicates cross order purpose.

	Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
	552	noSides	R	GroupSizeEncoding	74 (18)	
→	54	side	R	Side Enum (char)	0 (1)	Side of order.
→		<padding>			1 (1)	*
→	1	account	O	AccountOptional (uint32)	2 (4)	Account mnemonic of the order.
→	35501	enteringFirm	O	FirmOptional (uint32)	6 (4)	Identifies the broker firm that will enter orders.
→	11	clOrdID	R	ClOrdID (uint64)	10 (8)	Unique identifier of the order as assigned by the market participant.
	35510	deskID	O	DeskIDEncoding		Identifies the trading desk.
	5149	memo	O	MemoEncoding		Free UTF-8 format text field. This field may be used to convey client's relevant info. It's always echoed in the reports.

7.19 Message: ExecutionReport_New

Name	Template ID	Version	Block Length	Description
ExecutionReport_New	200	2	144	Execution Report - New message is sent in response to a NewOrderSingle or SimpleNewOrder messages, or also from a restated iceberg order.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (uint8)		MessageType.ExecutionReport_New. Constant: 21
35524	businessHeader	R	OutboundBusinessHeader	0 (18)	Common header to all outbound business messages.
54	side	R	Side Enum (char)	18 (1)	Side of order.
39	ordStatus	R	OrdStatus Enum (char)	19 (1)	Identifies current status of order.
11	clOrdID	R	ClOrdID (uint64)	20 (8)	Unique identifier of the order as assigned by the market participant.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
198	secondaryOrderID	R	OrderID (uint64)	28 (8)	Exchange-generated order identifier that changes for each order modification event, or quantity replenishment in disclosed orders.
48	securityID	R	SecurityID (uint64)	36 (8)	Security identification as defined by exchange.
22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID (Exchange Symbol). Constant: "8"
207	securityExchange	C	SecurityExchange (char)		Market to which the symbol belongs. Constant: "BVMF"
37	orderID	R	OrderID (uint64)	44 (8)	Unique identifier for order as assigned by the exchange.
1	account	O	AccountOptional (uint32)	52 (4)	Account mnemonic of the order.
17	execID	R	ExecID (uint64)	56 (8)	Unique identifier of execution message as assigned by the exchange – unique per instrument.
60	transactTime	R	UTCTimestampNanos	64 (8)	Time of execution/order creation; expressed in UTC.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35543	marketSegmentReceivedTime	O	UTCTimestampNanosOptional	72 (8)	Time of receipt of related inbound message in the market segment path.
35001	protectionPrice	O	PriceOptional	80 (8)	Conditionally returned on execution reports for Market and Stop Protect orders. This contains the final protection price limit at which any unmatched quantity will rest on the book.
75	tradeDate	R	LocalMktDate (uint16)	88 (2)	Indicates date of trading day (expressed in local time at place of trade). Sent in number of days since Unix epoch.
636	workingIndicator	R	Boolean Enum (uint8)	90 (1)	Indicates if an order has been triggered and is available for trading. Used with Stop (Limit, with protection) orders and the At the Close validity.
442	multiLegReportingType	O	MultiLegReportingType Enum (char)	91 (1)	Used to indicate what an Execution Report represents. Default value is 1 (Single Security).
40	ordType	R	OrdType Enum (char)	92 (1)	Order type.
59	timeInForce	R	TimeInForce Enum (char)	93 (1)	Specifies how long the order remains in effect.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
432	expireDate	O	LocalMktDateOptional (uint16)	94 (2)	Date of order expiration (last day the order can trade), always expressed in terms of the local market date.
38	orderQty	R	Quantity (uint64)	96 (8)	Quantity ordered.
44	price	O	PriceOptional	104 (8)	Price per share or contract. Conditionally required if the order type requires a price (not market orders and RLP).
99	stopPx	O	PriceOptional	112 (8)	The stop price of a stop limit order (Conditionally required if OrdType = 4).
110	minQty	O	QuantityOptional (uint64)	120 (8)	Minimum quantity of an order to be executed.
111	maxFloor	O	QuantityOptional (uint64)	128 (8)	Maximum number of shares or contracts within an order to be shown on the match engine at any given time.
548	crossID	O	CrossIDOptional (uint64)	136 (8)	ID of electronically submitted cross order by the institution (if in response to a cross order).
35510	deskID	O	DeskIDEncoding		Identifies the trading desk.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
5149	memo	O	MemoEncoding		Free UTF-8 format text field. This field may be used to convey client's relevant info. It's always echoed in the reports.

7.20 Message: ExecutionReport_Modify

Name	Template ID	Version	Block Length	Description
ExecutionReport_Modify	201	2	160	Execution Report - Modify message is sent in response to OrderCancelReplaceRequest or SimpleModifyOrder messages.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (uint8)		MessageType.ExecutionReport_Modify. Constant: 22
35524	businessHeader	R	OutboundBusinessHeader	0 (18)	Common header to all outbound business messages.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
54	side	R	Side Enum (char)	18 (1)	Side of order.
39	ordStatus	R	OrdStatus Enum (char)	19 (1)	Identifies current status of order.
11	clOrdID	R	ClOrdID (uint64)	20 (8)	Unique identifier of the order as assigned by the market participant.
198	secondaryOrderID	R	OrderID (uint64)	28 (8)	Exchange-generated order identifier that changes for each order modification event, or quantity replenishment in disclosed orders.
48	securityID	R	SecurityID (uint64)	36 (8)	Security identification as defined by exchange.
22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID (Exchange Symbol). Constant: "8"
207	securityExchange	C	SecurityExchange (char)		Market to which the symbol belongs. Constant: "BVMF"
151	leavesQty	R	Quantity (uint64)	44 (8)	Number of shares open for further execution, or unexecuted.
1	account	O	AccountOptional (uint32)	52 (4)	Account mnemonic of the order.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
17	execID	R	ExecID (uint64)	56 (8)	Unique identifier of execution message as assigned by the exchange – unique per instrument.
60	transactTime	R	UTCTimestampNanos	64 (8)	Time of execution/order creation; expressed in UTC.
14	cumQty	R	Quantity (uint64)	72 (8)	Total number of shares or contracts filled.
35543	marketSegmentReceivedTime	O	UTCTimestampNanosOptional	80 (8)	Time of receipt of related inbound message in the market segment path.
37	orderID	R	OrderID (uint64)	88 (8)	Unique identifier for order as assigned by the exchange.
41	origClOrdID	O	ClOrdIDOptional (uint64)	96 (8)	Value of *origClOrdID* field informed from the related request message.
35001	protectionPrice	O	PriceOptional	104 (8)	Conditionally returned on execution reports for Market and Stop Protect orders. This contains the final protection price limit at which any unmatched quantity will rest on the book.
75	tradeDate	R	LocalMktDate (uint16)	112 (2)	Indicates date of trading day (expressed in local time at place of trade). Sent in number of days since Unix epoch.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
636	workingIndicator	R	Boolean Enum (uint8)	114 (1)	Indicates if an order has been triggered and is available for trading. Used with Stop (Limit, with protection) orders and the At the Close validity.
442	multiLegReportingType	O	MultiLegReportingType Enum (char)	115 (1)	Used to indicate what an Execution Report represents. Default value is 1 (Single Security).
40	ordType	R	OrdType Enum (char)	116 (1)	Order type.
59	timeInForce	R	TimeInForce Enum (char)	117 (1)	Specifies how long the order remains in effect.
432	expireDate	O	LocalMktDateOptional (uint16)	118 (2)	Date of order expiration (last day the order can trade), always expressed in terms of the local market date.
38	orderQty	R	Quantity (uint64)	120 (8)	Quantity ordered.
44	price	O	PriceOptional	128 (8)	Price per share or contract. Conditionally required if the order type requires a price (not market orders and RLP).
99	stopPx	O	PriceOptional	136 (8)	The stop price of a stop limit order (Conditionally required if OrdType = 4).

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
110	minQty	O	QuantityOptional (uint64)	144 (8)	Minimum quantity of an order to be executed.
111	maxFloor	O	QuantityOptional (uint64)	152 (8)	Maximum number of shares or contracts within an order to be shown on the match engine at any given time.
35510	deskID	O	DeskIDEncoding		Identifies the trading desk.
5149	memo	O	MemoEncoding		Free UTF-8 format text field. This field may be used to convey client's relevant info. It's always echoed in the reports.

7.21 Message: ExecutionReport_Cancel

Name	Template ID	Version	Block Length	Description
ExecutionReport_Cancel	202	2	156	ExecutionReport - Cancel message is sent in response to Order Cancel Request as well as to report unsolicited cancellation of orders due to: Market Operations or Cancel on Disconnect mechanism.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (uint8)		MessageType.ExecutionReport_Cancel. Constant: 23
35524	businessHeader	R	OutboundBusinessHeader	0 (18)	Common header to all outbound business messages.
54	side	R	Side Enum (char)	18 (1)	Side of order.
39	ordStatus	R	OrdStatus Enum (char)	19 (1)	Identifies current status of order.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
11	clOrdID	R	ClOrdID (uint64)	20 (8)	Unique identifier of the order as assigned by the market participant.
198	secondaryOrderID	R	OrderID (uint64)	28 (8)	Exchange-generated order identifier that changes for each order modification event, or quantity replenishment in disclosed orders.
48	securityID	R	SecurityID (uint64)	36 (8)	Security identification as defined by exchange.
22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID (Exchange Symbol). Constant: "8"
207	securityExchange	C	SecurityExchange (char)		Market to which the symbol belongs. Constant: "BVMF"
14	cumQty	R	Quantity (uint64)	44 (8)	Total number of shares or contracts filled.
1	account	O	AccountOptional (uint32)	52 (4)	Account mnemonic of the order.
17	execID	R	ExecID (uint64)	56 (8)	Unique identifier of execution message as assigned by the exchange – unique per instrument.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
60	transactTime	R	UTCTimestampNanos	64 (8)	Time of execution/order creation; expressed in UTC.
35543	marketSegmentReceivedTime	O	UTCTimestampNanosOptional	72 (8)	Time of receipt of related inbound message in the market segment path.
37	orderID	R	OrderID (uint64)	80 (8)	Unique identifier for order as assigned by the exchange.
41	origClOrdID	O	ClOrdIDOptional (uint64)	88 (8)	Value of *origClOrdID* field informed from the related request message.
75	tradeDate	R	LocalMktDate (uint16)	96 (2)	Indicates date of trading day (expressed in local time at place of trade). Sent in number of days since Unix epoch.
636	workingIndicator	R	Boolean Enum (uint8)	98 (1)	Indicates if an order has been triggered and is available for trading. Used with Stop (Limit, with protection) orders and the At the Close validity.
378	execRestatementReason	O	ExecRestatementReason Enum (uint8)	99 (1)	Indicates reason of cancelation, if available.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
	<padding>			100 (4)	*
1369	massActionReportID	O	MassActionReportIDOptional (uint64)	104 (8)	Unique ID of Order Mass Action Report as assigned by the matching engine.
40	ordType	R	OrdType Enum (char)	112 (1)	Order type.
59	timeInForce	R	TimeInForce Enum (char)	113 (1)	Specifies how long the order remains in effect.
432	expireDate	O	LocalMktDateOptional (uint16)	114 (2)	Date of order expiration (last day the order can trade), always expressed in terms of the local market date.
38	orderQty	R	Quantity (uint64)	116 (8)	Quantity ordered.
44	price	O	PriceOptional	124 (8)	Price per share or contract. Conditionally required if the order type requires a price (not market orders and RLP).
99	stopPx	O	PriceOptional	132 (8)	The stop price of a stop limit order (Conditionally required if OrdType = 4).

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
110	minQty	O	QuantityOptional (uint64)	140 (8)	Minimum quantity of an order to be executed.
111	maxFloor	O	QuantityOptional (uint64)	148 (8)	Maximum number of shares or contracts within an order to be shown on the match engine at any given time.
35510	deskID	O	DeskIDEncoding		Identifies the trading desk.
5149	memo	O	MemoEncoding		Free UTF-8 format text field. This field may be used to convey client's relevant info. It's always echoed in the reports.

7.22 Message: ExecutionReport_Reject

Name	Template ID	Version	Block Length	Description
ExecutionReport_Reject	204	2	138	Execution Report - Reject message notifies the reason a client request was not accepted by Matching Engine.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (uint8)		MessageType.ExecutionReport_Reject. Constant: 25
35524	businessHeader	R	OutboundBusinessHeader	0 (18)	Common header to all outbound business messages.
54	side	R	Side Enum (char)	18 (1)	Side of order.
39	ordStatus	C	OrdStatus Enum (char)		Identifies current status of order. Constant: "8"

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
434	cxlRejResponseTo	O	CxlRejResponseTo Enum (uint8)	19 (1)	Identifies the type of request that this Cancel Reject is in response to.
11	clOrdID	R	ClOrdID (uint64)	20 (8)	Unique identifier of the order as assigned by the market participant.
198	secondaryOrderID	O	OrderIDOptional (uint64)	28 (8)	Exchange-generated order identifier that changes for each order modification event, or quantity replenishment in disclosed orders.
48	securityID	R	SecurityID (uint64)	36 (8)	Security identification as defined by exchange.
22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID (Exchange Symbol). Constant: "8"
207	securityExchange	C	SecurityExchange (char)		Market to which the symbol belongs. Constant: "BVMF"
103	ordRejReason	R	RejReason (uint32)	44 (4)	Code to identify reason for order rejection. Please refer to the error codes document for domain information.
60	transactTime	R	UTCTimestampNanos	48 (8)	Time of execution/order creation; expressed in UTC.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
17	execID	R	ExecID (uint64)	56 (8)	Unique identifier of execution message as assigned by the exchange – unique per instrument.
37	orderID	O	OrderIDOptional (uint64)	64 (8)	Unique identifier for order as assigned by the exchange.
41	origClOrdID	O	ClOrdIDOptional (uint64)	72 (8)	Value of origClOrdID field informed from the related request message.
1	account	O	AccountOptional (uint32)	80 (4)	Account mnemonic of the order.
40	ordType	R	OrdType Enum (char)	84 (1)	Order type.
59	timeInForce	R	TimeInForce Enum (char)	85 (1)	Specifies how long the order remains in effect.
432	expireDate	O	LocalMktDateOptional (uint16)	86 (2)	Date of order expiration (last day the order can trade), always expressed in terms of the local market date.
38	orderQty	R	Quantity (uint64)	88 (8)	Quantity ordered.
44	price	O	PriceOptional	96 (8)	Price per share or contract. Conditionally required if the order type requires a price (not market orders and RLP).

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
99	stopPx	O	PriceOptional	104 (8)	The stop price of a stop limit order (Conditionally required if OrdType = 4).
110	minQty	O	QuantityOptional (uint64)	112 (8)	Minimum quantity of an order to be executed.
111	maxFloor	O	QuantityOptional (uint64)	120 (8)	Maximum number of shares or contracts within an order to be shown on the match engine at any given time.
548	crossID	O	CrossIDOptional (uint64)	128 (8)	ID of electronically submitted cross order by the institution (if in response to a cross order).
2523	crossedIndicator	O	CrossedIndicator Enum (uint16)	136 (2)	Indicates cross order purpose.
35510	deskID	O	DeskIDEncoding		Identifies the trading desk.
5149	memo	O	MemoEncoding		Free UTF-8 format text field. This field may be used to convey client's relevant info. It's always echoed in the reports.
58	text	O	TextEncoding		Free ASCII format text string.

7.23 Message: ExecutionReport_Trade

Name	Template ID	Version	Block Length	Description
ExecutionReport_Trade	203	2	154	Execution Report – Trade/Trade Bust message is sent with order fills that were traded and processed on Matching Engine. Also, trade bust included on behalf of B3's desk.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (uint8)		MessageType.ExecutionReport_Trade. Constant: 24
35524	businessHeader	R	OutboundBusinessHeader	0 (18)	Common header to all outbound business messages.
54	side	R	Side Enum (char)	18 (1)	Side of order.
39	ordStatus	R	OrdStatus Enum (char)	19 (1)	Identifies current status of order.
11	clOrdID	O	ClOrdIDOptional (uint64)	20 (8)	Unique identifier of the order as assigned by the market participant.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
198	secondaryOrderID	R	OrderID (uint64)	28 (8)	Exchange-generated order identifier that changes for each order modification event, or quantity replenishment in disclosed orders.
48	securityID	R	SecurityID (uint64)	36 (8)	Security identification as defined by exchange.
22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID (Exchange Symbol). Constant: "8"
207	securityExchange	C	SecurityExchange (char)		Market to which the symbol belongs. Constant: "BVMF"
1	account	O	AccountOptional (uint32)	44 (4)	Account mnemonic of the order.
33	lastQty	R	Quantity (uint64)	48 (8)	Quantity of shares bought/sold on the last fill.
31	lastPx	R	Price	56 (8)	Price of last fill.
17	execID	R	ExecID (uint64)	64 (8)	Unique identifier of execution message as assigned by the exchange – unique per instrument.
60	transactTime	R	UTCTimestampNanos	72 (8)	Time of execution/order creation; expressed in UTC.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
151	leavesQty	R	Quantity (uint64)	80 (8)	Number of shares open for further execution, or unexecuted.
14	cumQty	R	Quantity (uint64)	88 (8)	Total number of shares or contracts filled.
1057	aggressorIndicator	R	Boolean Enum (uint8)	96 (1)	Identify whether the order initiator is an aggressor or not in the trade.
150	execType	R	ExecType Enum (char)	97 (1)	Describes the action that triggered this specific Execution Report - see the OrdStatus (39) tag for the current order status (e.g, Partially Filled).
1115	orderCategory	O	OrderCategory Enum (char)	98 (1)	Reason a trade occurred.
442	multiLegReportingType	O	MultiLegReportingType Enum (char)	99 (1)	Used to indicate what an Execution Report represents. Default value is 1 (Single Security).
1003	tradeID	R	TradeID (uint32)	100 (4)	Contains the unique identifier for this trade, per instrument + trading date, as assigned by the exchange.
375	contraBroker	R	Firm (uint32)	104 (4)	Identifies the contra broker firm.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
37	orderID	R	OrderID (uint64)	108 (8)	Unique identifier for order as assigned by the exchange.
75	tradeDate	R	LocalMktDate (uint16)	116 (2)	Indicates date of trading day (expressed in local time at place of trade). Sent in number of days since Unix epoch.
393	totNoRelatedSym	O	TotNoRelatedSym (uint8)	118 (1)	Number of leg fill notice messages sent with spread summary.
	<padding>			119 (1)	*
8	secondaryExecID	O	ExecIDOptional (uint64)	120 (8)	Unique identifier present in all messages associated with a spread transaction. This value allows linking spread summary fill notice, leg fill notices, and leg trade cancellation execution report messages generated from a spread transaction.
19	execRefID	O	ExecIDOptional (uint64)	128 (8)	Optionally sent when reporting a trade bust. Contains the identifier of the busted trade.
548	crossID	O	CrossIDOptional (uint64)	136 (8)	ID of electronically submitted cross order by the institution (if in response to a cross order).
2523	crossedIndicator	O	CrossedIndicator Enum (uint16)	144 (2)	Indicates cross order purpose.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
38	orderQty	R	Quantity (uint64)	146 (8)	Quantity ordered.
35510	deskID	O	DeskIDEncoding		Identifies the trading desk.
5149	memo	O	MemoEncoding		Free UTF-8 format text field. This field may be used to convey client's relevant info. It's always echoed in the reports.

7.24 Message: ExecutionReport_Forward

Name	Template ID	Version	Block Length	Description
ExecutionReport_Forward	205	2	152	Execution Report – Forward message is sent with order fills were traded and processed on Matching Engine for Forward exclusively (Termo).

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (uint8)		MessageType.ExecutionReport_Forward. Constant: 26
35524	businessHeader	R	OutboundBusinessHeader	0 (18)	Common header to all outbound business messages.
54	side	R	Side Enum (char)	18 (1)	Side of order.
39	ordStatus	R	OrdStatus Enum (char)	19 (1)	Identifies current status of order.
11	clOrdID	O	ClOrdIDOptional (uint64)	20 (8)	Unique identifier of the order as assigned by the market participant.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
198	secondaryOrderID	R	OrderID (uint64)	28 (8)	Exchange-generated order identifier that changes for each order modification event, or quantity replenishment in disclosed orders.
48	securityID	R	SecurityID (uint64)	36 (8)	Security identification as defined by exchange.
22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID (Exchange Symbol). Constant: "8"
207	securityExchange	C	SecurityExchange (char)		Market to which the symbol belongs. Constant: "BVMF"
1	account	O	AccountOptional (uint32)	44 (4)	Account mnemonic of the order.
32	lastQty	R	Quantity (uint64)	48 (8)	Quantity of shares bought/sold on the last fill.
31	lastPx	R	Price	56 (8)	Price of last fill.
17	execID	R	ExecID (uint64)	64 (8)	Unique identifier of execution message as assigned by the exchange – unique per instrument.
60	transactTime	R	UTCTimestampNanos	72 (8)	Time of execution/order creation; expressed in UTC.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
151	leavesQty	R	Quantity (uint64)	80 (8)	Number of shares open for further execution, or unexecuted.
14	cumQty	R	Quantity (uint64)	88 (8)	Total number of shares or contracts filled.
1003	tradeID	R	TradeID (uint32)	96 (4)	Contains the unique identifier for this trade, per instrument + trading date, as assigned by the exchange.
375	contraBroker	R	Firm (uint32)	100 (4)	Identifies the contra broker firm.
37	orderID	R	OrderID (uint64)	104 (8)	Unique identifier for order as assigned by the exchange.
1057	aggressorIndicator	R	Boolean Enum (uint8)	112 (1)	Identify whether the order initiator is an aggressor or not in the trade.
63	settlType	O	SettlType Enum (char)	113 (1)	Indicates who in the contract has control over evoking settlement.
75	tradeDate	R	LocalMktDate (uint16)	114 (2)	Indicates date of trading day (expressed in local time at place of trade). Sent in number of days since Unix epoch.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
5497	daysToSettlement	O	DaysToSettlementOptional (uint16)	116 (2)	Deadline for completing the forward deal. For Common, Dollar and Index contracts must be between 16 and 999. And maximum of 90 days for Flexible.
	<padding>			118 (2)	*
527	secondaryExecID	O	ExecIDOptional (uint64)	120 (8)	Unique identifier present in all messages associated with a spread transaction. This value allows linking spread summary fill notice, leg fill notices, and leg trade cancellation execution report messages generated from a spread transaction.
19	execRefID	O	ExecIDOptional (uint64)	128 (8)	Optionally sent when reporting a trade bust. Contains the identifier of the busted trade.
5706	fixedRate	O	PercentageOptional	136 (8)	Describes the interest to be paid by the forward buyer and received by the forward seller, in proportion to the agreed days to settlement. Expressed in decimal form. For example, 1% is expressed and sent as 0.01. One basis point is represented as 0.0001.
38	orderQty	R	Quantity (uint64)	144 (8)	Quantity ordered.
35510	deskID	O	DeskIDEncoding		Identifies the trading desk.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
5149	memo	O	MemoEncoding		Free UTF-8 format text field. This field may be used to convey client's relevant info. It's always echoed in the reports.

7.25 Message: OrderMassActionRequest

Name	Template ID	Version	Block Length	Description
OrderMassActionRequest	701	2	54	OrderMassActionRequest is sent by the client system to cancel working orders that belongs to a defined criteria as per client definition.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (uint8)		Message type = OrderMassActionRequest. Constant: 29
35524	businessHeader	R	InboundBusinessHeader	0 (18)	Common header to all inbound business messages.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
1373	massActionType	R	MassActionType Enum (uint8)	18 (1)	Specifies the type of action requested.
1374	massActionScope	R	MassActionScope Enum (uint8)	19 (1)	Specifies the scope of the action.
11	clOrdID	R	ClOrdID (uint64)	20 (8)	Unique identifier of the order as assigned by the market participant.
378	execRestatementReason	R	ExecRestatementReason ValidForMassCancel Enum (uint8)	28 (1)	Used to communicate event type which triggers the Order Mass Action Request.
35505	ordTagID	O	OrdTagID (uint8)	29 (1)	Identifies the order tag identification.
54	side	O	Side Enum (char)	30 (1)	Side of order.
	<padding>			31 (1)	*
6937	asset	O	AssetOptional (char)	32 (6)	Asset associated with the security, such as DOL, BGI, OZ1, WDL, CNI, etc.
48	securityID	O	SecurityIDOptional (uint64)	38 (8)	Security identification as defined by exchange.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID (Exchange Symbol). Constant: "8"
207	securityExchange	C	SecurityExchange (char)		Market to which the symbol belongs. Constant: "BVMF"
35508	investorID	O	InvestorID	46 (8)	Unique identifier of investor for mass cancel on behalf purposes.

7.26 Message: OrderMassActionReport

Name	Template ID	Version	Block Length	Description
OrderMassActionReport	702	2	72	OrderMassActionReport message is used to acknowledge an OrderMassActionRequest message.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (uint8)		Message type = OrderMassActionReport. Constant: 30
35524	businessHeader	R	OutboundBusinessHeader	0 (18)	Common header to all inbound business messages.
1373	massActionType	R	MassActionType Enum (uint8)	18 (1)	Specifies the type of action requested.
1374	massActionScope	R	MassActionScope Enum (uint8)	19 (1)	Specifies the scope of the action.
11	clOrdID	R	ClOrdID (uint64)	20 (8)	Unique identifier of the order as assigned by the market participant.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
1369	massActionReportID	R	MassActionReportID (uint64)	28 (8)	Unique ID of Order Mass Action Report as assigned by the matching engine.
60	transactTime	R	UTCTimestampNanos	36 (8)	Time of execution/order creation; expressed in UTC.
1375	massActionResponse	R	MassActionResponse Enum (char)	44 (1)	Specifies the action taken by matching engine when it receives the Order Mass Action Request.
1376	massActionRejectReason	O	MassActionRejectReason Enum (uint8)	45 (1)	Reason Order Mass Action Request was rejected.
378	execRestatementReason	O	ExecRestatementReasonValidForMassCancel Enum (uint8)	46 (1)	Used to communicate event type which triggers the Order Mass Action Request.
35505	ordTagID	O	OrdTagID (uint8)	47 (1)	Identifies the order tag identification.
54	side	O	Side Enum (char)	48 (1)	Side of order.
	<padding>			49 (1)	*

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
6937	asset	O	AssetOptional (char)	50 (6)	Asset associated with the security, such as DOL, BGI, OZ1, WDL, CNI, etc.
48	securityID	O	SecurityIDOptional (uint64)	56 (8)	Security identification as defined by exchange.
22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID (Exchange Symbol). Constant: "8"
207	securityExchange	C	SecurityExchange (char)		Market to which the symbol belongs. Constant: "BVMF"
35508	investorID	O	InvestorID	64 (8)	Unique identifier of investor for mass cancel on behalf purposes.
58	text	O	TextEncoding		Free ASCII format text string.

7.27 Message: BusinessMessageReject

Name	Template ID	Version	Block Length	Description
BusinessMessageReject	206	2	36	BusinessMessageReject message can reject an application-level message which fulfills session level rules but fails the business rules.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (uint8)		Message type = BusinessMessageReject. Constant: 14
35524	businessHeader	R	OutboundBusinessHeader	0 (18)	Common header to all outbound business messages.
372	refMsgType	R	MessageType Enum (uint8)	18 (1)	MsgType of the FIX message being referenced.
	<padding>			19 (1)	*
45	refSeqNum	R	SeqNum (uint32)	20 (4)	Message sequence number of rejected message.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
379	businessRejectRefID	O	BusinessRejectRefID (uint64)	24 (8)	The value of the business-level “ID” field on the message being referenced. Required unless the corresponding ID field was not specified.
380	businessRejectReason	R	RejReason (uint32)	32 (4)	Code to identify the reason of the rejection.
5149	memo	O	MemoEncoding		Free UTF-8 format text field. This field may be used to convey client's relevant info. It's always echoed in the reports.
58	text	O	TextEncoding		Free ASCII format text string.

7.28 Message: SecurityDefinitionRequest

Name	Template ID	Version	Block Length	Description
SecurityDefinitionRequest	300	2	41	The SecurityDefinitionRequest message creates a User Defined Spread (UDS) instrument. User-Defined Spreads provide users of the electronic trading platform the ability to create strategies composed by their choice of leg instruments, leg ratio and leg side.

	Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
	35	messageType	C	MessageType Enum (uint8)		Message type = SecurityDefinitionRequest. Constant: 27
	35524	businessHeader	R	InboundBusinessHeader	0 (18)	Common header to all inbound business messages.
	320	securityReqID	R	SecurityReqRespID (uint64)	18 (8)	Unique ID of a Security Definition Request.
	35503	senderLocation	R	SenderLocation (char)	26 (10)	Identifies the original location for routing orders.

	Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
	35502	enteringTrader	R	Trader (char)	36 (5)	Identifies the trader who is inserting an order.
	555	noLegs	R	GroupSizeEncoding	41 (30)	
→	600	legSymbol	R	Symbol (char)	0 (20)	Multileg instrument's individual security's Symbol. See Symbol (55) field for description.
→	616	legSecurityExchange	C	SecurityExchange (char)		Exchange code the leg security belongs to. Constant: "BVMF"
→	623	legRatioQty	R	RatioQty	20 (8)	The ratio of quantity for this individual leg relative to the entire multileg security.
→	624	legSide	O	Side Enum (char)	28 (1)	The side of this individual leg (multileg security). See Side (54) field for description and values.

7.29 Message: SecurityDefinitionResponse

Name	Template ID	Version	Block Length	Description
SecurityDefinitionResponse	301	2	83	The SecurityDefinitioresponse message is sent in response to an attempt to create a new security definition.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (uint8)		Message type = SecurityDefinitionResponse. Constant: 28
35524	businessHeader	R	OutboundBusinessHeader	0 (18)	Common header to all outbound business messages.
	<padding>			18 (2)	*
320	securityReqID	R	SecurityReqRespID (uint64)	20 (8)	Unique ID of a Security Definition Request.
48	securityID	R	SecurityID (uint64)	28 (8)	Security identification as defined by exchange.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID (Exchange Symbol). Constant: "8"
207	securityExchange	C	SecurityExchange (char)		Market to which the symbol belongs. Constant: "BVMF"
323	securityResponseType	R	SecurityResponseType Enum (uint8)	36 (1)	Type of Security Definition message response.
7534	securityStrategyType	O	SecurityStrategyType (char)	37 (3)	Indicates the type of Strategy created. This field is not sent on rejects.
55	symbol	R	Symbol (char)	40 (20)	B3 requires that this field is properly set. It contains the human readable form of the SecurityID tag, available in the Security List message in Market Data feed.
322	securityResponseID	R	SecurityReqRespID (uint64)	60 (8)	Unique ID of a Security Definition message.
35503	senderLocation	R	SenderLocation (char)	68 (10)	Identifies the original location for routing orders.
35502	enteringTrader	R	Trader (char)	78 (5)	Identifies the trader who is inserting an order.

7.30 Message: QuoteRequest

Name	Template ID	Version	Block Length	Description
QuoteRequest	401	2	100	The Quote Request message is used within the context of this Forward transaction in which two parties have completed a deal outside the Exchange and are initiating the negotiation process to formalize and execute this operation on the Exchange.

	Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
	35	messageType	C	MessageType Enum (uint8)		Message type = QuoteRequest. Constant: 31
	35524	businessHeader	R	BidirectionalBusinessHeader	0 (20)	Common header to all bidirectional business messages.
	48	securityID	R	SecurityID (uint64)	20 (8)	Security identification as defined by exchange.
	22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID (Exchange Symbol). Constant: "8"
	207	securityExchange	C	SecurityExchange (char)		Market to which the symbol belongs. Constant: "BVMF"

	Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
	117	quoteID	O	QuoteIDOptional (uint64)	28 (8)	Unique identifier for quote.
	1003	tradeID	O	TradeIDOptional (uint32)	36 (4)	Contains the unique identifier for this trade, per instrument + trading date, as assigned by the exchange.
	375	contraBroker	R	Firm (uint32)	40 (4)	Broker identifier as assigned by B3 used to indicate the counterparty brokerage firm in a Forward deal.
	60	transactTime	R	UTCTimestampNanos	44 (8)	Time of execution/order creation; expressed in UTC.
	44	price	R	Price	52 (8)	Price per share or contract. Conditionally required if the order type requires a price (not market orders and RLP).
	63	settlType	R	SettlType Enum (char)	60 (1)	Indicates who in the contract has control over evoking settlement.
	35004	executeUnderlyingTrade	O	ExecuteUnderlyingTrade Enum (char)	61 (1)	Specifies if a simultaneous trade of the underlying is to be performed. Required to indicate Termo Vista and Termo Vista Registered.
	38	orderQty	R	Quantity (uint64)	62 (8)	Quantity ordered.

	Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
	35503	senderLocation	R	SenderLocation (char)	70 (10)	Identifies the original location for routing orders.
	35502	enteringTrader	R	Trader (char)	80 (5)	Identifies the trader who is inserting an order.
	35506	executingTrader	R	Trader (char)	85 (5)	Identifies the trader who is executing an order.
	5706	fixedRate	R	Percentage	90 (8)	Describes the interest to be paid by the forward buyer and received by the forward seller, in proportion to the agreed days to settlement. Expressed in decimal form. For example, 1% is expressed and sent as 0.01. One basis point is represented as 0.0001.
	1171	privateQuote	C	Boolean Enum (uint8)		Specifies whether a quote is public, i.e., available to the market, or private, i.e. available to a specified counterparty only. Constant: 1 (true, Y, 1)
	5497	daysToSettlement	R	DaysToSettlement (uint16)	98 (2)	Deadline for completing the forward deal. For Common, Dollar and Index contracts must be between 16 and 999. And maximum of 90 days for Flexible.
	35511	noSides	R	GroupSizeEncoding	100 (5)	
→	54	side	R	Side Enum (char)	0 (1)	Side of order.

	Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
→	1	account	O	AccountOptional (uint32)	1 (4)	Account mnemonic of the order.
	131	quoteReqID	O	QuoteReqID		Unique identifier for quote request.
	35510	deskID	O	DeskIDEncoding		Identifies the trading desk.
	5149	memo	O	MemoEncoding		Free UTF-8 format text field. This field may be used to convey client's relevant info. It's always echoed in the reports.

7.31 Message: QuoteStatusReport

Name	Template ID	Version	Block Length	Description
QuoteStatusReport	402	2	111	The QuoteStatusReport message is to inform the current status of forward acceptance.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (uint8)		Message type = QuoteStatusReport. Constant: 32
35524	businessHeader	R	BidirectionalBusinessHeader	0 (20)	Common header to all bidirectional business messages.
300	quoteRejectReason	O	RejReasonOptional (uint32)	20 (4)	Reason Quote was rejected.
48	securityID	R	SecurityID (uint64)	24 (8)	Security identification as defined by exchange.
22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID (Exchange Symbol). Constant: "8"

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
207	securityExchange	C	SecurityExchange (char)		Market to which the symbol belongs. Constant: "BVMF"
117	quoteID	R	QuoteID (uint64)	32 (8)	Unique identifier for quote.
1003	tradeID	O	TradeIDOptional (uint32)	40 (4)	Contains the unique identifier for this trade, per instrument + trading date, as assigned by the exchange.
375	contraBroker	R	Firm (uint32)	44 (4)	Broker identifier as assigned by B3 used to indicate the counterparty brokerage firm in a Forward deal.
60	transactTime	R	UTCTimestampNanos	48 (8)	Time of execution/order creation; expressed in UTC.
297	quoteStatus	R	QuoteStatus Enum (uint8)	56 (1)	Identifies the status of the quote acknowledgement.
35006	quoteStatusResponseTo	O	QuoteStatusResponseTo Enum (char)	57 (1)	Identifies the type of request that a Quote Status Report is in response to.
1	account	O	AccountOptional (uint32)	58 (4)	Account mnemonic of the order.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
54	side	O	Side Enum (char)	62 (1)	Side of order.
63	settlType	O	SettlType Enum (char)	63 (1)	Indicates who in the contract has control over evoking settlement.
44	price	O	PriceOptional	64 (8)	Price per share or contract. Conditionally required if the order type requires a price (not market orders and RLP).
38	orderQty	R	Quantity (uint64)	72 (8)	Quantity ordered.
35503	senderLocation	R	SenderLocation (char)	80 (10)	Identifies the original location for routing orders.
35502	enteringTrader	R	Trader (char)	90 (5)	Identifies the trader who is inserting an order.
35506	executingTrader	R	Trader (char)	95 (5)	Identifies the trader who is executing an order.
5706	fixedRate	O	PercentageOptional	100 (8)	Interest rate of the forward trade.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35004	executeUnderlyingTrade	O	ExecuteUnderlyingTrade Enum (char)	108 (1)	Specifies if a simultaneous trade of the underlying is to be performed. Required to indicate Termo Vista and Termo Vista Registered.
1171	privateQuote	C	Boolean Enum (uint8)		Specifies whether a quote is public, i.e. available to the market, or private, i.e. available to a specified counterparty only. Constant: 1 (true, Y, 1)
5497	daysToSettlement	O	DaysToSettlementOptional (uint16)	109 (2)	Deadline for completing the forward deal. For Common, Dollar and Index contracts must be between 16 and 999. And maximum of 90 days for Flexible.
131	quoteReqID	O	QuoteReqID		Unique identifier for quote request.
35510	deskID	O	DeskIDEncoding		Identifies the trading desk.
5149	memo	O	MemoEncoding		Free UTF-8 format text field. This field may be used to convey client's relevant info. It's always echoed in the reports.
58	text	O	TextEncoding		Free ASCII format text string.

7.32 Message: Quote

Name	Template ID	Version	Block Length	Description
Quote	403	2	97	Quote message is used as the response to a QuoteRequest message, tradeable, and restricted tradeable quoting markets.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (uint8)		Message type = TermoQuote. Constant: 33
35524	businessHeader	R	BidirectionalBusinessHeader	0 (20)	Common header to all bidirectional business messages.
48	securityID	R	SecurityID (uint64)	20 (8)	Security identification as defined by exchange.
22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID (Exchange Symbol). Constant: "8"
207	securityExchange	C	SecurityExchange (char)		Market to which the symbol belongs. Constant: "BVMF"
117	quoteID	R	QuoteID (uint64)	28 (8)	Unique identifier for quote.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
60	transactTime	R	UTCTimestampNanos	36 (8)	Time of execution/order creation; expressed in UTC.
44	price	O	PriceOptional	44 (8)	Price per share or contract. Conditionally required if the order type requires a price (not market orders and RLP).
38	orderQty	R	Quantity (uint64)	52 (8)	Quantity ordered.
54	side	R	Side Enum (char)	60 (1)	Side of order.
63	settlType	R	SettlType Enum (char)	61 (1)	Indicates who in the contract has control over evoking settlement.
1	account	O	AccountOptional (uint32)	62 (4)	Account mnemonic of the order.
35503	senderLocation	R	SenderLocation (char)	66 (10)	Identifies the original location for routing orders.
35502	enteringTrader	R	Trader (char)	76 (5)	Identifies the trader who is inserting an order.
35506	executingTrader	R	Trader (char)	81 (5)	Identifies the trader who is executing an order.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
5706	fixedRate	R	Percentage	86 (8)	Interest rate of the forward trade.
35004	executeUnderlyingTrade	O	ExecuteUnderlyingTrade Enum (char)	94 (1)	Specifies if a simultaneous trade of the underlying is to be performed. Required to indicate Termo Vista and Termo Vista Registered.
1171	privateQuote	C	Boolean Enum (uint8)		Specifies whether a quote is public, i.e., available to the market, or private, i.e., available to a specified counterparty only. Constant: 1 (true, Y, 1)
5497	daysToSettlement	R	DaysToSettlement (uint16)	95 (2)	Deadline for completing the forward deal. For Common, Dollar and Index contracts must be between 16 and 999. And maximum of 90 days for Flexible.
131	quoteReqID	O	QuoteReqID		Unique identifier for quote request.
35510	deskID	O	DeskIDEncoding		Identifies the trading desk.
5149	memo	O	MemoEncoding		Free UTF-8 format text field. This field may be used to convey client's relevant info. It's always echoed in the reports.

7.33 Message: QuoteCancel

Name	Template ID	Version	Block Length	Description
QuoteCancel	404	2	60	The QuoteCancel message is used to cancel an earlier QuoteRequest message.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (uint8)		Message type = QuoteCancel. Constant: 34
35524	businessHeader	R	BidirectionalBusinessHeader	0 (20)	Common header to all bidirectional business messages.
48	securityID	R	SecurityID (uint64)	20 (8)	Security identification as defined by exchange.
22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID (Exchange Symbol). Constant: "8"
207	securityExchange	C	SecurityExchange (char)		Market to which the symbol belongs. Constant: "BVMF"
117	quoteID	O	QuoteIDOptional (uint64)	28 (8)	Unique identifier for quote.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
298	quoteCancelType	C	QuoteCancelType Enum (uint8)		Identifies the type of quote cancel. Constant: 5
1	account	O	AccountOptional (uint32)	36 (4)	Account mnemonic of the order.
35503	senderLocation	R	SenderLocation (char)	40 (10)	Identifies the original location for routing orders.
35502	enteringTrader	R	Trader (char)	50 (5)	Identifies the trader who is inserting an order.
35506	executingTrader	R	Trader (char)	55 (5)	Identifies the trader who is executing an order.
1171	privateQuote	C	Boolean Enum (uint8)		Specifies whether a quote is public, i.e., available to the market, or private, i.e. available to a specified counterparty only. Constant: 1 (true, Y, 1)
131	quoteReqID	O	QuoteReqID		Unique identifier for quote request.
35510	deskID	O	DeskIDEncoding		Identifies the trading desk.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
5149	memo	O	MemoEncoding		Free UTF-8 format text field. This field may be used to convey client's relevant info. It's always echoed in the reports.

7.34 Message: QuoteRequestReject

Name	Template ID	Version	Block Length	Description
QuoteRequestReject	405	2	103	The QuoteRequestReject message is used when a QuoteRequest is not accept by B3 due to missing or incorrect details to reject QuoteRequest messages for all quoting models.

	Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
	35	messageType	C	MessageType Enum (uint8)		Message type = QuoteRequestReject. Constant: 35
	35524	businessHeader	R	BidirectionalBusinessHeader	0 (20)	Common header to all bidirectional business messages.

	Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
	658	quoteRequestRejectReason	O	RejReasonOptional (uint32)	20 (4)	Reason Quote was rejected.
	48	securityID	R	SecurityID (uint64)	24 (8)	Security identification as defined by exchange.
	22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID (Exchange Symbol). Constant: "8"
	207	securityExchange	C	SecurityExchange (char)		Market to which the symbol belongs. Constant: "BVMF"
	117	quoteID	O	QuoteIDOptional (uint64)	32 (8)	Unique identifier for quote.
	1003	tradeID	O	TradeIDOptional (uint32)	40 (4)	Contains the unique identifier for this trade, per instrument + trading date, as assigned by the exchange.
	375	contraBroker	R	Firm (uint32)	44 (4)	Broker identifier as assigned by B3 used to indicate the counterparty brokerage firm in a Forward deal.
	60	transactTime	R	UTCTimestampNanos	48 (8)	Time of execution/order creation; expressed in UTC.
	35502	enteringTrader	R	Trader (char)	56 (5)	Identifies the trader who is inserting an order.

	Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
	63	settlType	O	SettlType Enum (char)	61 (1)	Indicates who in the contract has control over evoking settlement.
	44	price	O	PriceOptional	62 (8)	Price per share or contract. Conditionally required if the order type requires a price (not market orders and RLP).
	38	orderQty	O	QuantityOptional (uint64)	70 (8)	Quantity ordered.
	35503	senderLocation	R	SenderLocation (char)	78 (10)	Identifies the original location for routing orders.
	35506	executingTrader	R	Trader (char)	88 (5)	Identifies the trader who is executing an order.
	5706	fixedRate	O	PercentageOptional	93 (8)	Interest rate of the forward trade.
	1171	privateQuote	C	Boolean Enum (uint8)		Specifies whether a quote is public, i.e., available to the market, or private, i.e. available to a specified counterparty only. Constant: 1 (true, Y, 1)
	5497	daysToSettlement	O	DaysToSettlementOptional (uint16)	101 (2)	Deadline for completing the forward deal. For Common, Dollar and Index contracts must be between 16 and 999. And maximum of 90 days for Flexible.

	Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
	35511	noSides	R	GroupSizeEncoding	103 (5)	
→	54	side	R	Side Enum (char)	0 (1)	Side of order.
→	1	account	O	AccountOptional (uint32)	1 (4)	Account mnemonic of the order.
	131	quoteReqID	O	QuoteReqID		Unique identifier for quote request.
	35510	deskID	O	DeskIDEncoding		Identifies the trading desk.
	5149	memo	O	MemoEncoding		Free UTF-8 format text field. This field may be used to convey client's relevant info. It's always echoed in the reports.
	58	text	O	TextEncoding		Free ASCII format text string.

7.35 Message: PositionMaintenanceRequest

Name	Template ID	Version	Block Length	Description
PositionMaintenanceRequest	502	2	73	PositionMaintenanceRequest message allows the position owner (holder) to submit requests which will affect the position. Generally, the holder of the position or clearing organization is a central party but can also be a party providing investment services.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (uint8)		Message type = PositionMaintenanceRequest. Constant: 37
35524	businessHeader	R	InboundBusinessHeader	0 (18)	Common header to all inbound business messages.
710	posReqID	R	PosReqID (uint64)	18 (8)	Unique identifier for the position maintenance request.
48	securityID	R	SecurityID (uint64)	26 (8)	Security identification as defined by exchange.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID (Exchange Symbol). Constant: "8"
207	securityExchange	C	SecurityExchange (char)		Market to which the symbol belongs. Constant: "BVMF"
834	thresholdAmount	O	PriceOffsetOptional	34 (8)	Used to indicate the minimum acceptable offset between the Strike Price and the Market Price.
1	account	O	AccountOptional (uint32)	42 (4)	Account mnemonic of the order.
35503	senderLocation	R	SenderLocation (char)	46 (10)	Identifies the original location for routing orders.
709	posTransType	R	PosTransType Enum (uint8)	56 (1)	Identifies the type of position transaction.
715	clearingBusinessDate	R	LocalMktDate (uint16)	57 (2)	The 'Clearing Business Date' referred to by this request. It must be set with the current date.
719	contraryInstructionIndicator	R	Boolean Enum (uint8)	59 (1)	Used to indicate when a contrary instruction for exercise or abandonment is being submitted: The exercise should not happen to an ITM position or it should happen to an ATM or OTM

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
					position, always using the values of tags 709-PosTransType and 712-PosMaintAction to determine which operation will take place. Should not be submitted when false.
35502	enteringTrader	R	Trader (char)	60 (5)	Identifies the trader who is inserting an order.
703	posType	R	PosType Enum (char)		Used to identify the type of quantity. Constant: “E”.
704	longQty	R	Quantity (uint64)	65 (8)	Long quantity.
35510	deskID	O	DeskIDEncoding		Identifies the trading desk.
5149	memo	O	MemoEncoding		Free UTF-8 format text field. This field may be used to convey client's relevant info. It's always echoed in the reports.

7.36 Message: PositionMaintenanceCancelRequest

Name	Template ID	Version	Block Length	Description
PositionMaintenanceCancelRequest	501	2	65	PositionMaintenanceCancelRequest is a solicited cancel of PositionMaintenance message sent by client.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (uint8)		Message type = PositionMaintenanceCancelRequest. Constant: 36
35524	businessHeader	R	InboundBusinessHeader	0 (18)	Common header to all inbound business messages.
710	posReqID	R	PosReqID (uint64)	18 (8)	Unique identifier for the position maintenance request.
48	securityID	R	SecurityID (uint64)	26 (8)	Security identification as defined by exchange.
22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID (Exchange Symbol). Constant: "8"
207	securityExchange	C	SecurityExchange (char)		Market to which the symbol belongs. Constant: "BVMF"

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
713	origPosReqRefID	O	PosReqIDOptional (uint64)	34 (8)	Reference to the PosReqID (710) of a previous maintenance request that is being canceled.
714	posMaintRptRefID	O	PosMaintRptIDOptional (uint64)	42 (8)	Reference to a PosMaintRptID (721) from a previous Position Maintenance Report that is being canceled.
35503	senderLocation	R	SenderLocation (char)	50 (10)	Identifies the original location for routing orders.
35502	enteringTrader	R	Trader (char)	60 (5)	Identifies the trader who is inserting an order.

7.37 Message: PositionMaintenanceReport

Name	Template ID	Version	Block Length	Description
PositionMaintenanceReport	503	2	95	PositionMaintenanceReport message is sent owner of a position (holder) in response to a PositionMaintenanceRequest message and is used to confirm that a request has been successfully processed or rejected.

	Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
	35	messageType	C	MessageType Enum (uint8)		Message type = PositionMaintenanceReport. Constant: 38
	35524	businessHeader	R	OutboundBusinessHeader	0 (18)	Common header to all inbound business messages.
	710	posReqID	O	PosReqIDOptional (uint64)	18 (8)	Unique identifier for the position maintenance request.
	48	securityID	R	SecurityID (uint64)	26 (8)	Security identification as defined by exchange.

	Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
	22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID (Exchange Symbol). Constant: "8"
	207	securityExchange	C	SecurityExchange (char)		Market to which the symbol belongs. Constant: "BVMF"
	721	posMaintRptID	R	PosMaintRptID (uint64)	34 (8)	Unique identifier for this position report.
	709	posTransType	R	PosTransType Enum (uint8)	42 (1)	Identifies the type of position transaction.
	712	posMaintAction	R	PosMaintAction Enum (char)	43 (1)	Maintenance Action to be performed.
	722	posMaintStatus	R	PosMaintStatus Enum (char)	44 (1)	Status of Position Maintenance Request.
	1003	tradeID	O	TradeIDOptional (uint32)	45 (4)	The unique ID assigned to the trade entity once it is received or matched by the exchange or central counterparty.

	Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
	713	origPosReqRefID	O	PosReqIDOptional (uint64)	49 (8)	Reference to the PosReqID (710) of a previous maintenance request that is being canceled.
	581	accountType	O	AccountType Enum (uint8)	57 (1)	Type of account associated with an order.
	715	clearingBusinessDate	R	LocalMktDate (uint16)	58 (2)	The 'Clearing Business Date' referred to by this request. It must be set with the current date.
	834	thresholdAmount	O	PriceOffsetOptional	60 (8)	Used to indicate the minimum acceptable offset between the Strike Price and the Market Price.
	60	transactTime	R	UTCTimestampNanos	68 (8)	Time of execution/order creation; expressed in UTC.
	1	account	O	AccountOptional (uint32)	76 (4)	Account mnemonic of the order.
	35503	senderLocation	R	SenderLocation (char)	80 (10)	Identifies the original location for routing orders.
	723	posMaintResult	O	RejReasonOptional (uint32)	90 (4)	Identifies reason for rejection. Required when PosMaintStatus = 2.

	Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
	719	contraryInstructionIndicator	R	Boolean Enum (uint8)	94 (1)	Used to indicate when a contrary instruction for exercise or abandonment is being submitted :The exercise should not happen to an ITM position or it should happen to an ATM or OTM position, always using the values of tags 709-PosTransType and 712-PosMaintAction to determine which operation will take place. Should not be submitted when false.
	702	noPositions	R	GroupSizeEncoding	95 (17)	
→	703	posType	R	PosType Enum (char)	0 (1)	Used to identify the type of quantity.
→	704	longQty	O	QuantityOptional (uint64)	1 (8)	Long Quantity.
→	705	shortQty	O	QuantityOptional (uint64)	9 (8)	Short Quantity.
	35510	deskID	O	DeskIDEncoding		Identifies the trading desk.
	5149	memo	O	MemoEncoding		Free UTF-8 format text field. This field may be used to convey client's relevant info. It's always echoed in the reports.
	58	text	O	TextEncoding		Free ASCII format text string.

7.38 Message: AllocationInstruction

Name	Template ID	Version	Block Length	Description
AllocationInstruction	601	2	86	AllocationInstruction message submits a request to allocate (fully or partially) a non-allocated trade to block an issuer position, preventing it to be assigned to an exercise executed by a holder during current session.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (uint8)		Message type = AllocationInstruction. Constant: 39
35524	businessHeader	R	InboundBusinessHeader	0 (18)	Common header to all inbound business messages.
70	allocID	R	AllocID (uint64)	18 (8)	Unique identifier for this allocation instruction message.
48	securityID	R	SecurityID (uint64)	26 (8)	Security identification as defined by exchange.
22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID (Exchange Symbol). Constant: "8"

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
207	securityExchange	C	SecurityExchange (char)		Market to which the symbol belongs. Constant: "BVMF"
71	allocTransType	R	AllocTransType Enum (char)	34 (1)	Identifies allocation transaction type.
626	allocType	R	AllocType Enum (char)	35 (1)	Describes the specific type or purpose of an Allocation message.
857	allocNoOrdersType	R	AllocNoOrdersType Enum (char)	36 (1)	Indicates how the orders being booked and allocated by an Allocation Instruction.
53	quantity	R	Quantity (uint64)	37 (8)	Overall/total quantity (e.g., number of shares).
54	side	C	Side Enum (char)		Side of order. Constant: "1"
35503	senderLocation	R	SenderLocation (char)	45 (10)	Identifies the original location for routing orders.
35502	enteringTrader	R	Trader (char)	55 (5)	Identifies the trader who is inserting an order.
1003	tradeID	R	TradeID (uint32)	60 (4)	The unique ID assigned to the trade entity once it is received or matched by the exchange or central counterparty.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
75	tradeDate	O	LocalMktDateOptional (uint16)	64 (2)	Indicates date of trading day (expressed in local time at place of trade). Sent in number of days since Unix epoch.
467	individualAllocID	R	AllocID (uint64)	66 (8)	Unique identifier for a specific NoAllocs (78) repeating group instance (e.g., for an AllocAccount).
79	allocAccount	R	Account (uint32)	74 (4)	Sub-account mnemonic.
80	allocQty	R	Quantity (uint64)	78 (8)	Quantity allocated to specific sub-account.
35510	deskID	O	DeskIDEncoding		Identifies the trading desk.
5149	memo	O	MemoEncoding		Free UTF-8 format text field. This field may be used to convey client's relevant info. It's always echoed in the reports.

7.39 Message: AllocationReport

Name	Template ID	Version	Block Length	Description
AllocationReport	602	2	84	AllocationReport message is as response of AllocationInstruction message.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (uint8)		Message type = AllocationReport. Constant: 40
35524	businessHeader	R	OutboundBusinessHeader	0 (18)	Common header to all inbound business messages.
70	allocID	R	AllocID (uint64)	18 (8)	Unique identifier for this allocation instruction message.
48	securityID	R	SecurityID (uint64)	26 (8)	Security identification as defined by exchange.
22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID (Exchange Symbol). Constant: "8"
207	securityExchange	C	SecurityExchange (char)		Market to which the symbol belongs. Constant: "BVMF"

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
755	allocReportID	R	AllocReportID (uint64)	34 (8)	Unique identifier for this message.
71	allocTransType	R	AllocTransType Enum (char)	42 (1)	Identifies allocation transaction type.
794	allocReportType	R	AllocReportType Enum (char)	43 (1)	Describes the specific type or purpose of an Allocation Report message.
857	allocNoOrdersType	R	AllocNoOrdersType Enum (char)	44 (1)	Indicates how the orders being booked and allocated by an Allocation Instruction.
88	allocRejCode	O	RejReasonOptional (uint32)	45 (4)	Identifies reason for rejection.
53	quantity	R	Quantity (uint64)	49 (8)	Overall/total quantity (e.g. number of shares).
87	allocStatus	R	AllocStatus Enum (char)	57 (1)	Identifies status of allocation.
75	tradeDate	O	LocalMktDateOptional (uint16)	58 (2)	Indicates date of trading day (expressed in local time at place of trade). Sent in number of days since Unix epoch.
60	transactTime	R	UTCTimestampNanos	60 (8)	Time of execution/order creation; expressed in UTC.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
54	side	R	Side Enum (char)	68 (1)	Side of order.
35503	senderLocation	R	SenderLocation (char)	69 (10)	Identifies the original location for routing orders.
35502	enteringTrader	R	Trader (char)	79 (5)	Identifies the trader who is inserting an order.

8 COMPOSITE TYPES

8.1 Type: InboundBusinessHeader

Name	Version	Block Length	Description
InboundBusinessHeader	2	18	Header used for inbound business messages.

Name	Data Type	Size	Offset	Description
sessionID	SessionID	4	0	Client connection identification on the gateway assigned by B3.
msgSeqNum	SeqNum	4	4	Sequence number of a given SessionID/SessionVerID.
sendingTime	SendingTime	8	8	Time of message transmission, expressed in UTC.
marketSegmentID	MarketSegmentID	1	16	Identifies the market segment.
padding	char	1	17	Padding for alignment purpose.

8.2 Type: OutboundBusinessHeader

Name	Version	Block Length	Description
OutboundBusinessHeader	2	18	Header used for outbound business messages.

Name	Data Type	Size	Offset	Description
sessionID	SessionID	4	0	Client connection identification on the gateway assigned by B3.
msgSeqNum	SeqNum	4	4	Sequence number of a given SessionID/SessionVerID.
sendingTime	SendingTime	8	8	Time of message transmission, expressed in UTC.
possResend	PossResend	1	16	Indicates that message may contain information that has been sent under another sequence number.
padding	char	1	17	Padding for alignment purpose.

8.3 Type: BidirectionalBusinessHeader

Name	Version	Block Length	Description
BidirectionalBusinessHeader	2	20	Header used for business messages that can go inbound or outbound.

Name	Data Type	Size	Offset	Description
sessionID	SessionID	4	0	Client connection identification on the gateway assigned by B3.
msgSeqNum	SeqNum	4	4	Sequence number of a given SessionID/SessionVerID.
sendingTime	SendingTime	8	8	Time of message transmission, expressed in UTC.
possResend	PossResend	1	16	Indicates that message may contain information that has been sent under another sequence number.
marketSegmentID	MarketSegmentIDOptional	1	17	Optional identifier of the market segment.
padding	char	2	18	Padding for alignment purpose.

8.4 Type: ClientAppEncoding

Name	Version	Block Length	Description
ClientAppEncoding	2	N	Identifies the IP address, name and version declared for audit.

Name	Data Type	Size	Offset	Description
length	uint8	1	0	Length of client system information. Max value: 30.
varData	char	N	1	Client system information.

8.5 Type: CredentialsEncoding

Name	Version	Block Length	Description
CredentialsEncoding	2	N	Credentials to identify/authenticate the client. The format is a JSON with the following data: { "auth_type": "basic", "username": "session_id", "access_key": "somepassword" }

Name	Data Type	Size	Offset	Description
length	uint8	1	0	Length of credentials data. Max value: 128.
varData	char	N	1	Credentials data in UTF-8.

8.6 Type: CustodianInfo

Name	Version	Block Length	Description
CustodianInfo	2	12	Custodian information is required for going private offer.

Name	Data Type	Size	Offset	Description
custodian	uint32	4	0	Identifies the custodian.
custodyAccount	uint32	4	4	Identifies the custody account.
custodyAllocationType	uint32	4	8	Custody allocation type.

8.7 Type: DeltaInMillis

Name	Version	Block Length	Description
DeltaInMillis	2	8	Interval time expressed in milliseconds.

Name	Data Type	Size	Offset	Description
time	uint64	8	0	Interval time expressed in milliseconds.
unit	uint8	0	8	time unit (milliseconds). Constant: 3.

8.8 Type: DeskIDEncoding

Name	Version	Block Length	Description
DeskIDEncoding	2	N	Identify the client associated with the given account number. This information may be used to correlate the order entry messages with the messages at the back-office and clearing systems.

Name	Data Type	Size	Offset	Description
length	uint8	1	0	Length of trading desk identification. Max value: 20.
varData	char	N	1	Trading desk identification in ASCII format.

8.9 Type: FramingHeader

Name	Version	Block Length	Description
FramingHeader	2	4	Simple Open Framing Header. Used to frame both session and business messages.

Name	Data Type	Size	Offset	Description
messageLength	uint16	2	0	Message length (including Framing and SBE headers). Min value: 12. Max value: 2048.
encodingType	uint16	2	2	Encoding type of payload.

8.10 Type: GroupSizeEncoding

Name	Version	Block Length	Description
GroupSizeEncoding	2	3	Repeating group dimensions.

Name	Data Type	Size	Offset	Description
blockLength	uint16	2	0	Root block length.
numInGroup	uint8	1	2	Counter representing the number of entries in a repeating group.

8.11 Type: MemoEncoding

Name	Version	Block Length	Description
MemoEncoding	2	N	Free UTF-8 format text field. This field may be used to convey client's relevant info. It's always echoed in the reports.

Name	Data Type	Size	Offset	Description
length	uint8	1	0	Length of free format text field defined by client. Max value: 40.
varData	char	N	1	Free UTF-8 format text field defined by client.

8.12 Type: messageHeader

Name	Version	Block Length	Description
messageHeader	2	8	Message identifiers and length of message root.

Name	Data Type	Size	Offset	Description
blockLength	uint16	2	0	Length of the root of the FIX message contained before repeating groups or variable/conditions fields.
templateId	uint16	2	2	Template ID used to encode the message.
schemaId	uint16	2	4	ID of the system publishing the message.
version	uint16	2	6	Schema version.

8.13 Type: Percentage

Name	Version	Block Length	Description
Percentage	2	8	Percentage.

Name	Data Type	Size	Offset	Description
mantissa	int64	8	0	Mantissa (for fixed-point decimal numbers).
exponent	int8	0	8	Exponent (for fixed-point decimal numbers). Constant: -4.

8.14 Type: PercentageOptional

Name	Version	Block Length	Description
PercentageOptional	2	8	Percentage.

Name	Data Type	Size	Offset	Description
mantissa	int64	8	0	Mantissa (for fixed-point decimal numbers).
exponent	int8	0	8	Exponent (for fixed-point decimal numbers). Constant: -4.

8.15 Type: Price

Name	Version	Block Length	Description
Price	2	8	Mandatory price.

Name	Data Type	Size	Offset	Description
mantissa	int64	8	0	Mantissa (for fixed-point decimal numbers).
exponent	int8	0	8	Exponent (for fixed-point decimal numbers). Constant: -4.

8.16 Type: PriceOffsetOptional

Name	Version	Block Length	Description
PriceOffsetOptional	2	8	Price Offset (4 decimal places). Usually 3 places are enough, but FX requires 4.

Name	Data Type	Size	Offset	Description
mantissa	int64	8	0	Mantissa (for fixed-point decimal numbers).
exponent	int8	0	8	Exponent (for fixed-point decimal numbers). Constant: -4.

8.17 Type: PriceOptional

Name	Version	Block Length	Description
PriceOptional	2	8	Optional price.

Name	Data Type	Size	Offset	Description
mantissa	int64	8	0	Mantissa (for fixed-point decimal numbers).
exponent	int8	0	8	Exponent (for fixed-point decimal numbers). Constant: -4.

8.18 Type: QuoteReqID

Name	Version	Block Length	Description
QuoteReqID	2	N	Unique identifier for quote request.

Name	Data Type	Size	Offset	Description
length	uint8	1	0	Max value: 20.
varData	uint8	N	1	Free ASCII format text string representing unique identifier for quote request.

8.19 Type: RatioQty

Name	Version	Block Length	Description
RatioQty	2	8	Ratio of quantity relative to the whole thing.

Name	Data Type	Size	Offset	Description
mantissa	int64	8	0	Mantissa (for fixed-point decimal numbers).
exponent	int8	0	8	Exponent (for fixed-point decimal numbers). Constant: -7.

8.20 Type: SendingTime

Name	Version	Block Length	Description
SendingTime	2	8	Time of message transmission, expressed in UTC.

Name	Data Type	Size	Offset	Description
time	uint64	8	0	UTC timestamp with nanosecond precision (Unix Epoch).
unit	uint8	0	8	time unit (nanoseconds). Constant: 9.

8.21 Type: TextEncoding

Name	Version	Block Length	Description
TextEncoding	2	N	Free ASCII format text string.

Name	Data Type	Size	Offset	Description
length	uint8	1	0	Length of free format text string generated by exchange. Max value: 250.
varData	char	N	1	Free ASCII format text string generated by exchange.

8.22 Type: UTCTimestampNanos

Name	Version	Block Length	Description
UTCTimestampNanos	2	8	UTC timestamp with nanosecond precision.

Name	Data Type	Size	Offset	Description
time	uint64	8	0	UTC timestamp with nanosecond precision (Unix Epoch).
unit	uint8	0	8	time unit (nanoseconds). Constant: 9.

8.23 UTCTimestampNanosOptional

Name	Version	Block Length	Description
UTCTimestampNanosOptional	2	8	Optional UTC timestamp with nanosecond precision.

Name	Data Type	Size	Offset	Description
time	uint64	8	0	UTC timestamp with nanosecond precision (Unix Epoch).
unit	uint8	0	8	time unit (nanoseconds). Constant: 9.

8.24 InvestorID

Name	Version	Block Length	Description
InvestorID	2	8	Self-trade prevention investor identification is composed of the prefix and document.

Name	Data Type	Size	Offset	Description
prefix	uint16	2	0	Prefix is a user-defined value and needs to be within pre-defined range depending on the type of document.
<padding>		2	2	For future use. Until there, fill with nulls (0x00 0x00).
document	uint32	4	4	Document number whose value depends on the class of the investor (corporate resident, individual resident or non-resident investor). Max value: 999999999.

9 ENUMERATIONS

Null values are present in all types but should be used only for optional fields.

Enum Type	Type	Null Value	Description / Value Domain
AccountType	uint8	0	Type of Account associated with an order. 38 - REMOVE_ACCOUNT_INFORMATION 39 - REGULAR_ACCOUNT
AllocNoOrdersType	char	0	Indicates how the orders being booked and allocated by an Allocation Instruction. "0" - NOT_SPECIFIED
AllocReportType	char	0	Describes the specific type or purpose of an Allocation Report message. "8" - REQUEST_TO_INTERMEDIARY
AllocStatus	char	0	Identifies status of allocation. "0" - ACCEPTED "5" - REJECTED_BY_INTERMEDIARY
AllocTransType	char	0	Identifies allocation transaction type. "0" - NEW "2" - CANCEL
AllocType	char	0	Describes the specific type or purpose of an Allocation message. "8" - REQUEST_TO_INTERMEDIARY
Boolean	uint8	255	Boolean type 0 - FALSE_VALUE: false, N, 0 1 - TRUE_VALUE: true, Y, 1
CancelOnDisconnectType	uint8	255	Criteria used to initiate cancel on disconnect mechanism by the gateway. 0 - DO_NOT_CANCEL_ON_DISCONNECT_OR_TERMINATE 1 - CANCEL_ON_DISCONNECT_ONLY 2 - CANCEL_ON_TERMINATE_ONLY 3 - CANCEL_ON_DISCONNECT_OR_TERMINATE
CrossedIndicator	uint16	0	1001 - STRUCTURED_TRANSACTION 1002 - OPERATIONAL_ERROR 1003 - TWAP_VWAP
CxlRejResponseTo	uint8	255	Identifies the type of request that this cancel reject is in response to. 0 - NEW 1 - CANCEL 2 - REPLACE

EstablishRejectCode	uint8	255	Identifies the code of reject establishment. 0 - UNSPECIFIED 1 - CREDENTIALS 2 - UNNEGOTIATED 3 - ALREADY_ESTABLISHED 4 - SESSION_BLOCKED 5 - INVALID_SESSIONID 6 - INVALID_SESSIONVERID 7 - INVALID_TIMESTAMP 8 - INVALID_KEEPALIVE_INTERVAL 9 - INVALID_NEXTSEQNO 10 - ESTABLISH_ATTEMPTS_EXCEEDED 20 - ESTABLISH_NOT_ALLOWED 21 - DUPLICATE_SESSION_CONNECTION 22 - AUTHENTICATION_IN_PROGRESS 23 - PROTOCOL_VERSION_NOT_SUPPORTED
ExecRestatementReason ValidForSingleCancel	uint8EnumEncoding	0	Used to communicate a reason for a solicited cancel. 203 - CANCEL_ORDER_DUE_TO_OPERATIONAL_ERROR
ExecRestatementReason ValidForMassCancel	uint8	255	Used to communicate event type which triggers mass cancelation. 202 - ORDER_MASS_ACTION_FROM_CLIENT_REQUEST 207 - MASS_CANCEL_ORDER_DUE_TO_OPERATIONAL_ERROR_REQUEST
ExecRestatementReason	uint8EnumEncoding	0	Indicates reason of cancelation, if available. 8 - MARKET_OPTION 100 - CANCEL_ON_HARD_DISCONNECTION 101 - CANCEL_ON_TERMINATE 102 - CANCEL_ON_DISCONNECT_AND_TERMINATE 103 - SELF_TRADING_PREVENTION 105 - CANCEL_FROM_FIRMSOFT 107 - CANCEL_RESTING_ORDER_ON_SELF_TRADE 200 - MARKET_MAKER_PROTECTION 201 - RISK_MANAGEMENT_CANCELLATION 202 - ORDER_MASS_ACTION_FROM_CLIENT_REQUEST 203 - CANCEL_ORDER_DUE_TO_OPERATIONAL_ERROR 204 - ORDER_CANCELLED_DUE_TO_OPERATIONAL_ERROR

			205 - CANCEL_ORDER_FIRMSOFT_DUE_TO_OPERATIONAL_ERROR 206 - ORDER_CANCELLED_FIRMSOFT_DUE_TO_OPERATIONAL_ERROR 207 - MASS_CANCEL_ORDER_DUE_TO_OPERATIONAL_ERROR_REQUEST 208 - MASS_CANCEL_ORDER_DUE_TO_OPERATIONAL_ERROR_EFFECTIVE 209 - CANCEL_ON_MIDPOINT_BROKER_ONLY_REMOVAL
ExecType	char	0	Describes the action that triggered this specific Execution Report - see the OrdStatus (39) tag for the current order status (e.g., Partially Filled). "F" - TRADE "H" - TRADE_CANCEL
ExecuteUnderlyingTrade	char	0	Specifies if a simultaneous trade of the underlying is to be performed. "0" - NO_UNDERLYING_TRADE "1" - UNDERLYING_OPPOSING_TRADE
FlowType	uint8	255	Type of message flow from client to server or from server to client. 0 - NONE 1 - RECOVERABLE 2 - UNSEQUENCED 3 - IDEMPOTENT
MassActionRejectReason	uint8	255	Reason Order Mass Action Request was rejected. 0 - MASS_ACTION_NOT_SUPPORTED 8 - INVALID_OR_UNKNOWN_MARKET_SEGMENT 99 - OTHER
MassActionResponse	char	0	Specifies the action taken by matching engine when it receives the Order Mass Action Request. "0" - REJECTED "1" - ACCEPTED
MassActionScope	uint8	0	Specifies the scope of the action. All Day and MOC orders will be cancelled. GTC, GTD and MOA orders will not be cancelled. 6 - ALL_ORDERS_FOR_A_TRADING_SESSION
MassActionType	uint8	255	Specifies the type of action requested. 2 - RELEASE_ORDERS_FROM_SUSPENSION 3 - CANCEL_ORDERS 4 - CANCEL_AND_SUSPEND_ORDERS 5 - SESSION_GROUP_QUERY
MessageType	uint8	255	Defines message type. 0 - Negotiate 1 - NegotiateResponse 2 - NegotiateReject

			3 - Establish 4 - EstablishAck 5 - EstablishReject 6 - Terminate 9 - NotApplied 10 - RetransmitRequest 11 - Retransmission 12 - RetransmitReject 13 - Sequence 14 - BusinessMessageReject 15 - SimpleNewOrder 16 - SimpleModifyOrder 17 - NewOrderSingle 18 - OrderCancelReplaceRequest 19 - OrderCancelRequest 20 - NewOrderCross 21 - ExecutionReport_New 22 - ExecutionReport_Modify 23 - ExecutionReport_Cancel 24 - ExecutionReport_Trade 25 - ExecutionReport_Reject 26 - ExecutionReport_Forward 27 - SecurityDefinitionRequest 28 - SecurityDefinitionResponse 29 - OrderMassActionRequest 30 - OrderMassActionReport 31 - QuoteRequest 32 - QuoteStatusReport 33 - Quote 34 - QuoteCancel 35 - QuoteRequestReject 36 - PositionMaintenanceCancelRequest 37 - PositionMaintenanceRequest 38 - PositionMaintenanceReport 39 - AllocationInstruction 40 - AllocationReport

MultiLegReportingType	char	0	Used to indicate what an Execution Report represents. "1" - SINGLE_SECURITY "2" - INDIVIDUALLEG_OF_MULTILEG_SECURITY "3" - MULTILEG_SECURITY
NegotiationRejectCode	uint8	255	Identifies the code of reject negotiation. 0 - UNSPECIFIED 1 - CREDENTIALS 2 - FLOWTYPE_NOT_SUPPORTED 3 - ALREADY_NEGOTIATED 4 - SESSION_BLOCKED 5 - INVALID_SESSIONID 6 - INVALID_SESSIONVERID 7 - INVALID_TIMESTAMP 8 - INVALID_FIRM 20 - NEGOTIATE_NOT_ALLOWED 21 - DUPLICATE_SESSION_CONNECTION 22 - AUTHENTICATION_IN_PROGRESS 23 - PROTOCOL_VERSION_NOT_SUPPORTED
OrdStatus	char	0	Identifies current status of order. "0" - NEW "1" - PARTIALLY_FILLED "2" - FILLED "4" - CANCELED "5" - REPLACED "8" - REJECTED "C" - EXPIRED "R" - RESTATED "Z" - PREVIOUS_FINAL_STATE
OrdType	char	0	Order type. "1" - MARKET "2" - LIMIT "3" - STOP_LOSS "4" - STOP_LIMIT "K" - MARKET_WITH_LEFTOVER_AS_LIMIT "W" - RLP "P" - PEGGED_MIDPOINT

OrderCategory	char	0	Defines the type of interest behind a trade i.e., why a trade occurred. "B" - RESULT_OF_OPTIONS_EXERCISE "C" - RESULT_OF_ASSIGNMENT_FROM_AN_OPTIONS_EXERCISE "D" - RESULT_OF_AUTOMATIC_OPTIONS_EXERCISE "E" - RESULT_OF_MIDPOINT_ORDER "F" - RESULT_OF_BLOCK_BOOK_TRADE "G" - RESULT_OF_TRADE_AT_CLOSE "H" - RESULT_OF_TRADE_AT_AVERAGE
PosMaintAction	char	0	Maintenance Action to be performed. "1" - NEW "3" - CANCEL
PosMaintStatus	char	0	Status of Position Maintenance Request. "0" - ACCEPTED "2" - REJECTED "3" - COMPLETED "9" - NOT_EXECUTED
PosTransType	uint8	255	Identifies the type of position transaction. 1 - EXERCISE 105 - AUTOMATIC_EXERCISE 106 - EXERCISE_NOT_AUTOMATIC
PosType	char	0	Used to identify the type of quantity. "T" - TRANSACTION_QUANTITY "S" - START_OF_DAY_QTY "E" - OPTION_EXERCISE_QTY "B" - BLOCKED_QTY "U" - UNCOVERED_QTY "C" - COVERED_QTY
QuoteCancelType	uint8	0	Identifies the type of quote cancel. 5 - CANCEL_FOR_QUOTE_ID
QuoteStatus	uint8	255	Identifies the status of the quote acknowledgement. 7 - EXPIRED 0 - ACCEPTED 5 - REJECTED 9 - QUOTE_NOT_FOUND 10 - PENDING 11 - PASS

			17 - CANCELED
QuoteStatusResponseTo	char	0	Identifies the type of request that a Quote Status Report is in response to "0" - QUOTE "1" - QUOTE_REQUEST "2" - QUOTE_CANCEL "3" - QUOTE_REQUEST_REJECT
RetransmitRejectCode	uint8	255	Identifies the code of reject retransmission. 0 - OUT_OF_RANGE 1 - INVALID_SESSION 2 - REQUEST_LIMIT_EXCEEDED 3 - RETRANSMIT_IN_PROGRESS 4 - INVALID_TIMESTAMP 5 - INVALID_FROMSEQNO 9 - INVALID_COUNT 10 - THROTTLE_REJECT 11 - SYSTEM_BUSY
RoutingInstruction	uint8	0	Indicates additional order instruction. 1 - RETAIL_LIQUIDITY_TAKER 2 - WAIVED_PRIORITY 3 - BROKER_ONLY 4 - BROKER_ONLY_REMOVAL
SecurityIDSource	char	0	 "4" - ISIN "8" - EXCHANGE_SYMBOL
SecurityResponseType	uint8	255	Type of Security Definition message response. 1 - ACCEPT_SECURITY_PROPOSAL_AS_IS 5 - REJECT_SECURITY_PROPOSAL 2 - ACCEPT_SECURITY_AS_PROPOSAL_WITH_REVISIONS
SelfTradePreventionInstruction	uint8	255	Indicates which order should be canceled due to self-trade prevention. 0 - NONE 1 - CANCEL_AGGRESSOR_ORDER 2 - CANCEL_RESTING_ORDER 3 - CANCEL_BOTH_ORDERS
SettlType	char	0	Indicates who in the contract has control over evoking settlement. "0" - BUYERS_DISCRETION "8" - SELLERS_DISCRETION

			"X" - MUTUAL
Side	char	0	Side of order. "1" - BUY "2" - SELL
SimpleOrdType	char	0	Order type. "1" - MARKET "2" - LIMIT
SimpleTimeInForce	char	0	Specifies how long the order remains in effect. "0" - DAY "3" - IMMEDIATE_OR_CANCEL "4" - FILL_OR_KILL
TerminationCode	uint8	255	Identifies the code of termination. 0 - UNSPECIFIED 1 - FINISHED 2 - UNNEGOTIATED 3 - NOT_ESTABLISHED 4 - SESSION_BLOCKED 5 - NEGOTIATION_IN_PROGRESS 6 - ESTABLISH_IN_PROGRESS 10 - KEEPALIVE_INTERVAL_LAPSED 11 - INVALID_SESSIONID 12 - INVALID_SESSIONVERID 13 - INVALID_TIMESTAMP 14 - INVALID_NEXTSEQNO 15 - UNRECOGNIZED_MESSAGE 16 - INVALID_SOFH 17 - DECODING_ERROR 20 - TERMINATE_NOT_ALLOWED 21 - TERMINATE_IN_PROGRESS 23 - PROTOCOL_VERSION_NOT_SUPPORTED 30 - BACKUP_TAKEOVER_IN_PROGRESS
TimeInForce	char	0	Specifies how long the order remains in effect. "0" - DAY "1" - GOOD_TILL_CANCEL "3" - IMMEDIATE_OR_CANCEL "4" - FILL_OR_KILL

			"6" - GOOD_TILL_DATE "7" - AT_THE_CLOSE "A" - GOOD_FOR_AUCTION
possResend	uint8	255	Indicates that message may contain information that has been sent under another sequence number. 0 - FALSE_VALUE: false, N, 0 1 - TRUE_VALUE: true, Y, 1