MARKET DATA B3: BINARY UMDF

Message Reference – version 2.2.0

Last modified: August 7th, 2025

1 DESCRIPTION

This document describes the messages and values that can be sent on the new Market Data Platform in binary format.

1.1 Benefits

The new binary Market Data feed will provide the following benefits:

- SBE messaging protocol.
- Execution Summary message.

1.2 Contact Information

- Contact Services Department handles all requests for connectivity setup and general exchange supported services.
    - contratacao@b3.com.br
    - +55 11 2565-5081
- Certification and Testing Center: performs certification of all software solutions applying for EntryPoint connectivity.
    - tradingcertification@b3.com.br
    - +55 11 2565-5029
- Trading Support Department (GSN): provides real time connectivity monitoring and troubleshooting.
    - tradingsupport@b3.com.br
    - +55 11 2565-5021

2 CHANGE LOG

Date	Version	Description	Author
Mar. 4th, 2020	1.0.0	- First version.	RDRC, AYSF
Apr. 28th, 2020	1.1	- Inclusion of LastMsgSeqNumProcessed for Snapshot messages. - Reinclusion of Heartbeat Message (distinct from SequenceReset). - Inclusion of Book Reset Message. - Inclusion of Market Data Incremental Refresh – Price message. - Parts of this document are generated by processing the SBE template.	RDRC, EEW
Jun. 25th, 2020	1.2	- Removing duplicate messages. - Adding a message for Security Status and Phase. - Adding a LastTradePrice. - Adding messages for High Price and Low Price. - Template version: 1.1	EEW
Feb. 16th, 2022	1.3	- Removing MBE messages. - Streamlining several messages. - Restoring some fields from UMDF FIX/FAST spec. - Template version: 1.3	EEW, RNKH
Mar. 3rd, 2022	1.3.1	- Updating messages TheoreticalOpeningPrice and LastTradePrice. - Template version: 1.3.1	EEW
Apr. 6th, 2022	1.3.2	- Minor edits.	EEW
Apr. 28th, 2022	1.3.3	- Fields reordered and padding added. Some common fields are placed at fixed offsets. - Fields added to or removed from messages: ClosingPrice, DeleteOrder, EmptyBook, LastTradePrice, Trade, News. - Template Version: 1.3.3	EEW, RNKH
May 30th, 2022	1.4.0	- New messages ExecutionSummary and ExecutionStatistics, GroupPhaseStatus, MassDeleteOrders, ForwardTrade. - Fields added to or removed from messages: Trade. - Fields reordered. - Template Version: 1.4.0	RNKH, EEW
Jun. 23th, 2022	1.4.1	- Field and message paddings are shown in the message descriptions. - Template Version: 1.4.0	EEW
Jul. 7th, 2022	1.5.0	- New messages (ChannelReset, TradeBust). - New fields: RptSeq, LastRptSeq, NumberOfTrades. - Fields renamed: OrderID → SecondaryOrderID.	EEW, RKNH
Sep. 13th, 2022	1.5.1	- mDEntryTimestamp and rptSeq fields moved to the end of the message. - streamID field removed in all messages. - mDEntryPositionNo field is required now. - mDEntryPositionNo and enteringFirm fields moved just after mDEntrySize field in the Order_MBO message. - tradeDate field is required now and also included in the ExecutionStatistics message. - totNumReport field removed from SecurityDefinition message. - priceLimitType field in PriceBand message is required now.	RNKH, EEW

Date	Version	Description	Author
		- priceBandType field in PriceBand message is required now. - mDInsertTimestamp field in SnapshotFullRefresh_Orders_MBO message is required now. - tradeCondition field is removed from TradeBust message. - totNumStats field type (in SnapshotFullRefresh_Header message) is uint16 (from uint8).	
Nov. 17th, 2022	1.5.2	- LocalMktDate32 and LocalMktDate32Optional types added to support '9999-12-31' date in FIX. - UTCTimestampSeconds type added to support '9999-12-31 23:59:59 UTC' timestamp in FIX. - issueDate, maturityDate, startDate, endDate, settlDate, datedDate fields changed to LocalMktDate32 or LocalMktDate32Optional type. - securityValidityTimestamp field changed to UTCTimestampSeconds type. - OpeningPrice message length set from 42 to 44 (4-byte alignment).	RNKH
Jan. 6th, 2023	1.5.3	- Removed NO_CHANGE value in SecurityTradingEvent type (related fields are now optional). - Removed NO_PRICE_BAND value in PriceBandType type (related fields are now optional). - Removed PREVIOUS_VALUE value in SecurityTradingStatus type. - Removed PREVIOUS_VALUE value in TradingSessionSubID type. - Removed matchEventIndicator (tag 37035) field in the ExecutionSummary message. - mDEntrySize (tag 271) field in the DeleteOrder_MBO message is now optional. - mDEntrySize (tag 271) field in the AuctionImbalance message is now optional. - mDEntrySize (tag 271) and mDEntryPx (tag 270) fields in the TheoreticalOpeningPrice message are now optional. - priceBandType (tag 6939), priceLimitType (tag 1306) and priceBandMidpointPriceType (tag 37008) fields in the PriceBand message are now optional. - Adding 3 new items in the SecuritySubType domain related to Block Trading instruments.	RNKH
Feb. 16th, 2023	1.5.4	- Adding minCrossQty (tag 35561) field in the SecurityDefinition message. - More clarification on alignment and padding in SBE.	RNKH
Apr. 13th, 2023	1.5.4.1	- Correcting the declaration of exponent of Price type to -4 from -3 (documentation only).	RNKH
Apr. 28th, 2023	1.5.5	- Type of *mDEntryPx* field in the *ClosingPrice* message changed to 8 decimals. - Type of *netChgPrevDay* field in the *OpeningPrice* and *ExecutionStatistics* messages changed to 8 decimals. - *PriceType* enum changed: DECIMALS value removed, and null value changed from 255 to 0. - Primitive types removed from fields: changed to an equivalent declared type. - SecurityIDSource type converted from a custom type to enumeration, but the presence attribute is kept as constant in all but SecurityDefinition message.	RNKH
May 11th, 2023	1.5.6	- mDEntryType field position changed from 9 to 10 in DeleteOrder_MBO message (side 'field' in the same position across messages). - aggressorSide field position changed from 8 to 10 in ExecutionSummary message (side 'field' in the same position across messages).	RNKH
Jun. 7th, 2023	1.6.0	- Schema version changed from 6 to 7 to support optional trdSubType field. - tradeCondition type streamlined, non-regular trade types moved to TrdSubType type. - trdSubType field added to LastTradePrice, Trade and ForwardTrade messages with sinceVersion=7. - Size of LastTradePrice and ForwardTrade messages increased to 68 bytes to include trdSubType field. - Trade at Close and Trade at Average values included in the TrdSubType enum. - Domain of GovernanceIndicator type reduced to reflecting the same domain in the listed companies registration system. - ImbalanceCondition type created with the same original bit position for ImbalanceMoreBuyers and ImbalanceMoreSellers as they were in the TradeCondition type. - imbalanceCondition field (ImbalanceCondition type) used instead of tradeCondition field in the AuctionImbalance message.	RNKH

Date	Version	Description	Author
		- securitySubType field in SecurityDefinition message changed from SecuritySubType enum to uint16 type because new values are added frequently. We will update the Message Reference accordingly in the SecuritySubType values section. - ORDER_CROSS_ELIGIBLE and FLAG_RFQ_FOR_CROSS_ELIGIBLE values added in the domain of InstrAttribValue enum. - Explicit paddings were removed from DeleteOrder and ExecutionSummary messages.	
Nov., 17th, 2023	1.7.0	- Schema version changed from 7 to 8. - Templates: **12** (*SecurityDefinition*) and **22** (*PriceBand*) created to replace 4 (*SecurityDefinition*) and 20 (*PriceBand*) respectively. - Type of *minPriceIncrement* field in the new *SecurityDefinition* template changed from *PriceOptional* to *Fixed8*. - *indexPct* (tag 6919) and *indexTheoreticalQty* (tag 37021) fields removed from *noUnderlyings* group in the new *SecurityDefinition* template. - Type of *tradingReferencePrice* field in the new *PriceBand* template changed from *PriceOptional* to *Fixed8*. - Description of value 105 of *TrdSubType* type changed to: "RF_TRADE - Equities: RFQ Trade, Futures: Fixed Income Trade (RF).".	RNKH
Feb. 16th, 2024	1.7.0.1	- Price datatype definition fixed from 3 decimal to 4 decimal in section 4 (*Datatypes*).	RNKH
Feb. 27th, 2024	1.7.0.2	- Value 92 - Strategy Interest Rate included in the *SecuritySubType* field domain in section 13 (*SecuritySubType* values).	RNKH
Apr.,1st, 2024	1.8.0	- Schema version changed from 8 to 9. - 109 - SWEEP_TRADE added to *TrdSubType* type for Sweep and Cross support. - *SettlementPrice* (template id=28) and *OpenInterest* (template id=29) messages added. - More clarification in the description of *openCloseSettlFlag* field in the messages that uses this field.	RNKH
May 15th, 2024	1.8.0.1	- Typo *SecurityDefinition_4* fixed to *SecurityDefinition_12* in section 9.2.8.	RNKH
Aug. 14th, 2024	1.8.0.2	- Including a explicit 1-byte padding definition at the end of the *SettlementPrice* (template id=28) message.	RNKH
Aug. 15th, 2024	1.9.0	- Schema version changed from 9 to 10. - *impliedMarketIndicator* enum added to the end of *SecurityDefinition* message. - Implied flag defined at *matchEventIndicator* field in the *Order_MBO* message to inform if the order book entry is synthetic generated by the implied engine. - Implied flag defined at *matchEventIndicator* field in the *Trade* message to inform if the trade is resulting from a match that involves an implied order. - Template ids: 4 and 20, that were deprecated, are removed. - *SnapshotFullRefresh_Orders_MBO*: field *matchEventIndicator* included. - Descriptions of *transactTime*, *mdEntryTimestamp* and *maturityMonthYear* fields revised.	RNKH
Sep. 19th, 2024	1.9.0.1	- Complement the description of *mDEntrySize* field in the *DeleteOrder_MBO* message: "Absent if the deletion is the result of a matching event.". - Description of *matchEventIndicator* field equalized throughout the document.	RNKH
Oct. 15th, 2024	1.9.0.2	- More clarification for definition of *matchEventIndicator* field in each of message types that have this field.	RNKH
Apr. 16th, 2025	2.1.0	- **Schema version changed to 15.** - Order management based on price/secondaryOrderID. - *mDEntryPositionNo* field and *MDEntryPositionNo* type marked as deprecated and optional. - *mDEntrySize* field in the *DeleteOrder_MBO* is now present in all cases. - *transactTime* field replaces *mdEntryTimestamp* field in Order_MBO, DeleteOrder_MBO, MassDeleteOrders, ExecutionSummary, Trade, ForwardTrade and TradeBust messages.	RNKH

Date	Version	Description	Author
		- Optional *lastSequenceVersion* field (tag 37084) added at the end of *SnapshotFullRefresh_Header* message. - Optional *mDEntryPx* field of the deleted order included in the *DeleteOrder_MBO* message.	
Aug. 7th, 2025	2.2.0	- **Schema version changed to 16.** - *optPayoutType* enum added to the end of *SecurityDefinition* message. - *securitySubType* field: new value 160 (Crypto Asset). - MaturityMonthYear can be used by daily options. - *mDEntryPositionNo* field is replaced by a 4-byte padding in all messages (it was deprecated in version 15). - *mDEntryPrevSize* field (tag 37780) added to *Order_MBO* message. Present only in *mDUpdateAction*=CHANGE and represents the previous quantity of the order before the modification. - Changed behavior for the *Trade* and *ForwardTrade* message: if a trade is amended, the value of the *transactTime* field (tag 60) is set manually by MktOps.	EEW, RNKH

3 SIMPLE BINARY ENCODING

Binary UMDF is based on version 1.0 of the FIX Simple Binary Encoding (SBE). 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 complementary to other FIX standards for session protocol and application-level behavior.

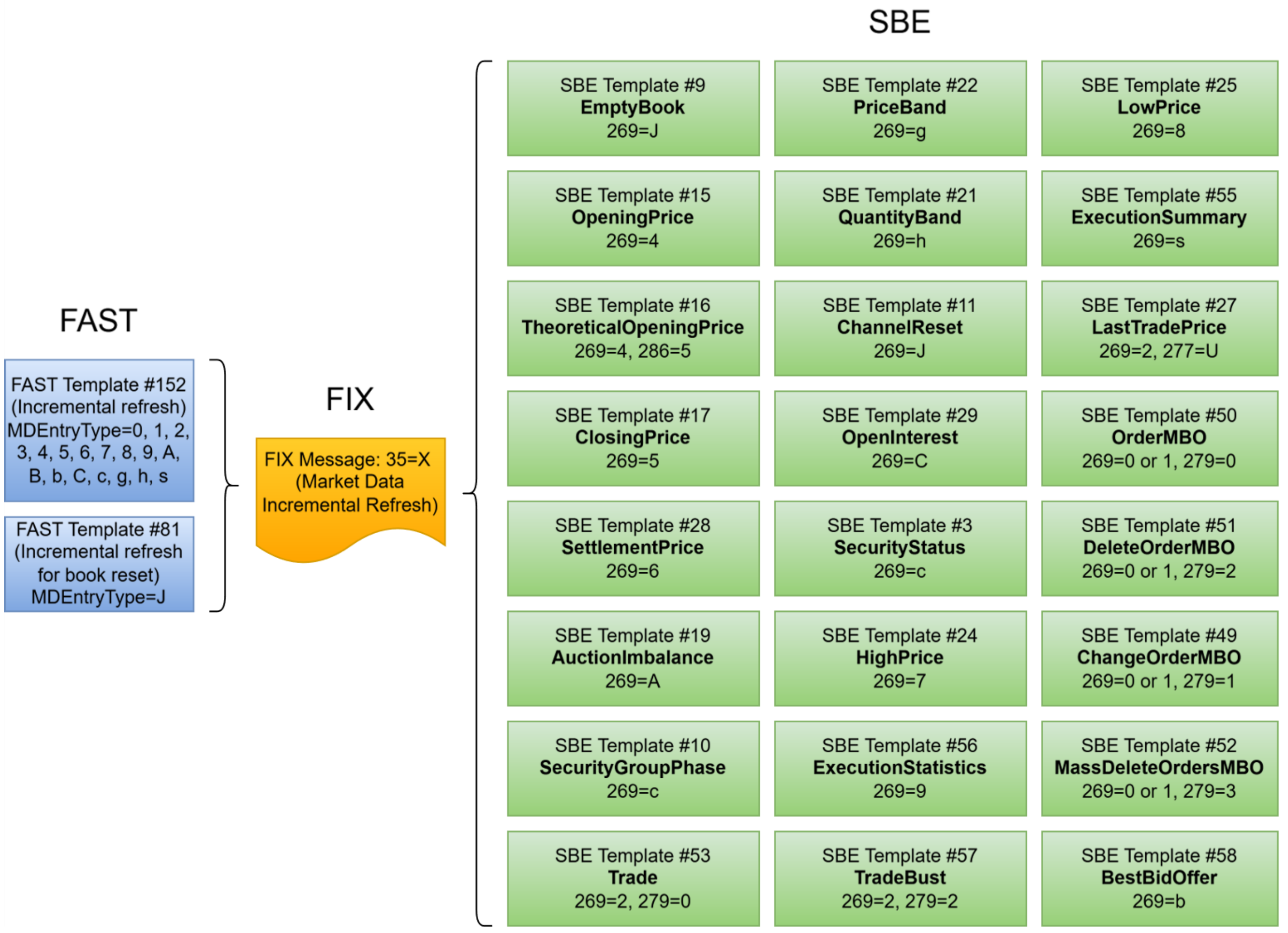
3.1 Differences between FIX/FAST and FIX/SBE

Feature	FIX/FAST	FIX/SBE	Comments
Message Size	Unlimited	Limited by packet size (1400 bytes, including headers)	A single message in FIX/FAST can be equivalent to several messages in FIX/SBE
Chunking (Multiple parts)	Required because messages can have unlimited size. It is done in the transport layer (technical header)	Application layer must handle multiple parts.	Only some messages (like News) have “parts” (continuations) in application layer.
Transport and Application Layer	MessageSequenceNumber and SendingTime are part of the message (application layer)	SequenceNumber (actually “Packet Sequence Number”) and SendingTime are sent in the packet (transport layer).	If a message must be retransmitted, it doesn’t need to be reencoded just because its “SendingTime” is updated.
Bandwidth	It is the main driver for the design of the FAST encoding.	SBE prioritizes ease of encoding and fast processing.	If SBE templates are correctly designed, bandwidth savings can be significant, but usually SBE messages are slightly larger than FAST messages.

Feature	FIX/FAST	FIX/SBE	Comments
Fixed Message Size	Almost all messages have variable message size.	Variable message size only for low-traffic, large messages; fixed message size is preferred.	Messages with fixed size have simpler and faster processing.
String Fields	String size is unlimited. Encoding for non-ASCII strings is not explicitly specified.	Fixed byte size for strings is preferred. Encoding is explicitly specified.	Some string fields were "shortened" to a more practical size.
Message Types	Few message types, with several message subtypes	One or more message types for each use case	Each MDEntryType in FIX/FAST corresponds to one or more SBE message types
Field Types	Few message types. The String type is overloaded.	Enums, sets and fixed-size strings help to specialize the field types.	Space savings; field validation; help to keep message size fixed.
Constant Fields	Constant fields (that occupy 0 bytes) are not very common in FAST because a single message has several use cases.	Constant fields are very useful in SBE because they replicate FIX fields (like MD-EntryType) without using space.	Space savings; speed.
Optional Fields	Encoded differently; preferred in FAST because sometimes omitted fields occupy just 1 or 2 bits in message.	Optional fields are encoded exactly like required fields (occupying the same space). The only difference is the special value for 'null'.	Because optional fields don't save space, lots of message types (according to use cases) are defined.
Default Values	FAST can define default values for fields or components of fields	SBE has no "default values", only "null values" for optional fields. The application can choose to replace the null value with the default value.	
Timestamps	UTC, Decimal, milliseconds. Ex: Jan 31, 2022 12:38:02.456 = 20220131123802456	Nanoseconds since Jan 1, 1970 00:00:00 UTC. Example: Jan 31, 2022 12:38:02.456000000 = 1643632682456000000	Faster processing because it avoids converting to and from human-readable notation.

Feature	FIX/FAST	FIX/SBE	Comments
Date and Time Fields	Date and time sometimes come in separate fields. Example: MDEntryDate=20220131 and MDEntryTime=123802456	Replaced by UTC Timestamps in a single field. Example: MDEntryTimestamp = 1643632682456000000	Faster processing because it avoids separating the same information in two distinct fields.
Repeating Groups with few repetitions	Faithfully replicates FIX (Text) definition	Some repeating groups were “linearized” or “flattened” if possible	Faster processing because it avoids processing repeating groups. Example: SecurityAltID vs ISINNumber and ClearingHouse-ID.
Repeating Groups that represent large lists of related information	Faithfully replicates FIX (Text) definition	The list is represented by a sequence of messages. Sometimes it requires a “header message” as well.	SecurityList (35=y) → several SecurityDefinition (35=d), each one with a single instrument. Snapshot (35=W) → SnapshotHeader and Incremental (35=X)
Messages with variable size	Faithfully replicates FIX (Text) definition	The average message can usually fit a single packet. Some exceptionally large messages require multiple parts.	Fields “PartCount” and “PartNumber” are present in News messages.
Trivially redundant information	Faithfully replicates FIX (Text) definition	Avoided if possible	Fields “Text” and “EncodedText” in News messages were combined into a single field “Text” with UTF-8 encoding.
Combined Fields	Mutually exclusive fields (for instance, MDEntry-Buyer and MDEntrySeller in an Order message) are present. Because optional fields are cheap, they are both defined.	Combine the information into a single field.	MDEntryBuyer and MDEntrySeller were combined in a single field “EnteringFirm” in an Order message

3.2 Mapping FIX/FAST Messages to FIX/SBE Messages



A single kind of FIX Message can be mapped to several types of FAST or SBE messages.

For instance, the FIX "Market Data Incremental Refresh" message is mapped to 2 distinct types of FAST Messages (template #152 for the general case, and #81 for the special use case "Book Reset" with no *SecurityID*). Because optional fields are cheap in FAST (sometimes using just a single bit in message), almost all fields can be defined as optional and very few templates are required for the same FIX message.

For SBE, we map each use case to a different SBE message type, because it helps to organize use cases and keeps message size small. SBE penalizes defining optional, unused fields, because they occupy the same space of mandatory fields, so it is better to define specialized types of messages.

Usually, each distinct *Market Data Entry Type* corresponds to a distinct use case (for instance, *MDEntryType* = 7 (Session High Price) is associated to the SBE message whose template is #24.

Sometimes, a single *Market Data Entry Type* can be associated to several different SBE messages, depending on use cases, or presence of certain fields. For instance, there are four different SBE messages for *MDEntryType* value = 2 (Trade): one for a real trade (template #53), one for a trade on forward ("termo") instruments (template #54), that includes two additional fields, one for the price of the last trade (template #27) and one for the removal operation ("trade bust", template #57).

There are two distinct *MDEntryTypes* values (0=Bid and 1=Offer) that are associated, in FIX, with a single use case (New/Modify/Delete orders). In Binary UMDF, there will be four SBE messages: one for new, one for modified orders, one for deleting a single order, and one for deleting all orders from a side of the book.

3.3 Handling of SBE Enums and Sets

Applications must tolerate enumeration values that are not specified in the client's template. For instance, let's say that a field with *SecurityType* field (that assumes the values from 1 to 17 today) is being sent by B3 with value 18, and the client's template does not list this value. The application can choose to ignore this value.

Applications must tolerate set choices that are not specified in the client's template. For instance, let's say that a field with *TradeCondition* field has its bit 10 set (*RFQTrade*), but the client's template is not updated to support it. The application can choose to ignore this value.

3.4 Data Endianness

All messages and headers are represented as little-endian, even packet headers (usually encoded as big-endian in network protocols), to make encoding consistent.

3.5 SBE, Optional Fields and Default Values, Empty Fields

Optional fields in SBE can hold 'null values', that an application can interpret as 'absence of contents (field is not set)'.

There are no 'default values' in SBE. Applications can replace 'null values' with 'default values' if required but it is not in the protocol specification.

Optional fields in SBE do not save space at all. If a field is defined as an optional *int32* (4 bytes), it always occupies 4 bytes, even though the contents of the field are 'null' (no defined value).

For optional fields, there is a value that represents the NULL value (the field is not set). It can be specified in the declaration of their types.

If not explicitly defined in the type, the ‘null’ value is assumed as default for the primitive type: for unsigned fields, it is the largest possible value (something like 0xFFFF… in hexadecimal); for signed fields, it is the most negative value (something like 0x8000… in hexadecimal); for char fields, it is always the binary value 0 (‘\0’).

For enumeration fields, it depends on the ‘encodingType’ attribute (for instance, for the *LotType* enum, whose encoding type is ‘*uint8*’, the ‘null’ value is 255; for the *SecurityUpdateAction* enum, whose encoding type is ‘char’, the ‘null’ value is NUL (‘\0’).

For some types, the encoding for the 'null value' is not the default, but it is specified in the ‘nullValue’ attribute for the type. For instance, the type of the '*enteringFirm*' field (tag 37501) is *FirmOptional*, whose ‘nullValue’ attribute is "0". For such field, the value "0" represents null, not zero.

There are two types of strings in SBE: variable-length strings and fixed-length strings. Null string and empty strings are encoded the same way. For variable-length strings, the field ‘length’ is 0 for null (empty) strings. For fixed-length strings, if the content is shorter than the specified length, must be padded with NUL (‘\0’) character.

From the SBE specification, the default values for *null* values are:

Primitive Type	Value	Decimal	Hexadecimal
int8	-128	-128	0x80
uint8	255	255	0xFF
int16	-32768	-32768	0x8000
uint16	65535	65535	0xFFFF
int32	-2^{31}	-2147483648	0x80000000
uint32	$2^{32}-1$	4294967295	0xFFFFFFFF
int64	-2^{63}	-9223372036854775808	0x8000000000000000
uint64	$2^{64}-1$	18446744073709551615	0xFFFFFFFFFFFFFFFF
char	$0\ (ASCII\ NULL)$	0	'\0'
decimal (int32 mantissa)	Mantissa: -2^{31}	-2147483648	0x80000000
decimal (int64 mantissa)	Mantissa: -2^{63}	-9223372036854775808	0x8000000000000000

3.6 Decimal 'null' values

For instance, the type 'Price' is defined as a Decimal; the mantissa type is '*int64*' and the exponent is fixed as -4. A price like '+12.34' is encoded as the long value '123400' (it is 12.34 times 10^4, or 10000) but the *null* price (that can be found in market orders, that have no defined price) is encoded as the special value 0x8000000000000000, or -9223372036854775808.

For some decimal fields, it is possible to have null values encoded as '0' instead. Typically, they are values that are strictly positive (cannot assume the value 0.0). Check the SBE template: searching for *nullValue* attribute in type declaration.

4 DATATYPES

Type	Description	Size (Bytes)
uint8	8-bit Unsigned Integer (Little Endian encoding)	1
int8	8-bit Signed Integer (Little Endian encoding)	1
uint16	16-bit Unsigned Integer (Little Endian encoding)	2
int16	16-bit Signed Integer (Little Endian encoding)	2
uint32	32-bit Unsigned Integer (Little Endian encoding)	4
int32	32-bit Signed Integer (Little Endian encoding)	4
uint64	64-bit Unsigned Integer (Little Endian encoding)	8
int64	64-bit Signed Integer (Little Endian encoding)	8
Boolean	True (1) or False (0)	1
char	1 ASCII character	1
char(N)	Fixed length string padded on the right with null bytes. Encoding is US-ASCII, except when specified by the template.	N
Price	A decimal fixed-point number with **scale = 4**, representing a price. For instance, $1.23 will be represented by 12300.	8
Qty	A 32-bit signed integer, representing a quantity.	8
UTCTimestampNanos	Number of nanoseconds since UNIX Epoch (January 1st, 1970, UTC) with millisecond accuracy. Example: 1582821143123000000 represents Thursday, February 27, 2020 16:32:23.123 in UTC.	8
ISINNumber	ISO 6166 12-letter ISIN (International Securities Identification Number). Example: BRB3SAACNOR6	12
LanguageCode	ISO 639-1 two-letter language code. Example: “pt” (Portuguese), “en” (English)	2
LocalMktDate	Local date (as opposed to UTC). Number of days since UNIX Epoch (January 1st, 1970). Example: 18319 represents February 27, 2020.	2

MonthYear	Local year and month (as opposed to UTC). Represented with two subfields: year (0000 – 9999) and month (1 – 12).	3
Currency	3-letter *alphabetic* ISO Currency Codes (ISO 4217). Example: BRL, USD.	3
Fixed(n)	A decimal fixed-point number with exponent -n (i.e., n places after the decimal separator). For instance, the number 42.28456973 will be represented as a Fixed(8) with mantissa = 4228456973 and exponent = -8.	8
Percentage	A decimal fixed-point number with exponent = -4 (Basis points). For instance, a percentage of 1% is represented as 0.01 (Mantissa = 100, Exponent = -4) and 1 basis point is represented as 0.0001 (Mantissa = 1, Exponent = -4)	8
Percentage9	A decimal fixed-point number with exponent = -9 (used for index instruments). A percentage of 0.4877802% is the number 0.004877802 (Mantissa = 4877802, exponent = -9)	8
NumInGroup	Counter representing the number of entries in a repeating group. Must be positive. Because the messages here can't exceed 1400 bytes, we use a 1-byte counter (from 0 to 255), and 2 bytes for the "block length".	3
Enum	A single choice of mutually exclusive values. Depending on the range of values, it can be represented in 1 or 2 bytes. For instance, the values for the enum representing Side (tag 54) are Buy (1) and Sell (2); for SecurityType (tag 167), are CS(1), PS(2), CASH (3), OPT (4), and so on. The specific values can be found in the SBE Template schema file.	1 or 2
VarString	A variable-length string, encoded in UTF-8	2 + N

4.1 Constant Values

Constant values (specified in the template as presence="constant") aren't transmitted, because the values are already specified by the template. The wire size (number of bytes required to transmit the field) is always 0 (zero).

4.2 Field Placement and Alignment

The ordering of the fields in this document follows the same order of the fields in the current version of the SBE Template.

The fields for the incremental refresh messages are carefully aligned, according to their data type, to maximize the speed of access. The most important fields are placed in the first 64 bytes of each message (first cache line).

Alignment is especially important for speeding up FPGA processing. It is guaranteed by judicious placement of the fields, and padding. (In SBE the alignment and padding are done by specifying the field offsets explicitly (see more information at https://www.fixtrading.org/standards/sbe-online/#message-body), and by carefully defining the block length of the message (block length must be greater than or equal to the sum of the sizes of all fields in the message or group: see more detail of padding at the end of a message at https://www.fixtrading.org/standards/sbe-online/#padding-at-end-of-a-message-or-group). There are no ‘dummy fields’ for padding; the unused space can be reclaimed by a specification update). We highlighted field offsets and block lengths that are different than the sum of size of the fields in the Message Reference document for clarification.

There are fields that can be found in most incremental messages, so they are placed in the same offset if possible.

Examples:

Offset	Field	Trade	Order	ClosingPrice	QuantityBand
0	SecurityID	☑	☑	☑	☑
8	MatchEventIndicator	☑	☑	☑	☑

4.3 Common Type Mappings

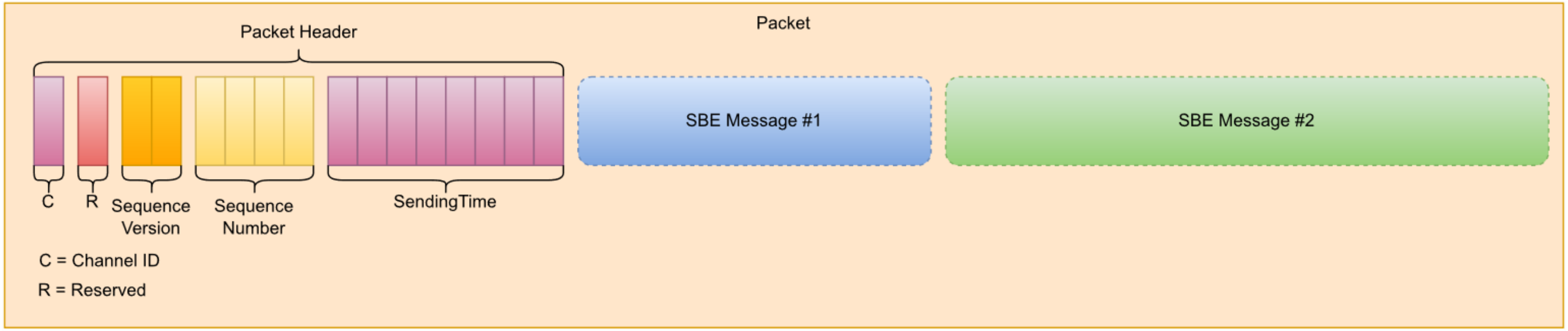
UMDF FIX/FAST	UMDF SBE	Reason	Example
UTCDateOnly and UTCTimeOnly	UTCTimestampNanos	Remove expensive conversions from decimal formats in Market Data; add nanosecond precision	Field MDEntryDate (tag 273) and MDEntryTime (274) were combined and mapped to MDEntryTimestamp (37033)
FIX MultipleValueString	Set	Save space (each alternative is represented by one bit). Validate the domain	Field TradeCondition (tag 277)
String or Integer fields with a limited domain	Enum	Save space (no unlimited reserved space) Validate the domain.	Field MDUpdateAction

5 HEADERS

5.1 Packet Header

Each packet (datagram) has one ***Packet Header***, and one or more messages inside.

The packet header is in *little-endian* format (the least significant values come first). The total size is 16 bytes.



Packet Header has the following fields:

Name	Type	Size (bytes)	Description
ChannelID	uint8	1	Channel ID
Reserved	uint8	1	Reserved.
SequenceVersion	uint16	2	Packet Sequence Version. Starts with 1 every week, incremented in case of failover events.
SequenceNumber	uint32	4	Packet Sequence Number.
SendingTime	uint64	8	UTC date and time of message transmission, in nanoseconds since Unix epoch (Jan 1st., 1970), accurate to microseconds.

5.2 Message Header

Each message in the packet starts with a *Message Header* that consists of the *Framing Header* and the ***SBE Message Header***.

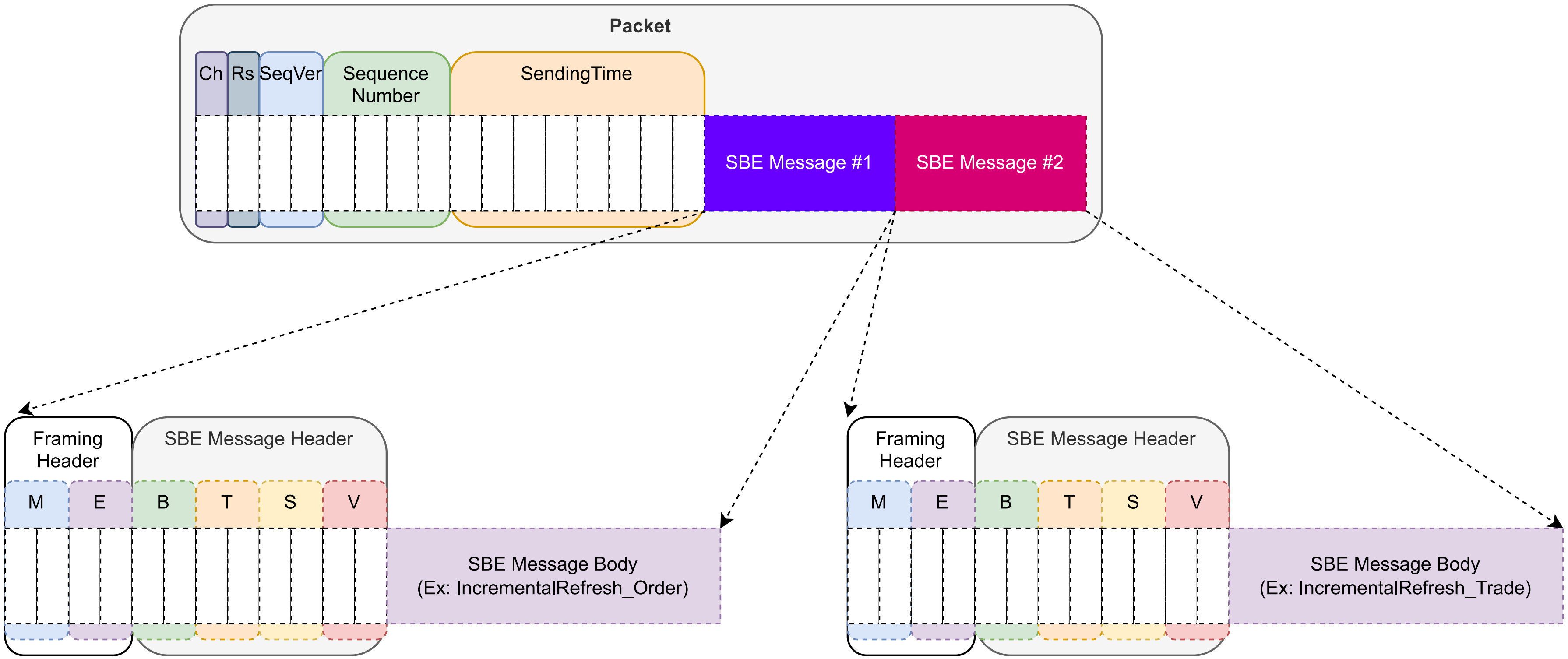
The message header is in little-endian format (the least significant values come first). The total size is 12 bytes.

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.

6 SAMPLE PACKET

A packet with two SBE messages is shown below.



This is the binary representation of a packet containing two SBE messages.

```
         +-------------------------------------------------+
         | 0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f |
+--------+-------------------------------------------------+----------------+
|00000000| 37 00 01 00 b1 68 de 3a 00 c8 98 65 f4 ac eb 15 |7....h.:...e....|
|00000010| 4c 00 50 eb 40 00 32 00 02 00 0F 00 a4 92 78 48 |L.P.@.2.......xH|
|00000020| 17 00 00 00 80 01 31 00 00 00 00 00 00 61 bc 00 |......1......a..|
|00000030| 00 00 00 00 78 e0 01 00 00 00 00 00 00 c8 98 65 |....x..........e|
|00000040| f4 ac eb 15 00 00 00 00 00 00 00 00 0a 00 00 00 |................|
|00000050| 39 30 00 00 00 00 00 00 00 00 00 00 44 00 50 eb |90..........D.P.|
|00000060| 38 00 35 00 02 00 0A 00 a4 92 78 48 17 00 00 00 |8.5.......xH....|
|00000070| 80 01 04 00 00 00 00 00 00 61 bc 00 00 00 00 00 |.........a......|
|00000080| 00 00 00 00 00 00 00 00 d0 07 00 00 81 e3 01 00 |................|
|00000090| 39 30 00 00 74 4a 45 00 00 c8 98 65 f4 ac eb 15 |90..tJE....e....|
+--------+-------------------------------------------------+----------------+
```

Legend:

Packet Header – 16-byte header for each packet.

Framing Header – 4-byte header for each SBE message.

SBE Message Header – 8-byte header for each SBE Body.

6.1 Packet Header

Offset	Length	Field	Hex bytes	Decoded value
0000	1	ChannelID	37	55 (Channel #55)
0001	1	Reserved	00	-
0002	2	SequenceVersion	01 00	1
0004	4	SequenceNumber	b1 68 de 3a	0x3ade68b1 = 987654321
0008	8	SendingTime	00 c8 98 65 f4 ac eb 15	0x15ebacf46598c800 = 1579546260000000000 = Jan 20, 2020 18:51:00.000000000

6.2 Framing Header #1

Offset	Length	Field	Hex bytes	Decoded value
0010	2	MessageLength	4C 00	0x004C = 76
0012	2	EncodingType	50 eb	SBE 1.0 Little-Endian

6.3 SBE Message Header #1

Offset	Length	Field	Hex bytes	Decoded value
0014	2	BlockSize	40 00	0x0040 = 64
0016	2	TemplateID	32 00	0x0032 = 50 (Order_50)
0018	2	SchemaID	02 00	0x0002 = 2
001A	2	SchemaVersion	0F 00	0x000F = 15
001C … 0053	72	SBE Message Body	a4 92 78…	

6.4 Framing Header #2

Offset	Length	Field	Hex bytes	Decoded value
0058	2	MsgLength	44 00	0x0044 = 68
005a	2	Encoding	50 eb	SBE 1.0 Little-Endian

6.5 SBE Message Header #2

Offset	Length	Field	Hex bytes	Decoded value
005C	2	BlockSize	38 00	0x0038 = 56
005E	2	TemplateID	35 00	0x0035 = 53 (Trade_53)
0060	2	SchemaID	02 00	0x0002 = 2
0062	2	SchemaVersion	0F 00	0x000F = 15
0064 … 009F	56	SBE Message Body	a4 92 78…	

7 SBE APPLICATION MESSAGES

7.1 Template

Name	Description	Semantic Version	Schema Version	Package	Byte Order
b3-market-data-messages.xml	SBE Template for Market Data Messages	2.2.0	16	b3.umdf.mbo.sbe	Little Endian

8 TYPES

Name	Data Type	Size	Description
Asset	char	6	Asset associated with the security, such as DOL, BGI, OZ1, WDL, CNI, etc.
blockLength	uint16	2	Root block length.
CFICode	char	6	Classification of Financial Instruments (CFI code) values, which indicate the type of security using the ISO 10962 standard.
channelNumber	uint8	1	Channel number.
ClearingHouseID	uint64	8	ClearingHouseID (SecurityAltID).
CountryCode	char	2	ISO 3166-1 alpha-2 country code.
Currency	char	3	3-letter alphabetic ISO Currency Codes (ISO 4217). Example: BRL, USD.
day	uint8	1	Day of month (1 to 31)
encodingType	uint16	2	Identifier of the encoding used in the message payload.
exponent	int8	0	Exponent (for fixed-point decimal numbers).
FirmOptional	uint32	4	Identifies the broker firm.
ISINNumber	char	12	ISIN Number (SecurityAltID).
LanguageCode	char	2	ISO 639-1 two-letter language code.
length	uint16	2	Length of a string, in bytes. For instance, the string 'Ação', converted to UTF-8, has 6 bytes, so length = 6.
LocalMktDate	uint16	2	Local date (as opposed to UTC). Number of days since UNIX Epoch (January 1st, 1970). Example: 18319 represents February 27, 2020.
LocalMktDateOptional	uint16	2	Optional local date (as opposed to UTC). Number of days since UNIX Epoch (January 1st, 1970). Example: 18319 represents February 27, 2020.
LocalMktDate32	Int32	4	Local date (as opposed to UTC). Number of days since UNIX Epoch (January 1st, 1970). Example: 18319 represents February 27, 2020.

Name	Data Type	Size	Description
LocalMktDate32Optional	int32	4	Optional local date (as opposed to UTC). Number of days since UNIX Epoch (January 1st, 1970). Example: 18319 represents February 27, 2020.
mantissa	int64	8	Mantissa (for fixed-point decimal numbers).
MarketSegmentID	uint8	1	Identifies the market segment. Required for all tradable instruments. Not present in equity indexes, ETF indexes, BTB and Option Exercise.
messageLength	uint16	2	Overall message length including headers to support framing.
month	uint8	1	Month (1 to 12)
NewsID	uint64	8	News ID.
NumberOfTrades	uint32	4	Contains the number of trades executed in the session.
numInGroup	uint8	1	A counter representing the number of entries in a repeating group.
OrderID	uint64	8	Unique identifier for Order as assigned by the exchange.
packetLength	uint16	2	Packet Length.
Quantity	int64	8	Quantity in order/trade.
QuantityOptional	int64	8	Optional quantity in order/trade.
QuantityVolume	int64	8	Volume Quantity.
QuantityVolumeOptional	int64	8	Optional volume quantity.
reserved	uint8	1	Reserved.
RptSeq	uint32	4	Sequence number per instrument update. It can be used to synchronize the snapshot with the incremental feed if the client is only interested in a subset of the channel's instruments.
SecurityExchange	char	4	Security Exchange Code.
SecurityExchangeBVMF	char	0	SecurityExchange (constant BVMF).
SecurityGroup	char	3	Indicates the group this instrument belongs to.
SecurityID	uint64	8	Security ID as defined by B3's Trading System.
SecurityIDOptional	uint64	8	Optional Security ID as defined by B3's Trading System.
SecurityStrategyType	char	3	Strategy type definition. Required for strategy instruments.
sendingTime	uint64	8	Sending Time as number of nanoseconds since epoch (1970-01-01 00:00:00 UTC).
SeqNum	uint32	4	Sequence number inside the given channel.

Name	Data Type	Size	Description
SeqNum1	uint32	0	Sequence Number, fixed to 1.
sequenceNumber	uint32	4	Packet sequence number (reset to 1 when sequenceVersion increments).
sequenceVersion	uint16	2	Sequence version (incremented weekly or when market data engine failover happens).
SettlType	uint16	2	Indicates order settlement period in days. (e.g. 0, D1, D2, D3, D60, D120 etc). If present, SettlDate (64) overrides this field.
Symbol	char	20	Ticker symbol.
time	uint64	8	UTC timestamp with nanosecond precision (Unix Epoch).
TradeID	uint32	4	Contains the unique identifier for this trade per instrument + trading date, as assigned by the exchange. Required if reporting a Trade.
UInt16	uint16	2	2-byte unsigned integer, from 0 to 65535.
UInt16NULL	uint16	2	2-byte unsigned integer, from 1 to 65535, NULL (optional) value = 0.
UInt32	uint32	4	4-byte unsigned integer, from 0 to 4294967295 (2^32-1).
UInt32NULL	uint32	4	4-byte unsigned integer, from 1 to 4294967295 (2^32-1), NULL (optional) value = 0.
UInt64NULL	uint64	8	8-byte unsigned integer, from 1 to 18446744073709551615 (2^64-1), NULL (optional) value = 0.
UInt8	uint8	1	1-byte unsigned integer, from 0 to 255.
UInt8NULL	uint8	1	1-byte signed integer, from -128 to 127, NULL (optional) value = 0.
unit	uint8	0	time unit (nanoseconds).
varData	uint8	N	Bytes of the string, encoded in UTF-8.
week	uint8	1	Week of month (1 to 5).
year	uint16	2	4-digit year.

9 MESSAGES

9.1 Message in Streams

MessageType	TemplateID	Application Message	Incremental Stream	Snapshot Stream	Instrument definition Stream
4	1	SequenceReset	✓	✓	✓
0	2	Sequence	✓	✓	✓
f	3	SecurityStatus	✓	✓	
B	5	News	✓		
X	9	EmptyBook	✓		
f	10	SecurityGroupPhase	✓	✓	
X	11	ChannelReset	✓		
d	12	SecurityDefinition	✓		✓
X	15	OpeningPrice	✓	✓	
X	16	TheoreticalOpeningPrice	✓	✓	
X	17	ClosingPrice	✓	✓	
X	19	AuctionImbalance	✓	✓	
X	22	PriceBand	✓	✓	
X	21	QuantityBand	✓	✓	
X	24	HighPrice	✓	✓	
X	25	LowPrice	✓	✓	
X	27	LastTradePrice	✓	✓	

MessageType	TemplateID	Application Message	Incremental Stream	Snapshot Stream	Instrument definition Stream
X	28	SettlementPrice	√	√	
X	29	OpenInterest	√	√	
X	50	Order_MBO	√	√	
X	51	DeleteOrder_MBO	√		
X	52	MassDeleteOrders_MBO	√		
X	53	Trade	√		
X	54	ForwardTrade	√		
X	55	ExecutionSummary	√		
X	56	ExecutionStatistics	√	√	
X	57	TradeBust	√		
X	49	ChangeOrder_MBO	√		
X	58	BestBidOffer	√		
W	30	SnapshotFullRefresh_Header		√	
W	71	SnapshotFullRefresh_Orders_MBO		√	

9.2 Message Definitions

All messages are summarized below:

Presence of the field in the message can be:

- **R** = required.
- **O** = optional (if it is not present, assume the null value defined in the related type or, if not explicitly defined it is the default null value of the primitive type. See optional fields and default values section).
- **C** = constant.

9.2.1 Message: SequenceReset_1

Name	Template ID	Version	Encoded Length	Description
SequenceReset_1	1	16	0	Used to reset the incremental stream or indicate the loop on instrument definition or snapshot recovery is restarting

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (char)		Constant: "4" (Sequence Reset).
1128	applVerID	C	ApplVerID Enum (uint8)		Specifies the service pack release being applied at message level. Constant: 9 (FIX 5.0 SP2).
36	newSeqNo	C	SeqNum1 (uint32)		New sequence number. Always one. Constant: 1

9.2.2 Message: Sequence_2

Name	Template ID	Version	Encoded Length	Description
Sequence_2	2	16	4	Sent in incremental, snapshot and instrument list feeds in periods of no activity.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (char)		Constant: "0" (Sequence message for heartbeat).
1128	applVerID	C	ApplVerID Enum (uint8)		Specifies the service pack release being applied at message level. Constant: 9 (FIX 5.0 SP2).
35526	nextSeqNo	R	SeqNum (uint32)	0 (4)	The next application sequence number in the feed. Always 1 (one) for snapshot replay and instrument replay feeds.

9.2.3 Message: SecurityStatus_3

Name	Template ID	Version	Encoded Length	Description
SecurityStatus_3	3	16	36	Trading status for instruments

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (char)		Constant: "f" (Security Status).
1128	applVerID	C	ApplVerID Enum (uint8)		Specifies the service pack release being applied at message level. Constant: 9 (FIX 5.0 SP2).
48	securityID	R	SecurityID (uint64)	0 (8)	Security ID as defined by B3. For the SecurityID list, see the Security Definition message in Market Data feed.
22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID. Constant: “EXCHANGE_SYMBOL” (8).
207	securityExchange	C	SecurityExchangeBVMF (char)		Market to which the symbol belongs. Constant: "BVMF"

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
37035	matchEventIndicator	R	MatchEventIndicator Set (uint8)	8 (1)	Set of indicators that identify some market data events. Here are the possible bits applied to this message when it is set: Bit 7: Last message for the event.
336	tradingSessionID	R	TradingSessionID Enum (uint8)	9 (1)	Identifier for trading session.
326	securityTradingStatus	R	SecurityTradingStatus Enum (uint8)	10 (1)	Status related to a given instrument.
1174	securityTradingEvent	O	SecurityTradingEvent Enum (uint8)	11 (1)	Identifies an event related to a Trading. This tag is also used to mark when an instrument state is kept separate from the group phase, or when the instrument state follows the default group phase (stops having a separate, defined state). Always sent when tag 48 is present.
75	tradeDate	R	LocalMktDate (uint16)	12 (2)	Used to specify the trading date for which a set of market data applies.
	<padding>			14 (2)	*
342	tradSesOpenTime	O	UTCTimestampNanos	16 (8)	Estimated end of the current auction. Only present when SecurityTradingStatus=21 (Pre-open/Reserved).
60	transactTime	R	UTCTimestampNanos	24 (8)	Timestamp when status of the security changed.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
83	rptSeq	O	RptSeq (uint32)	32 (4)	Sequence number per instrument update. (Zeroed in snapshot feed)

* Padding is implemented by declaring "offset" attribute in next field.

9.2.4 Message: News_5

Name	Template ID	Version	Encoded Length	Description
News_5	5	16	36	Conveys market information of B3 market surveillance notifications and news produced by agencies.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (char)		Constant: "B" (News).
1128	applVerID	C	ApplVerID Enum (uint8)		Specifies the service pack release being applied at message level. Constant: 9 (FIX 5.0 SP2).
48	securityID	O	SecurityIDOptional (uint64)	0 (8)	Security ID as defined by B3. For the SecurityID list, see the Security Definition message in Market Data feed.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID. Constant: "EXCHANGE_SYMBOL" (8).
207	securityExchange	C	SecurityExchangeBVMF (char)		Market to which the symbol belongs. Constant: "BVMF"
37035	matchEventIndicator	R	MatchEventIndicator Set (uint8)	8 (1)	Set of indicators that identify some market data events. Here are the possible bits applied to this message when it is set: Bit 7: Last message for the event.
6940	newsSource	R	NewsSource Enum (uint8)	9 (1)	Source of the News.
1474	languageCode	O	LanguageCode (char)	10 (2)	Indicates the language the news is in. If null, it's 'pt' = Portuguese.
37709	partCount	R	UInt16 (uint16)	12 (2)	Total number of parts for the text of a News message. Usually 1.
37710	partNumber	R	UInt16 (uint16)	14 (2)	Number of the part for this message. Starts from 1.
1472	newsID	O	NewsID (uint64)	16 (8)	Unique identifier for News message. Included in the News messages sent in the Unified News Channel. Not sent for trading engine News messages.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
42	origTime	O	UTCTimestampNanos	24 (8)	Time of message origination.
37777	totalTextLength	R	UInt32 (uint32)	32 (4)	Total size, in bytes, for the text of a News message.
148	headline	R	VarString	2 + N	The headline of a News message.
58	text	R	VarString	2 + N	Free format text string.
149	uRLLink	R	VarString	2 + N	A URL (Uniform Resource Locator) link to additional information (e.g. http://www.b3.com.br).

9.2.5 Message: EmptyBook_9

Name	Template ID	Version	Encoded Length	Description
EmptyBook_9	9	16	20	Market Data Incremental Refresh - Empty Book

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (char)		Constant: "X" (Market Data Incremental Refresh).
1128	applVerID	C	ApplVerID Enum (uint8)		Specifies the service pack release being applied at message level. Constant: 9 (FIX 5.0 SP2).
279	mDUpdateAction	C	MDUpdateAction Enum (uint8)		Update Action (NEW) - always replace. Constant: 0 (New).
269	mDEntryType	C	MDEntryType Enum (char)		Constant: "J" (Empty Book).
48	securityID	R	SecurityID (uint64)	0 (8)	Security ID as defined by B3. For the SecurityID list, see the Security Definition message in Market Data feed.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID. Constant: "EXCHANGE_SYMBOL" (8).
207	securityExchange	C	SecurityExchangeBVMF (char)		Market to which the symbol belongs. Constant: "BVMF"
37035	matchEventIndicator	R	MatchEventIndicator Set (uint8)	8 (1)	Set of indicators that identify some market data events. Here are the possible bits applied to this message when it is set: Bit 5: Message is sent during recovery process. Bit 7: Last message for the event.
	<padding>			9 (3)	*
37033	mDEntryTimestamp	R	UTCTimestampNanos	12 (8)	Date and time of market data entry.

* Padding is implemented by declaring "offset" attribute in next field.

9.2.6 Message: SecurityGroupPhase_10

Name	Template ID	Version	Encoded Length	Description
SecurityGroupPhase_10	10	16	32	Trading status for security groups

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (char)		Constant: "f" (Security Status).
1128	applVerID	C	ApplVerID Enum (uint8)		Specifies the service pack release being applied at message level. Constant: 9 (FIX 5.0 SP2).
1151	securityGroup	R	SecurityGroup (char)	0 (3)	Security Group.
	<padding>			3 (5)	*
37035	matchEventIndicator	R	MatchEventIndicator Set (uint8)	8 (1)	Set of indicators that identify some market data events. Here are the possible bits applied to this message when it is set: Bit 7: Last message for the event.
336	tradingSessionID	R	TradingSessionID Enum (uint8)	9 (1)	Identifier for trading session.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
625	tradingSessionSubID	R	TradingSessionSubID Enum (uint8)	10 (1)	Phase related to a given SecurityGroup.
1174	securityTradingEvent	O	SecurityTradingEvent Enum (uint8)	11 (1)	Indicates if the trading session changed in order to reset some statistics for this group.
75	tradeDate	R	LocalMktDate (uint16)	12 (2)	Used to specify the trading date for which a set of market data applies.
	<padding>			14 (2)	*
342	tradSesOpenTime	O	UTCTimestampNanos	16 (8)	Estimated end of the current auction. Only present when TradingSessionSubID=21 (Pre-open/Reserved).
60	transactTime	R	UTCTimestampNanos	24 (8)	Timestamp when phase of the security group changed.

* Padding is implemented by declaring "offset" attribute in next field.

9.2.7 Message: ChannelReset_11

Name	Template ID	Version	Encoded Length	Description
ChannelReset_11	11	16	12	Channel Reset (remove all instruments, empty all books and statistics)

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (char)		Constant: "X" (Market Data Incremental Refresh).
1128	applVerID	C	ApplVerID Enum (uint8)		Specifies the service pack release being applied at message level. Constant: 9 (FIX 5.0 SP2).
279	mDUpdateAction	C	MDUpdateAction Enum (uint8)		Update Action (NEW) - always replace. Constant: 0 (New).
269	mDEntryType	C	MDEntryType Enum (char)		Constant: "J" (Empty Book).
37035	matchEventIndicator	R	MatchEventIndicator Set (uint8)	0 (1)	Set of indicators that identify some market data events. Here are the possible bits applied to this message when it is set: Bit 5: Message is sent during recovery process. Bit 7: Last message for the event.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
	<padding>			1 (3)	*
37033	mDEntryTimestamp	R	UTCTimestampNanos	4 (8)	Date and time of market data entry.

* Padding is implemented by declaring "offset" attribute in next field.

9.2.8 Message: SecurityDefinition_12

Name	Template ID	Version	Encoded Length	Description
SecurityDefinition_12	12	16	232	Security Definition.

	Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
	35	messageType	C	MessageType Enum (char)		Constant: "d" (Security Definition).
	1128	applVerID	C	ApplVerID Enum (uint8)		Specifies the service pack release being applied at message level. Constant: 9 (FIX 5.0 SP2).
	48	securityID	R	SecurityID (uint64)	0 (8)	Security ID as defined by B3.
	207	securityExchange	R	SecurityExchange (char)	8 (4)	Exchange Code.
	22	securityIDSource	R	SecurityIDSource (char)	12 (1)	Identifies the class of the SecurityID.

	Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
	1151	securityGroup	R	SecurityGroup (char)	13 (3)	Security Group.
	55	symbol	R	Symbol (char)	16 (20)	Instrument's ticker symbol.
	980	securityUpdateAction	R	SecurityUpdateAction Enum (char)	36 (1)	Action used when updating the security.
	167	securityType	R	SecurityType Enum (uint8)	37 (1)	Instrument's security type.
	762	securitySubType	R	UInt16 (uint16)	38 (2)	Instrument's security sub type. Updated list of values defined at *SecuritySubType* Enum.
	393	totNoRelatedSym	R	UInt32 (uint32)	40 (4)	Total number of instruments to be returned in the current replay loop.
	969	minPriceIncrement	O	Fixed8 (int64)	44 (8)	Number of minimum price increments.
	202	strikePrice	O	PriceOptional (int64)	52 (8)	Strike price of an option.

	Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
	231	contractMultiplier	O	Fixed8 (int64)	60 (8)	Specifies the ratio or multiply factor to convert from “nominal” units (e.g. contracts) to total units (e.g. shares) (e.g. 1.0, 100, 1000, 0.00025 etc).
	37012	priceDivisor	O	Fixed8 (int64)	68 (8)	Value that divides the Price field to produce the actual order price (based on Step of Quotation). (e.g. 1, 100, 1000, etc). Default value is 1. Also used for index instruments to disseminate the index reducer, in this case, there is no default value.
	6938	securityValidityTimestamp	R	UTCTimestampSeconds	76 (8)	Indicates the UTC timestamp when trading for this security expires, i.e. when it is not eligible to trade anymore. Different from MaturityDate.
	7595	noSharesIssued	O	UInt64NULL (uint64)	84 (8)	Share capital.
	37037	clearingHouseID	O	ClearingHouseID (uint64)	92 (8)	Clearing House ID.
	9749	minOrderQty	O	QuantityOptional (int64)	100 (8)	Minimum quantity for an order.
	9748	maxOrderQty	O	QuantityOptional (int64)	108 (8)	Maximum quantity for an order.

	Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
	1231	minLotSize	O	QuantityOptional (int64)	116 (8)	Minimum lot size allowed based on lot type specified in LotType(1093). Used for the equities segment.
	562	minTradeVol	O	QuantityOptional (int64)	124 (8)	The minimum trading volume for the security.
	37010	corporateActionEventId	O	UInt32NULL (uint32)	132 (4)	Corporate Action Event ID.
	225	issueDate	R	LocalMktDate32 (int32)	136 (4)	The date on which the security is issued/activated.
	541	maturityDate	O	LocalMktDate32Optional (int32)	140 (4)	Date of instrument maturity.
	470	countryOfIssue	O	CountryCode (char)	144 (2)	ISO 3166-1 alpha-2 country code.
	916	startDate	O	LocalMktDate32Optional (int32)	146 (4)	Start date of a financing deal, i.e. the date the buyer pays the seller cash and takes control of the collateral.
	917	endDate	O	LocalMktDate32Optional (int32)	150 (4)	End date of a financing deal, i.e. the date the seller reimburses the buyer and takes back control of the collateral.

	Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
	63	settlType	O	SettlType (uint16)	154 (2)	Order settlement period in days.
	64	settlDate	O	LocalMktDate32Optional (int32)	156 (4)	Specific date of trade settlement.
	873	datedDate	O	LocalMktDate32Optional (int32)	160 (4)	The date of the security activation, if different from the IssueDate.
	37026	isinNumber	O	ISINNumber (char)	164 (12)	ISIN Number.
	6937	asset	R	Asset (char)	176 (6)	Asset associated to the security.
	461	cfiCode	R	CFICode (char)	182 (6)	Classification of Financial Instruments (CFI code) values, which indicate the type of security using the ISO 10962 standard.
	200	maturityMonthYear	O	MaturityMonthYear	188 (5)	Day, week, month and year of the maturity (used for standardized futures and options).
	667	contractSettlMonth	O	MaturityMonthYear	193 (5)	Specifies when the contract will settle.

	Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
	15	currency	R	Currency (char)	198 (3)	Currency code.
	947	strikeCurrency	O	Currency (char)	201 (3)	Currency of option's strike price.
	120	settlCurrency	O	Currency (char)	204 (3)	Currency used for the settlement.
	7534	securityStrategyType	O	SecurityStrategyType (char)	207 (3)	Strategy type definition.
	1093	lotType	O	LotType Enum (uint8)	210 (1)	Lot type.
	5151	tickSizeDenominator	O	UInt8 (uint8)	211 (1)	Number of decimals for pricing the instrument.
	460	product	R	Product Enum (uint8)	212 (1)	Type of product.
	1194	exerciseStyle	O	ExerciseStyle Enum (uint8)	213 (1)	Exercise Style.
	201	putOrCall	O	PutOrCall Enum (uint8)	214 (1)	Indicates whether an option contract is a put or call.

	Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
	423	priceType	O	PriceType Enum (uint8)	215 (1)	Price type of the instrument.
	1300	marketSegmentID	O	MarketSegmentID (uint8)	216 (1)	Market segment.
	37011	governanceIndicator	O	GovernanceIndicator Enum (uint8)	217 (1)	Corporative Governance Level Indicator.
	37015	securityMatchType	O	SecurityMatchType Enum (uint8)	218 (1)	Type of matching that occurred.
	893	lastFragment	O	Boolean Enum (uint8)	219 (1)	Indicates whether this message is the last in the sequence of messages.
	1377	multiLegModel	O	MultiLegModel Enum (uint8)	220 (1)	Defines whether the security is pre-defined or user-defined. Used for multileg security only.
	1378	multiLegPriceMethod	O	MultiLegPriceMethod Enum (uint8)	221 (1)	Defines the method used when applying the multileg price to the legs. When this tag is set, it indicates spreads that have leg prices generated by the trading engine.
	35561	minCrossQty	O	QuantityOptional (int64)	222 (8)	Minimum quantity of a cross order for the security. Since schema version 6.

	Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
	1144	impliedMarketIndicator	O	ImpliedMarketIndicator	230 (1)	Indicates that an implied order can be created for the instrument. Since schema version 10.
	1482	optPayoutType	O	OptPayoutType	231(1)	Indicates the type of payout that will result from an in-the-money option. Since schema version 16.
	711	**noUnderlyings**	R	GroupSizeEncoding	232 (28)	Underlying instruments.
→	309	underlyingSecurityID	R	SecurityID (uint64)	0 (8)	Underlying instrument's security ID.
→	305	underlyingSecurityIDSource	C	SecurityIDSource Enum (char)		Underlying instrument's Security ID Source. Constant: “EXCHANGE_SYMBOL” (8).
→	308	underlyingSecurityExchange	C	SecurityExchangeBVMF (char)		Underlying instrument's Exchange Code. Constant: "BVMF"
→	311	underlyingSymbol	R	Symbol (char)	8 (20)	Underlying instrument's ticker symbol.
	555	**noLegs**	R	GroupSizeEncoding	* (38)	Instrument legs.
→	602	legSecurityID	R	SecurityID (uint64)	0 (8)	Leg's security ID.

	Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
→	603	legSecurityIDSource	C	SecurityIDSource Enum (char)		Leg's Security ID Source. Constant: “EXCHANGE_SYMBOL” (8).
→	616	legSecurityExchange	C	SecurityExchangeBVMF (char)		Leg's Exchange Code. Constant: "BVMF"
→	623	legRatioQty	R	RatioQty	8 (8)	Ratio of quantity for this leg relative to the entire security.
→	609	legSecurityType	R	SecurityType Enum (uint8)	16 (1)	Leg's security type.
→	624	legSide	R	Side Enum (uint8)	17 (1)	Side of this leg.
→	600	legSymbol	R	Symbol (char)	18 (20)	Leg symbol.
	870	**noInstrAttribs**	R	GroupSizeEncoding	* (2)	Specifies the number of the application ID occurrences (number of channels).
→	871	instrAttribType	R	InstrAttribType Enum (uint8)	0 (1)	Code to represent the type of instrument attributes.
→	872	instrAttribValue	R	InstrAttribValue Enum (uint8)	1 (1)	Attribute value appropriate to the InstrAttribType (871) field.

	Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
	107	securityDesc	R	TextEncoding		Non-normative textual description for the financial instrument.

9.2.9 Message: OpeningPrice_15

Name	Template ID	Version	Encoded Length	Description
OpeningPrice_15	15	16	44	Carries the summary information about opening trading session events per market data stream.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (char)		Constant: "X" (Market Data Incremental Refresh).
1128	applVerID	C	ApplVerID Enum (uint8)		Specifies the service pack release being applied at message level. Constant: 9 (FIX 5.0 SP2).
48	securityID	R	SecurityID (uint64)	0 (8)	Security ID as defined by B3. For the SecurityID list, see the Security Definition message in Market Data feed.
22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID. Constant: “EXCHANGE_SYMBOL” (8).
207	securityExchange	C	SecurityExchangeBVMF (char)		Market to which the symbol belongs. Constant: "BVMF"

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
37035	matchEventIndicator	R	MatchEventIndicator Set (uint8)	8 (1)	Set of indicators that identify some market data events. Here are the possible bits applied to this message when it is set: Bit 7: Last message for the event.
279	mDUpdateAction	R	MDUpdateAction Enum (uint8)	9 (1)	Update Action (NEW or DELETE).
269	mDEntryType	C	MDEntryType Enum (char)		Entry type: Opening Price. Constant: "4" (Opening Price).
286	openCloseSettlFlag	R	OpenCloseSettlFlag Enum (uint8)	10 (1)	Identifies if the opening price represents or not a daily opening price.
	<padding>			11 (1)	*
270	mDEntryPx	R	Price	12 (8)	Value of the statistics.
451	netChgPrevDay	O	PriceOffset8Optional	20 (8)	Net change from previous trading day’s closing price vs. last traded price.
75	tradeDate	R	LocalMktDate (uint16)	28 (2)	Used to specify the trading date for which a set of market data applies.
37033	mDEntryTimestamp	R	UTCTimestampNanos	30 (8)	Date and time of market data entry.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
83	rptSeq	O	RptSeq (uint32)	38 (4)	Sequence number per instrument update. Zeroed in snapshot feed.
	<padding>			42 (2)	**

* Padding is implemented by declaring "offset" attribute in next field.

** Padding is implemented by fixing "blockLength" attribute in the sbe:message.

9.2.10 Message: TheoreticalOpeningPrice_16

Name	Template ID	Version	Encoded Length	Description
TheoreticalOpeningPrice_16	16	16	40	The theoretical opening price is also sent on this block and is calculated and updated based on the orders presented in the book during every auction including the pre-opening / pre-closing auction.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (char)		Constant: "X" (Market Data Incremental Refresh).
1128	applVerID	C	ApplVerID Enum (uint8)		Specifies the service pack release being applied at message level. Constant: 9 (FIX 5.0 SP2).
48	securityID	R	SecurityID (uint64)	0 (8)	Security ID as defined by B3. For the SecurityID list, see the Security Definition message in Market Data feed.
22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID. Constant: “EXCHANGE_SYMBOL” (8).
207	securityExchange	C	SecurityExchangeBVMF (char)		Market to which the symbol belongs. Constant: "BVMF"

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
37035	matchEventIndicator	R	MatchEventIndicator Set (uint8)	8 (1)	Set of indicators that identify some market data events. Here are the possible bits applied to this message when it is set: Bit 7: Last message for the event.
279	mDUpdateAction	R	MDUpdateAction Enum (uint8)	9 (1)	Update Action (NEW or DELETE).
269	mDEntryType	C	MDEntryType Enum (char)		Entry type: Opening Price. Constant: "4" (Opening Price).
286	openCloseSettlFlag	C	OpenCloseSettlFlag Enum (uint8)		Indicates this is a theoretical opening price. Constant: 5 (Theoretical Price).
75	tradeDate	R	LocalMktDate (uint16)	10 (2)	Used to specify the trading date for which a set of market data applies.
270	mDEntryPx	O	PriceOptional	12 (8)	Theoretical Opening Price.
271	mDEntrySize	O	QuantityOptional (int64)	20 (8)	Theoretical Opening Quantity.
37033	mDEntryTimestamp	R	UTCTimestampNanos	28 (8)	Date and time of market data entry.
83	rptSeq	O	RptSeq (uint32)	36 (4)	Sequence number per instrument update. Zeroed in snapshot feed.

9.2.11 Message: ClosingPrice_17

Name	Template ID	Version	Encoded Length	Description
ClosingPrice_17	17	16	36	Summary information about closing trading sessions per market data stream.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (char)		Constant: "X" (Market Data Incremental Refresh).
1128	applVerID	C	ApplVerID Enum (uint8)		Specifies the service pack release being applied at message level. Constant: 9 (FIX 5.0 SP2).
48	securityID	R	SecurityID (uint64)	0 (8)	Security ID as defined by B3. For the SecurityID list, see the Security Definition message in Market Data feed.
22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID. Constant: "EXCHANGE_SYMBOL" (8).
207	securityExchange	C	SecurityExchangeBVMF (char)		Market to which the symbol belongs. Constant: "BVMF"

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
37035	matchEventIndicator	R	MatchEventIndicator Set (uint8)	8 (1)	Set of indicators that identify some market data events. Here are the possible bits applied to this message when it is set: Bit 7: Last message for the event.
279	mDUpdateAction	C	MDUpdateAction Enum (uint8)		Update Action (NEW) - always replace. Constant: 0 (New).
269	mDEntryType	C	MDEntryType Enum (char)		Entry type: Closing Price. Constant: "5" (Closing Price).
286	openCloseSettlFlag	R	OpenCloseSettlFlag Enum (uint8)	9 (1)	Identifies if the closing price represents a daily or entry from previous business day.
	<padding>			10 (2)	*
270	mDEntryPx	R	Price8	12 (8)	Closing price. May be adjusted by corporate events.
9325	lastTradeDate	O	LocalMktDateOptional (uint16)	20 (2)	Date the instrument last traded.
75	tradeDate	R	LocalMktDate (uint16)	22 (2)	Used to specify the trading date for which a set of market data applies.
37033	mDEntryTimestamp	R	UTCTimestampNanos	24 (8)	Date and time of market data entry.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
83	rptSeq	O	RptSeq (uint32)	32 (4)	Sequence number per instrument update. Zeroed in snapshot feed.

* Padding is implemented by declaring "offset" attribute in next field.

9.2.12 Message: AuctionImbalance_19

Name	Template ID	Version	Encoded Length	Description
AuctionImbalance_19	19	16	32	Carries auction imbalance information, indicating the remaining quantity and to which side (buyer or seller) the auction is pending towards.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (char)		Constant: "X" (Market Data Incremental Refresh).
1128	applVerID	C	ApplVerID Enum (uint8)		Specifies the service pack release being applied at message level. Constant: 9 (FIX 5.0 SP2).
48	securityID	R	SecurityID (uint64)	0 (8)	Security ID as defined by B3. For the SecurityID list, see the Security Definition message in Market Data feed.
22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID. Constant: “EXCHANGE_SYMBOL” (8).
207	securityExchange	C	SecurityExchangeBVMF (char)		Market to which the symbol belongs. Constant: "BVMF"

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
37035	matchEventIndicator	R	MatchEventIndicator Set (uint8)	8 (1)	Set of indicators that identify some market data events. Here are the possible bits applied to this message when it is set: Bit 7: Last message for the event.
279	mDUpdateAction	R	MDUpdateAction Enum (uint8)	9 (1)	Updsecate Action (NEW or DELETE).
269	mDEntryType	C	MDEntryType Enum (char)		Entry type: Imbalance. Constant: "A" (Imbalance).
37277	imbalanceCondition	R	ImbalanceCondition Set (uint16)	10 (2)	IMBALANCE_MORE_BUYERS, IMBALANCE_MORE_SELLERS, All bits off => BALANCED.
271	mDEntrySize	O	QuantityOptional (int64)	12 (8)	Remaining auction quantity.
37033	mDEntryTimestamp	R	UTCTimestampNanos	20 (8)	Date and time of market data entry.
83	rptSeq	O	RptSeq (uint32)	28 (4)	Sequence number per instrument update. Zeroed in snapshot feed.

9.2.13 Message: PriceBand_22

Name	Template ID	Version	Encoded Length	Description
PriceBand_20	22	16	48	Price Banding (tunnel).

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (char)		Constant: "X" (Market Data Incremental Refresh).
1128	applVerID	C	ApplVerID Enum (uint8)		Specifies the service pack release being applied at message level. Constant: 9 (FIX 5.0 SP2).
48	securityID	R	SecurityID (uint64)	0 (8)	Security ID as defined by B3. For the SecurityID list, see the Security Definition message in Market Data feed.
22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID. Constant: “EXCHANGE_SYMBOL” (8).
207	securityExchange	C	SecurityExchangeBVMF (char)		Market to which the symbol belongs. Constant: "BVMF"

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
37035	matchEventIndicator	R	MatchEventIndicator Set (uint8)	8 (1)	Set of indicators that identify some market data events. Here are the possible bits applied to this message when it is set: Bit 7: Last message for the event.
279	mDUpdateAction	C	MDUpdateAction Enum (uint8)		Update Action (NEW) - always replace. Constant: 0 (New).
269	mDEntryType	C	MDEntryType Enum (char)		Entry type: Price Band. Constant: "g" (Price band).
6939	priceBandType	O	PriceBandType Enum (uint8)	9 (1)	Indicates the type of price banding (tunnel).
1306	priceLimitType	O	PriceLimitType Enum (uint8)	10 (1)	Describes how the price limits are expressed.
37008	priceBandMidpointPriceType	O	PriceBandMidpointPriceType Enum (uint8)	11 (1)	Band Midpoint Type, used with Auction Price Banding. Only sent for Rejection and Auction Bands when PriceLimitType (1306) equals to 2 (Percentage).
1148	lowLimitPrice	O	PriceOptional	12 (8)	Allowable low limit price for the trading day. A key parameter in validating order price. Used as the lower band for validating order prices. Orders submitted with prices below the lower limit will be rejected.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
1149	highLimitPrice	O	PriceOptional	20 (8)	Allowable high limit price for the trading day. A key parameter in validating order price. Used as the upper band for validating order prices. Orders submitted with prices above the upper limit will be rejected.
1150	tradingReferencePrice	O	Fixed8 (int64)	28 (8)	Reference price for the current trading price range. The value may be the reference price, settlement price or closing price of the prior trading day. Sent only for Economic Indicators.
37033	mDEntryTimestamp	R	UTCTimestampNanos	36 (8)	Date and time of market data entry.
83	rptSeq	O	RptSeq (uint32)	44 (4)	Sequence number per instrument update. Zeroed in snapshot feed.

9.2.14 Message: QuantityBand_21

Name	Template ID	Version	Encoded Length	Description
QuantityBand_21	21	16	40	Quantity Band.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (char)		Constant: "X" (Market Data Incremental Refresh).
1128	applVerID	C	ApplVerID Enum (uint8)		Specifies the service pack release being applied at message level. Constant: 9 (FIX 5.0 SP2).
48	securityID	R	SecurityID (uint64)	0 (8)	Security ID as defined by B3. For the SecurityID list, see the Security Definition message in Market Data feed.
22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID. Constant: “EXCHANGE_SYMBOL” (8).
207	securityExchange	C	SecurityExchangeBVMF (char)		Market to which the symbol belongs. Constant: "BVMF"

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
37035	matchEventIndicator	R	MatchEventIndicator Set (uint8)	8 (1)	Set of indicators that identify some market data events. Here are the possible bits applied to this message when it is set: Bit 7: Last message for the event.
279	mDUpdateAction	C	MDUpdateAction Enum (uint8)		Update Action (NEW) - always replace. Constant: 0 (New).
269	mDEntryType	C	MDEntryType Enum (char)		Entry type: Quantity Band. Constant: "h" (Quantity band).
	<padding>			9 (3)	*
37003	avgDailyTradedQty	O	QuantityVolumeOptional (int64)	12 (8)	Daily average shares traded within 30 days – equity market only. Previously known as DailyAvgShares30D. Always 0 for Derivatives.
1140	maxTradeVol	O	QuantityVolumeOptional (int64)	20 (8)	The maximum order quantity that can be submitted for a security. The value is the minimum between % of shares issued and % of average traded quantity within 30 days.
37033	mDEntryTimestamp	R	UTCTimestampNanos	28 (8)	Date and time of market data entry.
83	rptSeq	O	RptSeq (uint32)	36 (4)	Sequence number per instrument update. Zeroed in snapshot feed.

* Padding is implemented by declaring "offset" attribute in next field.

9.2.15 Message: HighPrice_24

Name	Template ID	Version	Encoded Length	Description
HighPrice_24	24	16	32	The highest price traded for the security in the trading session.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (char)		Constant: "X" (Market Data Incremental Refresh).
1128	applVerID	C	ApplVerID Enum (uint8)		Specifies the service pack release being applied at message level. Constant: 9 (FIX 5.0 SP2).
48	securityID	R	SecurityID (uint64)	0 (8)	Security ID as defined by B3. For the SecurityID list, see the Security Definition message in Market Data feed.
22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID. Constant: “EXCHANGE_SYMBOL” (8).
207	securityExchange	C	SecurityExchangeBVMF (char)		Market to which the symbol belongs. Constant: "BVMF"

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
37035	matchEventIndicator	R	MatchEventIndicator Set (uint8)	8 (1)	Set of indicators that identify some market data events. Here are the possible bits applied to this message when it is set: Bit 7: Last message for the event.
279	mDUpdateAction	R	MDUpdateAction Enum (uint8)	9 (1)	Update Action (NEW or DELETE).
269	mDEntryType	C	MDEntryType Enum (char)		Entry type: Trading Session High Price. Constant: "7" (Session High Price).
75	tradeDate	R	LocalMktDate (uint16)	10 (2)	Used to specify the trading date for which a set of market data applies.
270	mDEntryPx	R	Price	12 (8)	Trading Session High Price.
37033	mDEntryTimestamp	R	UTCTimestampNanos	20 (8)	Date and time of market data entry.
83	rptSeq	O	RptSeq (uint32)	28 (4)	Sequence number per instrument update. Zeroed in snapshot feed.

9.2.16 Message: LowPrice_25

Name	Template ID	Version	Encoded Length	Description
LowPrice_25	25	16	32	The lowest price traded for the security in the trading session.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (char)		Constant: "X" (Market Data Incremental Refresh).
1128	applVerID	C	ApplVerID Enum (uint8)		Specifies the service pack release being applied at message level. Constant: 9 (FIX 5.0 SP2).
48	securityID	R	SecurityID (uint64)	0 (8)	Security ID as defined by B3. For the SecurityID list, see the Security Definition message in Market Data feed.
22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID. Constant: “EXCHANGE_SYMBOL” (8).
207	securityExchange	C	SecurityExchangeBVMF (char)		Market to which the symbol belongs. Constant: "BVMF"

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
37035	matchEventIndicator	R	MatchEventIndicator Set (uint8)	8 (1)	Set of indicators that identify some market data events. Here are the possible bits applied to this message when it is set: Bit 7: Last message for the event.
279	mDUpdateAction	R	MDUpdateAction Enum (uint8)	9 (1)	Update Action (NEW or DELETE).
269	mDEntryType	C	MDEntryType Enum (char)		Entry type: Trading Session Low Price. Constant: "8" (Session Low Price).
75	tradeDate	R	LocalMktDate (uint16)	10 (2)	Used to specify the trading date for which a set of market data applies.
270	mDEntryPx	R	Price	12 (8)	Trading Session Low Price.
37033	mDEntryTimestamp	R	UTCTimestampNanos	20 (8)	Date and time of market data entry.
83	rptSeq	O	RptSeq (uint32)	28 (4)	Sequence number per instrument update. Zeroed in snapshot feed.

9.2.17 Message: LastTradePrice_27

Name	Template ID	Version	Encoded Length	Description
LastTradePrice_27	27	16	68	The latest price traded for the security in the trading session

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (char)		Constant: "X" (Market Data Incremental Refresh).
1128	applVerID	C	ApplVerID Enum (uint8)		Specifies the service pack release being applied at message level. Constant: 9 (FIX 5.0 SP2).
48	securityID	R	SecurityID (uint64)	0 (8)	Security ID as defined by B3. For the SecurityID list, see the Security Definition message in Market Data feed.
22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID. Constant: “EXCHANGE_SYMBOL” (8).
207	securityExchange	C	SecurityExchangeBVMF (char)		Market to which the symbol belongs. Constant: "BVMF"

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
37035	matchEventIndicator	R	MatchEventIndicator Set (uint8)	8 (1)	Set of indicators that identify some market data events. Here are the possible bits applied to this message when it is set: Bit 4: Trade resulted from an implied generated order. Bit 7: Last message for the event.
279	mDUpdateAction	C	MDUpdateAction Enum (uint8)		Update Action (NEW) - always replace. Constant: 0 (New).
269	mDEntryType	C	MDEntryType Enum (char)		Entry type: Last Trade Price. Constant: "2" (Trade or Trade Summary or Last Trade Price).
336	tradingSessionID	R	TradingSessionID Enum (uint8)	9 (1)	Identifier for trading session.
277	tradeCondition	R	TradeCondition Set (uint16)	10 (2)	Set of conditions describing a trade.
270	mDEntryPx	R	Price	12 (8)	Last Trade Price.
271	mDEntrySize	R	Quantity (int64)	20 (8)	Quantity or volume represented by the Market Data Entry.
1003	tradeID	R	TradeID (uint32)	28 (4)	Contains the unique identifier for this trade per instrument + trading date, as assigned by the exchange. Required if reporting a Trade.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
288	mDEntryBuyer	O	FirmOptional (uint32)	32 (4)	For reporting trades (buying party).
289	mDEntrySeller	O	FirmOptional (uint32)	36 (4)	For reporting trades (selling party).
75	tradeDate	R	LocalMktDate (uint16)	40 (2)	Used to specify the trading date for which a set of market data applies.
37033	mDEntryTimestamp	R	UTCTimestampNanos	42 (8)	Date and time of market data entry.
83	rptSeq	O	RptSeq (uint32)	50 (4)	Sequence number per instrument update. Zeroed in snapshot feed.
287	sellerDays	O	UInt16NULL (uint16)	54 (2)	Specifies the number of days that may elapse before delivery of the security. Only used for trades in forward market.
37014	mDEntryInterestRate	O	Percentage	56 (8)	Interest Rate of the Termo Trade. Expressed in decimal form. For example, 1% points is expressed and sent as 0.01. One basis point is represented as 0.0001.
829	trdSubType	O	TrdSubType Enum (uint8)	64 (1)	Sub type of trade assigned to a trade. Since schema version 7.
	<padding>			65 (3)	**

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
37034	mDInsertTimestamp	R	UTCTimestampNanos	68 (8)	Timestamp when the trade event occurred. Typically, it has the same value as the *transactTime* field, except when set manually by MktOps. Since schema version 16.

** Padding is implemented by fixing "blockLength" attribute in the sbe:message.

9.2.18 Message: SettlementPrice_28

Name	**Template ID**	**Version**	**Encoded Length**	**Description**
SettlementPrice_28	28	16	36	Settlement price or the previous day's adjusted closing price.

Tag	**Tag Name**	**Presence**	**Data Type (Primitive Type)**	**Offset (Size)**	**Comments**
35	messageType	C	MessageType Enum (char)		Constant: "X" (Market Data Incremental Refresh).
1128	applVerID	C	ApplVerID Enum (uint8)		Specifies the service pack release being applied at message level. Constant: 9 (FIX 5.0 SP2).

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
48	securityID	R	SecurityID (uint64)	0 (8)	Security ID as defined by B3. For the SecurityID list, see the Security Definition message in Market Data feed.
22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID (Exchange Symbol). Constant: "8" (null)
207	securityExchange	C	SecurityExchangeBVMF (char)		Market to which the symbol belongs. Constant: "BVMF"
37035	matchEventIndicator	R	MatchEventIndicator Set (uint8)	8 (1)	Set of indicators that identify some market data events. Here are the possible bits applied to this message when it is set: Bit 7: Last message for the event.
279	mDUpdateAction	C	MDUpdateAction Enum (uint8)		Update Action (NEW) - always replace. Constant: 0 (New).
269	mDEntryType	C	MDEntryType Enum (char)		Entry type: Settlement Price. Constant: "6" (Settlement Price).
	<padding>			9 (1)	*
75	tradeDate	R	LocalMktDate (uint16)	10 (2)	Used to specify the trading date for which a set of market data applies.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
270	mDEntryPx	R	Price	12 (8)	Settlement Price.
37033	mDEntryTimestamp	R	UTCTimestampNanos	20 (8)	Date and time of market data entry.
286	openCloseSettlFlag	R	OpenCloseSettlFlag Enum (uint8)	28 (1)	Identifies if the settlement price represents a daily, preliminary or an entry from previous business day.
423	priceType	R	PriceType Enum (uint8)	29 (1)	Code to represent the price type.
731	settlPriceType	R	SettlPriceType Enum (uint8)	30 (1)	Type of settlement price: FINAL, THEORETICAL or UPDATED.
83	rptSeq	O	RptSeq (uint32)	31 (4)	Sequence number per instrument update. Zeroed in snapshot feed.
	<padding>			35 (1)	**

* Padding is implemented by declaring "offset" attribute in next field.

** Padding is implemented by fixing "blockLength" attribute in the sbe:message.

9.2.19 Message: OpenInterest_29

Name	Template ID	Version	Encoded Length	Description
OpenInterest_29	29	16	32	Total number of contracts in a commodity or options market that are still open.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (char)		Constant: "X" (Market Data Incremental Refresh).
1128	applVerID	C	ApplVerID Enum (uint8)		Specifies the service pack release being applied at message level. Constant: 9 (FIX 5.0 SP2).
48	securityID	R	SecurityID (uint64)	0 (8)	Security ID as defined by B3. For the SecurityID list, see the Security Definition message in Market Data feed.
22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID (Exchange Symbol). Constant: "8" (null)
207	securityExchange	C	SecurityExchangeBVMF (char)		Market to which the symbol belongs. Constant: "BVMF"

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
37035	matchEventIndicator	R	MatchEventIndicator Set (uint8)	8 (1)	Set of indicators that identify some market data events. Here are the possible bits applied to this message when it is set: Bit 7: Last message for the event.
279	mDUpdateAction	C	MDUpdateAction Enum (uint8)		Update Action (NEW) - always replace. Constant: 0 (New).
269	mDEntryType	C	MDEntryType Enum (char)		Entry type: Open Interest. Constant: "C" (Open Interest).
	<padding>			9 (1)	*
75	tradeDate	R	LocalMktDate (uint16)	10 (2)	Used to specify the trading date for which a set of market data applies.
271	mDEntrySize	R	Quantity (int64)	12 (8)	Indicates volume of contracts currently open.
37033	mDEntryTimestamp	R	UTCTimestampNanos	20 (8)	Date and time of market data entry.
83	rptSeq	O	RptSeq (uint32)	28 (4)	Sequence number per instrument update. Zeroed in snapshot feed.

* Padding is implemented by declaring "offset" attribute in next field.

9.2.20 Message: Order_MBO_50

Name	Template ID	Version	Encoded Length	Description
Order_MBO_50	50	16	72	Disseminates the creation/amendment of an order.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (char)		Constant: "X" (Market Data Incremental Refresh).
1128	applVerID	C	ApplVerID Enum (uint8)		Specifies the service pack release being applied at message level. Constant: 9 (FIX 5.0 SP2).
48	securityID	R	SecurityID (uint64)	0 (8)	Security ID as defined by B3. For the SecurityID list, see the Security Definition message in Market Data feed.
22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID. Constant: “EXCHANGE_SYMBOL” (8).
207	securityExchange	C	SecurityExchangeBVMF (char)		Market to which the symbol belongs. Constant: "BVMF"

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
37035	matchEventIndicator	R	MatchEventIndicator Set (uint8)	8 (1)	Set of indicators that identify some market data events. Here are the possible bits applied to this message when it is set: Bit 4: Implied generated order. Bit 5: Message is sent during recovery process. Bit 7: Last message for the event.
279	mDUpdateAction	R	MDUpdateAction Enum (uint8)	9 (1)	Update Action (NEW).
269	mDEntryType	R	MDEntryType Enum (char)	10 (1)	Entry Type (BID or OFFER).
	<padding>			11 (1)	*
270	mDEntryPx	O	PriceOptional	12 (8)	Price of the order per share or contract (not MOA/MOC orders).
271	mDEntrySize	R	Quantity (int64)	20 (8)	Displayed quantity or volume represented by the Market Data Entry.
	<padding>			28 (4)	*

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
37501	enteringFirm	O	FirmOptional (uint32)	32 (4)	Identifies the broker firm.
37034	mDInsertTimestamp	R	UTCTimestampNanos	36 (8)	The date and time when the order was inserted or re-inserted into the order book or manually altered by MktOps.
198	secondaryOrderID	R	OrderID (uint64)	44 (8)	Exchange-generated identifier of the order that updates with each order modification event involving a loss of priority, price change, or quantity replenishment in disclosed orders. This identifier reflects the order's priority relative to other orders at the same price-level in an ascending manner, where smaller values indicate higher priority.
83	rptSeq	O	RptSeq (uint32)	52 (4)	Sequence number per instrument update. Zeroed in snapshot feed.
60	transactTime	R	UTCTimestampNanos	56 (8)	Timestamp when the order event occurred in the matching process.
37780	mDEntryPrevSize	O	QuantityOptional	64 (8)	Previously displayed quantity or volume represented by the Market Data Entry. Present only in when mDUpdateAction=CHANGE and represents the previous quantity of the order before the modification. Absent for new order. Since schema version 16.

* Padding is implemented by declaring "offset" attribute in next field.

9.2.21 Message: DeleteOrder_MBO_51

Name	Template ID	Version	Encoded Length	Description
DeleteOrder_MBO_51	51	16	52	Disseminates the deletion of a new order.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (char)		Constant: "X" (Market Data Incremental Refresh).
1128	applVerID	C	ApplVerID Enum (uint8)		Specifies the service pack release being applied at message level. Constant: 9 (FIX 5.0 SP2).
48	securityID	R	SecurityID (uint64)	0 (8)	Security ID as defined by B3. For the SecurityID list, see the Security Definition message in Market Data feed.
22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID. Constant: “EXCHANGE_SYMBOL” (8).
207	securityExchange	C	SecurityExchangeBVMF (char)		Market to which the symbol belongs. Constant: "BVMF"
37035	matchEventIndicator	R	MatchEventIndicator Set (uint8)	8 (1)	Set of indicators that identify some market data events. Here are the possible bits applied to this message when it is set:

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
					Bit 4: Implied generated order. Bit 7: Last message for the event.
	<padding>			9 (1)	*
279	mDUpdateAction	C	MDUpdateAction Enum (uint8)		Update Action (DELETE). Constant: 2 (Delete).
269	mDEntryType	R	MDEntryType Enum (char)	10 (1)	Entry Type (BID or OFFER).
	<padding>			11 (5)	*
271	mDEntrySize	R	QuantityOptional (int64)	16 (8)	Last remaining quantity of the order immediately before the deletion.
198	secondaryOrderID	R	OrderID (uint64)	24 (8)	Exchange-generated identifier of the order that is deleted from the book. This identifier reflects the order's priority relative to other orders at the same price level in an ascending manner, where smaller values indicate higher priority.
60	transactTime	R	UTCTimestampNanos	32 (8)	Timestamp when the delete order event occurred in the matching process.
83	rptSeq	O	RptSeq (uint32)	40 (4)	Sequence number per instrument update. Zeroed in snapshot feed.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
270	mDEntryPx	O	PriceOptional	44 (8)	Price of the deleted order per share or contract (not MOA/MOC orders). Since schema version 15.

* Padding is implemented by declaring "offset" attribute in next field.

9.2.22 Message: MassDeleteOrders_MBO_52

Name	Template ID	Version	Encoded Length	Description
MassDeleteOrders_MBO_52	52	16	28	Disseminates mass deletion of orders.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (char)		Constant: "X" (Market Data Incremental Refresh).
1128	applVerID	C	ApplVerID Enum (uint8)		Specifies the service pack release being applied at message level. Constant: 9 (FIX 5.0 SP2).
48	securityID	R	SecurityID (uint64)	0 (8)	Security ID as defined by B3. For the SecurityID list, see the Security Definition message in Market Data feed.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID. Constant: “EXCHANGE_SYMBOL” (8).
207	securityExchange	C	SecurityExchangeBVMF (char)		Market to which the symbol belongs. Constant: "BVMF"
37035	matchEventIndicator	R	MatchEventIndicator Set (uint8)	8 (1)	Set of indicators that identify some market data events. Here are the possible bits applied to this message when it is set: Bit 7: Last message for the event.
279	mDUpdateAction	R	MDUpdateAction Enum (uint8)	9 (1)	Update Action (**DELETE_THRU**).
269	mDEntryType	R	MDEntryType Enum (char)	10 (1)	Entry Type (BID or OFFER).
	<padding>			11 (5)	*
60	transactTime	R	UTCTimestampNanos	16 (8)	Timestamp when the mass delete order events occurred in the matching process.
83	rptSeq	O	RptSeq (uint32)	24 (4)	Sequence number per instrument update. Zeroed in snapshot feed.

* Padding is implemented by declaring "offset" attribute in next field.

9.2.23 Message: Trade_53

Name	Template ID	Version	Encoded Length	Description
Trade_53	53	16	56	Relays trade information on one instrument.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (char)		Constant: "X" (Market Data Incremental Refresh).
1128	applVerID	C	ApplVerID Enum (uint8)		Specifies the service pack release being applied at message level. Constant: 9 (FIX 5.0 SP2).
48	securityID	R	SecurityID (uint64)	0 (8)	Security ID as defined by B3. For the SecurityID list, see the Security Definition message in Market Data feed.
22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID. Constant: “EXCHANGE_SYMBOL” (8).
207	securityExchange	C	SecurityExchangeBVMF (char)		Market to which the symbol belongs. Constant: "BVMF"

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
37035	matchEventIndicator	R	MatchEventIndicator Set (uint8)	8 (1)	Set of indicators that identify some market data events. Here are the possible bits applied to this message when it is set: Bit 4: Trade resulted from an implied generated order. Bit 7: Last message for the event.
279	mDUpdateAction	C	MDUpdateAction Enum (uint8)		Update Action (NEW). Constant: 0 (New).
269	mDEntryType	C	MDEntryType Enum (char)		Entry type: Trade. Constant: "2" (Trade or Trade Summary or Last Trade Price).
336	tradingSessionID	R	TradingSessionID Enum (uint8)	9 (1)	Identifier for trading session.
277	tradeCondition	R	TradeCondition Set (uint16)	10 (2)	Set of conditions describing a trade.
270	mDEntryPx	R	Price	12 (8)	Price of the Market Data Entry.
271	mDEntrySize	R	Quantity (int64)	20 (8)	Quantity or volume represented by the Market Data Entry.
1003	tradeID	R	TradeID (uint32)	28 (4)	Contains the unique identifier for this trade per instrument + trading date, as assigned by the exchange. Required if reporting a Trade.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
288	mDEntryBuyer	O	FirmOptional (uint32)	32 (4)	For reporting trades (buying party).
289	mDEntrySeller	O	FirmOptional (uint32)	36 (4)	For reporting trades (selling party).
75	tradeDate	R	LocalMktDate (uint16)	40 (2)	Used to specify the trading date for which a set of market data applies.
829	trdSubType	O	TrdSubType Enum (uint8)	42 (1)	Sub type of trade assigned to a trade. Since schema version 7.
	<padding>			43 (1)	*
60	transactTime	R	UTCTimestampNanos	44 (8)	Timestamp when the trade event occurred in the matching process. If a trade is amended, the value of the field is set manually by MktOps.
83	rptSeq	O	RptSeq (uint32)	52 (4)	Sequence number per instrument update. Zeroed in snapshot feed.

* Padding is implemented by declaring "offset" attribute in next field.

9.2.24 Message: ForwardTrade_54

Name	Template ID	Version	Encoded Length	Description
ForwardTrade_54	54	16	68	Relays trade information on one Forward instrument

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (char)		Constant: "X" (Market Data Incremental Refresh).
1128	applVerID	C	ApplVerID Enum (uint8)		Specifies the service pack release being applied at message level. Constant: 9 (FIX 5.0 SP2).
48	securityID	R	SecurityID (uint64)	0 (8)	Security ID as defined by B3. For the SecurityID list, see the Security Definition message in Market Data feed.
22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID. Constant: “EXCHANGE_SYMBOL” (8).
207	securityExchange	C	SecurityExchangeBVMF (char)		Market to which the symbol belongs. Constant: "BVMF"

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
37035	matchEventIndicator	R	MatchEventIndicator Set (uint8)	8 (1)	Set of indicators that identify some market data events. Here are the possible bits applied to this message when it is set: Bit 7: Last message for the event.
279	mDUpdateAction	C	MDUpdateAction Enum (uint8)		Update Action (NEW). Constant: 0 (New).
269	mDEntryType	C	MDEntryType Enum (char)		Entry type: Trade. Constant: "2" (Trade or Trade Summary or Last Trade Price).
336	tradingSessionID	R	TradingSessionID Enum (uint8)	9 (1)	Identifier for trading session.
277	tradeCondition	R	TradeCondition Set (uint16)	10 (2)	Set of conditions describing a trade.
270	mDEntryPx	R	Price	12 (8)	Price of the Market Data Entry.
271	mDEntrySize	R	Quantity (int64)	20 (8)	Quantity or volume represented by the Market Data Entry.
1003	tradeID	R	TradeID (uint32)	28 (4)	Contains the unique identifier for this trade per instrument + trading date, as assigned by the exchange. Required if reporting a Trade.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
288	mDEntryBuyer	O	FirmOptional (uint32)	32 (4)	For reporting trades (buying party).
289	mDEntrySeller	O	FirmOptional (uint32)	36 (4)	For reporting trades (selling party).
75	tradeDate	R	LocalMktDate (uint16)	40 (2)	Used to specify the trading date for which a set of market data applies.
60	transactTime	R	UTCTimestampNanos	42 (8)	Timestamp when the trade event occurred in the matching process. If a trade is amended, the value of the field is set manually by MktOps.
83	rptSeq	O	RptSeq (uint32)	50 (4)	Sequence number per instrument update. Zeroed in snapshot feed.
287	sellerDays	O	UInt16NULL (uint16)	54 (2)	Specifies the number of days that may elapse before delivery of the security. Only used for trades in forward market.
037014	mDEntryInterestRate	O	Percentage	56 (8)	Interest Rate of the Termo Trade. Expressed in decimal form. For example, 1% points are expressed and sent as 0.01. One basis point is represented as 0.0001.
829	trdSubType	O	TrdSubType Enum (uint8)	64 (1)	Sub type of trade assigned to a trade. Since schema version 7.
	<padding>			65 (3)	**

** Padding is implemented by fixing "blockLength" attribute in the sbe:message.

9.2.25 Message: ExecutionSummary_55

Name	Template ID	Version	Encoded Length	Description
ExecutionSummary_55	55	15	64	Relays execution summary information on one instrument.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (char)		Constant: "X" (Market Data Incremental Refresh).
1128	applVerID	C	ApplVerID Enum (uint8)		Specifies the service pack release being applied at message level. Constant: 9 (FIX 5.0 SP2).
48	securityID	R	SecurityID (uint64)	0 (8)	Security ID as defined by B3. For the SecurityID list, see the Security Definition message in Market Data feed.
22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID. Constant: “EXCHANGE_SYMBOL” (8).
207	securityExchange	C	SecurityExchangeBVMF (char)		Market to which the symbol belongs. Constant: "BVMF"

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
279	mDUpdateAction	C	MDUpdateAction Enum (uint8)		Update Action (NEW) - always replace. Constant: 0 (New).
269	mDEntryType	C	MDEntryType Enum (char)		Entry type: Execution Summary. Constant: "s" (Execution Summary for related Trades).
	<padding>			8 (2)	*
2446	aggressorSide	R	AggressorSide Enum (uint8)	10 (1)	Which side is aggressor of all fills.
	<padding>			11 (1)	*
31	lastPx	R	Price	12 (8)	Price of the last fill (i.e. worst price of this match).
1365	fillQty	R	Quantity (int64)	20 (8)	Quantity of all fills.
37779	tradedHiddenQty	O	QuantityOptional (int64)	28 (8)	Total quantity of matched passive orders that is not displayed to the market.
84	cxlQty	O	QuantityOptional (int64)	36 (8)	Total quantity canceled during matching process (e.g. due to self-trade).
2445	aggressorTime	R	UTCTimestampNanos	44 (8)	Timestamp of aggressive order resulting in match event

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
83	rptSeq	O	RptSeq (uint32)	52 (4)	Sequence number per instrument update. Zeroed in snapshot feed.
60	transactTime	R	UTCTimestampNanos	56 (8)	Timestamp when the matching event occurred in the matching process.

* Padding is implemented by declaring "offset" attribute in next field.

9.2.26 Message: ExecutionStatistics_56

Name	Template ID	Version	Encoded Length	Description
ExecutionStatistics_56	56	16	52	Relays execution summary statistics information on one instrument.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (char)		Constant: "X" (Market Data Incremental Refresh).
1128	applVerID	C	ApplVerID Enum (uint8)		Specifies the service pack release being applied at message level. Constant: 9 (FIX 5.0 SP2).
48	securityID	R	SecurityID (uint64)	0 (8)	Security ID as defined by B3. For the SecurityID list, see the Security Definition message in Market Data feed.
22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID. Constant: “EXCHANGE_SYMBOL” (8).
207	securityExchange	C	SecurityExchangeBVMF (char)		Market to which the symbol belongs. Constant: "BVMF"

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
37035	matchEventIndicator	R	MatchEventIndicator Set (uint8)	8 (1)	Set of indicators that identify some market data events. Here are the possible bits applied to this message when it is set: Bit 7: Last message for the event.
279	mDUpdateAction	C	MDUpdateAction Enum (uint8)		Update Action (NEW) - always replace. Constant: 0 (New).
269	mDEntryType	C	MDEntryType Enum (char)		Entry type: Execution Statistics. Constant: "9" (Execution Statistics for related Trades).
336	tradingSessionID	R	TradingSessionID Enum (uint8)	9 (1)	Identifier for trading session.
75	tradeDate	R	LocalMktDate (uint16)	10 (2)	Used to specify the trading date for which a set of market data applies.
1020	tradeVolume	R	QuantityVolume (int64)	12 (8)	Total traded volume for the session.
37778	vwapPx	O	PriceOptional	20 (8)	Volume-weighted average price.
451	netChgPrevDay	O	PriceOffset8Optional	28 (8)	Net change from previous trading day’s closing price vs. last traded price.
37073	numberOfTrades	R	NumberOfTrades (uint32)	36 (4)	Number of trades executed in the session.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
37033	mDEntryTimestamp	R	UTCTimestampNanos	40 (8)	Date and time of market data entry.
83	rptSeq	O	RptSeq (uint32)	48 (4)	Sequence number per instrument update. Zeroed in snapshot feed.

9.2.27 Message: TradeBust_57

Name	Template ID	Version	Encoded Length	Description
TradeBust_57	57	16	48	Relays trade bust (trade reversal) information on one instrument.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (char)		Constant: "X" (Market Data Incremental Refresh).
1128	applVerID	C	ApplVerID Enum (uint8)		Specifies the service pack release being applied at message level. Constant: 9 (FIX 5.0 SP2).
48	securityID	R	SecurityID (uint64)	0 (8)	Security ID as defined by B3. For the SecurityID list, see the Security Definition message in Market Data feed.
22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID. Constant: “EXCHANGE_SYMBOL” (8).
207	securityExchange	C	SecurityExchangeBVMF (char)		Market to which the symbol belongs. Constant: "BVMF"

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
37035	matchEventIndicator	R	MatchEventIndicator Set (uint8)	8 (1)	Set of indicators that identify some market data events. Here are the possible bits applied to this message when it is set: Bit 4: Bust of a trade resulted from an implied generated order. Bit 7: Last message for the event.
279	mDUpdateAction	C	MDUpdateAction Enum (uint8)		Update Action (DELETE) = always delete. Constant: 2 (Delete).
269	mDEntryType	C	MDEntryType Enum (char)		Entry type: Trade Bust. Constant: "u" (Trade busted by Market Supervision).
336	tradingSessionID	R	TradingSessionID Enum (uint8)	9 (1)	Identifier for trading session.
	<padding>			10 (2)	*
270	mDEntryPx	R	Price	12 (8)	Price of the Market Data Entry.
271	mDEntrySize	R	Quantity (int64)	20 (8)	Quantity or volume represented by the Market Data Entry.
1003	tradeID	R	TradeID (uint32)	28 (4)	Contains the unique identifier for this trade per instrument + trading date, as assigned by the exchange. Required if reporting a Trade.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
75	tradeDate	R	LocalMktDate (uint16)	32 (2)	Used to specify the trading date for which a set of market data applies.
	<padding>			34 (2)	*
60	transactTime	R	UTCTimestampNanos	36 (8)	Timestamp when the trade bust event occurred in the matching process.
83	rptSeq	O	RptSeq (uint32)	44 (4)	Sequence number per instrument update. Zeroed in snapshot feed.

* Padding is implemented by declaring "offset" attribute in next field.

9.2.28 Message: SnapshotFullRefresh_Header_30

Name	Template ID	Version	Encoded Length	Description
SnapshotFullRefresh_Header_30	30	16	34	Header for the snapshot of a single instrument

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
35	messageType	C	MessageType Enum (char)		Constant: "W" (Market Data Full Refresh).
1128	applVerID	C	ApplVerID Enum (uint8)		Specifies the service pack release being applied at message level. Constant: 9 (FIX 5.0 SP2).
48	securityID	R	SecurityID (uint64)	0 (8)	Security ID as defined by B3. For the SecurityID list, see the Security Definition message in Market Data feed.
22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID. Constant: “EXCHANGE_SYMBOL” (8).
207	securityExchange	C	SecurityExchangeBVMF (char)		Market to which the symbol belongs. Constant: "BVMF"

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
369	lastMsgSeqNumProcessed	R	SeqNum (uint32)	8 (4)	The last processed packet sequence number of the incremental channel as of the time the snapshot was generated. This value is used to synchronize the snapshot with the incremental feed.
911	totNumReports	R	UInt32 (uint32)	12 (4)	Total number of snapshots to be returned in the current replay loop.
37071	totNumBids	R	UInt32 (uint32)	16 (4)	Total number of bid orders that constitute this snapshot.
37072	totNumOffers	R	UInt32 (uint32)	20 (4)	Total number of ask orders that constitute this snapshot.
37070	totNumStats	R	UInt16 (uint16)	24 (2)	Total number of statistics (incremental and security status messages) that constitute this snapshot.
	<padding>			26 (2)	*
37083	lastRptSeq	O	RptSeq (uint32)	28 (4)	Last processed RptSeq (sequence number per instrument update) for this instrument. Can be used to synchronize the snapshot with the incremental feed if the client is only interested in a subset of the channel's instruments.

Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
37084	lastSequenceVersion	O	UInt16NULL (uint16)	32 (2)	**[NEW]** The last processed packet sequence version of the incremental channel as of the time the snapshot was generated. This value is used to synchronize the snapshot with the incremental feed. Since schema version 15.

* Padding is implemented by declaring "offset" attribute in next field.

9.2.29 Message: SnapshotFullRefresh_Orders_MBO_71

Name	Template ID	Version	Encoded Length	Description
SnapshotFullRefresh_Orders_MBO_71	71	16	11 + N	Partial list of orders for the snapshot of a single instrument.

	Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
	35	messageType	C	MessageType Enum (char)		Constant: "W" (Market Data Full Refresh).
	1128	applVerID	C	ApplVerID Enum (uint8)		Specifies the service pack release being applied at message level. Constant: 9 (FIX 5.0 SP2).

	Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
	48	securityID	R	SecurityID (uint64)	0 (8)	Security ID as defined by B3. For the SecurityID list, see the Security Definition message in Market Data feed.
	22	securityIDSource	C	SecurityIDSource Enum (char)		Identifies the class of the SecurityID. Constant: “EXCHANGE_SYMBOL” (8).
	207	securityExchange	C	SecurityExchangeBVMF (char)		Market to which the symbol belongs. Constant: "BVMF"
	268	**noMDEntries**	R	GroupSizeEncoding	8 (42)	Partial list of orders.
→	270	mDEntryPx	O	PriceOptional	0 (8)	Price per share or contract. Conditionally required if the order type requires a price (not market orders).
→	271	mDEntrySize	R	Quantity (int64)	8 (8)	Displayed quantity or volume represented by the Market Data Entry.
→		*<padding>*			16 (4)	*
→	37501	enteringFirm	O	FirmOptional (uint32)	20 (4)	Identifies the broker firm.

	Tag	Tag Name	Presence	Data Type (Primitive Type)	Offset (Size)	Comments
→	37034	mDInsertTimestamp	R	UTCTimestampNanos	24 (8)	The date and time when the order was inserted or re-inserted into the order book or manually altered by MktOps.
→	198	secondaryOrderID	R	OrderID (uint64)	32 (8)	Exchange-generated identifier of the order that updates with each order modification event involving a loss of priority, price change, or quantity replenishment in disclosed orders. This identifier reflects the order's priority relative to other orders at the same price-level in an ascending manner, where smaller values indicate higher priority.
→	269	mDEntryType	R	MDEntryType Enum (char)	40 (1)	Entry Type (BID or OFFER).
→	37035	matchEventIndicator	O	MatchEventIndicator Set (uint8)	41 (1)	Set of indicators that identify some market data events. Here are the possible bits applied to this message when it is set: Bit 4: Implied generated order. Since schema version 10.

10 COMPOSITE TYPES

10.1 Type: Fixed8

Name	Version	Encoded Length	Description
Fixed8	16	8	Decimal with constant exponent -8.

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: -8

10.2 Type: FramingHeader

Name	Version	Encoded Length	Description
FramingHeader	16	4	Framing Header (Compact SOFH).

Name	Data Type	Size	Offset	Description
messageLength	uint16	2	0	Overall message length including headers to support framing.
encodingType	uint16	2	2	Identifier of the encoding used in the message payload.

10.3 Type: GroupSizeEncoding

Name	Version	Encoded Length	Description
GroupSizeEncoding	16	3	Repeating group dimensions.

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

10.4 Type: MaturityMonthYear

Name	Version	Encoded Length	Description
MaturityMonthYear	16	5	Year, month, day and week of the maturity (used for standardized futures and options).

Name	Data Type	Size	Offset	Description
year	uint16	2	0	4-digit year
month	uint8	1	2	Month (1 to 12)
day	uint8	1	3	Day of month (1 to 31)
week	uint8	1	4	Week of month (1 to 5)

10.5 Type: messageHeader

Name	Version	Encoded Length	Description
messageHeader	16	9	Message identifiers and length of message root.

Name	Data Type	Size	Offset	Description
blockLength	uint16	2	0	Root block length.
templateId	uint16	2	2	Template ID.
schemaId	uint16	2	4	Schema ID.
version	uint16	2	6	Schema Version.

10.6 Type: PacketHeader

Name	Version	Encoded Length	Description
PacketHeader	16	16	Packet header.

Name	Data Type	Size	Offset	Description
channelNumber	uint8	1	0	Channel number.
reserved	uint8	1	1	Reserved.

Name	Data Type	Size	Offset	Description
sequenceVersion	uint16	2	2	Sequence version (incremented weekly or when market data engine failover happens).
sequenceNumber	uint32	4	4	Packet sequence number (reset to 1 when sequenceVersion increments).
sendingTime	uint64	8	8	Sending Time as number of nanoseconds since epoch (1970-01-01 00:00:00 UTC).

10.7 Type: Percentage

Name	Version	Encoded Length	Description
Percentage	16	8	Percentage (4 decimal places).

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

10.8 Type: Price

Name	Version	Encoded Length	Description
Price	16	8	Price (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

10.9 Type: Price8

Name	Version	Encoded Length	Description
Price8	16	8	Price (8 decimal places). For prices subjected to be adjusted from corporate events.

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: -8

10.10 Type: PriceOffset8Optional

Name	Version	Encoded Length	Description
PriceOffset8Optional	16	8	Price Offset (8 decimal places). Offsets related to prices subjected to be adjusted from corporate events.

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: -8

10.11 Type: PriceOptional

Name	Version	Encoded Length	Description
PriceOptional	16	8	Optional Price (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

10.12 Type: RatioQty

Name	Version	Encoded Length	Description
RatioQty	16	8	Ratio of quantity relative to the whole thing.

Name	Data Type	Size	Offset	Description
mantissa	int64	8	0	mantissa
exponent	int8	0	8	exponent. Constant: -7

10.13 Type: TextEncoding

Name	Version	Encoded Length	Description
TextEncoding	16	N	Variable-length short UTF-8 String.

Name	Data Type	Size	Offset	Description
length	uint8	1	0	
varData	char	N	1	

10.14 Type: UTCTimestampNanos

Name	Version	Encoded Length	Description
UTCTimestampNanos	16	8	UTC timestamp with nanosecond precision (Unix Epoch)

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

10.15 Type: UTCTimestampSeconds

Name	Version	Encoded Length	Description
UTCTimestampSeconds	16	8	UTC timestamp with second precision (Unix Epoch)

Name	Data Type	Size	Offset	Description
time	int64	8	0	UTC timestamp with second precision (Unix Epoch).
unit	uint8	0	8	time unit (seconds). Constant: 0

10.16 Type: VarString

Name	Version	Encoded Length	Description
VarString	16	N	Variable-length long UTF-8 String.

Name	Data Type	Size	Offset	Description
length	uint16	2	0	Length of a string, in bytes. For instance, the string 'Ação', converted to UTF-8, has 6 bytes, so length = 6.
varData	uint8	N	2	Bytes of the string, encoded in UTF-8.

11 SETS

11.1 Set: MatchEventIndicator

Name	Description	Version	Encoding Type	Encoded Length
MatchEventIndicator	Set of indicators that identify some market data events.	16	uint8	1

Name	Bit Number	Description
Reserved	0	0=Reserved for future use.
Reserved	1	0=Reserved for future use.
Reserved	2	0=Reserved for future use.
Reserved	3	0=Reserved for future use.
Implied	4	1=Implied generated order or trade, 0=Not implied. Since schema version 10.
RecoveryMsg	5	1=Message is sent during recovery process (QuoteCondition=R), 0=Not.
Reserved	6	0=Reserved for future use.
EndOfEvent	7	1=Last message for the event, 0=Not last.

11.2 Set: TradeCondition

Name	Description	Version	Encoding Type	Encoded Length
TradeCondition	Set of conditions describing a trade.	16	uint16	2

Name	Bit Number	Description
OpeningPrice	0	1=Opening Price (R), 0=Not.
Crossed	1	1=Crossed (X), 0=Not.
LastTradeAtTheSamePrice	2	1=Last Trade at the Same Price (L), 0=Not.
OutOfSequence	3	1=Out of sequence (Not last trade), 0=Last trade.
TradeOnBehalf	6	1=Marketplace Entered Trade (Trade on behalf) (2), 0=Not.
RegularTrade	13	1=Regular Trade, 0=Special trade type (see *TrdSubType* enum).
BlockTrade	14	1=Block Trade (see TrdSubType enum for details), 0=Not.

11.3 Set: ImbalanceCondition

Name	Description	Version	Encoding Type	Encoded Length
ImbalanceCondition	Set of conditions describing a imbalance.	16	uint16	2

Name	Bit Number	Description
ImbalanceMoreBuyers	8	1=Imbalance more buyers (P), 0=Not.
ImbalanceMoreSellers	9	1=Imbalance more sellers (Q), 0=Not.

12 ENUMERATIONS

Enum Type	Type	Description / Value Domain
AggressorSide	uint8	Which side is aggressor of this trade. 0 - NO_AGGRESSOR: Trade has no aggressor. 1 - BUY: Buy. 2 - SELL: Sell.
ApplVerID	uint8	Specifies the service pack release being applied at message level. 0 - FIX27: FIX 2.7. 1 - FIX30: FIX 3.0. 2 - FIX40: FIX 4.0. 3 - FIX41: FIX 4.1. 4 - FIX42: FIX 4.2. 5 - FIX43: FIX 4.3. 6 - FIX44: FIX 4.4. 7 - FIX50: FIX 5.0. 8 - FIX50SP1: FIX 5.0 SP1. 9 - FIX50SP2: FIX 5.0 SP2.
Boolean	uint8	Boolean type. 0 - FALSE_VALUE: false, N, 0. 1 - TRUE_VALUE: true, Y, 1.
ExerciseStyle	uint8	Type of exercise of a derivatives security. 0 - EUROPEAN: European. 1 - AMERICAN: American.
GovernanceIndicator	uint8	Corporative governance level indicator. Required for cash equities. 0 - No: Unspecified. 1 - N1: Level 1. 2 - N2: Level 2. 4 - NM: New Market. 5 - MA: Bovespa Mais.

		6 - MB: Over the counter (SOMA Market). 7 - M2: Bovespa Mais Level 2.
ImpliedMarketIndicator	uint8	Indicates that an implied order can be created for the instrument. 0 - NOT_IMPLIED: Not implied. 1 – IMPLIED: Implied enabled.
InstrAttribType	uint8	Code to represent the type of instrument attributes. 24 - TRADE_TYPE_ELIGIBILITY: Trade type eligibility details for security. 34 - GTD_GTC_ELIGIBILITY: Eligibility for GTD/GTC.
InstrAttribValue	uint8	Code to represent the type of instrument attributes. 1 - ELECTRONIC_MATCH_OR_GTD_GTC_ELIGIBLE: Electronic Match Eligible (871=24) or GTD/GTC Eligible (871=34). 2 - ORDER_CROSS_ELIGIBLE: Order Cross Eligible (871=24). 3 - BLOCK_TRADE_ELIGIBLE: Block Trade Eligible (871=24). 14 - FLAG_RFQ_FOR_CROSS_ELIGIBLE: Request for Quote (RFQ) for Cross Eligible (871=24). 17 - NEGOTIATED_QUOTE_ELIGIBLE: Negotiated Quote Eligible (871=24).
LotType	uint8	Describes the lot type for the instruments. Used for the Equities segment. 1 - ODD_LOT: Odd lot. 2 - ROUND_LOT: Round lot. 3 - BLOCK_LOT: Block lot.
MDEntryType	char	Type of the Market Data Entry. "0" - BID: Bid. "1" - OFFER: Offer. "2" - TRADE: Trade or Trade Summary or Last Trade Price. "3" - INDEX_VALUE: Index Value. "4" - OPENING_PRICE: Opening Price. "5" - CLOSING_PRICE: Closing Price. "6" - SETTLEMENT_PRICE: Settlement Price. "7" - SESSION_HIGH_PRICE: Session High Price. "8" - SESSION_LOW_PRICE: Session Low Price. "9" - EXECUTION_STATISTICS: Execution Statistics for related Trades. "A" - IMBALANCE: Imbalance. "B" - TRADE_VOLUME: Trade Volume. "C" - OPEN_INTEREST: Open Interest.

		"J" - EMPTY_BOOK: Empty Book. "c" - SECURITY_TRADING_STATE_PHASE: Security Trading State / Phase. "g" - PRICE_BAND: Price band. "h" - QUANTITY_BAND: Quantity band. "D" - COMPOSITE_UNDERLYING_PRICE: Composite Underlying Price. "s" - EXECUTION_SUMMARY: Execution Summary for related Trades. "v" - VOLATILITY_PRICE: Volatility price. "u" - TRADE_BUST: Trade busted by Market Supervision. “b” – BEST_BID_OFFER: Best Bid and Offer.
MDUpdateAction	uint8	Types of Market Data update action. 0 - NEW: New. 1 - CHANGE: Change. 2 - DELETE: Delete. 3 - DELETE_THRU: Delete Thru - only for MBO. 4 - DELETE_FROM: Delete From - only for MBO. 5 - OVERLAY: Overlay - not used.
MessageType	char	FIX Message Type. "0" - Sequence: Sequence message for heartbeat. "4" - SequenceReset: Sequence Reset. "X" - MarketDataIncrementalRefresh: Market Data Incremental Refresh. "f" - SecurityStatus: Security Status. "d" - SecurityDefinition: Security Definition. "B" - News: News. "W" - MarketDataSnapshotFullRefresh: Market Data Full Refresh.
MultiLegModel	uint8	Specifies the type of multileg order. Defines whether the security is pre-defined or user-defined. 0 - PREDEFINED: Predefined Multileg Security. 1 - USER_DEFINED: User-Defined Multileg Security.
MultiLegPriceMethod	uint8	Code to represent how the multileg price is to be interpreted when applied to the legs. 0 - NET_PRICE: Net Price. 1 - REVERSED_NET_PRICE: Reversed Net Price. 2 - YIELD_DIFFERENCE: Yield Difference. 3 - INDIVIDUAL: Individual.

		4 - CONTRACT_WEIGHTED_AVERAGE_PRICE: Contract Weighted Average Price. 5 - MULTIPLIED_PRICE: Multiplied Price.
NewsSource	uint8	Source for the news. 0 - OTHER: Other news source. 1 - DCM: DCM. 2 - BBMNET: BBMNet. 3 - MARKET_SURVEILLANCE: MarketSurveillance. 4 - INTERNET: Internet. 5 - DPR_VE: DPR-VE. 19 - MKT_OPS_FX_AGENCY: Mkt Ops FX Agency. 20 - MKT_OPS_DERIVATIVES_AGENCY: Mkt Ops Derivatives Agency. 11 - OVER_THE_COUNTER_NEWS_AGENCY: Over-the-counter News Agency. 13 - ELECTRONIC_PURCHASE_EXCHANGE: Electronic Purchase Exchange. 14 - CBLC_NEWS_AGENCY: CBLC News Agency. 15 - BOVESPA_INDEX_AGENCY: BOVESPA – Index Agency. 16 - BOVESPA_INSTITUTIONAL_AGENCY: BOVESPA – Institutional Agency. 17 - MKT_OPS_EQUITIES_AGENCY: Mkt Ops Equities Agency. 18 - BOVESPA_COMPANIES_AGENCY: BOVESPA – Companies Agency.
OpenCloseSettlFlag	uint8	Flag that identifies if the opening/closing/settlement price is related to theoretical, daily, previous business day or just an updated value. 0 - DAILY: Daily settlement entry. 1 - SESSION: Session settlement entry. 3 - EXPECTED_ENTRY: Expected entry (preliminary price). 4 - ENTRY_FROM_PREVIOUS_BUSINESS_DAY: Entry from previous business day. 5 - THEORETICAL_PRICE: Theoretical Price.
OptPayoutType	UInt8NULL	Indicates the type of payout that will result from an in-the-money option. 1 – VANILLA: Vanilla. 2 – CAPPED: Capped. 3 – BINARY: Digital (Binary).
PriceBandMidpointPriceType	uint8	Band Midpoint Type, used with Auction Price Banding. 0 - LAST_TRADED_PRICE: Last traded price. 1 - COMPLEMENTARY_LAST_PRICE: Complementary last price. 2 - THEORETICAL_PRICE: Theoretical Price.

PriceBandType	uint8	Indicates the type of price banding (tunnel). Used for Price Banding when tags LowLimitPrice and HighLimitPrice are sent. 1 - HARD_LIMIT: Hard Limit. 2 - AUCTION_LIMITS: Auction Limits. 3 - REJECTION_BAND: Rejection Band. 4 - STATIC_LIMITS: Static Limits.
PriceLimitType	uint8	Describes how the price limits are expressed. The default value is “0” (Price Unit). 0 - PRICE_UNIT: Price Unit. 1 - TICKS: Ticks. 2 - PERCENTAGE: Percentage.
PriceType	UInt8NULL	Code to represent the price type. If absent (zero), the default value is DECIMALS. 1 - PERCENTAGE: Percentage. 2 - PU: Per unit (i.e., per share or contract). 3 - FIXED_AMOUNT: Fixed amount (absolute value).
Product	uint8	Indicates the type of product the security is associated with. 2 - COMMODITY: Commodity. 3 - CORPORATE: Corporate Fixed Income. 4 - CURRENCY: Currency. 5 - EQUITY: Equity. 6 - GOVERNMENT: Public debt. 7 - INDEX: Index. 15 - ECONOMIC_INDICATOR: Economic indicator. 16 - MULTILEG: Multileg.
PutOrCall	uint8	Indicates whether an option contract is a put or call. 0 - PUT: Put Option. 1 - CALL: Call Option.
SecurityIDSource	char	Identifies class or source of the *SecurityID* <tag 48> value. "4" - ISIN "8" - EXCHANGE_SYMBOL
SecurityMatchType	uint8	Type of matching that occurred. Required for Special Auctions. 8 - ISSUING_BUY_BACK_AUCTION: Issuing/Buy Back Auction.
SecuritySubType	uint16	The subtype of the instrument. 4 - FX_SPOT: FX spot.

		10 - GOLD: Gold. 20 - INDEX: Index. 30 - INTEREST_RATE: Interest rate. 40 - FX_RATE: FX rate. 50 - FOREIGN_DEBT: Foreign debt. 60 - AGRICULTURAL: Agricultural. 70 - ENERGY: Energy. 80 - ECONOMIC_INDICATOR: Economic Indicator. 90 - STRATEGY: Strategy. 91 - STRATEGY_DOLLAR_SPOT: Strategy Dollar Spot. 100 - FUTURE_STYLE_OPTION: Future-style Option. 110 - VOLATILITY: Volatility. 120 - SWAP: Swap. 130 - MINI_CONTRACT: MiniContract. 140 - FINANCIAL_ROLLOVER: Financial RollOver. 141 - AGRICULTURAL_ROLLOVER: Agricultural RollOver. 142 - STOCK_ROLLOVER: Stock RollOver. 150 - TARGET_RATE: Target Rate. 160 - CRYPTO_ASSET: Crypto Asset. 190 - CARBON_CREDIT: Carbon credit. 1001 - ORDINARY_RIGHTS_DO: Ordinary Rights (DO). 1002 - PREFERRED_RIGHTS_DP: Preferred Rights (DP). 1003 - COMMON_SHARES_ON: Common Shares (ON). 1004 - PREFERRED_SHARES_PN: Preferred Shares (PN). 1005 - CLASS_A_PREFERRED_SHARES_PNA: Class A preferred shares (PNA). 1006 - CLASS_B_PREFERRED_SHARES_PNB: Class B preferred shares (PNB). 1007 - CLASS_C_PREFERRED_SHARES_PNC: Class C preferred shares (PNC). 1008 - CLASS_D_PREFERRED_SHARES_PND: Class D preferred shares (PND). 1009 - ORDINARY_RECEIPTS_ON_REC: Ordinary Receipts (ON REC). 1010 - PREFERRED_RECEIPTS_PN_REC: Preferred Receipts (PN REC). 1020 - TRADE_AT_CLOSE: Trade at Close (TAC). 1021 - TRADE_AT_AVERAGE: Trade at Average (TAA).

		1022 - BLOCK TRADING - MIDPOINT ORDER BOOK. 1023 - BLOCK TRADING - BOOK OF BLOCK TRADE. 1024 - BLOCK TRADING - REQUEST FOR QUOTE (RFQ). 1100 - COMMON_FORWARD: Common Forward. 1101 - FLEXIBLE_FORWARD: Flexible Forward. 1102 - DOLLAR_FORWARD: Dollar Forward. 1103 - INDEX_POINTS_FORWARD: Index Points Forward. 1104 - NON_TRADABLE_ETF_INDEX: Non-tradable ETF Index. 1105 - PREDEFINED_COVERED_SPREAD: Predefined Covered Spread. 1106 - TRADABLE_ETF: Tradable ETF. 1107 - NON_TRADABLE_INDEX: Non-tradable Index. 1108 - USER_DEFINED_SPREAD: User defined spread. 1109 - EXCHANGE_DEFINED_SPREAD: Exchange defined spread (not currently in use). 1110 - SECURITY_LOAN: Security Loan. 1111 - TRADABLE_INDEX: Tradable Index. 1112 - BRAZILIAN_DEPOSITARY_RECEIPT: Brazilian Depositary Receipt. 1113 - FUND: Fund. 1114 - OTHER_RECEIPT: Other Receipt. 1115 - OTHER_RIGHT: Other Right. 1116 - UNIT: UNIT. 1117 - CLASS_E_PREFERRED_SHARE_PNE: Class E Preferred Share (PNE). 1118 - CLASS_F_PREFERRED_SHARE_PNF: Class F Preferred Share (PNF). 1119 - CLASS_G_PREFERRED_SHARE_PNG: Class G Preferred Share (PNG). 1120 - WARRANT: Warrant. 1122 - NON_TRADABLE_SECURITY_LENDING: Non-tradable Security Lending. 1123 - FOREIGN_INDEX_ETF: Foreign Index ETF. 1124 - FIXED_INCOME_ETF: Fixed Income ETF. 1125 - GOVERNMENT_ETF: Government ETF. 1126 - IPO_FOLLOW_ON: IPO or Follow on. 1127 - GROSS_AUCTION: Gross Auction. 1128 - NET_AUCTION: Net Auction. 1129 - TRADABLE_INDEX_PARTNERSHIP: Tradable Index in Partnership.

		1130 - NONTRADABLE_INDEX_PARTNERSHIP: Nontradable Index in Partnership. 1131 - NONTRADABLE_FIXED_INCOME_ETF: Nontradable Fixed Income ETF. 1135 - TRADABLE_REAL_ESTATE_ETF: Tradable Real Estate ETF. 1136 - NON_TRADABLE_INDEX_REAL_ESTATE_ETF: Non-tradable Index in Real Estate ETF. 1300 - OUTRIGHT_PURCHASE: Outright purchase. 1301 - SPECIFIC_COLLATERAL_REPO: Specific collateral repo. 1302 - DEBENTURE: Debenture. 1303 - REAL_STATE_RECEIVABLE_CERTIFICATE: Real State Receivable Certificate. 1304 - AGRIBUSINESS_RECEIVABLE_CERTIFICATE: Agribusiness Receivable Certificate. 1305 - PROMISSORY_NOTE: Promissory Note. 1306 - LETRA_FINANCEIRA: Letra Financeira. 1307 - AMERICAN_DEPOSITARY_RECEIPT: American Depositary Receipt. 1308 - UNIT_INVESTMENT_FUND: Unit Investment Fund. 1309 - RECEIVABLE_INVESTMENT_FUND: Receivable Investment Fund. 1310 - OUTRIGHT_T1: Outright T+1. 1311 - REPO_T1: Repo T+1. 1312 - NON_TRADABLE_GROSS_SETTLEMENT: Non-tradable gross settlement. 1313 - NON_TRADABLE_NET_SETTLEMENT: Non-tradable net settlement. 1314 - ETF_PRIMARY_MARKET: ETF Primary Market. 1316 - SHARES_PRIMARY_MARKET: Shares Primary Market. 1317 - RIGHTS_PRIMARY_MARKET: Rights Primary Market. 1318 - UNIT_PRIMARY_MARKET: Unit Primary Market. 1319 - FUND_PRIMARY_MARKET: Fund Primary Market. 1320 - FOREIGN_INDEX_ETF_PRIMARY_MARKET: Foreign Index ETF Primary Market. 1321 - WARRANT_PRIMARY_MARKET: Warrant Primary Market. 1322 - RECEIPT_PRIMARY_MARKET: Receipt Primary Market. 1999 - OTHERS: Others. 2002 - GERMAN_PUBLIC_DEBTS: German Public Debts. 4000 - INFRASTRUCTURE_INVESTMENT_FUND: Infrastructure Investment Fund. 4001 - MULTIMARKET_INVESTMENT_FUND: Multimarket Investment Fund. 4002 - FIXED_INCOME_INVESTMENT_FUND: Fixed Income Investment Fund. 4003 - CURRENCY_INVESTMENT_FUND: Currency Investment Fund .

SecurityTradingEvent	uint8	Identifies an event related to a Trade. This tag is also used to mark when an instrument state is kept separate from the group phase, or when the instrument state follows the default group phase (stops having a separate, defined state). 4 - TRADING_SESSION_CHANGE: Change of Trading Session. 101 - SECURITY_STATUS_CHANGE: Security Status maintained separately from Group Status. 102 - SECURITY_REJOINS_SECURITY_GROUP_STATUS: Security Status following Group Status.
SecurityTradingStatus	uint8	Status related to a given Instrument or phase related to a SecurityGroup where the instrument belongs to. 2 - PAUSE: Trading Halt (Pause). 4 - CLOSE: No-Open (Close). 17 - OPEN: Ready To Trade (Open). 18 - FORBIDDEN: Not Available For Trading (Forbidden). 20 - UNKNOWN_OR_INVALID: Unknown Or Invalid. 21 - RESERVED: Pre Open (Reserved). 101 - FINAL_CLOSING_CALL: Final Closing Call.
SecurityType	uint8	Indicates the type of the security. 1 - CASH: Rights, etc. 2 - CORP: Corporate Fixed Income. 3 - CS: Common Stock. 4 - DTERM: Derivative Forward or 'Termo'. 5 - ETF: Exchange Traded Fund. 6 - FOPT: Future Options. 7 - FORWARD: Equity Forward or 'Termo'. 8 - FUT: Futures. 9 - INDEX: Non-Tradable index. 10 - INDEXOPT 11 - MLEG: Multileg Instrument. 12 - OPT: Option. 13 - OPTEXER: Option Exercise. 14 - PS: Preferred Stock. 15 - SECLOAN 16 - SOPT: Spot Options. 17 - SPOT: Spot Market.
SecurityUpdateAction	char	Action used when updating the security.

		"A" - ADD: Add. "D" - DELETE: Delete. "M" - MODIFY: Modify.
Side	uint8	Side of order. 1 - BUY 2 - SELL
TradingSessionID	uint8	Phase related to a SecurityGroup where the instrument belongs to. 1 - REGULAR_TRADING_SESSION: Regular day session. 6 - NON_REGULAR_TRADING_SESSION: Non-Regular Session (after market).
TradingSessionSubID	uint8	Status related to a given Instrument or phase related to a SecurityGroup where the instrument belongs to. 2 - PAUSE: Trading Halt (Pause). 4 - CLOSE: No-Open (Close). 17 - OPEN: Ready To Trade (Open). 18 - FORBIDDEN: Not Available For Trading (Forbidden). 20 - UNKNOWN_OR_INVALID: Unknown Or Invalid. 21 - RESERVED: Pre Open (Reserved). 101 - FINAL_CLOSING_CALL: Final Closing Call.
TrdSubType	UInt8NULL	101 - MULTI_ASSET_TRADE: Multi Asset Trade (Termo Vista). 102 - LEG_TRADE: Leg Trade (UDS/EDS). 103 - MIDPOINT_TRADE: Midpoint Trade (MP). 104 - BLOCK_BOOK_TRADE: Block Book Trade (PT). 105 - RF_TRADE: Equities: RFQ Trade, Futures: Fixed Income Trade (RF). 106 - RLP_TRADE: RLP Trade (RL). 107 - TAC_TRADE: Trade at Close Trade (TC). 108 - TAA_TRADE: Trade at Average Trade (TA). 109 - SWEEP_TRADE: Sweep Trade (SW).

13 SECURITYSUBTYPE VALUES

SecuritySubType	**The subtype of the instrument:** 4 - FX_SPOT: FX spot.

	10 - GOLD: Gold. 20 - INDEX: Index. 30 - INTEREST_RATE: Interest rate. 40 - FX_RATE: FX rate. 50 - FOREIGN_DEBT: Foreign debt. 60 - AGRICULTURAL: Agricultural. 70 - ENERGY: Energy. 80 - ECONOMIC_INDICATOR: Economic Indicator. 90 - STRATEGY: Strategy. 91 - STRATEGY_DOLLAR_SPOT: Strategy Dollar Spot. 92 - Strategy Interest Rate. 100 - FUTURE_STYLE_OPTION: Future-style Option. 110 - VOLATILITY: Volatility. 120 - SWAP: Swap. 130 - MINI_CONTRACT: Mini Contract. 140 - FINANCIAL_ROLLOVER: Financial RollOver. 141 - AGRICULTURAL_ROLLOVER: Agricultural RollOver. 142 - STOCK_ROLLOVER: Stock RollOver. 150 - TARGET_RATE: Target Rate. 160 - Crypto Asset. 190 - CARBON_CREDIT: Carbon credit. 1001 - ORDINARY_RIGHTS_DO: Ordinary Rights (DO). 1002 - PREFERRED_RIGHTS_DP: Preferred Rights (DP). 1003 - COMMON_SHARES_ON: Common Shares (ON). 1004 - PREFERRED_SHARES_PN: Preferred Shares (PN). 1005 - CLASS_A_PREFERRED_SHARES_PNA: Class A preferred shares (PNA). 1006 - CLASS_B_PREFERRED_SHARES_PNB: Class B preferred shares (PNB). 1007 - CLASS_C_PREFERRED_SHARES_PNC: Class C preferred shares (PNC). 1008 - CLASS_D_PREFERRED_SHARES_PND: Class D preferred shares (PND). 1009 - ORDINARY_RECEIPTS_ON_REC: Ordinary Receipts (ON REC). 1010 - PREFERRED_RECEIPTS_PN_REC: Preferred Receipts (PN REC). 1020 - TRADE_AT_CLOSE: Trade at Close (TAC).

	1021 - TRADE_AT_AVERAGE: Trade at Average (TAA). 1022 - BLOCK TRADING - MIDPOINT ORDER BOOK. 1023 - BLOCK TRADING - BOOK OF BLOCK TRADE. 1024 - BLOCK TRADING - REQUEST FOR QUOTE (RFQ). 1100 - COMMON_FORWARD: Common Forward. 1101 - FLEXIBLE_FORWARD: Flexible Forward. 1102 - DOLLAR_FORWARD: Dollar Forward. 1103 - INDEX_POINTS_FORWARD: Index Points Forward. 1104 - NON_TRADABLE_ETF_INDEX: Non-tradable ETF Index. 1105 - PREDEFINED_COVERED_SPREAD: Predefined Covered Spread. 1106 - TRADABLE_ETF: Tradable ETF. 1107 - NON_TRADABLE_INDEX: Non-tradable Index. 1108 - USER_DEFINED_SPREAD: User defined spread. 1109 - EXCHANGE_DEFINED_SPREAD: Exchange defined spread (not currently in use). 1110 - SECURITY_LOAN: Security Loan. 1111 - TRADABLE_INDEX: Tradable Index. 1112 - BRAZILIAN_DEPOSITARY_RECEIPT: Brazilian Depositary Receipt. 1113 - FUND: Fund. 1114 - OTHER_RECEIPT: Other Receipt. 1115 - OTHER_RIGHT: Other Right. 1116 - UNIT: Unit. 1117 - CLASS_E_PREFERRED_SHARE_PNE: Class E Preferred Share (PNE). 1118 - CLASS_F_PREFERRED_SHARE_PNF: Class F Preferred Share (PNF). 1119 - CLASS_G_PREFERRED_SHARE_PNG: Class G Preferred Share (PNG). 1120 - WARRANT: Warrant. 1122 - NON_TRADABLE_SECURITY_LENDING: Non-tradable Security Lending. 1123 - FOREIGN_INDEX_ETF: Foreign Index ETF. 1124 - FIXED_INCOME_ETF: Fixed Income ETF. 1125 - GOVERNMENT_ETF: Government ETF. 1126 - IPO_FOLLOW_ON: IPO or Follow on. 1127 - GROSS_AUCTION: Gross Auction. 1128 - NET_AUCTION: Net Auction.

	1129 - TRADABLE_INDEX_PARTNERSHIP: Tradable Index in Partnership. 1130 - NONTRADABLE_INDEX_PARTNERSHIP: Non-tradable Index in Partnership. 1131 - NONTRADABLE_FIXED_INCOME_ETF: Non-tradable Fixed Income ETF. 1135 - TRADABLE_REAL_ESTATE_ETF: Tradable Real Estate ETF. 1136 - NON_TRADABLE_INDEX_REAL_ESTATE_ETF: Non-tradable Index in Real Estate ETF. 1137 - CRYPTOCURRENCY ETF: Crypto-currency ETF. 1138 - CURRENCY ETF: Currency ETF. 1139 - FOREIGN FIXED INCOME ETF: Foreign Fixed Income ETF. 1150 - BDR EQUITY ETF: Brazilian Depositary Receipt Equity ETF. 1151 - BDR COMMODITY ETF: Brazilian Depositary Receipt Commodity ETF. 1152 - BDR FIXED INCOME ETF: Brazilian Depositary Receipt Fixed Income ETF. 1153 - BDR DEBT: Brazilian Depositary Receipt Debt. 1154 - BDR CURRENCY ETF: Brazilian Depositary Receipt Currency ETF. 1200 – COLLATERAL: Collateral. 1300 - OUTRIGHT_PURCHASE: Outright purchase. 1301 - SPECIFIC_COLLATERAL_REPO: Specific collateral repo. 1302 - DEBENTURE: Debenture. 1303 - REAL_STATE_RECEIVABLE_CERTIFICATE: Real State Receivable Certificate. 1304 - AGRIBUSINESS_RECEIVABLE_CERTIFICATE: Agribusiness Receivable Certificate. 1305 - PROMISSORY_NOTE: Promissory Note. 1306 - LETRA_FINANCEIRA: Financial notes. 1307 - AMERICAN_DEPOSITARY_RECEIPT: American Depositary Receipt. 1308 - UNIT_INVESTMENT_FUND: Unit Investment Fund. 1309 - RECEIVABLE_INVESTMENT_FUND: Receivable Investment Fund. 1310 - OUTRIGHT_T1: Outright T+1. 1311 - REPO_T1: Repo T+1. 1312 - NON_TRADABLE_GROSS_SETTLEMENT: Non-tradable gross settlement. 1313 - NON_TRADABLE_NET_SETTLEMENT: Non-tradable net settlement. 1314 - ETF_PRIMARY_MARKET: ETF Primary Market. 1316 - SHARES_PRIMARY_MARKET: Shares Primary Market. 1317 - RIGHTS_PRIMARY_MARKET: Rights Primary Market. 1318 - UNIT_PRIMARY_MARKET: Unit Primary Market.

	1319 - FUND_PRIMARY_MARKET: Fund Primary Market. 1320 - FOREIGN_INDEX_ETF_PRIMARY_MARKET: Foreign Index ETF Primary Market. 1321 - WARRANT_PRIMARY_MARKET: Warrant Primary Market. 1322 - RECEIPT_PRIMARY_MARKET: Receipt Primary Market. 1999 - OTHERS: Others. 2000 - BANKING TERM DEPOSIT: Bank Term Deposit. 2001 - BRAZILLIAN PUBLIC DEBTS: Brazilian Public Debts. 2002 - GERMAN_PUBLIC_DEBTS: German Public Debts. 4000 - INFRASTRUCTURE_INVESTMENT_FUND: Infrastructure Investment Fund. 4001 - MULTIMARKET_INVESTMENT_FUND: Multimarket Investment Fund. 4002 - FIXED_INCOME_INVESTMENT_FUND: Fixed Income Investment Fund. 4003 - CURRENCY_INVESTMENT_FUND: Currency Investment Fund. 4004 - AGRO INVESTMENT FUND (FII): Agro Investment Fund (FII). 4005 - AGRO INVESTMENT FUND (FIDC): Agro Investment Fund (FIDC). 4006 - AGRO INVESTMENT FUND (FIP): Agro Investment Fund (FIP). 5000 - NONTRADABLE SECURITY FIXED INCOME: Non-tradable Security Fixed Income.