

# Cloud Stream User Manual

Version: 1.0

Related to: Cloud Stream Release 1.0

Last update: 16-Jan-2023

## Table of contents

<b>1</b>	<b>INTRODUCTION.....</b>	<b>3</b>
<b>2</b>	<b>AUTHENTICATION .....</b>	<b>4</b>
<b>3</b>	<b>MESSAGING – FEED SUBSCRIPTION.....</b>	<b>5</b>
<b>4</b>	<b>SERVICE AVAILABILITY .....</b>	<b>6</b>
<b>5</b>	<b>DATA AND SERVICE MESSAGES .....</b>	<b>7</b>
<b>6</b>	<b>APPENDIX A – MICROPRODUCTS FEED DESCRIPTION.....</b>	<b>8</b>
<b>7</b>	<b>APPENDIX B – TRADEGATE DESCRIPTION .....</b>	<b>14</b>
<b>8</b>	<b>APPENDIX C – KAIKO DESCRIPTION.....</b>	<b>18</b>
<b>8.1</b>	<b>GENERAL REFERENCE DATA PROCESSING RULES.....</b>	<b>18</b>
<b>8.2</b>	<b>MESSAGE STRUCTURE DESCRIPTION.....</b>	<b>19</b>
8.2.1	Message type: Market Data Report.....	19
8.2.2	Message type: Asset.....	19
8.2.3	Message type: Exchange .....	20
8.2.4	Message type: Instrument .....	20
8.2.5	Message type: Pool .....	21
8.2.6	Message type: Trade .....	22
8.2.7	Message type: TopOfBookUpdate .....	23
8.2.8	Message type: MarketDataUpdate .....	24
8.2.9	Message type: MarketDataSnapshot .....	25
<b>9</b>	<b>CHANGE LOG .....</b>	<b>26</b>

---

## 1 Introduction

Cloud Stream allows customers to easily retrieve real-time public market data feeds of selected DBAG products via cloud-based solution. The solution is designed to minimize the entry barrier for market data access by using standard solutions and technologies:

- Connectivity is available via public internet or via VPC access in AWS Frankfurt region by using WebSocket technology
- Both binary (Google Protocol Buffers – GPB) and ascii (JSON) data encoding is supported
- Message content is – as far as possible – aligned to FIX specification; thus, field names and valid values can be easily understood and in addition, supported valid values are documented in the GPB protocol description.
- Cloud Stream solution is furthermore designed for support any kind of data feeds, beside trades, quotes, aggregates (L2) orderbook also incremental data feeds with full depth orderbook data.

This document describes the message layouts and the technical details of the streaming solution.

This manual relates to the interface version number 001.000.000.

---

## 2 Authentication

The streaming WebSocket API is secured by using standard Bearer authentication method and a short living access token – which is only required during establishing the connection. The token can be generated by the following URL:

```
https://md.deutsche-boerse.com/login
```

The POST request has to provide the username / password as JSON object in the body part:

```
{
  "username": "customer_id",
  "password": "secure_pwd"
}
```

In addition, the HTTP Header field "Content-Type" has to be set to json:

```
Content-Type: "application/json"
```

The response will contain beside the access token some additional fields; for accessing the stream, only the access token field is relevant:

```
{
  "AccessToken": "<token>"
  "ExpiresIn": <seconds>
  "TokenType": "Bearer"
}
```

The real-time stream can be reached by the following URL:

```
wss://md.deutsche-boerse.com/stream?format= <json|binary>
```

The access token has to be provided in the HTTP Authentication header field:

```
"Authentication": „Bearer <token>“
```

### Differences between json and binary format:

You can choose during connecting to the feed between binary and json format. Depending on the targeted performance, binary format is faster to process and consumes less bandwidth, than json, and is to be preferred in general.

### 3 Messaging – feed subscription

In order to retrieve market data feed, the customer has to select the interested data stream. For this reason, after the physical connection has been established, customer has to *subscribe* to the stream of interest. This can be done sending the Subscribe request to the server:

Fieldname	Description	Content
event	Event (action) name	"subscribe"
requestId	Optional request id, will be returned to the client as part of the response message.	Identification provided by the customer.
stream	List of stream name and stream offset in case of data recovery.	e.g. <pre>{"stream": "md-microproducts", "startTime": "1659008334123456789"}</pre> startTime is in nanoseconds since epoch, TZ=UTC

#### Unsubscribe

If the customer is no more interested to receive data from the stream, the unsubscribe request can be used:

Fieldname	Description	Content
event	Event (action) name	"unsubscribe"
requestId	Optional request id, will be returned to the client as part of the response message.	Identification provided by the customer.
stream	List of stream names.	e.g. <pre>"stream": ["md-microproducts"]</pre>

#### 4 Service availability

The service will be technically available 24x7 without interruption; planned maintenance will be announced in advance. The respective data made available via Cloud Stream is only available and updated if the relevant trading venue or other source(s) of the data is open for business and/or trading.

Service	Telephone	Email
Functional Support	+49 (0)69 – 211 - 11540	datafeeds@deutsche-boerse.com
Contractual Support	+49-(0)69 – 211 - 13440	data.services@deutsche-boerse.com

## 5 Data and service messages

The message, which will be sent to the customer based on his subscription were defined in a generic mode in order, any kind of messages can be transported by the same technical interface. The 'stream message' contains the following fields:

Field Name	Type	Description
subs	string	Subscription information – stream or topic name
seq	UInt64	Sequence number of the message in the stream; this number has to be used on case of message recovery.
messages	object	List of messages – the list will contain usually only a single message.

## 6 Appendix A – Microproducts feed description

Message type: Market Data

FIX Tag	FIX Field Name	Req'd	FAST Data Type	Description
35	MsgTyp	Y	enum	Message type '0' = MarketDataIncrementalRefresh ('X') '1' = MarketDataSnapshotFullRefresh ('W')
< ApplSeqCtrl > group starts				
1180	> ApplID	Y	UInt32	Application ID
1181	> ApplSeqNum	Y	UInt64	Application Sequence Number
< ApplSeqCtrl > group ends				
< Instrument > group starts				
1301	> MktID	Y	string	Market Identifier
55	> Sym	Y	string	Symbol – instrument identification
48	> ID	Y	string	SecurityID
22	> Src	N	enum	Security ID Source '0' = ISIN '1' = Exchange Symbol '2' = Synthetic
167	> SecTyp	Y	enum	Security Type '0' = NO_SECURITYTYPE '1' = Future '2' = Option '3' = Multileg Instrument '4' = Index '5' = Exchange traded commodity '6' = Exchange traded note '7' = Common stock '8' = Repurchase '9' = Cash '10' = Foreign Exchange Contract



				'11' = Bond '12' = Mutual Fund '13' = Investment Fund '14' = Interest Rate Swap '15' = Subscription Rights '16' = Warrant '99' = Other
15	> Ccy	N	string	Currency
200	> MMY	N	string	Maturity Month Year
30866	> CntrDate	N	Int32	Contract Date
	> Ct	N	enum	Contract Type '0' = Standard ('S') '1' = Flexible ('F')
201	> PutCall	N	enum	Put or Call '0' = Put '1' = Call
25034	> CntrGenNr	N	Int32	Contract Generation Number
1193	> SettlMeth	N	enum	Settlement method '0' = Cash ('C') '1' = Physical ('P')
2578	> OrigStrkPx	N	Decimal	Original Strike Price
1194	> ExerStyle	N	enum	Exercise Style '0' = European ('E') '1' = American ('A')
< Instrument > group ends				
< Data > group starts				
< Bid > group starts				
270	> Px	N	Decimal	Bid Price – MDEntryPrice
271	> Sz	N	Decimal	Bid Size – MDEntrySize
346	> NumOfOrds	N	Int32	Number Of Orders

< Bid > group ends				
< Offer > group starts				
270	> Px	N	Decimal	Offer Price – MDEntryPrice
271	> Sz	N	Decimal	Offer Size – MDEntrySize
346	> NumOfOrds	N	Int32	Number Of Orders
< Offer > group ends				
336	SesID	N	enum	Trading Session ID '1' = Day '3' = Morning '5' = Evening '6' = Afterhours '7' = Holiday
625	SesSub	N	enum	Trading Session Sub ID '1' = PreTrading '3' = Continuous '4' = Closing '5' = PostTrading '6' = ScheduledIntradayAuction '7' = Quiescent '8' = AnyAuction '103' = ContinuousAuctionIssuer '104' = ContinuousAuctionSpecialist
2447	FastMktInd	N	bool	Fast Market Indicator
326	TrdgStat	N	enum	Security Trading Status '2' = TradingHalt '200' = Closed '201' = Restricted '202' = Book '203' = Continuous '204' = OpeningAuction '205' = OpeningAuctionFreeze

				'206' = IntradayAuction '207' = IntradayAuctionFreeze '208' = CurcuitBreakerAuction '209' = CurcuitBreakerAuctionFreeze '210' = ClosingAuction '211' = ClosingAuctionFreeze '212' = IPOAuction '213' = IPOAuctionFreeze '214' = PreCall '215' = Call '216' = Freeze '217' = TradeAtClose
2705	MktCond	N	enum	Market Condition '0' = normal market '1' = stressed market
270	Px	N	Decimal	Price
271	Sz	N	Decimal	Size
828	TrdTyp	N	enum	Trade type
2449	NumOfBuyOrds	N	Int32	Number of Buy orders
2450	NumOfSellOrds	N	Int32	Number of Sell orders
1024	MDOriTyp	N	enum	Origin of the market data '0' = Book '1' = Off-Book
15	Ccy	N	string	Currency
278	MDID	N	string	Market data entry ID (match step ID)
880	MtchID	N	String	Trade Match ID
279	UpdtAct	N	enum	Update action type '0' = New '1' = Change '2' = Delete
277	TrdCond	N	enum	Trade condition

				'0' = 'U' // FIX Exchange Last '1' = 'R' // FIX Opening Price '2' = 'AJ' // FIX Official Close Price '3' = 'AW' // FIX Last Auction Price '4' = 'AX' // FIX High Price '5' = 'AY' // FIX Low Price '6' = 'BD' // FIX Previous Closing Price '7' = 'BB' // FIX Midpoint Price '8' = 'BC' // FIX Trading On Terms Of issue '9' = 'SA' // FIX Special Auction '10' = 'TC' // FIX Trade At Close '11' = 'k' // FIX Out of Sequence '12' = 'a' // FIX Volume Only
965	Status	N	enum	Instrument Status '1' = Active '2' = Inactive '10' = Published '11' = Pending Deletion
2705	MktCond	N	enum	Stressed Market Ind '0' = Normal '1' = Stressed '2' = Exceptional
423	PxTyp	N	int	Code to represent the price type '1' = percentage '2' = unit
202	StrkPx	N	decimal	Current strike price
	StlPx	N	decimal	Settlement price
	SettlCcy	N	string	Settlement currency
	Int	N	decimal	Open Interest
1020	TrdVol	N	decimal	Total volume
< Bids > sequence starts				

270	>> Px		Decimal	Bid price
271	>> Sz		Decimal	Bid size
346	>> NumOfOrds		Int32	Number of orders on bid side
< Bids > sequence ends				
< Offers > sequence starts				
270	>> Px		Decimal	Offer price
271	>> Sz		Decimal	Offer size
346	>> NumOfOrds		Int32	Number of orders on offer side
< Offers > sequence ends				
	Pap	N	Decimal	Potential Auction Price
	Opn	N	Decimal	Opening Price
	Cls	N	Decimal	Closing Price
273	Tm	Y	UInt64	MD Entry Time
< Data > group ends				

## 7 Appendix B – Tradegate description

Message type: Market Data

FIX Tag	FIX Field Name	Req'd	FAST Data Type	Description
35	MsgTyp	Y	enum	Message type 'X' = MarketDataIncrementalRefresh 'W' = MarketDataSnapshotFullRefresh
< ApplSeqCtrl > group starts				
1180	> ApplID	Y	UInt32	Application ID
1181	> ApplSeqNum	Y	UInt64	Application Sequence Number
< ApplSeqCtrl > group ends				
< Instrument > group starts				
1301	> MktID	Y	string	Market Identifier 'XGAT'
55	> Sym	Y	String	Symbol – instrument identification
167	> SecTyp	Y	enum	Security Type 0 = NO_SECURITYTYPE // None 1= FUT // Future 2 = OPT // Option 3 = MLEG // Multileg Instrument 4 = INDEX // Index 5 = ETC // Exchange traded commodity 6 = ETN // Exchange traded note 7 = CS // Common Stock 8 = REPO // Repurchase 9 = CASH // Repurchase 10 = FOR // Foreign Exchange Contract 11 = BOND = 11; // Bond 12 = MF // Mutual Fund

				13 = FUN // Investment Fund 14 = IRS // Interest Rate Swap 15 = SR // Subscription Rights 16 = WAR // Warrant 99 = OTHER // Other
< Instrument > group ends				
< Data > group starts				
< Quote.Bid > group starts				
270	> Px	N	Decimal	Bid Price – MDEntryPrice
271	> Sz	N	Decimal	Bid Size – MDEntrySize
1070	> MDQteTyp	N	Enum	Bid – MDQuoteType 0 = INDICATIVE (Taxe) 1 = TRADEABLE (Quote)
< Quote.Bid > group ends				
< Quote.Offer > group starts				
270	> Px	N	Decimal	Offer Price – MDEntryPrice
271	> Sz	N	Decimal	Offer Size – MDEntrySize
1070	> MDQteTyp	N	enum	Offer – MDQuoteType 0 = INDICATIVE (Taxe) 1 = TRADEABLE (Quote)
< Quote.Offer > group ends				
326	TrdgStat	N	enum	SecurityTradingStatus 0 = UNDEFINED 2 = TRADINGHALT 203 = CONTINUOUS 204 = OPENINGAUCTION 208 = CIRCUITBREAKERAUCTION
965	Status	N	enum	SecurityStatus 0 = UNDEFINED 1 = ACTIVE

				9 = SUSPENDED
273	Tm	Y	UInt64	MDEntryTime (Quotation Time)
270	Px	N	Decimal	Price (Last Trade)
271	Sz	N	Decimal	Size (Last Qty)
273	Tm	Y	UInt64	MD Entry Time (Last Trade Time)
423	PxTyp	N	enum	PriceType 0 = UNDEFINED 1 = PERCENTAGE 2 = PER_UNIT 9 = YIELD 22 = BASIS_POINT
828	TrdTyp	N	enum	Trade type 1107 = IPOAUCTIONTRADE
15	Ccy	N	String	Currency
6	AvgPx	N	Decimal	AvgPx (Average Price)
1020	TrdVol	N	Decimal	TradeVolume
	Ttt	N	Decimal	TotalTurnover
2490	TrdNum	N	Int32	TradeNumber
278	MDID	N	String	Market data entry ID
880	MtchID	N	String	Trade Match ID
279	UpdtAct	N	enum	Update action type 0 = NEW 1 = CHANGE 2 = DELETE
277	TrdCond	N	enum	Trade condition 0 = U // FIX Exchange Last 1 = R // FIX Opening Price 4 = AX // FIX High Price 5 = AY // FIX Low Price



270	MDEntryPx	N	Decimal	ClosePx (Tradegate Close price)
140	PrevClsPx	N	Decimal	PrevClosePx
344	ClsTim	N	UInt64	Official Close Timestamp
343	PreClsTim	N	UInt64	Previous Official Close Timestamp
15	Ccy	N	String	Currency
< Data > group ends				

---

## 8 Appendix C – KAIKO description

For KAIKO the stream specification in the subscribe and unsubscribe has to be detailed in the following way:

Stream / Topic name	Description
crypto-<exchange>.<feed>.<level>.<product>	<p>Based on the type of the feed, e.g. incremental vs. snapshot, by using the &lt;feed&gt; type, customer can distinguish among different content and can subscribe only to the incremental data and skip snapshot messages.  <b>feed</b> := 'inc'   'snp'</p> <p>Level 1 and level 2 data will distinguish between trade and top of book resp. market depth messages. E.g. <b>crypto-cbse.inc.lev2.*</b> for full orderbook market depth.  <b>level</b> := 'lev1'   'lev2'</p> <p>Each exchange stream will be 'split' into several products based on the definitions top5-top15. E.g. <b>crypto-cbse.inc.lev1.premium</b> [or crypto-cbse.inc.lev1.top5]</p> <p><b>product</b> := 'top5'   'top10'   'top15'   'top20'   'others'</p>
crypto-reference	(common) Reference data stream for all exchanges and assets.

### 8.1 General reference data processing rules

A snapshot cycle consists of (see figure x):

- A market data report message (Evt = 1 = "StartOfReferenceData").
- A sequence of a data messages of type exchange, asset, pool and instrument
- Finally, market data report message (Evt = 2 = "EndOfReferenceData").

Each data message has its own unique message sequence number

Start of reference data	Exchanges			Assets			Pools			Instruments			End of reference data				
MarketDataReport Evt = 1; Cnt = 761	E <sub>1</sub> <sup>1</sup>	E <sub>2</sub> <sup>2</sup>	...	E <sub>n</sub> <sup>89</sup>	A <sub>1</sub> <sup>90</sup>	A <sub>2</sub> <sup>91</sup>	...	A <sub>k</sub> <sup>251</sup>	P <sub>1</sub> <sup>252</sup>	P <sub>2</sub> <sup>253</sup>	...	P <sub>j</sub> <sup>321</sup>	I <sub>1</sub> <sup>322</sup>	I <sub>2</sub> <sup>323</sup>	...	I <sub>m</sub> <sup>761</sup>	MarketDataReport Evt = 2; Cnt = 761

**Figure 1** Entire snapshot cycle of the KAIKO reference data stream

with:

E<sub>x</sub>: Exchange x

A<sub>x</sub>: Asset x

P<sub>x</sub>: Pool x and

I<sub>x</sub>: Instrument x

This reference data snapshot cycle will be repeated every 15 minutes and repeated cycles may contain more (in case of newly added) or less (in case of expired) instruments. For this reason, customer's application should start receiving the reference data stream e.g. 15 minutes in the past, wait for the 'StartOfReferenceData' message which will also contain the total number of messages as part of the reference data cycle; then receive all reference data information and finish the process by receiving the 'EndOfReferenceData' information.

## 8.2 Message structure description

The tables below contain detailed description of each message type of the KAIKO feed:

- Reference data messages: MarketDataReport, Asset, Exchange, Instrument, Pool
- Market data messages: TopOfBookUpdate, Trade, MarketDataUpdate, MarketDataSnapshot

### 8.2.1 Message type: Market Data Report

FIX Tag	FIX Field Name	Req'd	FAST Data Type	Description
	Evt	Y	Enum	Event 0 = UNKNOWN_EVENT 1 = START_OF_REFERENCE_DATA 2 = END_OF_REFERENCE_DATA
	Cnt	Y	Int32	total message count of the current reference data cycle

### 8.2.2 Message type: Asset

FIX Tag	FIX Field Name	Req'd	FAST Data Type	Description
	> Seq	Y	Int32	Seq
	> Code	Y	String	AssetCode

	> Name	Y	String	Name
1938	> AssetClass	N	enum	AssetClass 0 = UNKNOWN_AC 101 = CRYPTOCURRENCY 102 = FIAT

### 8.2.3 Message type: Exchange

FIX Tag	FIX Field Name	Req'd	FAST Data Type	Description
	> Seq	Y	Int32	Seq
	> ExCode	Y	String	AssetCode
	> Name	Y	String	Name
	> ExClass	N	enum	ExchangeClass 0 = UNDEFINED 1 = CENTRAL 2 = DECENTRAL

### 8.2.4 Message type: Instrument

FIX Tag	FIX Field Name	Req'd	FAST Data Type	Description
	> Seq	Y	Int32	Seq
< InstrumentID > group starts				
	>> InstCode	Y	UInt32	Seq
	>> ExCode	Y	String	AssetCode
	>> Class	N	enum	InstrumentClass 0 = UNKNOWN 1 = SPOT 2 = FUTURE 3 = PERPETUAL_FUTURE 4 = OPTION

				5 = OPTION_COMBO 6 = FUTURE_COMBO
< InstrumentID > group ends				
	> BaseAsset	Y	String	BaseAsset (e.g. 'ada')
	> QuoteAsset	Y	String	QuoteAsset (e.g. 'btc')
	> ExchPairCod	N	enum	ExchangePairCode (e.g. 'adabtc')
	> StrTm	N	UInt64	MD Entry Time
	> EndTm	N	UInt64	MD Entry Time
	>> Pools	N	String	List of underlying pools
	> Expiry	N	UInt64	Expiry
	> StrkPx	N	DoubleValue	StrikePrice
	> PutCall	N	Enum	PutOrCall 0 = PUT 1 = CALL
	> Mult	N	DoubleValue	ContractMultiplier
	> Stream	N	String	DBAG's crypto product classification 'top5', 'top10' etc.

### 8.2.5 Message type: Pool

FIX Tag	FIX Field Name	Req'd	FAST Data Type	Description
	> Seq	Y	UInt32	Seq
	> Address	Y	String	Adress (e.g. 0x0002e63328169d7f6ea121f1e32e4f620abf0352)
	> Name	Y	String	Name (e.g. 'wNXM-ICHI-0.003')
	> Protocol	N	String	Protocol (e.g. 'usp3')
	> Type	N	String	Type (e.g. 'liquidity_pool')
	> Fee	N	String	Fee (e.g. '0.003')

	>> Tokens	N	String	Tokens (e.g. 'wNXM', 'ICHI')
	>> UndrTokens	N	String	UndrTokens
	> TickSpacing	N	String	TickSpacing (e.g. '60')
	> Weights	N	String	Weights

### 8.2.6 Message type: Trade

FIX Tag	FIX Field Name	Req'd	FAST Data Type	Description
	SeqID	Y	String	SeqID (e.g. 'ccaunvs5c7oora13vhlq')
< InstrumentID > group starts				
	> InstCode	Y	UInt32	Seq
	> ExCode	Y	String	AssetCode
	> Clss	N	enum	InstrumentClass 0 = UNKNOWN 1 = SPOT 2 = FUTURE 3 = PERPETUAL_FUTURE 4 = OPTION 5 = OPTION_COMBO 6 = FUTURE_COMBO
< InstrumentID > group ends				
279	UpdtAct	Y	Int32	MUpdateAction 0 = NEW 1 = CHANGE 2 = DELETE
278	MDID	Y	Int32	MDEntryID (e.g. '401511317')
270	Px	Y	Double	Price
271	Sz	Y	Double	Size
2446	AgrsrSide	Y	Int32	AggressorSideValue

				0 = NO_AGGRESSOR 1 = BUY 2 = SELL
273	Tm	Y	UInt64	MD Entry Time

### 8.2.7 Message type: TopOfBookUpdate

FIX Tag	FIX Field Name	Req'd	FAST Data Type	Description
	SeqID	Y	String	SeqID (e.g. 'ccaunvs5c7oora13vhlq')
< InstrumentID > group starts				
	> InstCode	Y	UInt32	Seq
	> ExCode	Y	String	AssetCode
	> Clss	N	enum	InstrumentClass 0 = UNKNOWN 1 = SPOT 2 = FUTURE 3 = PERPETUAL_FUTURE 4 = OPTION 5 = OPTION_COMBO 6 = FUTURE_COMBO
< InstrumentID > group ends				
269	Typ	N	Int32	MDEntryType 0 = BID 1 = OFFER
279	UpdtAct	Y	Int32	MDUpdateAction 0 = NEW 1 = CHANGE 2 = DELETE
270	Px	N	Decimal	Bid or Offer Price – MDEntryPrice
271	Sz	N	Decimal	Bid or Offer Size – MDEntrySize

273	Tm	Y	UInt64	MD Entry Time
-----	----	---	--------	---------------

### 8.2.8 Message type: MarketDataUpdate

MarketDataUpdate is currently not yet activated.

FIX Tag	FIX Field Name	Req'd	FAST Data Type	Description
	SeqID	Y	String	SeqID (e.g. 'ccaunvs5c7oora13vhlg')
< InstrumentID > group starts				
	> InstCode	Y	UInt32	Seq
	> ExCode	Y	String	AssetCode
	> Clss	N	enum	InstrumentClass 0 = UNKNOWN 1 = SPOT 2 = FUTURE 3 = PERPETUAL_FUTURE 4 = OPTION 5 = OPTION_COMBO 6 = FUTURE_COMBO
< InstrumentID > group ends				
269	Typ	N	Int32	MDEntryType 0 = BID 1 = OFFER
279	UpdtAct	Y	Int32	MDUpdateAction 0 = NEW 1 = CHANGE 2 = DELETE
270	Px	N	Decimal	Bid or Offer Price – MDEntryPrice
271	Sz	N	Decimal	Bid or Offer Size – MDEntrySize
273	Tm	Y	UInt64	MD Entry Time



## 8.2.9 Message type: MarketDataSnapshot

MarketDataSnapshot is currently not yet activated.

FIX Tag	FIX Field Name	Req'd	FAST Data Type	Description
< InstrumentID > group starts				
	> InstCode	Y	UInt32	Seq
	> ExCode	Y	String	AssetCode
	> Clss	N	enum	InstrumentClass 0 = UNKNOWN 1 = SPOT 2 = FUTURE 3 = PERPETUAL_FUTURE 4 = OPTION 5 = OPTION_COMBO 6 = FUTURE_COMBO
< InstrumentID > group ends				
	LstSeqID	Y	String	SeqID (e.g. 'ccaunvs5c7oora13vhlg')
Repeating group < Bids > starts				
270	> Px	N	Decimal	Bid Price
271	> Sz	N	Decimal	Bid Size
Repeating group < Bids > ends				
Repeating group < Offers > starts				
270	> Px	N	Decimal	Offer Price
271	> Sz	N	Decimal	Offer Size
Repeating group < Offers > ends				
273	Tm	Y	UInt64	MD Entry Time

## 9 Change log

No	Chapter, page	Date	Change
1.0	General	January 13, 2023	Initial version for publication