

# Cloud Stream User Manual

Version: 3.3

Valid as of: 01-Jul-2024

Last update: 03-Jun-2024

## Table of contents

<b>1</b>	<b>INTRODUCTION.....</b>	<b>3</b>
<b>2</b>	<b>AUTHENTICATION AND AUTHORIZATION.....</b>	<b>4</b>
<b>3</b>	<b>MESSAGING – FEED SUBSCRIPTION.....</b>	<b>5</b>
<b>4</b>	<b>SERVICE AVAILABILITY .....</b>	<b>7</b>
<b>5</b>	<b>DATA AND SERVICE MESSAGES .....</b>	<b>8</b>
5.1	GOOGLE PROTOCOL BUFFERS .....	8
5.2	JSON FORMAT.....	8
<b>6</b>	<b>APPENDIX A – EUREX MICRO DERIVATIVES AND EUREX CRYPTOCURRENCY DERIVATIVES FEED .....</b>	<b>9</b>
<b>7</b>	<b>APPENDIX B – TRADEGATE FEED .....</b>	<b>15</b>
<b>8</b>	<b>APPENDIX C – KAIKO FEED.....</b>	<b>19</b>
8.1	GENERAL REFERENCE DATA PROCESSING RULES.....	20
8.2	MESSAGE STRUCTURE DESCRIPTION.....	20
8.2.1	Message type: Market Data Report.....	21
8.2.2	Message type: Asset.....	21
8.2.3	Message type: Exchange .....	21
8.2.4	Message type: Instrument .....	22
8.2.5	Message type: Pool .....	23
8.2.6	Message type: Trade .....	23
8.2.7	Message type: TopOfBookUpdate .....	24
8.2.8	Message type: MarketDataUpdate .....	25
8.2.9	Message type: MarketDataSnapshot .....	26
8.2.10	Message type: Rate .....	27
<b>9</b>	<b>APPENDIX D – XETRA AND DBDX SPOT FEED .....</b>	<b>28</b>
<b>10</b>	<b>CHANGE LOG .....</b>	<b>37</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 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 001.000.006.

---

## 2 Authentication and Authorization

The streaming WebSocket API is secured by using standard Bearer authorization 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|proto>
```

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

```
Authorization: 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

<https://github.com/Deutsche-Boerse/Cloud.Stream.Client/tree/main/proto/src/client.proto>

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. "md-microproducts" "md-tradegate" (see Appendices)	e.g. { "stream": "md-microproducts", "startTime": "1659008334123456789" } startTime is in nanoseconds since epoch, TZ=UTC or { "stream": "md-microproducts", "startSeq": 12345 } startSeq is an integer to be used to start at the first message having the sequence number or the next one available

#### 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. "md-microproducts" "md-tradegate" (see Appendices)	e.g. "stream": ["md-microproducts"]

### Example

```
{"event": "subscribe", "requestId": 123456789, "subscribe": {"stream": [{"stream": "md-tradegate"}]}}  
{"event": "unsubscribe", "requestId": 123456789, "unsubscribe": {"stream": ["md-tradegate"]}}
```

The Request to Subscribe or Unsubscribe will be replied with Response containing a Status field that will inform about the result.

The count and order of Response messages is the same as the one in repeated streams field of the Request.

Client is disconnected in case the Request is malformed.

### Example for Status=OK

```
{"subs": "md-tradegate", "messages": [{"@type": "type.googleapis.com/Client.Response", "requestId": "123456789"}]}
```

#### **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.

<b>Service</b>	<b>Telephone</b>	<b>Email</b>
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 messages which will be sent to the customer based on his subscription were defined in a generic mode so that any kind of messages can be transported by the same technical interface. The StreamMessage 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.

### 5.1 Google Protocol Buffers

The usage of Google Protocol Buffers is forcing some standard. Please be recommended to read <https://protobuf.dev/programming-guides/proto3/#default> for the usage of default values. E.g. values for enum fields are not sent, if they are default values ('0'=).

### 5.2 JSON Format

The same is valid for using JSON format. Please be recommended to read <https://protobuf.dev/programming-guides/proto3/#json> for the usage of JSON format.



## 6 Appendix A – Eurex Micro Derivatives and Eurex Cryptocurrency Derivatives feed

[https://github.com/Deutsche-Boerse/Cloud.Stream.Client/tree/main/proto/src/md\\_cef.proto](https://github.com/Deutsche-Boerse/Cloud.Stream.Client/tree/main/proto/src/md_cef.proto)

Stream Name for **Eurex Micro Derivatives** is “md-microproducts”.

Stream Name for **Eurex Cryptocurrency Derivatives** is “md-cryptoproducts”.

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 Security Type '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' = Scheduled Intraday Auction '7' = Quiescent '8' = AnyAuction '103' = Continuous Auction Issuer '104' = Continuous Auction Specialist
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	enum	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 feed

[https://github.com/Deutsche-Boerse/Cloud.Stream.Client/tree/main/proto/src/md\\_cef.proto](https://github.com/Deutsche-Boerse/Cloud.Stream.Client/tree/main/proto/src/md_cef.proto)

Stream Name for **Tradegate** is “md-tradegate”.

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 Security Type // 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 // Cash 10 = FOR // Foreign Exchange Contract 11 = BOND // Bond 12 = MF // Mutual Fund

				13 = FUN // Investment Fund 14 = IRS // Interest Rate Swap 15 = SR // Subscription Right 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
270	Px	N	decimal	Price (Last Trade)
271	Sz	N	decimal	Size (Last Qty)
273	Tm	N	uint64	MD Entry Time - Trade message: Last Trade Time - Quote message: Quotation 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 feed

[https://github.com/Deutsche-Boerse/Cloud.Stream.Client/tree/main/proto/src/md\\_crypto.proto](https://github.com/Deutsche-Boerse/Cloud.Stream.Client/tree/main/proto/src/md_crypto.proto)

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

Stream / Topic name	Description
<b>crypto-&lt;exchange&gt;.&lt;feed&gt;.&lt;level&gt;.&lt;product&gt;</b>	<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>
<b>crypto-reference</b>	(common) Reference data stream for all exchanges and assets.

Designed to bring greater transparency to pricing in the crypto currency markets, the Benchmark Rates are aggregated price feeds across platforms based on the vast majority of executed trades on centralized and/or decentralized exchanges. Reference rates and price rates distributed by MD+S will be included as separate product into the existing CEF Cloud Streaming platform.

The following six products should be offered without elements that they are composed of:

Stream / Topic name	Description
<b>crypto_bench.reference.btc</b>	Reference rates for Bitcoin (BTC)
<b>crypto_bench.reference.eth</b>	Reference rates for Ethereum (ETH)
<b>crypto_bench.reference.other</b>	Reference rates for all assets, except BTC and ETH
<b>crypto_bench.price.btc</b>	Price rates for BTC
<b>crypto_bench.price.eth</b>	Price rates for ETH
<b>crypto_bench.price.other</b>	Price rates for all assets, except BTC and ETH

Stream / Topic name	Description
	crypto_bench.*.* should be able to receive all data using a single subscription request.

## 8.1 General reference data processing rules

A snapshot cycle consists of (see Figure 1 ):

- 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	> AssetCls	N	enum	AssetClass 0 = Unknown Asset Class 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
	> ExCls	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
	>> Clss	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')
	> StrtTm	N	uint64	MD Entry Time
	> EndTm	N	uint64	MD Entry Time
	>> Pools	N	string	List of underlying pools
	> Expiry	N	uint64	Expiry
	> StrkPx	N	double	StrikePrice
	> PutCall	N	enum	PutOrCall 0 = PUT 1 = CALL
	> Mult	N	double	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. 0x0002e63328169d7feea121f1e32e4f620abf0352)
	> 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	enum	MDUpdateAction 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	enum	MDEntryType 0 = BID 1 = OFFER
279	UpdtAct	Y	enum	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	enum	MDEntryType 0 = BID 1 = OFFER
279	UpdtAct	Y	enum	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

### 8.2.10 Message type: Rate

FIX Tag	FIX Field Name	Req'd	FAST Data Type	Description
	Ind	N	enum	IndexTyp 0 = PRICE_RATE 1 = REFERENCE_RATE
	IndexCode	N	string	IndexCode
	BaseAsset	N	string	BaseAsset
	QuoteAsset	N	string	QuoteAsset
	Cty	N	enum	Commodity 0 = REAL_TIME 1 = FIXING
	StrtTm	N	uint64	StrtTm
	EndTm	N	uint64	EndTm
	Px	N	double	Px
273	Tm	N	uint64	MDEntryTime

## 9 Appendix D – Xetra and DBDX Spot feed

[https://github.com/Deutsche-Boerse/Cloud.Stream.Client/tree/main/proto/src/md\\_cef.proto](https://github.com/Deutsche-Boerse/Cloud.Stream.Client/tree/main/proto/src/md_cef.proto)

Stream Name for **Xetra ETF ETP** is “md-xetraetfep”.

Stream Name for **DBDX Crypto Spot** is “md-dbdx”.

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 Security Type '5' = Exchange traded commodity '6' = Exchange traded note '17' = Exchange traded fund
15	> Ccy	N	string	Currency
455	> AltID	N	string	SecurityAltID

456	> AltIDSrc	N	enum	SecurityAltIDSource '4' = ISIN
7703	> MktSeg	N	string	MarketSegment
969	> MinPxIncr	N	decimal	MinPriceIncrement
980	> UpdActn	N	enum	SecurityUpdateAction '0' = NEW '1' = DELETE '2' = MODIFY
779	> LastUpdateTm	N	uint64	LastUpdateTime
< Event > sequence starts				
865	>> EventType	N	Enum	EventType '0' = UNDEFINED '6' = INACTIVATION '7' = LAST_ELIGIBLE_TRADE_DATE '28' = FIRST_ELIGIBLE_TRADE_DATE
866	>> Dt	N	uint32	EventDate
< Event > sequence ends				
< 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 Side
1070	> MDQteTyp	N	enum	MDQuoteType Bid Side '0' = INDICATIVE '1' = TRADEABLE
269	> Typ	N	enum	MDEntryType '0' = BID '11' = MARKET_BID

< 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 Side
1070	> MDQteTyp	N	enum	MDQuoteType Offer Side '0' = INDICATIVE '1' = TRADEABLE
269	> Typ	N	enum	MDEntryType '1' = OFFER '12' = MARKET_OFFER
< Offer > group ends				
423	PxTyp	N	enum	Code to represent the price type '0' = UNDEFINED '1' = PERCENTAGE '2' = PER_UNIT '9' = YIELD '22' = BASIS_POINT
965	Status	N	enum	SecurityStatus '0' = UNDEFINED '1' = ACTIVE '2' = INACTIVE '4' = EXPIRED '5' = DELISTED '6' = KNOCKED_OUT '9' = SUSPENDED '10' = PUBLISHED '11' = PENDING_DELETION '12' = KNOCKED_OUT_AND_SUSPENDED
336	SesID	N	enum	TradingSessionID '0' = UNDEFINED

				'1' = DAY '2' = HALFDAY '3' = MORNING '4' = AFTERNOON '5' = EVENING '6' = AFTERHOURS '7' = HOLIDAY
625	SesSub	N	enum	TradingSessionSubID '0' = UNDEFINED '1' = PRETRADING '3' = CONTINUOUS '4' = CLOSING '5' = POSTTRADING '6' = SCHEDULEDINTRADAYAUCTION '7' = QUIESCENT '8' = ANYAUCTION '103' = CONTINUOUSAUCTIONISSUER '104' = CONTINUOUSAUCTIONSPECIALIST
2447	FastMktInd	N	bool	FastMarketIndicator 0 = FALSE 1 = TRUE
326	TrdgStat	N	enum	SecurityTradingStatus '0' = UNDEFINED '1' = OPENING_DELAY '2' = TRADINGHALT '3' = RESUME '19' = NOT_TRADED_ON_THIS_MARKET '23' = FAST_MARKET '200' = CLOSED '201' = RESTRICTED '202' = BOOK '203' = CONTINUOUS '204' = OPENINGAUCTION '205' = OPENINGAUCTIONFREEZE

				<p>'206' = INTRADAYAUCTION</p> <p>'207' = INTRADAYAUCTIONFREEZE</p> <p>'208' = CIRCUITBREAKERAUCTION</p> <p>'209' = CIRCUITBREAKERAUCTIONFREEZE</p> <p>'210' = CLOSINGAUCTION</p> <p>'211' = CLOSINGAUCTIONFREEZE</p> <p>'212' = IPOAUCTION</p> <p>'213' = IPOAUCTIONFREEZE</p> <p>'214' = PRECALL</p> <p>'215' = CALL</p> <p>'216' = FREEZE</p> <p>'217' = TRADEATCLOSE</p>
2705	MktCond	N	enum	<p>Market Condition</p> <p>'0' = NORMAL</p> <p>'1' = STRESSED</p> <p>'2' = EXCEPTIONAL</p>
25045	TesStatus	N	enum	<p>SecurityStatus</p> <p>'0' = UNDEFINED</p> <p>'1' = ACTIVE</p> <p>'2' = INACTIVE</p> <p>'4' = EXPIRED</p> <p>'5' = DELISTED</p> <p>'6' = KNOCKED_OUT</p> <p>'9' = SUSPENDED</p> <p>'10' = PUBLISHED</p> <p>'11' = PENDING_DELETION</p> <p>'12' = KNOCKED_OUT_AND_ SUSPENDED</p>
2542	MktSegStat	N	enum	<p>MarketSegmentStatus</p> <p>'0' = UNDEFINED</p> <p>'1' = ACTIVE</p> <p>'2' = INACTIVE</p> <p>'3' = PUBLISHED</p>
270	Px	N	decimal	Price



271	Sz	N	decimal	Size
828	TrdTyp	N	enum	TradeType '0' = REGULARTRADE '1' = BLOCKTRADE '2' = EFP '12' = EXCHANGEFORSWAP '50' = PORTFOLIOCOMPRESSIONTRADE '54' = OTC '55' = EXCHANGEBASISFACILITY '1000' = VOLATRADE '1001' = EFPFINTRADE '1002' = EFPINDEXFUTURESTRIDE '1004' = BLOCKTRADEATMARKET '1006' = XETRAEUREXENLIGHTTRIGGEREDTRADE '1007' = BLOCKQTPIPTRIDE '1100' = OPENINGAUCTIONTRADE '1101' = INTRADAYAUCIONTRADE '1102' = VOLATILITYAUCTIONTRADE '1103' = CLOSINGAUCTIONTRADE '1104' = CROSSAUCTIONTRADE '1107' = IPOAUCTIONTRADE '1108' = LIQUIDITYIMPROVEMENTCROSS
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' = MDOT_BOOK '1' = MDOT_OFF_BOOK
15	Ccy	N	string	Currency
278	MDID	N	string	Market data entry ID (match step ID)
880	MtchID	N	string	TradeMatchID
279	UpdtAct	N	enum	Update action type '0' = NEW

				'1' = DELETE '2' = MODIFY
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
< Bids > sequence starts				
270	>> Px		decimal	Bid price
271	>> Sz		decimal	Bid size
346	>> NumOfOrds		int32	Number of orders on bid side
1070	>> MDQteTyp	N	enum	MDQuoteType Bid Side '0' = INDICATIVE '1' = TRADEABLE
269	>> Typ	N	enum	MDEntryType '1' = OFFER '12' = MARKET_OFFER
< 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

1070	>> MDQteTyp	N	enum	MDQuoteType Offer Side '0' = INDICATIVE '1' = TRADEABLE
269	>> Typ	N	enum	MDEntryType '1' = OFFER '12' = MARKET_OFFER
< Offers > sequence ends				
	OpenPx	N	decimal	Opening Price
	ClosePx	N	decimal	Closing Price
332	HighPx	N	decimal	High Price
333	LowPx	N	decimal	Low Price
	AvgPx	N	decimal	Average Price
1020	TrdVol	N	decimal	TotalVolume
	Ttt	N	decimal	TotalTurnover
2490	TrdNum	N	int32	TradeNumber
2490	TrdNumTes	N	int32	TradeNumberTes
344	ClsTim	N	uint64	CloseTime
	RefPx	N	decimal	Reference Price
234	Val	N	string	StipulationValue - 'CD' (Cum Dividend) - 'XD' (Ex Dividend)
< Pxs > sequence starts				
269	MDEntryType	N	enum	MDEntryType '4' = Opening Price '6' = Close Price '7' = High Price '8' = Low Price '9' = Average price 'M' = Previous Close Price

270	MDEntryPx	N	decimal	MD EntryPrice
271	MDEntrySz	N	decimal	MD Entry Size
273	MDEntryTime	N	uint64	MD Entry Time
< Pxs > sequence ends				
273	Tm	Y	uint64	MD Entry Time
< Data > group ends				

---

## 10 Change log

No	Chapter, page	Date	Change
1.0	General	November 25, 2022	Initial version for publication
1.1	Appendix A 2 Authentication and Authorization, 3 Feed Subscription	April 12, 2023	Adjustments for micro options, adjustments for Authorization header field, and example for Feed Subscription
1.2	2 Authentication and Authorization	April 13, 2023	Adjustments of values of the format parameter (json   proto)
2.0	Appendix D	June 21, 2023	Added Appendix D for Xetra ETF & ETP
3.0	Appendix B, 8.2.10 Chapter 3 Appendix A	August 28, 2023	Added new Kaiko message type Rate, new streams Extended stream field content for Subscribe message Added Cryptoproducts
3.1	Page 5, Appendix A, B, D	October 09, 2023	Adjustment of stream names and other minor corrections
3.2	Page 3 Page 9, Appendix A	March 04, 2024	Interface version number and connectivity availability option is updated. Changed product names.
3.3	Page 28, Appendix D	June 03, 2024	Appendix D adjustment for DBDX Crypto Spot