Xetra Release 17.0

Xetra Market Data Interface

Final Version

p

Table of Contents

1 Introduction

Xetra Market Data Interface provides netted market data in FIX 5.0 SP2 (and approved Extension Packs) over FAST 1.2 format over multicast.

The interface provides easier access to Xetra and will enable customers to replace their current VALUES API infrastructure.

1.1 Purpose and Readership

The purpose of this document is to provide a functional and technical description of the Xetra Market Data Interface as a basis for software development.

This document is intended for system designers and programmers who wish to develop/adapt their client application to interact with the services offered by the Xetra Market Data Interface.

1.2 Interface Landscape

The Xetra Market Data Interface disseminates price level aggregated, netted depth data in multicast format and will become the standard Market Data offering for Xetra in the future. Market data is provided in FIX 5.0 SP2 (and approved Extension Packs) over FAST 1.2 format. Like the Enhanced Broadcast Solution it is a low footprint architecture where customers have a free choice of the platform and the operating systems. The main differences are:

Features	Xetra Enhanced Broadcast Solution	Xetra Market Data Interface
General Description	High Performance, Low Latency, Real Time Market Data	Standard FIX Market Data Interface for netted Market Data
Network Protocol	UDP Multicast	UDP Multicast
Application Layer (Semantics)	Proprietary	FIX 5.0 SP2
Presentation Layer (Syntax)	FAST 1.1	FAST 1.2
Bandwidth Requirements	Dedicated Leased Line connection with a bandwidth of minimum 10 Mbit/s	Approximately 250 kBit/s consumption Provided via Multi-Interface Channel or via Enhanced Broadcast Solution connection
Market Depth	Up to 20	Continuous Trading: 5 Continuous Auction: 1 or 5
Reference Data	Yes	Not included[1].
State Changes	Yes	Yes
All Trade Prices (ATP)	Provided via a dedicated feed per broadcast group.	Provided per backend via Ticker feed.
Snapshots	Yes	

[1] Note: The Xetra Market Data interface will not provide any instrument reference data. Clients are asked to retrieve instrument reference data via the Enhanced Broadcast Solution interface or via the Xetra Instrument Reference Data file provided in the Members Only section on Xetra website.

Features	Xetra Enhanced Broadcast Solution	Xetra Market Data Interface
Deltas/Incrementals	Yes	Common Snapshot and Incremental/Delta Feed, netted, with last trade prices; last trade prices are disseminated in the Snapshot and Incremental/Delta feed. Snapshot and Incremental/Delta Feed is netted, i.e. delivered on the same feed and with regularly repeated Snapshot messages, on which all respectively following Incremental/Delta messages have to refer.
Ticker Feed	Not included.	Yes

1.3 Document Overview

General Introduction

- Chapter 2 "Infrastructure Requirements" highlights the basic infrastructure requirements for accessing the services offered by the Xetra Market Data Interface.
- Chapter 3 "Overview" provides an overview of the Xetra Market Data Interface explaining some commonly used terms.

Xetra Market Data Interface Structure

- Chapter 4 "Data Feeds" introduces the Xetra Market Data Interface feeds and describes the interface data dissemination schedule during a Xetra business day.

Message Structures

- Chapter 5 "Structure of Messages" explains the Xetra Market Data Interface message structures in detail and describes the classification of the disseminated messages.

Appendices provide additional information on the Xetra Market Data Interface.

- Chapter 6 "Appendix - Glossary of Terms" provides an alphabetical listing of some commonly used terms.
- Chapter 7 "Appendix - Data Field Dictionary" contains the explanation of data fields, their formats and allowed value ranges.
- Chapter 8 "Appendix - Mapping of VALUES to FIX standard" provides a table how to map VALUES information format to FIX information format.
- Chapter 9 "Appendix - Interface Limits" provides a list of system limits for the Xetra Market Data Interface.
- Chapter 10 "Appendix - How to Use" covers the description of the operation of an example client system using the Xetra Market Data Interface.
- Chapter 11 "Appendix - Message Encoding" describes the formats used for the encoding of data fields.
- Chapter 12 "Appendix - Xetra Member Homepage" describes the Xetra homepage where you can find information on the Xetra Market Data Interface.

1.4 Change Log

Content changes in comparison to Xetra Release 16.0 Final Version of this document. It is recommended not to rely only on this change log, but to read in full the new version of the document.

Starting from January 2, 2018 onwards, all time stamps will be formatted in UTC format.

2 Infrastructure Requirements

2.1 Hardware and Network Infrastructure

The Xetra Market Data Interface disseminates market information over a multicast network set up by Xetra. A router capable of handling IP multicast is required as the communication equipment for accessing the Xetra Market Data Interface. The multicast management protocol is IGMPv2. Utilizing IGMPv3 it has to be considered that the IGMPv2 compatibility mode is enabled.

Members subscribing to the information can contact the Xetra Customer Technical Support CTS to obtain further details on configuring their network equipment.

2.2 Software Infrastructure

Members need to have a standard FAST template based decoder in order to be able to use the Xetra Market Data Interface. Alternatively members may implement the FAST decoder by themselves.

Xetra will not provide any client software for accessing the services offered. Member systems for Xetra Market Data Interface can be based on any platform capable of receiving multicasts. Members will have the opportunity to develop and use specific third party applications that fit their requirements.

3 Overview

The Xetra Market Data Interface is a flexible, transparent, UDP based interface which disseminates market information to members over a multicast network. The information disseminated by the Xetra Market Data Interface includes in different feeds Order Book data, Ticker data, Cross Requests, Market Resets and State Changes.

The messaging protocol used by the Xetra Market Data Interface is fully compliant to the Financial Information eXchange (FIX) protocol version 5.0 SP2 and will have message formats customized to fit the Xetra business requirements. The Xetra Market Data Interface will conform to the FAST (FIX Adapted for Streaming) protocol version 1.2 principles for efficient bandwidth utilization.

The interface provides members with the information in form of broadcast feeds. From these broadcast feeds, members will be able to receive the information that meets their requirements. The interface is designed so that members will only receive information from those feeds they have joined.

The Xetra Market Data Interface will be supported in parallel to the Enhanced Broadcast Solution and the public market data dissemination over VALUES.

Due to the common technical architecture, customers of the Xetra Enhanced Broadcast Solution interface will automatically have access to the broadcast feeds and multicast addresses of Xetra Market Data interface.

3.1 Definition of Commonly Used Terms

In the following some further definitions of terms frequently used in the context of the Xetra Market Data Interface are given.

- Member/Exchange Participant. The term "Member" refers to an organization that is authorized by Deutsche Börse AG, Eurex Bonds or a Partner Exchange (e.g. Wiener Börse AG or Irish Stock Exchange) to receive the services offered through the Xetra Market Data Interface.
- Securities and instruments. "Security" and "Instrument" refer to an individual tradable component in Xetra.
- Broadcast. Broadcasts contain information about events affecting the market. The information comprised in these broadcasts is limited to public information.
- Broadcast groups. Certain functional instrument groups are aggregated to a single broadcast group. The instruments of this instrument group are sent over the same multicast addresses.
- Feeds. One or more tradable instruments on Xetra may be grouped together to form a logical group. Data pertaining to all instruments of such a group will be disseminated over one set of multicast addresses and constitute a feed. Xetra Market Data Interface offers three kinds of feeds, the maintenance feed “0” (Fast Market, Market Reset, Cross Request, State Changes, Knock-Out, Instrument Suspension), the Ticker feed “1” and the feeds for market data (snapshot, delta/Incremental; both on the same feed).
- Delta or Incremental Broadcasts. They are provided for an order book affecting event (such as order entry or a trade).

- Snapshot broadcasts. A snapshot broadcast contains all current market information for a given instrument, regardless of recent movements and will be sent regularly on the same feed in addition to the "real time" delta broadcasts.

  Members have to receive and interpret snapshots all the time since no delta message is sent if a snapshot is sent at the time a delta could be sent. The frequency of dissemination is calculated from the network load condition in real time. The expected member application behaviour in detail is explained in chapter 11, Appendix – How to Use.
- State Change broadcast. These broadcasts inform the receiver of any exchange or system or instrument state change.
- Joining. Joining a feed means electronically connecting to the information source. Once joined to a source, all information that is disseminated by the source will be received involuntarily and will continue until the feed is quitted.
- Financial Information eXchange (FIX) protocol. The FIX Protocol is a messaging standard developed specifically for the real-time electronic exchange of securities transactions. FIX is a public domain specification owned and maintained by FIX Protocol Ltd.
- FIX Adapted for Streaming^{SM} (FAST Protocol ^{SM}). FAST is a standard developed by Data Representation and Transport Subgroup of FPL's Market Data Optimization Working Group. FAST uses proven data compression techniques that leverages knowledge about data content and data formats.
- Recoverable Information. Recoverability of information in the context of this document is availability of the most recent value of a field on the common snapshot/delta feed in form of order book snapshots. The Xetra Market Data Interface does not provide retransmission/query services.
- Unrecoverable Information. For information sent as market event the most recent value is available through the interface in form of snapshot and delta/incremental messages. If the packet carrying a delta information does not reach the member system, the information should be considered as lost. The member will be able to detect the loss of such information as such information is sequenced and then, recover it using a snapshot.
- Order Book (ODB). A digital book of orders for a tradable unit (instrument) available for matching and maintained by Xetra.
- Trading Session: The phrase "Trading Session" means a trading session within a FIX context throughout this document.

For a more detailed list of definitions, see table in section 7, Appendix - Glossary of Terms.

3.2 Points for Special Attention

Listed below are some points for special attention regarding the data content published by the Xetra Market Data Interface.

- Order book rules for auctions: Order book information disseminated during auctions depends on the instrument set-up. Please find more details in the corresponding Xetra market model documents.
- All Trade Price (ATP) data is disseminated via Ticker Feed.

- A News feed will not be supported by Xetra Market Data Interface.
- Reference data will not be disseminated via the Xetra Market Data Interface.
- Quote Requests will not be disseminated via the Xetra Market Data Interface.
- No beacon messages are sent for the Xetra Market Data Interface.
- No start of service or end of service messages are sent for Xetra Market Data Interface.

4 Data Feeds

The Xetra Market Data Interface will disseminate information in three kinds of feeds in a live-live concept meaning that the data will be disseminated in separate feeds over two services called service A and service B.

These three feeds are:

- State Changes.
- Ticker.
- Market Data.

Due to the inherently unreliable nature of the UDP protocol, data packets may be lost. Members are advised to join both services (A and B) to reduce the probability of data loss.

4.1 State Changes Feed

State Change and other information for the Xetra Market Data Interface are provided by the following messages:

- *Exchange State Change/Fast Market Change/Market Reset* message.
- *System State Change* message.
- *Instrument State Change* message.
- *Cross Request* message.

4.2 Ticker Feed

This feed provides ticker data for Xetra internal and external instruments as well as Xetra trade price information. The following messages are provided via this feed:

- *Ticker* message.
- *All Trade Price* message.
- *Price without Turnover* message.

4.3 Market Data Feed

The current state of the order book is distributed using the order book information messages. Members have to build and maintain their own picture of the order book from these messages:

- *Order Book Snapshot* message.
  Snapshots contain complete order book information up to a depth defined by Xetra (see Xetra Member Homepage). Snapshots provide information about the actual instrument status, the details of the last trade and the days' statistical information for the instrument.

- *Order Book Delta/Incremental message*.
  Deltas should be interpreted as commands issued by the exchange. A member has to alter/update their copy of the order book for each instrument based on the delta messages received.

In the Xetra Market Data Interface Snapshot and Delta messages are sent on the same feed, in contrast to the Enhanced Broadcast Solution, where Snapshot and Delta messages are sent on different feeds. More information about maintaining the global market picture can be found in chapter 11, Appendix – How to Use.

Snapshots and Deltas/Incrementals carry a common and continuous sequence number.

The Market Data Feed is published for each Xetra Broadcast group separately. For identification of a relevant market data feed the Market (MIC Code) and the Xetra broadcast group needs to be identified.

Xetra publishes the multicast addresses for the different markets and broadcast groups in the file "Xetra Multicast Addresses for Production and Simulation" in the Xetra Member Section.

4.4 Interface Schedule

The Xetra Market Data Interface will be based on a "push only" architecture and the data will be disseminated in feeds. Member applications will have to join a particular multicast address and port combination for connecting to a feed. Whenever the interface sends data over a feed, all client applications joined to it will receive these data packets.

Please note that the information regarding feeds which member applications have to join for receiving market information, will be provided by Xetra Member Homepage, cf. chapter 13.

A mapping between the trading phases and the functional timeline of the Market Data Interface is shown below:

Xetra Exchange Period	Xetra Market Data Interface	Member System
Start of business day	No service.	Read feed information from Xetra Member Homepage.
Before Pre-trade	Start of Market Data, State Changes and Ticker Feed without notification.	Join the feeds.
During the trading day	Continue broadcasting the information.	Receive the disseminated information.
After end of trading	Service is brought down without notification message	Leave the feeds. End of trading day

5 FIX/FAST Implementation

All messages, all field data types and encoding formats disseminated by the Xetra Market Data Interface conform to the FAST standard 1.2. The FAST 1.2 specification is provided as an extension to the FAST 1.1 specification. The documents can be found under the following links:

FAST Specification (Version 1.1) and FAST version 1.2 Extension Proposal

www.fixtrading.org > Standards > FIX Standards > FIX Adapted for Streaming (FAST)

5.1 Structure of Messages

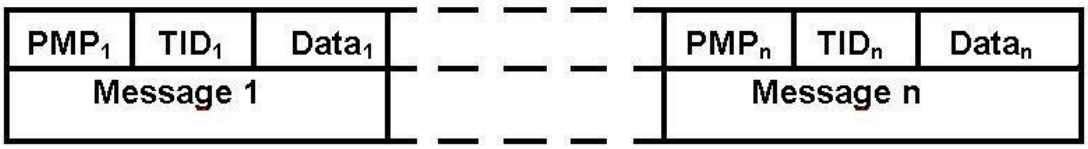
The Xetra Market Data Interface disseminates data in UDP datagrams in network byte order also known as big Endian byte order. Every UDP datagram sent through the Xetra Market Data Interface will have the following structure:

- The UDP datagram starts with the Packet Header message.
- Followed by a FAST Reset message.
- Followed by the actual message (Msg_1).
- Possibly followed by one or more messages (Msg_2 – Msg_n).

Each message shown in the picture above has the following sub structure:

- PMP (Presence Map).
- TID (Template ID).
- Data part.

This is shown in the following diagram:



The UDP protocol adds at least a 28 byte header to every packet (20 byte IP header plus 8 byte UDP protocol header).

Every UDP datagram will be complete, i.e. there will be absolutely no dependency across datagrams. This is required to compensate for the unreliable nature of the UDP protocol.

5.2 FAST Terminology

5.2.1 FAST Reset Message

The Xetra Market Data Interface uses global dictionary scope for FAST operators[2]. All the operators share the same dictionary regardless of the template and application type. The FAST reset message is inserted at the start of every data part in order to explicitly reset all the dictionaries.

5.2.2 Presence Map (PMP)

The presence map is a bit combination indicating the presence or absence of a field (or group) in the message body. The allocation of a bit for a field in the presence map is governed by the FAST field encoding rules[3].

If the presence map bit is set**,** it indicates that the field is sent within the message and if not, it indicates that the field is omitted.

If a certain field does not need a presence map bit, the table value will be "N.A." (not applicable).

5.2.3 Template Identifier (TID)

The template identifier is represented by a number (integer) and points to a specific FAST template which describes the layout and the characteristics of the message to be encoded. The FAST XML file can be downloaded in the Xetra Member Section.

FAST uses templates to reduce redundancies within a message by using the following methods:

- The order of fields within the FAST message is fixed, so the field meaning is defined by its position in the message and there is no need to transfer the field tag to describe the field value.

- The templates specify the order and occurrence of message fields like type, presence and operators.

The following table lists the message types and their corresponding template identifiers. The template IDs in the list are not sequenced, there are a couple of gaps due to the already existing templates for the Enhanced Broadcast Solution in Eurex and due to FAST rules as to template numbering[4].

[2] The dictionary scope should always be derived from the template definition.

[3] see section 10.5 of the FAST specification document at www.fixtrading.org > Standards > FIX Standards > FIX Adapted for Streaming (FAST)

[4] To learn more about the template identifiers rules, read appendix 2.2 of the Session Control Protocol FAST document at www.fixtrading.org > Standards > FIX Standards > FIX Adapted for Streaming (FAST)

Message	Template ID
Packet Header message	37
Delta / Incremental message	40
Snapshot message	41
System State Change message	42
Exchange State Change / Fast Market Change / Market Reset message	43
Instrument State Change message	44
Ticker message	45
Cross Request message	47
All Trade Price Data Message	48
Price without Turnover Message	49

The FAST Reset Message template[5] (TID=120) is not included in the FAST Message Templates file. This TID is reserved in the main FAST specification and allocated by the SCP 1.1 specification. A conforming decoder must treat this as predefined, so it is not necessary to mention it in the template file. Nevertheless, the “reset message” is sent out for initialization of the dictionary.

5.2.4 Dictionaries

A dictionary is a cache in which previous values are stored. FAST operators (->5.2.6) make use of the previous values.

5.2.5 Stop bit encoding

Most FAST fields are stop bit encoded; each byte consists of seven data bits for data transfer and a stop bit to indicate the end of a field value. Exceptions from this rule are Byte Vectors, as they are used in the packet header of Xetra Market Data Interface.

5.2.6 FAST operators

Field operators are used to remove redundancies in the data values. Message templates are the metadata for the message and are provided earlier.

When the messages arrive, the receiving application has complete knowledge of the message layout via the template definition; it is able to determine the field values of the incoming message.

[5] Standard FAST template:
<template name="Reset" scp:reset="yes" id="120"><typeRef name="Reset"/></template>

The following FAST operators are used in Xetra Market Data Interface:

- delta.
- copy.
- constant.
- default.
- increment.

For more information on the new FAST 1.2 features please refer to:

www.fixtrading.org > Standards > FIX Standards > FIX Adapted for Streaming (FAST)

5.3 Decoding the FAST message

The FAST messages need to be decoded by means of the FAST templates. The FAST templates provide all necessary information to decode a message such as data types (e.g. uInt32), field names (e.g. MsgType), FIX tags (e.g. 35) and FAST operators (e.g. increment). The FAST templates also contain information about repeating groups (sequences)

5.4 Transfer decoding

Transfer decoding describes the process of how the fields are decoded from the FAST format. For further information, please refer to section 10 of the FAST Specification Version 1.1. Transfer encoding describes the opposite process.

5.5 Composing the Actual FIX message

A typical FAST decoder would not deliver FIX messages in tag-value syntax after the decoding process. In order to compose FIX messages, applications need to apply additional rules. The sequence of FIX-fields after composing the FIX-message on participants' side is not governed by the FIX-layout of the messages, i.e. the fields names of the FIX-message do not need to be in the same sequence.

The FIX message in tag-value syntax, however, needs to fulfill the minimum requirement:

- *BeginString (8)* in the Standard Header must be the first tag in the message.
- *BodyLength (9)* in the Standard Header must be the second tag in the message.
- *MsgType (35)* in the Standard Header must be the third tag in the message.
- *CheckSum (10)* Standard Trailer must be the last tag in the message.

5.6 New features in FAST version 1.2

The following new features from the FAST 1.2 protocol are used:

- **New Type Definition Syntax:** This allows the separation of the “type definitions” from the “type usage” within template definitions.
- **Enumeration:** This feature can be used when there is a fixed set of valid values for a single field.
- **Set (multi-value field):** This feature can be used when there is a fixed set of valid values which could be sent together as a bit combination instead of using a repeating group. An example for a set would be the field TradeCondition (277) in the Order Book Delta/Incremental Message. Sets are used to define the valid values for fields.
- **Timestamp Data Type:** The use of this feature allows native support of time stamp fields which becomes increasingly important for the Xetra market data interface. A time stamp is an integer that represents a number of time units since an epoch. Please note that, starting from January 2, 2018 onwards, all time stamps will be formatted in UTC format.

5.7 Data types

The Xetra implementation of FAST utilizes the following FAST data types:

- Decimal.
- Length.
- String.
- uInt32/uInt64.
- Byte vector.
- Set.
- Enum.
- Timestamp.

6 Message Formats

Messages are logically divided into service messages that do not contain any market information and market data messages.

Some special comments that should not be forgotten:

- The Xetra Market Data Interface maintains the sequence numbers per feed.
- Price and quantity fields are sent as scaled numbers with a mantissa of 8 bytes and an exponent of 4 bytes.

6.1 Service Messages

Service messages do not carry any market information. These messages are sent for the purpose of synchronization or to indicate the status of the service.

For more information on the FAST implementation, refer to the XML template on the Xetra Member Homepage.

6.1.1 FAST Reset Message

The template ID = 120 is not included in the FAST Message Templates file. This TID is reserved in the main FAST specification and allocated by the FAST Session Control Protocol specification (SCP 1.1).

Note: A conforming decoder must be able to deal with the FAST Reset message even though it is not mentioned in the template file. Once the FAST Reset message is sent out, the dictionary needs to be initialized.

6.1.2 Packet Header

Delivered in: Every UDP datagram

The Packet Header is a technical header used for identification of datagrams and is sent out on a channel basis. Every host stamps outgoing datagrams with a sequence number (field: PacketSeqNum).

One method to identify duplicates between Service A and B is by the use of the field PacketSeqNum which is unique per senderCompID; a faster way is to perform a memory comparison on the first 9 bytes of the packet header.

This method eliminates the need to even decode the header in order to determine, if it has already been processed. This is especially useful to applications using both Service A and Service B feeds, allowing them to determine that a packet has already been processed without incurring any decoding overhead at all.

The datagram sequence number (PacketSeqNum) cannot be used to detect data loss on the application level. The receiver application has to use the sequence number (MsgSeqNum) in the messages for this purpose

As the packet header message is not defined in the FIX standard, the FIX Tags are not shown in the table below. The following layout is available after FAST decoding of the packet header:

Packet Header Message			
Field Name	**Descriptive Name**	**Presence**	**Field Format**
TID	Template ID	Mandatory	Template ID type (8.30)
senderCompID	Identifier of the message disseminating source (host)	Mandatory	SenderCompId type (8.28)
PacketSeqNum	Sequence number of the datagram	Mandatory	PacketSeqNum (8.20)
SendingTime	Time stamp	Mandatory	SendingTime type (8.29)

The following picture shows the structure of the packet header before FAST decoding:

Presence map	Template ID	sender CompID	Length (4)	PacketSeqNum	Length (5)	SendingTime
1 byte	1 byte	1 byte	1 byte	4 bytes	1 byte	5 bytes

The Packet Header Message is FAST-encoded, but is guaranteed to always be exactly 14 bytes in length. The Packet Header Template ID (TID) is always present. To ensure that the message remains fixed length the SendingTime and PacketSeqNum fields are raw values encoded as Byte Vectors in network order.

The TID and senderCompID (datagram sequence number) are range-constrained to a single byte, and the remaining elements of the header are encoded as fixed-length byte vectors.

6.2 Market Data Messages

6.2.1 Order Book Snapshot Message

Delivered on: Market Data feed.

The snapshot messages will be disseminated over the market feed together with the delta / incremental messages. The following principles are valid for the design of the snapshot messages.

- Snapshot messages include top-of-book and price-level information (depth ≥ 1).
- Orders are aggregated per price level and not distributed individually.
- Snapshot messages provide complete information about **only one** instrument up to a specific level.
- Order book information is combined with statistical and trade information in a single message.
- Price levels are provided explicitly (by a number) and do not need to be inferred through the price itself.
- During the pre-trading and post-trading phases when no price levels exist for an instrument, an empty book will be disseminated.
- Delta and snapshot messages use the same sequence number series.
- Xetra Market Data Interface applications must process Order Book Snapshot Messages and Order Book Delta/Incremental Messages: The snapshot message might provide new order book information that has not been sent in a delta/incremental message. This is indicated in the snapshot message by the field RefreshIndicator (1187).

FIX tag	Field Name	Req'd	Field Format	Description
35	MsgType	Y	See 8.16	W = MarketDataSnapshotFullRefresh
34	MsgSeqNum	Y	See 8.15	The sequence number of the message is incremented per feed.
48	SecurityID	Y	See 8.23	Internal identifier assigned to each instrument
49	SenderCompID	Y	See 8.28	Source ID of sender
1187	RefreshIndicator	Y	See 8.22	Set by the sender to tell the receiver to perform a refresh
779	LastUpdateTime	Y	See 8.6	Time of last change for SecurityID
268	NoMDEntries	Y	See 8.17	Defines the size of the repeating group.

FIX tag	Field Name	Req'd	Field Format	Description
269	> MDEntryType	Y	See 8.11	0 = Bid 1 = Offer 2 = Trade 4 = Opening price 5 = Closing Price 7 = Trading Session High Price 8 = Trading Session Low Price J = Empty book Q = Auction Clearing Price; potential auction price b = Market Bid c = Market Ask
625	> TradingSessionSubID	Y	See 8.34	Identifier for Trading Session sub-level phases
326	> SecurityTradingStatus	Y	See 8.27	Trading Status of securities
276	> QuoteCondition	N	See 8.21	Together with MDEntryType 0 or 1 it defines a surplus 'Z' or a matching range 'Q'
270	> MDEntryPx	N	See 8.8	Price
271	> MDEntrySize	N	See 8.9	Quantity
346	> NumberOfOrders	N	See 8.18	Number of orders on this level
1023	> MDPriceLevel	N	See 8.12	Book level
277	> TradeCondition	N	See 8.32	Together with MDEntryType 2 it defines the type of trade U = Exchange last R = Opening price AX = High price AY = Low price AJ = Official closing price AW = Last auction price V = Final price of session AZ = BEST price BB = Midpoint price BC = From subscription period V = Special Auction; e.g. for Federal Bonds the Special Auction indicator is used for prices determined with Bundesbank participation
273	> MDEntryTime	N	See 8.10	Time of entry, official timestamp for trades
6139	> TotalNumOfTrades	N	See 8.31	The cumulative total of the number of Trades for the current day in a given instrument.
1020	> TradeVolume	N	See 8.33	Cumulative volume of units traded in the day.

6.2.2 Order Book Delta/Incremental Message

Delivered on: Market Data feed.

The order book delta messages will be disseminated over the market feed together with the snapshots. The following principles are valid for the design of the delta messages.

- Delta/incremental messages include partial price-level information (depth ≥ 1).
- Orders are aggregated per price level and not distributed individually.
- Delta/incremental messages provide complete information about only one instrument for one or more levels.
- Delta messages carry statistical and order book information.
- Price levels are provided explicitly (by a number) and do not need to be inferred through the price itself.
- During the pre-trading and post-trading phases when no price levels exist for an instrument, an empty book will be disseminated.
- The sequence number is common with the snapshot messages on the same feed.

The following values are valid for the *mdUpdateAction* field:

- New: Creating a price level, adds the new price at the specified mdPriceLevel say x. All price levels y where y >= x are shifted to y + 1.
- Change: Changing a price level, replaces the price and/or quantity of the price level specified by the mdPriceLevel with the information sent in the message.
- Delete: Deleting a price level, removes the price at the level specified by mdPriceLevel say x. All price levels y where y > x are shifted to y - 1.
- Delete From: Deletes all price levels from mdPriceLevel >= x to maximum price levels maintained in the order book for the instrument.
- Delete Thru: Deletes all price levels from price level = '1' to mdPriceLevel = 'x'. All price levels y where y > x are shifted to y - x.
- For all delete actions (delete, delete from, delete thru) the fields mdEntryPx, mdEntrySize and numberOfOrders have to be ignored in the repeating group noMDEntries.

FIX tag	Field Name	Req'd	Field Format	Description
35	MsgType	Y	See 8.16	X = MarketDataIncrementalRefresh
34	MsgSeqNum	Y	See 8.15	The sequence number of the message is incremented per feed.
49	SenderCompID	Y	See 8.28	Source ID of sender
779	LastUpdateTime	Y	See 8.6	Time of last change for SecurityID
268	NoMDEntries	Y	See 8.17	Defines the size of the repeating group

FIX tag	Field Name	Req'd	Field Format	Description
269	> MDEntryType	Y	See 8.11	0 = Bid 1 = Offer 2 = Trade 4 = Opening price 5 = Closing Price 7 = Trading Session High Price 8 = Trading Session Low Price Q = Auction Clearing price; potential auction price b = Market Bid c = Market Ask
48	> SecurityID	Y	See 8.23	Internal identifier assigned to each instrument
625	> TradingSessionSubID	Y	See 8.34	Identifier for Trading Session sub-level phases
326	> SecurityTradingStatus	Y	See 8.27	Trading Status of securities
279	> MDUpdateAction	Y	See 8.14	0 = New 1 = Change 2 = Delete 3 = Delete thru 4 = Delete From Instruction to clear book will be "Delete from price level 1" for bid and offer side
276	> QuoteCondition	N	See 8.21	Together with MDEntryType 0 or 1 it defines a surplus 'Z' or a matching range 'Q'
270	> MDEntryPx	N	See 8.8	Price
271	> MDEntrySize	N	See 8.9	Quantity
346	> NumberOfOrders	N	See 8.18	Number of orders on this level
1023	> MDPriceLevel	N	See 8.12	Book level
277	> TradeCondition	N	See 8.32	Together with MDEntryType 2 it defines the type of trade U = Exchange last R = Opening price AX = High price AY = Low price AJ = Official closing price AW = Last auction price V = Final price of session AZ = BEST price BB = Midpoint price BC = From subscription period V = Special Auction; e.g. for Federal Bonds the Special Auction indicator is used for prices determined with Bundesbank participation
273	> MDEntryTime	N	See 8.10	Time of entry, official timestamp for trades
6139	> TotalNumOfTrades	N	See 8.31	The cumulative total of the number of Trades for the current day in a given instrument.
1020	> TradeVolume	N	See 8.33	Cumulative volume of units traded in the day.

6.2.3 Ticker Message

Delivered on: Ticker feed.

FIX tag	Field Name	Req'd	Field Format	Description
35	MsgType	Y	See 8.16	X = MarketDataIncrementalRefresh
34	MsgSeqNum	Y	See 8.15	The sequence number of the message is incremented per ticker message.
49	SenderCompID	Y	See 8.28	Source ID of sender
268	NoMDEntries	Y	See 8.17	Defines the size of the repeating group
279	> MDUpdateAction	Y	See 8.14	0= New ; always "new"
48	> SecurityID	Y	See 8.23	ISIN of instrument
270	> MDEntryPx	Y	See 8.8	Price or index value
271	> MDEntrySize	N	See 8.9	Quantity, not set for indices
273	> MDEntryTime	N	See 8.10	Time of market data entry
15	> Currency	N	See 8.1	Price currency
1500	> MDStreamID	Y	See 8.13	Name of the price source

6.2.4 All Trade Price Message

Delivered on: Ticker feed.

FIX tag	Field Name	Req'd	Field Format	Description
35	MsgType	Y	See 8.16	X = MarketDataIncrementalRefresh
34	MsgSeqNum	Y	See 8.15	The sequence number of the message is incremented per ALL TRADE PRICE message from a source ID.
49	SenderCompID	Y	See 8.28	Source ID of sender
268	NoMDEntries	Y	See 8.17	Defines the size of the repeating group
269	> MDEntryType	Y	See 8.11	Always used. 2 = Trade
279	> MDUpdateAction	Y	See 8.14	0 = New 1 = Change 2 = Delete
48	> SecurityID	Y	See 8.23	Internal Identifier for Instrument

FIX tag	Field Name	Req'd	Field Format	Description
625	> TradingSessionSubID	N	See 8.34	2 = Opening Auction 3 = Continuous Trading 4 = Closing Auction 6 = Intraday Auction (incl. Continuous Auction) 100 = End-of-day Auction
326	> SecurityTradingStatus	N	See 8.27	17 = Ready to trade 111 = Regular volatility interrupt 111 is used to define that the price is determined by a volatility interruption during continuous trading. For Price without Turnover quotes always 17 (Ready to trade) is used.
270	> MDEntryPx	Y	See 8.8	Trade price value
271	> MDEntrySize	N	See 8.9	Trade quantity
273	> MDEntryTime	Y	See 8.10	Time of price determination
277	> TradeCondition	N	See 8.32	U = Exchange Last Price AZ = BEST Price BB = Midpoint Price BC = Price from Subscription Period V = Special Auction; e.g. for Federal Bonds the Special Auction indicator is used for prices determined with Bundesbank participation.
278	> MDEntryID	Y	See 8.7	Transaction match ID

6.2.5 Price Without Turnover Message

Delivered on: Ticker feed.

FIX tag	Field Name	Req'd	Field Format	Description
35	MsgType	Y	See 8.16	X = MarketDataIncrementalRefresh
34	MsgSeqNum	Y	See 8.15	The sequence number of the message is incremented per ALL TRADE PRICE message from a source ID.
49	SenderCompID	Y	See 8.28	Source ID of sender
268	NoMDEntries	Y	See 8.17	Defines the size of the repeating group
269	> MDEntryType	Y	See 8.11	Always used. 2 = Trade
279	> MDUpdateAction	Y	See 8.14	0 = New 1 = Change 2 = Delete
48	> SecurityID	Y	See 8.23	Internal Identifier for Instrument
625	> TradingSessionSubID	N	See 8.34	3 = Continuous Trading 6 = Intraday Auction (incl. Continuous Auction) 100 = End-of-day Auction

FIX tag	Field Name	Req'd	Field Format	Description
326	> SecurityTradingStatus	N	See 8.27	17 = Ready to trade
270	> MDEntryPx	Y	See 8.8	Trade price value
273	> MDEntryTime	Y	See 8.10	Time of price determination
277	> TradeCondition	N	See 8.32	V = Special Auction; e.g. for Federal Bonds the Special Auction indicator is used for prices determined with Bundesbank participation
278	> MDEntryID	Y	See 8.7	Transaction match ID

6.2.6 Exchange State Change / Fast Market Change / Market Reset Message

Delivered on: State Changes feed.

When Fast Market is ON for an instrument type: Every following update of an instrument of this instrument type will have the Fast market flag within the field *SecurityMassTradingStatus*.

The message of type "CO" is part of FIX 5.0 SP2 (EP106).

FIX tag	Field Name	Req'd	Field Format	Description
35	MsgType	Y	See 8.16	Message Type. CO = SecurityMassStatus
34	MsgSeqNum	Y	See 8.15	The sequence number of the message is incremented per maintenance feed message.
49	SenderCompId	Y	See 8.28	Source ID of sender
1679	SecurityMassTradingStatus	Y	See 8.25	2: Trading Halt 17: Ready to Trade 18: Not available for trading 23: Fast Market
1680	SecurityMassTradingEvent	Y	See 8.24	2: Market Reset, Trading resumes. 5: Change of Trading Subsession; Exchange segment state change. 6: Change of SecurityTradingStatus; Fast Market change.
1544	InstrumentScopeProductComplex	N	See 8.2	This tag is restricted to instruments in trading model "Continuous Auction" only. The target state is always delivered with TradingSessionSubID (625). Mutually exclusive with (1616, 1547) and 1545.

FIX tag	Field Name	Reqʼd	Field Format	Description
625	TradingSessionSubID	N	See 8.34	1: Pre-Trading 3: Trading 5: Post-Trading 7: Quiescent Used for exchange state change in trading model “Continuous auction” together with InstrumentScopeProductComplex (1544). The combination of SecurityMassTradingStatus (1679) and TradingSessionSubID (625) of 2: Trading Halt and 7: Quiescent means that the Trading is on Halt for this Exchange Segment. The combination of 18: Not available for Trading and 7: Quiescent means that the Trading is finished for this business day.
1616	InstrumentScopeSecurityExchange	N	See 8.3	Exchange, on which Fast Market is enabled. Together with InstrumentScopeSecurityType (1547).
1547	InstrumentScopeSecurityType	N	See 8.5	Fast Market. The value corresponds to field “instTypCod” in reference data: Reference data ⇔ Security Type: “EQU” ⇔ “STOCK” ”BON” ⇔ “BOND” “WAR” ⇔ “WAR” “BAS” ⇔ “BAS” Mutually exclusive with (1544 InstrumentScopeProductComplex, 625 TradingSessionSubID).
1545	InstrumentScopeSecurityGroup	N	See 8.4	1 … 1007 Technical Market Reset. The value corresponds to the field “setId” in reference data. Mutually exclusive with (1616 InstrumentScopeSecurityExchange, 1547 InstrumentScopeSecurityType) and (1544 InstrumentScopeProductComplex, 625 TradingSessionSubID).
60	TransactTime	Y	See 8.36	Timestamp when the business transaction represented by the message occurred.

6.2.7 System State Change Message

Delivered on: State Changes feed.

FIX tag	Field Name	Req'd	Field Format	Description
35	MsgType	Y	See 8.16	h = TradingSessionStatus
34	MsgSeqNum	Y	See 8.15	The sequence number of the message is incremented per maintenance feed message.
49	SenderCompID	Y	See 8.28	Source ID of sender
340	TradSesStatus	Y	See 8.35	1 = Halt
60	TransactTime	Y	See 8.36	Timestamp when the business transaction represented by the message occurred.

6.2.8 Instrument State Change

Delivered on: State Changes feed.

Comprises the following instrument state change information:

- Suspension/Release.
- Knock-Out/Revoke.
- Midpoint book freeze/unfreeze.

FIX tag	Field Name	Req'd	Field Format	Description
35	MsgType	Y	See 8.16	f = SecurityStatus
34	MsgSeqNum	Y	See 8.15	The sequence number of the message is incremented per maintenance feed message.
49	SenderCompID	Y	See 8.28	Source ID of sender
48	SecurityID	Y	See 8.23	Internal identifier for instrument
965	SecurityStatus	Y	See 8.26	1=Active 3=Knocked out 4=Knocked out revoke 'Active' always for suspend and release

FIX tag	Field Name	Req'd	Field Format	Description
326	SecurityTradingStatus	Y	See 8.27	17 = Ready to trade (also Midpoint unfreeze) 21 = Pre-Open 107 = Suspended 126 = Midpoint book freeze 'Ready to trade' always for knocked out actions 'Pre-Open' with 'Active' (965) is released
60	TransactTime	Y	See 8.36	Timestamp when the business transaction represented by the message occurred.

6.2.9 Cross Request Message

Delivered on: State Changes feed.

FIX tag	Field Name	Req'd	Field Format	Description
35	MsgType	Y	See 8.16	U21 = CrossRequest
34	MsgSeqNum	Y	See 8.15	The sequence number of the message is incremented per feed.
49	SenderCompID	Y	See 8.28	Source ID of sender
48	SecurityID	Y	See 8.23	Internal identifier for instrument
38	OrderQty	Y	See 8.19	Requested quantity, can be zero
60	TransactTime	Y	See 8.36	Timestamp when the business transaction represented by the message occurred.

7 Appendix - Glossary of Terms

Term	Explanation
Ask	A price in the order book at which a financial instrument can be bought.
Bid	A price in the order book at which a financial instrument can be sold.
Datagram	Datagram is a unit of information sent by the source. A datagram could be split into one or more network packets.
Delta Message	A delta message carries only the changes in the information sent earlier on the entity that it describes. Also called Incremental message.
Empty Book	An order book which does not contain any information (price levels).
ETS	Enhanced Transaction Solution
FAST	FIX Adapted for Streaming[SM] (FAST Protocol[SM]).
FIX	Financial Information eXchange.
Gap	Missing information related to trades or changes to the order book.
Incremental Message	Other word for a Delta Message, see there.
ITA2	Encoding International Telegraph Alphabet (Standard) 2.
Last Auction Price	This is the price of the last auction for the instrument.
Last Trade Price	Last trade price is the price of the last (most recent) trade for the instrument.
Live-Live Concept	Live - live concept means that the same data is disseminated in parallel over two distinct sets of feeds.
Market Ask	Best ask is a market order.
Market Bid	Best bid is a market order.
Multicast	Many-to-many network architecture as against one-to-one (unicast).
Netting	Collect data in a defined interval before sending.
Opening Price	This is the price of the first trade for the instrument. This price is recoverable through the snapshot feed.
Order	A contractually-binding request to other market participants to buy or sell a specific quantity of a financial instrument at a defined price.

Term	Explanation
Order Book (ODB)	Contains all current orders for an instrument, according to their trading restrictions and execution conditions.
Packet	Packet is generally referred to a single set of data bits carried by the network.
PIM	Protocol Independent Multicast.
Potential Auction Price	This is a theoretical value published in case of a cross order book, when the best bid and best ask prices are not revealed. This single price represents both the buy and sell side of the ODB. This price is unrecoverable.
Price	Public price information (bid/ask price, bid/ask quantities, traded quantity, daily high/low, etc).
Quote	Simultaneous entry of a limit buy and a limit sell order for the same instrument.
Snapshot Message	A snapshot message carries complete (all available) information about the entity that it describes.
Feed	A feed is a collection of one or more instruments and is the basic unit of subscription.
Trading Range	Specialist/Market Maker provides indicative quote (Trading Model Continuous Auction).
UDP	User Datagram Protocol.
VALUES	Virtual Access Link Using Exchange Services. It is the programmable interface providing connectivity to Deutsche Börse's electronic trading platforms (Eurex and Xetra®). VALUES API provides a single point of entry to the full range of the exchange functionality.

8 Appendix - Data Field Dictionary

The following section provides specific information on the Xetra Market Data Interface data fields in alphabetic order, listing valid values, ranges or giving an example value for the field, and adding the messages using the fields.

8.1 Currency

Description: Currency code of the instrument.
This field identifies the currency in ISO 4217 representation.
Type: ASCII Character String

Value Examples	Meaning
"EUR"	Euro
"GBP"	British Pound
"USD"	US Dollar

Where sent:

- Ticker Message.

8.2 InstrumentScopeProductComplex

Description: Name of the exchange segment.
Type: ASCII Character String

Value Example
"XEXC"

Where sent:

- Exchange State Change / Fast Market Change / Market Reset Message

8.3 InstrumentScopeSecurityExchange

Description: Exchange name.
Type: ASCII Character String

Value Example
"XETR"

Where sent:

- Exchange State Change / Fast Market Change / Market Reset Message

8.4 InstrumentScopeSecurityGroup

Description: Technical grouping of instruments.
Type: Unsigned long (uint32)

Value Range
1..1007

Where sent:

- Exchange State Change / Fast Market Change / Market Reset Message

8.5 InstrumentScopeSecurityType

Description: Security Type
Type: ASCII Character String

Value	Meaning
“STOCK”	Equities
“WAR”	Warrants
“BOND”	Bonds
“BAS”	Basis instruments

Where sent:

- Exchange State Change / Fast Market Change / Market Reset Message

8.6 LastUpdateTime

Description: The timestamp when the message was created by the Xetra Market Data Interface, in the format microseconds since midnight in Coordinated Universal Time (UTC)[6].
Type: Unsigned long (uint64)

Value Range	Meaning
0..86399999999	Microseconds since midnight

Where sent:

- Snapshot Message
- Delta / Incremental Message

[6] Till December 29, 2017 the timestamps will contain the time since midnight in Central European Time (CET/CEST).

8.7 MDEntryID

Description: Identifier of trade
Type: Unsigned long (uint32)

Value Range

0..>4294967295

Where sent:

- All Trade Price Message

8.8 MDEntryPx

Description: Market data price field. Float field expressed as exponent (4 bytes) and mantissa (8 bytes) fields.
Type: Decimal

Value Example

exponent = -2
mantissa = 2555
resulting value = 25.55

Where sent:

- Snapshot Message
- Delta / Incremental Message
- Ticker Message
- All Trade Price Message

8.9 MDEntrySize

Description: Quantity of market data (trade resp. order)
Type: Decimal

Value Example

exponent = -2
mantissa = 2555
resulting value = 25.55

Where sent:

- Snapshot Message
- Delta / Incremental Message
- Ticker Message
- All Trade Price Message

8.10 MDEntryTime

Description: Time of market data entry (order/trade) in the format milliseconds since midnight in Coordinated Universal Time (UTC) [7].

Type: Unsigned long (uint32)

Value Range	Meaning
0..86399999	Milliseconds since midnight

Where sent:

- Snapshot Message
- Delta / Incremental Message
- Ticker Message
- All Trade Price Message

8.11 MDEntryType

Description: Type of market data entry

Type: ASCII Character String

Value	Meaning
0	Bid
1	Offer
2	Trade
4	Opening price
5	Closing price
7	Trading Session High Price
8	Trading Session Low Price
J	Empty book
Q	Auction Clearing price
b	Market Bid
c	Market Ask

Where sent:

- Snapshot Message

[7] Till December 29, 2017 the timestamps will contain the time since midnight in Central European Time (CET/CEST).

- Delta / Incremental Message
- All Trade Price Message

8.12 MDPriceLevel

Description: The type of market data entry level
Type: Unsigned long (int32)

Value	Meaning
1	Top of book
1…10	Depth level

Where sent:

- Snapshot Message
- Delta / Incremental Message

8.13 MDStreamID

Description: Price source identifier
Type: ASCII Character String

Value (e.g.)	Status
RTR	Reuters
BUL	Bulgarian Stock Exchange
EUB	Eurex Bonds Indices
ETR	Xetra Frankfurt
FRA	Xetra Frankfurt 2
EUR	Eurex Frankfurt
IDX	Deutsche Börse Selected Indices
STX	STOXX Indices

NB: Generally, values not listed in this table are possible, too.

Where sent:

- Ticker Message

8.14 MDUpdateAction

Description: Market Data Update Action Code

Type: ASCII Character String

Value	Meaning
0	New
1	Change
2	Delete
3	Delete thru
4	Delete from

Where sent:

- Delta / Incremental Message
- Ticker Message
- All Trade Price Message

8.15 MsgSeqNum

Description: Sequence number to assist in identifying data loss.
Type: Unsigned long (int32)

Value Range
0..4294967295

Where sent:

- Snapshot Message
- Delta / Incremental Message
- Ticker Message
- All Trade Price Message
- Cross Request
- System State Change
- Exchange State Change / Fast Market Change / Market Reset Message
- Instrument State Change

8.16 MsgType

Description: Identifier of message
Type: ASCII Character String

Value Examples
“h”, “f”, “CO”, “U21”

Where sent:

- Snapshot Message

- Delta / Incremental Message
- Ticker Message
- All Trade Price Message
- Cross Request
- System State Change
- Exchange State Change / Fast Market Change / Market Reset Message
- Instrument State Change

8.17 NoMDEntries

Description: Counter for number of market data entries
Type: Unsigned long (length field)

Value Range

0..4294967295

Where sent:

- Snapshot Message
- Delta / Incremental Message
- Ticker Message
- All Trade Price Message

8.18 NumberOfOrders

Description: Counter for number of orders
Type: Unsigned long (int32)

Value Range

0..4294967295

Where sent:

- Snapshot Message
- Delta / Incremental Message

8.19 OrderQty

Description: Quantity of order
Type: Decimal

Value Example

exponent = -2
mantissa = 2555
resulting value = 25.55

Where sent:

- Cross Request Message

8.20 PacketSeqNum

Description: Sequence Number in the Packet Header message. Every host stamps outgoing datagrams with a sequence number.
Type: Unsigned long64 (byteVector)

Value Range
0..>4294967295

Where sent:

- Packet Header Message

8.21 QuoteCondition

Description: Together with MDEntryType 0 or 1 it defines a surplus 'Z' or a matching range 'Q'
Type: ASCII Character String

Value	Meaning
Z	Order imbalance
Q	Trading range

Where sent:

- Snapshot Message
- Delta / Incremental Message

8.22 RefreshIndicator

Description: Indicates a mandatory refresh.
Type: Boolean

Value	Meaning
Y	Mandatory refresh
Other	Process as required

The refresh indicator signalizes whether a delta message was omitted or not in the snapshot. If the refresh indicator is 'Y' a delta was omitted and the snapshot must be processed, otherwise the snapshot can be processed as required.

Where sent:

- Order Book Snapshot Message.

8.23 SecurityID

Description: Instrument identifier
Type: Ticker Message: ASCII Character String.
Other messages: Unsigned long (uint64)

Value Ranges

ISIN string resp. 0 ... >4294967295

Where sent:

- Snapshot Message
- Delta / Incremental Message
- Ticker Message
- All Trade Price Message
- Cross Request Message
- Instrument State Change

8.24 SecurityMassTradingEvent

Description: Security Mass Trading Event
Type: Unsigned long (uint32)

Value	Meaning
2	Market Reset; Trading resumes
5	Change of trading subsession; Exchange Segment state change
6	Change of security trading status; Fast Market Change.

Where sent:

- Exchange State Change / Fast Market Change / Market Reset Message

8.25 SecurityMassTradingStatus

Description: Security Mass Trading Status
Type: Unsigned long (uint32)

Value	Meaning
2	Trading Halt
17	Ready to trade
18	Not available for trading
23	Fast Market

Where sent:

- Exchange State Change / Fast Market Change / Market Reset Message

8.26 SecurityStatus

Description: Current state of instrument
Type: ASCII Character String

Value	Meaning
1	Active
3	Knocked-out
4	Knocked-out Revoke

'Active' always for suspend and release.

Where sent:

- Instrument State Change

8.27 SecurityTradingStatus

Description: Trading Status of securities
Type: Unsigned long (uint32)

Value	Status
2	Trading Halt
17	Ready to trade
18	Not available for trading
19	Not traded on this market
21	Pre-open
23	Fast Market

Value	Status
100	Regular pre order book balancing
101	Regular order book balancing
102	Regular freeze
103	Fast pre order book balancing
104	Fast order book balancing
105	IPO
106	IPO freeze
107	Suspended
108	Add
109	Deleted
110	Regular potential volatility interrupt
111	Regular volatility interrupt
112	Regular extended volatility interrupt
113	Extended volatility interrupt frozen
114	Fast potential volatility interrupt
115	Fast Volatility interrupt
116	Fast extended volatility interrupt
117	Regular potential market order interrupt
118	Regular market order interrupt
119	Fast potential market order interrupt
120	Fast market order interrupt
121	Volatility interrupt with potential market order interrupt
122	Fast market volatility interrupt with potential market order interrupt
123	Market order interrupt with potential volatility interrupt
124	Fast market order interrupt with potential volatility interrupt
125	In between auction
126	Midpoint book freeze

Where sent:

- Snapshot Message
- Delta / Incremental Message

- All Trade Price Message
- Instrument State Change

8.28 SenderCompID

Description: Identifier of the message disseminating source (host). Members should maintain the sequence numbers of the incoming messages per disseminating source.
Type: Unsigned long (uint32)

Cf. Appendix – How to Use.

Value Range
0..4294967295

Where sent:

- Snapshot Message
- Delta / Incremental Message
- Ticker Message
- All Trade Price Message
- Cross Request
- System State Change
- Exchange State Change / Fast Market Change / Market Reset Message
- Instrument State Change

8.29 SendingTime

Description: The time stamp when the message was created by the Market Data Interface.
Type: Unsigned long64 (byteVector)

Value Range	Meaning
0..86399999999	Microseconds since midnight UTC[8]

Where sent:

- Packet Header Message

8.30 TID

Description: The identifier used to indicate the FAST template to be used for decoding

[8] Till December 29, 2017 the timestamps will contain the time since midnight in Central European Time (CET/CEST).

Type: Unsigned long (uint64)

Value Range

0..>4294967295

Cf. table of Template IDs on page Seite 16.

Where sent:

- Packet Header Message

8.31 TotalNumOfTrades

Description: The cumulative total of the number of Trades for the current day in a given instrument.

Type: Unsigned long (uint64)

Value Range

0..>4294967295

Where sent:

- Snapshot Message
- Delta / Incremental Message

8.32 TradeCondition

Description: Together with MDEntryType 2 it defines the type of trade

Type: ASCII Character String (MultipleStringValue)

Value	Status
U	Exchange last
R	Opening price
AX	High price
AY	Low price
AJ	Official closing price
AW	Last auction price
V	Final price of session
AZ	BEST price
BB	Midpoint price
BC	Price from subscription period ("Handel per Erscheinen")

Value	Status
BD	Currently not used.

For continuous trading:

V = Price determined in intra day closing auction.

If no intra day closing auction is scheduled the closing price is disseminated.

For continuous auction:

V = Special Auction Price; for federal bonds the Special Auction indicator is used for prices determined with Bundesbank participation.

Where sent:

- Snapshot Message
- Delta / Incremental Message
- All Trade Price Message
- Price Without Turnover Message

8.33 TradeVolume

Description: Cumulative volume of units traded in the day
Type: Decimal

Value Example

exponent = -2
mantissa = 2555
resulting value = 25.55

Where sent:

- Snapshot Message
- Delta / Incremental Message

8.34 TradingSessionSubID

Description: Identifier for Trading Session sub-level phases
Type: ASCII Character String

Value	Status
1	Pre-Trading
2	Opening or opening auction
3	(Continuous) Trading
4	Closing or closing auction
5	Post-Trading

Value	Status
6	Scheduled Intraday Auction
7	Quiescent
101	Mini Auction
102	Liquidity Interruption

Where sent:

- Snapshot Message
- Delta / Incremental Message
- All Trade Price Message
- Exchange State Change / Fast Market Change / Market Reset Message

8.35 TradSesStatus

Description: Status of the trading session
Type: Unsigned long (uint32)

Value Range

1 = HALT

The corresponding value in Enhanced Broadcast Solution is 3="HALT".
Messages for "Online" or "Batch" are not sent.
Where sent:

- System State Change Message

8.36 TransactTime

Description: Representation of time in the format milliseconds since midnight UTC[9].
Type: Unsigned long (uint64)

Value Range	Meaning
0..86399999	Milliseconds since midnight

Where sent:

- Cross Request Message
- System State Change Message
- Exchange State Change / Fast Market Change / Market Reset Message
- Instrument State Change

[9] Till December 29, 2017 the timestamps will contain the time since midnight in Central European Time (CET/CEST).

9 Appendix – Additional Information Related to FIX Valid Values

9.1 FIX Market Data Entries for MDI Events

Consider the following notes when reading the following tables:

Statistics are attached to the trade entry only if it happens to be the last trade of the interval and have their own entries otherwise.

Last trade of the interval might change only a few statistics, i.e. there will be a mixture of entry types and trade conditions representing current statistics.

With the exception of state change without book change statistics will only be part of the delta message if one or more trades within the interval caused a change in one or more of the statistics.

Volume statistics are sent as dedicated fields attached to the Last Trade entry (not as a separate MDEntryType).

Volume statistics do not apply to trades other than those marked with TradeCondition U=Exchange Last or AW=Last Auction Price.

Last Trade and Last Auction Price can have multiple TradeCondition values set but are always separate entries (U and AW can never both be set as TradeCondition).

Imbalance information is only shown during auctions in case of a crossed book.

Surplus can never be on both sides, i.e. snapshot only contains one or the other.

Surplus and executable volume are not shown unless the market imbalance indicator is set.

9.1.1 Snapshots

Market Model Event	FIX Fields and Values for Snapshots				
	MDEntryType (269)	MDEntryPx (270)	MDEntry Size (271)	TradeCondition (277)	QuoteCondition (276)
Book is now empty	J = Empty Book	-	-	-	-
State change without book change	-	-	-	-	-
Orders/Quotes					
Buy Limit Orders/Quotes	0 = Bid	Bid Price	Bid Qty	-	-
Sell Limit Orders/Quotes	1 = Offer	Ask Price	Ask Qty	-	-
Buy Market Order	b = Market Bid	-	Bid Qty	-	-
Sell Market Order	c = Market Offer	-	Ask Qty	-	-
Trades (last trade within an interval or last auction information)					

Market Model Event	FIX Fields and Values for Snapshots				
	MDEntryType (269)	MDEntryPx (270)	MDEntry Size (271)	TradeCondition (277)	QuoteCondition (276)
Last Trade in interval is a regular trade without change of statistics	2 = Trade	Traded Price	Traded Qty	U=Exchange Last	-
Last Trade in interval is a regular trade and changes statistics	2 = Trade	Traded Price	Traded Qty	U=Exchange Last R=Opening Price AJ=Closing Price V=Final Price AX=High Price AY=Low Price	-
Last Trade in interval is a special trade (cannot change statistics)	2 = Trade	Traded Price	Traded Qty	U=Exchange Last AZ=Xetra BEST BB=Midpoint BC=Subscription V = Special Auction resp. Bundesbank	-
Last auction price	2 = Trade	Auction Price	Total Traded Qty	AW=Last Auction Price R=Opening Price AJ=Closing Price V=Final Price AX=High Price AY=Low Price	-
Statistics (when they have not been caused by the last trade of the current interval)					
Opening Price	4 = Opening Price	Opening Price	-	-	-
Closing Price	5 = Closing Price	Closing Price	-	-	-
Intraday closing auction price	5 = Closing Price	Auction Price	-	V=Final Price	-
Identical closing and Intraday closing auction price	5 = Closing Price	Auction Price	-	AJ=Closing Price V=Final Price	-
Current/New High Price	7 = Session High Price	High Price	-	-	-
Current/New Low Price	8 = Session Low Price	Low Price	-	-	-
Auctions					
Crossed Book Information	Q = Auction Clearing Price	Potential Price	Executable Qty	-	-

Market Model Event	FIX Fields and Values for Snapshots				
	MDEntryType (269)	MDEntryPx (270)	MDEntry Size (271)	TradeCondition (277)	QuoteCondition (276)
Surplus on the bid side	0 = Bid	-	Surplus Qty	-	Z=Order Imbalance
Surplus on the ask side	1 = Offer	-	Surplus Qty	-	Z=Order Imbalance
Uncrossed book information (bid)	0 = Bid	Bid Price	Bid Qty	-	-
Uncrossed book information (ask)	1 = Offer	Ask Price	Ask Qty	-	-
Matching range on the ask side	1 = Offer	Ask Price	Ask Qty	-	Q=Trading Range
Matching range on the bid side	0 = Bid	Bid Price	Bid Qty	-	Q=Trading Range

9.1.2 Deltas/Incrementals

Market Model Event	FIX Fields and Values for Deltas				
	MDEntryType (269)	MDEntryPx (270)	MDEntry Size (271)	TradeCondition (277)	QuoteCondition (276)
Book is now empty	J = Empty Book	-	-	-	-
State change without book change	7 = Session High Price	-	-	-	-
Orders/Quotes					
Buy Limit Orders/Quotes	0 = Bid	Bid Price	Bid Qty	-	-
Sell Limit Orders/Quotes	1 = Offer	Ask Price	Ask Qty	-	-
Buy Market Order	b = Market Bid	-	Bid Qty	-	-
Sell Market Order	c = Market Offer	-	Bid Qty	-	-
Trades (last trade within an interval or last auction information)					
Last Trade in interval is a regular trade without change of statistics	2 = Trade	Traded Price	Traded Qty	U=Exchange Last	-

Market Model Event	FIX Fields and Values for Deltas				
	MDEntryType (269)	**MDEntryPx (270)**	**MDEntry Size (271)**	**TradeCondition (277)**	**QuoteCondition (276)**
Last Trade in interval is a regular trade and changes statistics	2 = Trade	Traded Price	Traded Qty	U=Exchange Last R=Opening Price AJ=Closing Price V=Final Price AX=High Price AY=Low Price	-
Last Trade in interval is a special trade (cannot change statistics)	2 = Trade	Traded Price	Traded Qty	U=Exchange Last AZ=Xetra BEST BB=Midpoint BC=Subscription V=Special Auction resp. Bundesbank	-
Last auction price	2 = Trade	Auction Price	Total Traded Qty	AW=Last Auction Price R=Opening Price AJ=Closing Price V=Final Price AX=High Price AY=Low Price	-
Statistics (when they have not been caused by the last trade of the current interval)					
Opening Price	4 = Opening Price	Opening Price	-	-	-
Closing Price	5 = Closing Price	Closing Price	-	-	-
Intraday closing auction price	5 = Closing Price	Auction Price	-	V=Final Price	-
Identical closing and Intraday closing auction price	5 = Closing Price	Auction Price	-	AJ=Closing Price V=Final Price	-
Current/New High Price	7 = Session High Price	High Price	-	-	-
Current/New Low Price	8 = Session Low Price	Low Price	-	-	-
Auctions					
Crossed Book Information	Q = Auction Clearing Price	Potential Price	Executable Qty	-	-
Surplus on the bid side	0 = Bid	-	Surplus Qty	-	Z=OrderImbalance
Surplus on the ask side	1 = Offer	-	Surplus Qty	-	Z=OrderImbalance

Market Model Event	FIX Fields and Values for Deltas				
	MDEntryType (269)	MDEntryPx (270)	MDEntry Size (271)	TradeCondition (277)	QuoteCondition (276)
Uncrossed book information (bid)	0 = Bid	Bid Price	Bid Qty	-	-
Uncrossed book information (ask)	1 = Offer	Ask Price	Ask Qty	-	-
Matching range on the ask side	1 = Offer	Ask Price	Ask Qty	-	Q=Trading Range
Matching range on the bid side	0 = Bid	Bid Price	Bid Qty	-	Q=Trading Range

9.2 Mapping of VALUES to FIX for State Change information

Mapping of VALUES prcsStsValCod to FIX standard:

Name	SecurityTradingStatus	TradingSessionSubId
START	2=Trading Halt	7=Quiescent
PRETR	21-Pre-Open	1=Pre-Trading
OCALL	17=Ready to trade 23=Fast Market	2=Opening Auction
ICALL	17=Ready to trade 23=Fast Market	6=Intraday Auction
CCALL	17=Ready to trade 23=Fast Market	4=Closing Auction
ECALL	17=Ready to trade 23=Fast Market	100=End of day auction
OIPO	105=IPO	2=Opening Auction
OFRZ	106=IPO freeze	2=Opening Auction
IIPO	105=IPO	6=Intraday Auction
IFRZ	106=IPO freeze	6=Intraday Auction
OPOBB	100=Regular POBB 103=Fast POBB	2=Opening Auction
IPOBB	100=Regular POBB 103=Fast POBB	6=Intraday Auction
CPOBB	100=Regular POBB 103=Fast POBB	4=Closing Auction

Name	SecurityTradingStatus	TradingSessionSubId
EPOBB	100=Regular POBB 103=Fast POBB	100=End of day auction
OOBB	101=Regular OBB 104=Fast OBB	2=Opening Auction
IOBB	101=Regular OBB 104=Fast OBB	6=Intraday Auction
COBB	101=Regular OBB 104=Fast OBB	4=Closing Auction
EOBB	101=Regular OBB 104=Fast OBB	100=End of day auction
TRADE	17=Ready to trade 23=Fast Market	3=Continuous Trading
BETW	125= In between auction	7=Quiescent
POSTR	2=Trading Halt	5=Post Trading
ENDTR	18=Not available for trading	7=Quiescent
HALT	2=Trading Halt	3=Continuous Trading
SUSP	107=suspended	3=Trading
VOLA	111= Regular Volatility Interrupt 112 =Regular Extended Volatility Interrupt 115= Fast Volatility Interrupt 116 = Fast Extended Volatility Interrupt 113 = Extended Volatility Interrupt Frozen	3=Continuous Trading
ADD	108=Added to trading	7=Quiescent
DEL	109=Deleted from trading	7=Quiescent
HOL	19=Not traded on this market	7=Quiescent
XPREC	21=Pre-open	6=Intraday auction (continuous)
XCALL	17=Ready to trade	6=Intraday auction (continuous)
XFRZ	102=Regular (freeze)	6=Intraday auction (continuous)
MIDFR	126=Midpoint book freeze	6=Intraday auction (continuous)
MIDUF	17=Midpoint book unfreeze	6=Intraday auction (continuous)

Volatility and Market order interruptions will be considered as changes of the security trading status within a trading session sub id.

Name	SecurityTradingStatus	SecurityStatus
MIDFR	126=Midpoint book freeze	1=Active
MIDUF	17=Midpoint book unfreeze	1=Active

VOLA	MOI	SecurityTradingStatus for TradingSessionSubID =2,6,4,100
P		110=Potential Volatility Interrupt (Regular) 114=Potential Volatility Interrupt (Fast)
V		111=Volatility Interrupt (Regular) 115=Volatility Interrupt (Fast)
V	P	121=Volatility Interrupt and potential Market order Interruption (Regular) 122=Volatility Interrupt and potential Market order Interruption (Fast)
E		112=Extended Volatility Interrupt (Regular) 116=Extended Volatility Interrupt (Fast)
F		113=Extended Volatility Interrupt Frozen
	P	117=Potential Market order Interruption (Regular) 119=Potential Market order Interruption (Fast)
	M	118=Market order Interruption (Regular) 120=Market order Interruption (Fast)
P	M	123=Market order Interruption and potential Volatility Interrupt (Regular) 124=Market order Interruption and potential Volatility Interrupt (Fast)

Example: TradingSessionSubID is 6 (intraday auction), instrument type is in regular trade, the last entered order would lead to a volatility interrupt. Then security trading status changes from 17 (Ready to trade) to 110 (Potential Volatility Interrupt (regular)).

Mapping concerning Mini Auction and Liquidity Interruption:

TradingSessionSubID (tag=625)	SecurityTradingStatus (tag=326)	
101 Mini Auction	17 Ready to trade	Price Validation Auction (Mini Auction) in regular trading
	23 Fast Market	Price Validation Action Mini Auction) in fast market trading
	110 Regular potential volatility interrupt	Price Validation Auction (Mini Auction) with potential volatility interrupt in regular trading
	114 Fast potential volatility interrupt	Price Validation Auction (Mini Auction) with potential volatility interrupt in fast market trading
102 Liquidity Interruption	17 Ready to trade	Liquidity Improvement Auction (Liquidity Interrupt) in regular trading
	23 Fast Market	Liquidity Improvement Auction (Liquidity Interrupt) in fast market trading
	110 Regular potential volatility interrupt	Liquidity Improvement Auction (Liquidity Interrupt) with potential volatility interrupt in regular trading
	114 Fast potential volatility interrupt	Liquidity Improvement Auction (Liquidity Interrupt) with potential volatility interrupt in fast market trading

10 Appendix – Interface Limits

Although the protocol is very generic and supports unlimited data size, depending on the scope of the message and the resulting data property, some limits can be stated for the Xetra Market Data Interface.

Limit	Descriptive	Name / Message	Value
MAX LIVE LIVE SERVICES	The Xetra Market Data Interface disseminates data simultaneously in duplicate over two services, 'A' and 'B'	noFeeds	2
MAX ODB DEPTH	Maximum depth of order book supported by Market Data Interface.		10
MAX ODB ENTRIES	Maximum entries for order book data. (MAX ODB DEPTH * 2)	noMDEntries / Snapshot and Delta	20
MAX PRICE ENTRIES	Maximum entries for price data (opening price, last auction price, high price, low price, closing price, valuation price)	noEntriesPrc / Snapshot and Delta	7
MAX STRING SIZE	Maximum size of any string disseminated through the interface. Market Data Interface disseminates mainly binary data.	N.A.	16

11 Appendix – How to Use

This section explains how member applications should use the Xetra Market Data Interface. A member application will be used to explain this in more detail.

Please note that the operations shown below are basic and members are expected to integrate the operations described below into their trading system.

Due to multiple environments, the same instrument (ISIN) can be processed on different market sources. The instrument identifier (ISIX) is only distinct within its context. It means that the same instrument identifier processed on different market sources can represent different instruments. For that reason the version number in the Packet Header message (see chapter 6.1.1) has to be considered to determine the market source.

The version number field “versNo” in the Packet Header Message has been removed. The Packet Header can be derived from multicast address and port number. Additionally, the sequence number is defined as fixed length since Xetra Release 13.0. So each packet header has now always a fixed length of 14 bytes. Thus it can easily be used to detect duplicates.

11.1 Start of Day

11.1.1 Get Reference Data

Members receive information on the available feeds via Xetra Member Homepage (reference data file), via common report engine (reference data file) or via the reference feed of the current Enhanced Broadcast Solution.

The complete Xetra instrument reference data may be retrieved via three different sources:

- Enhanced Broadcast Solution:
  Xetra instrument reference data is disseminated via the Instrument Reference Data message (TID=3) of the Enhanced Broadcast Solution Reference Feed.
- Xetra website:
  For details see Appendix on Xetra Member Homepage.
- Common Report Engine:
  An instrument reference data file is provided via Xetra Backend on the Common Report Engine for download.

11.1.2 Receive Snapshots and Deltas/Incrementals

Members have to join the market feed for the instruments which they require and after receiving the order book snapshots, members should **not** leave this feed as Snapshot and Incremental updates are sent on the same feed. Members are expected to stay connected to this feed throughout the day.

The following steps need to be performed for the market feeds:

- Join the feeds of interest.
- Process datagrams containing data related to the instruments of interest. Discard all unwanted datagrams.
- Accept all incoming snapshot messages and build a local copy of the order book.

- Apply the incoming deltas to the local copy of the order book.

Please note that due to a limited number of multicast address and port pairs information of more than one instrument group can be disseminated on one multicast address and port pair for market feeds. Members will have to filter the information for the sets and instruments of their interest.

11.2 Messages Sequence Numbers Synchronization

Due to the unreliable nature of the UDP protocol, the order in which the broadcasts arrive, whether snapshots, deltas or other is random. Furthermore the UDP packets can be duplicated at the network.

Member applications must be prepared for the fact that multicast does not guarantee the correct sequencing of messages.

To be able to keep track of what has been received in each of the feeds, sequence numbers have been added to the messages per feed. These sequence numbers along with the source identifier (where is this message coming from) can be used to detect lost messages.

11.2.1 Snapshot - Delta Synchronization and Order Book recovery

Snapshot and Delta/Incremental messages are sent on the same feed and therefore carry the same continuously increasing sequence number.

Using Xetra Market Data Interface you can not ignore subsequent snapshots as in Enhanced Broadcast Solution. The reason is, that delta messages are omitted if a snapshot message is sent at that time.

The refresh indicator signalizes whether a delta message was omitted or not in the snapshot. If the refresh indicator is ‘Y’ a delta was omitted and the snapshot must be processed, otherwise the snapshot can be processed as required.

If any gaps are detected in the sequence numbers you have to wait for the next snapshot message to get a proper view of the order book.

11.3 Events Demanding Special Response

Due to the transparent operation of the Xetra Market Data Interface, there are situations when the interface will indicate that there has been some technical interruption or rearrangement of services at Xetra.

Members are expected to adjust to the prevailing situation at that point of time.

Listed below are the situations that may require special response from the member applications. Members should tailor their applications to respond to these events which will be classified as "Normal" or "Exceptional".

Please note that this list is not exhaustive so any changes to this list will be published.

11.3.1 Source Identifier senderCompID

Type	Exceptional
Where sent	Every Packet Header message sent out of the Xetra Market Data Interface carries a source identifier field.
Value	Changes from the previous value being sent for the instrument.

Description	When the source of information to the Xetra Market Data Interface changes, this value is changed accordingly. Members should maintain the market information for each instrument per source identifier.

11.3.2 SenderCompID

Type	Exceptional
Where sent	Every FAST message sent out of the Xetra Market Data Interface carries a SenderCompID field.
Value	Changes from the previous value being sent for the instrument.
Description	When the sender of information to the Xetra Market Data Interface changes, namely the Xetra trade matching system, this value is changed accordingly. There can be situations when this value alternates between two or more values. Members should maintain the market information for each instrument per SenderCompID.

11.3.3 Sequence Number

Type	Normal
Where sent	The Xetra Market Data Interface sequences messages per feed.
Value	Non-consecutive sequence numbers.
Description	This can occur due to the unreliable nature of the UDP protocol. As the interface sequences all messages per feed, any message loss will be known to the member immediately. If members wish to recover their view of the order book from this situation, they have to wait for the next snapshot message.

Type	Exceptional
Where sent	The Xetra Market Data Interface sequences messages per feed.
Value	Sequence numbers abruptly restart.
Description	This can occur due to two reasons: (1) If the sender/source of market data changes. Every time the Xetra Market Data Interface starts receiving data from a new sender/source, it restarts sequencing the messages. (2) Technical malfunction. If there is some technical malfunction and if the interface is unable to recover the old market information, the Xetra Market Data Interface can choose to restart sequencing the messages.

11.3.4 Host Fail-over

Internally, the Xetra Market Data Interface platform is concentrated on a single host, i.e. during normal processing, each feed and its associated instruments are processed by a single host, and it is this host that generates the Xetra Market Data Interface messages for these instruments.

If this host fails, another host takes over and is then responsible for all feeds. The SenderCompID is changed to the new host id and the sequence number starts anew with 1. During a business day a SenderCompID value is not used again in case of a fallback.

11.3.5 Calculation of message frequency

Since Xetra Market Data Interface offers no inquiry or other retransmission mechanism, a recovery mechanism for the Market Feed had to be introduced. This recovery mechanism does not work by offering data previously sent, but by offering frequently sent snapshots as integral part of the Market Feed. Recovery for a market feed then means, that a connected client application can establish a complete view of the data, published via the feed, only from the feed itself. For the recovery of a Market Feed it is necessary to send periodical snapshots for all instruments of this feed.

12 Appendix – Message Encoding

This chapter explains the FAST concepts important for understanding the Xetra Market Data Interface data encoding logic. The encoding adheres to the FAST 1.1 specification with the FAST 1.2 extensions.

Please note that FIX protocol and concepts may change in the future. Therefore it is important to describe/understand the FAST concepts which the Xetra Market Data Interface is based on.

12.1 FIX Adapted for Streaming℠ (FAST Protocol℠)

FAST makes use of proven data compression techniques that take advantage of the knowledge about the data content and the data formats.

For the latest descriptions about FAST, please refer to

www.fixtrading.org > Standards > FIX Standards > FIX Adapted for Streaming (FAST)

FAST proposes a two-step data compression method instead of a generic compression technique. The compression techniques are customized to the data content and minimize the repetition of fields to ensure that only the required bytes are used for the transmission of data. These techniques can be applied not only to FIX messages but to messages in any format.

Please note that this document does not contain the complete FAST specification but only explains the concepts relevant to the Xetra Market Data Interface.

12.1.1 Commonly Used Terms in FAST Context Explained

Term	Explanation
Application type	Represents the type of a group or a message in applications using FAST.
Field Instructions	Each field instruction has a name and a type. The name must be unique within the group. The type can be either primitive or sequence or group. Example: <uInt32 name="**sendingtime**" />
Group	A group is a named type comprising an unordered set of fields. A group appearing at the top most level of a feed is also referred to as a message.

Term	Explanation
Instructions	There are two categories of instructions: • Field instructions specify how to encode fields of the instance to the feed. • Template reference instructions provide the means for defining parts of a template by reference to other templates. Encoding and decoding are performed within the context of an instruction. An instruction context consists of: • set of templates • current template • set of application types • current application type • set of dictionaries • optional initial value Each field instruction has a name and a type. The name identifies the context of the application type. The optional presence attribute indicates whether the field is mandatory or optional. If the attribute is not specified, the field is mandatory.
Presence map	A sequence of bits. Fields of the presence map segment use the bits as specified by the current template. Not every field needs a bit in the presence map (see FAST documentation). When present, bit n indicates whether the corresponding field in the template is sent (‘1’) or not (‘0’).
Primitive field	A field that is not a group or sequence and can have a field operator. This operator specifies the optimization operation for the field.
Primitive type	Are ASCII strings, Unicode string, uInt32, int32, uInt64, int64, decimal (8 bytes mantissa plus 4 bytes exponent) and byte vector.
Segments	Consist of the following: • segment identifier • presence map • set of field instructions In general, a segment is a complete set of information. A FAST message is the highest level segment. A segment stop bit is used to link data bytes together belonging to a field. This is the Most Significant Bit (MSB) of a data byte. If this is unset (0), it indicates that the following data byte is part of the current field and if set (1), it indicates that the current byte is the last byte of the data field.

Term	Explanation
Sequence	Comprises a length and an ordered set of elements. Each element is of group type and must not have identical group types.
Template	Specifies how to encode an instance of an application type or a part thereof as a feed of bytes.

12.2 Field Instructions

Field instructions refer to how the fields will be sent in the information feed. For further information, please refer to the FAST Specifications Version 1.1 at

www.fixtrading.org > Standards > FIX Standards > FIX Adapted for Streaming (FAST)

12.3 Field Operators

Field operators are used to remove the existing redundancies in the data formats. Message data fields can be related to one another. Message templates will be the meta-data for the message and will be provided via public download. When the messages arrive, the receiver application will have a complete knowledge of the message layout and needs to work only with the template and the incoming message to determine the field value.

Field instructions can optionally have operators which specify how the receiver program should determine the value of the field.

For further information, please refer to section 6 of the FAST Specification Version 1.1.

12.4 Transfer Encoding

Transfer Encoding refers to how the fields will be encoded in an information feed.

For further information, please refer to section 10 of the FAST Specification Version 1.1.

13 Appendix – The Xetra Member Homepage

On the member section of the Xetra Member Homepage **www.deutsche-boerse.com** you will find the following information.

13.1 Xetra Market Data Interface – Multicast Addresses

An Excel sheet with reference data for Xetra Market Data Interface, i.e. the available feeds and their IP addresses (Maintenance feed, Ticker feed, Market Feeds) is provided. The currently provided market depth per market feed is also contained in this document.

13.2 Xetra Market Data Interface – Templates, Final Version

The FAST templates for Xetra Market Data Interface in XML format.
Please visit there:

Trading & Clearing > Member sections > Xetra Member Section
> Xetra/Floor > Xetra Documentation > Xetra 17.0
> Interfaces > Xetra Market Data Interface

13.3 Instrument Reference Data file in ASCII format

For production:
Trading & Clearing > Member sections > Xetra Member Section
> Xetra Information > Xetra Instruments

For simulation:
Trading & Clearing > Member sections > Xetra Member Section
> Xetra/Floor > Xetra Simulation > Instrumente

Xetra publishes a daily updated file including all tradable instruments on Xetra on its Xetra website: http://www.xetra.com/xetra-en/instruments/instruments