

# **Xetra Release 16.0**

## **Enhanced Broadcast Solution – Interface Specification**

### **Final Version**

© Deutsche Börse AG

All proprietary rights and interest in this Xetra® publication shall be vested in Deutsche Börse AG and all other rights including, but without limitation to, patent, registered design, copyright, trade mark, service mark, connected with this publication shall also be vested in Deutsche Börse AG. Whilst all reasonable care has been taken to ensure that the details contained in this publication are accurate and not misleading at the time of publication, no liability is accepted by Deutsche Börse AG for the use of information contained herein in any circumstances connected with actual trading or otherwise. Neither Deutsche Börse AG, nor its servants nor agents, is responsible for any errors or omissions contained in this publication which is published for information only and shall not constitute an investment advice. This brochure is not intended for solicitation purposes but only for the use of general information. All descriptions, examples and calculations contained in this publication are for guidance purposes only and should not be treated as definitive. Deutsche Börse AG reserves the right to alter any of its rules or product specifications, and such an event may affect the validity of information contained in this publication.

® Registered trademark of Deutsche Börse AG

---

---

## Table of Contents

<b>1</b>	<b>Introduction</b>	<b>7</b>
<b>1.1</b>	<b>Changes in Xetra Release 16.0 in comparison to Xetra Release 15.0</b>	<b>8</b>
1.1.1	Software related changes	8
1.1.1.1	Packet Header	8
1.1.1.2	Instrument Reference Data	8
1.1.1.3	Order Book	8
1.1.1.3.1	Delta / Incremental	8
1.1.1.3.2	Snapshot	8
1.1.1.3.3	All Trade Price Stream	8
1.1.2	Documentation related changes	8
<b>2</b>	<b>Infrastructure Requirements</b>	<b>10</b>
<b>2.1</b>	<b>Hardware and Network Infrastructure</b>	<b>10</b>
<b>2.2</b>	<b>Software Infrastructure</b>	<b>10</b>
<b>3</b>	<b>Overview</b>	<b>11</b>
<b>3.1</b>	<b>Definition of Commonly Used Terms</b>	<b>11</b>
<b>3.2</b>	<b>Disseminated Market Information</b>	<b>12</b>
3.2.1	Exceptions	13
<b>4</b>	<b>Broadcast Streams</b>	<b>14</b>
<b>4.1</b>	<b>Reference Data Stream</b>	<b>15</b>
<b>4.2</b>	<b>Snapshot Stream</b>	<b>15</b>
<b>4.3</b>	<b>Delta/Incremental Stream</b>	<b>15</b>
<b>4.4</b>	<b>All Trade Price Stream</b>	<b>15</b>
<b>4.5</b>	<b>State Changes Stream</b>	<b>16</b>
<b>4.6</b>	<b>Interface Schedule</b>	<b>16</b>
<b>5</b>	<b>Structure of Messages</b>	<b>17</b>
<b>5.1</b>	<b>Structure of Service Messages</b>	<b>19</b>
5.1.1	Packet Header Message	19
5.1.2	Functional Beacon Message	20
5.1.3	Technical Beacon Message	20
5.1.4	Start/End of Services	21
5.1.5	Start/End of Instrument/Maintenance Reference Data Information	21
<b>5.2</b>	<b>Structure of Data Messages</b>	<b>21</b>
5.2.1	Timestamp Information	21
5.2.2	Reference Information	24
5.2.2.1	Instrument Reference Data	24
5.2.2.2	Maintenance Reference Data	27
5.2.3	Order Book Information	28
5.2.3.1	Order Book Snapshot	28
5.2.3.2	Order Book Delta/Incremental	31
5.2.4	Trade Information	35
5.2.4.1	All Trade Price Message	35
5.2.4.2	Trade Reversals	36
5.2.5	State Changes	36

---

---

5.2.5.1	State Change Message	37
5.2.5.2	Instrument State Change Message	37
5.2.5.3	Technical Market Reset Message	37
5.2.5.4	Cross Request Message	38
<b>6</b>	<b>Appendix – Glossary of Terms</b>	<b>39</b>
<b>7</b>	<b>Appendix – Data Field Dictionary</b>	<b>41</b>
<b>7.1</b>	<b>Action Code Type</b>	<b>41</b>
<b>7.2</b>	<b>Address Type</b>	<b>41</b>
<b>7.3</b>	<b>BonAcrIntCalcMd</b>	<b>41</b>
<b>7.4</b>	<b>BonCrtCpnDat Type</b>	<b>42</b>
<b>7.5</b>	<b>BonCurPoolFact Type</b>	<b>42</b>
<b>7.6</b>	<b>bonFlatInd Type</b>	<b>43</b>
<b>7.7</b>	<b>bonMrtyDat Type</b>	<b>43</b>
<b>7.8</b>	<b>bonNxtCpnDat Type</b>	<b>43</b>
<b>7.9</b>	<b>bonYldTrdInd Type</b>	<b>43</b>
<b>7.10</b>	<b>Book Type</b>	<b>44</b>
<b>7.11</b>	<b>Business Date Type</b>	<b>44</b>
<b>7.12</b>	<b>clgLoc Type</b>	<b>44</b>
<b>7.13</b>	<b>Cmex Indicator Type</b>	<b>45</b>
<b>7.14</b>	<b>coupon rate Type</b>	<b>45</b>
<b>7.15</b>	<b>Currency Code Type</b>	<b>45</b>
<b>7.16</b>	<b>dnomCurrCod Type</b>	<b>45</b>
<b>7.17</b>	<b>endTickPriceRange Type</b>	<b>46</b>
<b>7.18</b>	<b>Entry Level Type</b>	<b>46</b>
<b>7.19</b>	<b>entryType</b>	<b>46</b>
<b>7.20</b>	<b>Exchange Id Type</b>	<b>48</b>
<b>7.21</b>	<b>Exchange Segment Identifier Type</b>	<b>48</b>
<b>7.22</b>	<b>frstTrdDat Type</b>	<b>49</b>
<b>7.23</b>	<b>General Price Type</b>	<b>49</b>
<b>7.24</b>	<b>homeMkt Type</b>	<b>49</b>
<b>7.25</b>	<b>Indicator Type</b>	<b>50</b>
<b>7.26</b>	<b>Indicator Collection Type</b>	<b>50</b>
<b>7.27</b>	<b>Instrument Group Type</b>	<b>51</b>
<b>7.28</b>	<b>Instrument Status Type</b>	<b>51</b>
<b>7.29</b>	<b>Instrument Type</b>	<b>53</b>
<b>7.30</b>	<b>instShtNam Type</b>	<b>53</b>
<b>7.31</b>	<b>instSubTypCod Type</b>	<b>53</b>
<b>7.32</b>	<b>inSubscription Type</b>	<b>54</b>
<b>7.33</b>	<b>ISIN Code Type</b>	<b>54</b>
<b>7.34</b>	<b>isix Type</b>	<b>54</b>
<b>7.35</b>	<b>issrMemblId Type</b>	<b>54</b>
<b>7.36</b>	<b>issrSubGrp Type</b>	<b>55</b>

---

---

7.37	issueDat Type	55
7.38	knockOutInd Type	55
7.39	lglMktSegCod Type	56
7.40	licensetype Type	57
7.41	lstTrdDat Type	57
7.42	Market Order Interruption Indicator Type	57
7.43	maxOrdrValBest Type	57
7.44	maxSrpQty Type	58
7.45	minHiddOrdrVal Type	58
7.46	minIceQty Type	58
7.47	minMidPntOrdrVal Type	58
7.48	minOrdrSiz Type	59
7.49	minPeakQty Type	59
7.50	vdoMinExecType	59
7.51	minTrdbUnitType	59
7.52	mktSegCod Type	60
7.53	Mnemonic Type	61
7.54	Number of Trades Type	61
7.55	Number of Values Type	61
7.56	OptiGatewayLocId Type	61
7.57	PerformanceIndicator	62
7.58	Port Type	62
7.59	postTrdAty Type	62
7.60	Price Action Type	63
7.61	priceTypeCod Type	63
7.62	Quantity Type	63
7.63	referencePrice Type	64
7.64	refMkt Type	64
7.65	reportingMarket Type	64
7.66	rondLotQty Type	64
7.67	SendingTime Type	65
7.68	Sequence Number Type	65
7.69	Set Id Type	65
7.70	setlCal Type	66
7.71	setlCurrCod Type	66
7.72	setlPeriodFlg Type	66
7.73	Source Id Type	66
7.74	specMemblId Type	67
7.75	specSubGrp Type	67
7.76	sprdFact Type	67
7.77	sprdMinQty Type	68
7.78	sprdTypCod Type	68

---

---

<b>7.79</b>	<b>State Type</b>	<b>68</b>
<b>7.80</b>	<b>Stream Depth Type</b>	<b>69</b>
<b>7.81</b>	<b>Stream Service Type</b>	<b>69</b>
<b>7.82</b>	<b>Stream Type</b>	<b>70</b>
<b>7.83</b>	<b>Template Id Type</b>	<b>70</b>
<b>7.84</b>	<b>tickIncrement Type</b>	<b>71</b>
<b>7.85</b>	<b>Timestamp Type</b>	<b>72</b>
<b>7.86</b>	<b>tradCal Type</b>	<b>72</b>
<b>7.87</b>	<b>Transaction ID Type</b>	<b>72</b>
<b>7.88</b>	<b>trdMdlTypCod Type</b>	<b>73</b>
<b>7.89</b>	<b>UnderlyingSecurityID Type</b>	<b>73</b>
<b>7.90</b>	<b>untOfQuotn Type</b>	<b>73</b>
<b>7.91</b>	<b>Vola Indicator Type</b>	<b>73</b>
<b>7.92</b>	<b>warSeg Type</b>	<b>74</b>
<b>7.93</b>	<b>warStrPrc Type</b>	<b>75</b>
<b>7.94</b>	<b>warTyp Type</b>	<b>75</b>
<b>7.95</b>	<b>wknNo Type</b>	<b>75</b>
<b>7.96</b>	<b>xetraIssrMnem Type</b>	<b>75</b>
<b>8</b>	<b>Appendix - Interface Limits</b>	<b>77</b>
<b>9</b>	<b>Appendix - How to Use</b>	<b>78</b>
<b>9.1</b>	<b>Start of Day</b>	<b>78</b>
9.1.1	Connect to the Reference Data Stream	78
9.1.2	Receive Snapshot	78
9.1.3	Process Delta	79
<b>9.2</b>	<b>Messages Sequence Numbers Synchronization</b>	<b>79</b>
9.2.1	Snapshot - Delta Synchronization	79
9.2.2	All Trade Price Synchronization	81
9.2.2.1	All Trade Price Stream	81
9.2.2.2	Delta – All Trade Price Synchronization	81
9.2.2.3	Snapshot – All Trade Price Synchronization	81
<b>9.3</b>	<b>Order Book Recovery</b>	<b>81</b>
9.3.1	Loss of Packets	82
9.3.1.1	Receive Snapshot for an Instrument	82
9.3.1.2	Leave the Snapshot Stream	82
9.3.1.3	Continue Operation with the Delta Stream	82
<b>9.4</b>	<b>Events Demanding Special Response</b>	<b>82</b>
9.4.1	Gap Indicator	82
9.4.2	Trade Gap Indicator	83
9.4.3	Source Identifier	83
9.4.4	Sequence Number	84
9.4.5	Start/End of Service Message	84
9.4.6	Host Fail-over	84
9.4.7	Market Reset	86
<b>10</b>	<b>Appendix - Message Encoding</b>	<b>87</b>
<b>10.1</b>	<b>FIX Adapted for SStreaming<sup>SM</sup> (FAST Protocol<sup>SM</sup>)</b>	<b>87</b>

---

---

10.1.1	Commonly Used Terms in FAST Context Explained	87
<b>10.2</b>	<b>Field Instructions</b>	<b>89</b>
<b>10.3</b>	<b>Field Operators</b>	<b>89</b>
<b>10.4</b>	<b>Transfer Encoding</b>	<b>89</b>
<b>11</b>	<b>Appendix – The Xetra Member Homepage</b>	<b>90</b>

---

---

## 1 Introduction

Xetra disseminates market information to members through a high performance, functionally enhanced broadcast interface known as the Xetra Enhanced Broadcast Solution. The Enhanced Broadcast Solution interface is a low latency solution with a highly granular dissemination model and is based on the concepts of the FIX (Financial Information eXchange) Protocol. Members are able to effectively use the Enhanced Broadcast Solution interface to develop applications that fit their requirements.

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 Enhanced Broadcast Solution interface.

### General Introduction

- Chapter 2 “Infrastructure Requirements” highlights the basic infrastructure requirements for accessing the services offered by the Enhanced Broadcast Solution interface.
- Chapter 3 “Overview” provides an overview of the Enhanced Broadcast Solution interface explaining some commonly used terms.

### Enhanced Broadcast Solution Interface Structure

- Chapter 4 “Broadcast Streams” introduces the Enhanced Broadcast Solution interface streams and describes the interface data dissemination schedule during a Xetra business day.

### Message Structures

- Chapter 5 “Structure of Messages” explains the Enhanced Broadcast Solution interface message structures in detail and describes the classification of the disseminated messages.

**Appendices** provide additional information on the Enhanced Broadcast Solution 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 – Interface Limits” provides a list of system limits for the Enhanced Broadcast Solution interface.
  - Chapter 9 “Appendix – How to Use” covers the description of the operation of an example client system using the Enhanced Broadcast Solution interface.
  - Chapter 10 “Appendix – Message Encoding” describes the formats used for the encoding of data fields.
  - Chapter 11 “Appendix – The Xetra Member Homepage” tells where to get the FAST message XML templates and further information from.
-

---

## **1.1 Changes in Xetra Release 16.0 in comparison to Xetra Release 15.0**

### **1.1.1 Software related changes**

#### **1.1.1.1 Packet Header**

Template ID (TID) is changed to 34.

#### **1.1.1.2 Instrument Reference Data**

The Volume Discovery Minimum Execution Quantity “vdoMinExec” field is added.

The Matching Instruction Cross Id Indicator “MatchInstCrossIdInd” is added to the indicator collection.

#### **1.1.1.3 Order Book**

##### **1.1.1.3.1 Delta / Incremental**

The canceled quantity of a resting order due to a prevented in house crossing (aka self match prevention) is delivered in the sequence “EntriesTradePrices” member of the sequence “EntriesTrade”

##### **1.1.1.3.2 Snapshot**

None

##### **1.1.1.3.3 All Trade Price Stream**

None

### **1.1.2 Documentation related changes**

All instrument status types belonging to the removed trading model block crossing – QPREC,QPRCX,QPRCC,QPRZ, QCALL,QIPO,QPOBB,QOBB – are removed from the description of the instrument status type (7.28).

New legal market segments are added to the lglMktSegCod Type (7.39),

060 – Budapest Stock Exchange MTF

318 – Budapest Stock Exchange Regulated Market

735 – Prague Stock Exchange Official Market

736 – Prague Stock Exchange MTF

738 – Prague Stock Exchange Free Market.

Please note: Further additions to lglMktSegCod can be made at a later point in time. The application must be able to handle codes which are not listed.

New market segment codes are added to mktSegCod Type (7.52)

PRX – PSE Qualified investors funds

PRY – PSE Collective investment funds

Please note: Further additions to mktSegCod can be made at a later point in time. The application must be able to handle codes which are not listed.

---



---

Order Book Snapshot: For instruments traded in continuous auction model, the Instrument trading phase is displayed.

Order Book Delta, Incremental: Allow MPO\_PRC, BEST\_PRC as entryTypes

---

## **2 Infrastructure Requirements**

### **2.1 Hardware and Network Infrastructure**

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

Members subscribing to the information can contact the Xetra Network Support 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 Enhanced Broadcast Solution 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 Enhanced Broadcast Solution can be based on any platform capable of receiving multicasts. Members will have the opportunity to develop and use specific applications that fit their requirements.

---

---

### 3 Overview

The Xetra Enhanced Broadcast Solution is a flexible, transparent, UDP based interface which disseminates market information to members over a multicast network. The information disseminated by the Enhanced Broadcast Solution interface includes order book, statistical and system status information.

The messaging protocol used by the Enhanced Broadcast Solution interface is based on the concepts of the FIX (Financial Information eXchange) Protocol but has message formats customized to fit the Xetra business requirements. The Enhanced Broadcast Solution interface will conform to the FIX Adapted for Streaming (FAST) protocol principles for efficient bandwidth utilization.

The interface provides members with the information in form of broadcast streams. From these broadcast streams, 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 streams they have joined.

The Enhanced Broadcast Solution interface will be supported in parallel to the existing public market data dissemination over VALUES and to the multicast Market Data Interface, which are customized for low bandwidth usage.

#### 3.1 Definition of Commonly Used Terms

Following are definitions of terms frequently used in the context of the Xetra Enhanced Broadcast Solution.

- Enhanced Broadcast Solution. The new broadcast concept bases on Multicast Infrastructure.
  - Member/Exchange Participant. The term “Member” refers to an organization that is authorized by Deutsche Börse AG, Eurex Bonds or a Partner Exchange to receive the services offered through the Enhanced Broadcast Solution 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.
  - Channels. Xetra offers multiple market depths for every instrument. Every channel offers a particular order book depth range that members can join.
  - Streams. 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 stream. The system has five kinds of streams, the reference stream, the three available streams for market data (snapshot, delta/Incremental and All Trade Price) and the system state changes stream.
  - Delta/Incremental Broadcasts. They are provided for a market-affecting event (such as order entry or a trade). Only changes to the market are broadcast through this stream.
  - Snapshot broadcasts. A snapshot broadcast contains all current market information for a given instrument, regardless of recent movements and will be sent on a different stream in addition to the “real time” delta broadcasts.
-

Snapshot broadcasts will be sent less frequently than the delta broadcasts. Members should join this stream just long enough to receive one snapshot enabling them to establish a baseline market picture. After establishing the baseline market picture, members should leave the snapshot broadcasts stream and rely on delta broadcasts to maintain it. Staying joined to the snapshot stream consumes larger bandwidth but gives no additional information. The frequency of dissemination is calculated from the network load condition in real time.

The expected member application behaviour is explained in chapter 9, Appendix - How to Use.

- All Trade Price broadcasts. The whole Xetra trade price chain, including normal order book trades, midpoint and Xetra BEST trades will be disseminated over the All Trade Price stream.
- State changes broadcast. These broadcasts inform the receiver of any exchange or system state change.
- Cross Request broadcasts. The cross request broadcast, disseminated via the state change stream, announces a possible cross trade.
- Joining. Joining a stream 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 stream 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<sup>SM</sup> (FAST Protocol <sup>SM</sup>). 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 snapshot stream in form of order book snapshots. The Xetra Enhanced Broadcast Solution 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 delta/incremental messages since the snapshot messages are always sent a little bit later. If the packet carrying the delta information does not reach the member system, the information should be considered as lost. In most cases though, 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.

## 3.2 Disseminated Market Information

### Reference Data Information

Instrument reference data information messages provide a description of all the available instruments for which the Enhanced Broadcast Solution interface will broadcast price updates. Instrument reference data information will be available from a pre-defined multicast address and ports and contains information on the granularity of the data offered for each instrument and the corresponding multicast addresses and ports for the channels.

Maintenance reference data messages provide the multicast address of the state changes stream.

---

---

## Order Book and Statistical Information

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.

- **Snapshot.** Snapshots contain complete order book information up to the depth indicated in the reference information. The snapshot message should be used only for the creation of the market picture at the beginning of a trading day and for its recovery in case of a data loss.  
Snapshots provide information about the actual instrument status, the details of the last trade and the days' statistical information for the instrument.
- **Delta/Incremental.** 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.

Under normal conditions members will be able to maintain their copy of the order book by joining the delta broadcast stream and applying the received messages.

More information about maintaining the global market picture can be found in chapter 9, Appendix - How to Use.

### Trade Information

Trade Price Information: The Xetra All Trade Price broadcast stream contains the Xetra trade price information excluding OTC trade prices. All Trade Price messages will be disseminated over the All Trade Price stream. The complete trade price information is also sent via the delta stream.

### Status Information

System and Exchange state change. This stream will inform the members of a system or exchange state change.

State change messages will not be sent per instrument. The instrument status changes throughout the day are received via the snapshot and the delta stream.

### Ticker data

The Ticker data stream is not a part of Enhanced Broadcast Solution, but a part of Market Data Information service (MDI). MDI is available to all users of Enhanced Broadcast Solution. Please lookup MDI manual to get more information.

## 3.2.1 Exceptions

Listed below are some rules or exceptions pertaining to the data content published by the Enhanced Broadcast Solution interface.

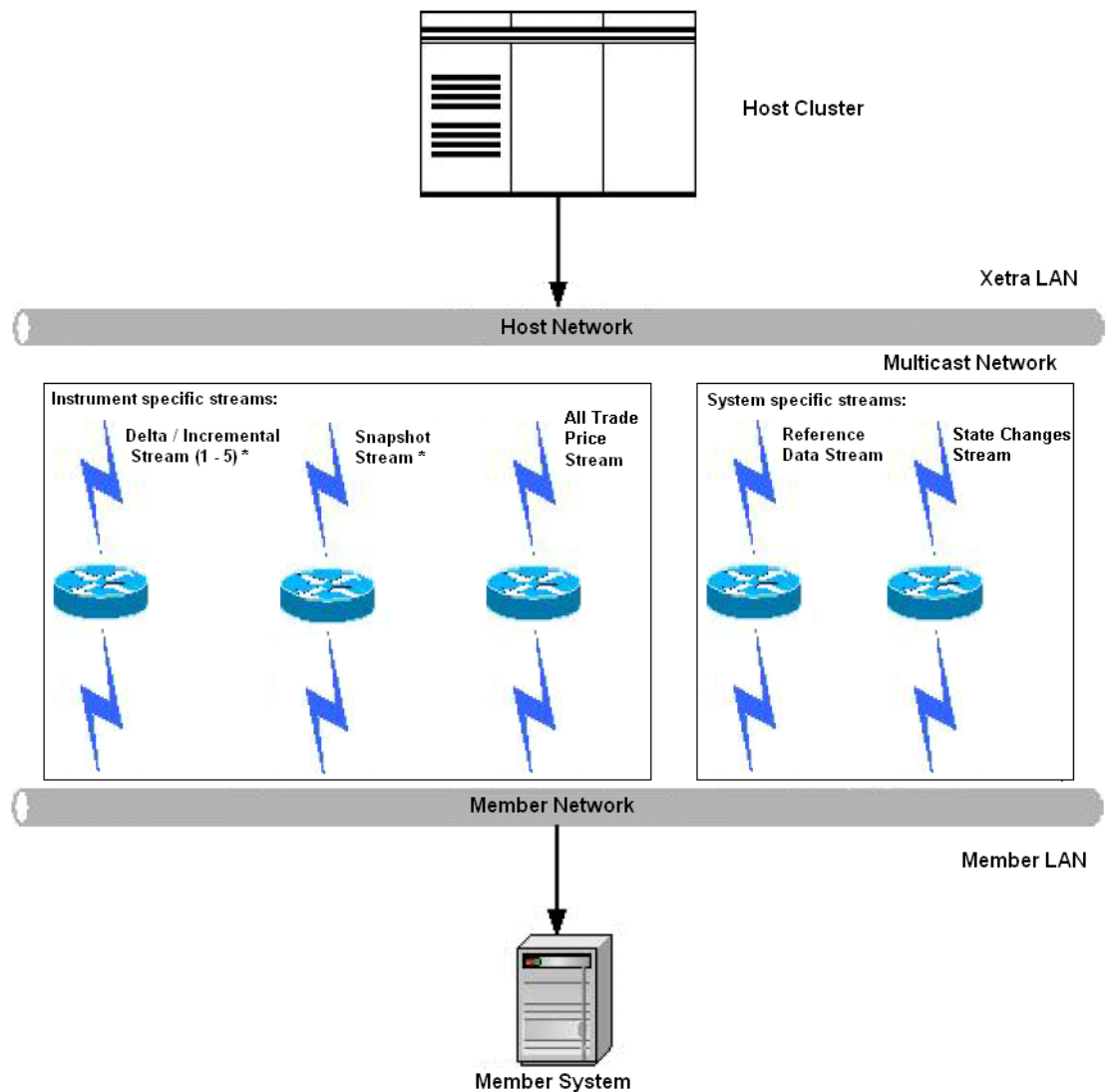
- Order book rules for auctions: Order book information disseminated during auctions depends on the instrument set-up. Please find more details in the corresponding Xetra market model documents.
  - Gap indication: The Enhanced Broadcast Solution interface will normally disseminate un-netted price information. However, in exceptional situations, e.g., due to heavy processing load some of the price level changes may not be sent. For these situations a gap indicator will be sent in the next message.
-

## 4 Broadcast Streams

The Xetra Enhanced Broadcast Solution interface will disseminate information in five kinds of streams in a live-live concept meaning that the data will be disseminated in separate streams over two services called service A and service B.

These five streams are:

- Reference Data. This stream is the first one to be connected to since it disseminates the addresses for all the other streams.
- Snapshot
- Delta / Incremental
- All Trade Prices
- State changes



\* can differ depending on the instrument

---

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. Every stream will consist of one or more channels and will follow certain rules for dissemination of data.

#### **4.1 Reference Data Stream**

Member applications have to join the reference data stream at the start of each business day. To make the bandwidth available for other streams this stream should be unsubscribed once all the reference data has been received. Each market source has its own reference data stream.

The reference data stream disseminates two types of information, instrument reference data and maintenance reference data. Both provide reference information for configuration data. This data is disseminated the whole day cyclically. The reference data is deemed as static for a business day.

The instrument reference information provided by this stream contains the multicast channel information (i.e. multicast addresses and ports) of the snapshot, delta and All Trade Price streams. The maintenance reference data provided by this stream contains the multicast channel information (i.e. multicast addresses and ports) of the state changes stream. Members must, therefore, join these streams and receive all necessary information before being able to join the state changes stream.

#### **4.2 Snapshot Stream**

Members should join this stream to receive complete order book snapshots. The information provided by this stream needs to be used for creating the base order book structure. After snapshots for all instruments have been received, members should quit this stream. This stream can also be used for recovery purposes in case of a data loss.

Order book snapshots carry the sequence number for the delta/incremental channel, providing the receiver with sufficient information to position the snapshots exactly amongst the delta messages.

Snapshot information will be available for all instruments throughout the business day but should be joined only for starting, unless there is a data loss.

#### **4.3 Delta/Incremental Stream**

Members should remain connected to this stream throughout the trading day for receiving delta messages. All incoming delta messages should be applied to the copy of the order book maintained by the member applications in order to have the latest information.

#### **4.4 All Trade Price Stream**

The Xetra All Trade Price stream is the Xetra public broadcast data stream for dissemination of information about every single trade.

Members should remain connected to this stream throughout the trading day for receiving All Trades Prices messages.

---

## 4.5 State Changes Stream

The state change messages contain the system, exchanges and special instrument states. The current system and exchange states are sent cyclically and event-driven. An event occurs as soon as an exchange or system change happens. The special instrument states are suspend/unsuspend, knocked out/knocked out revoke and midpoint book freeze/unfreeze.

Members should remain connected to this stream throughout the trading day for receiving the system and exchange State Change messages.

## 4.6 Interface Schedule

The Enhanced Broadcast Solution interface will be based on a “push only” architecture and the data will be disseminated in streams consisting of one or more multicast channels. Member applications will have to join a particular multicast address and port combination for connecting to a stream. Whenever the interface sends data over a stream, all client applications joined to it will receive these data packets.

Please note that the information regarding streams and channels which member applications have to join for receiving market information, will be provided by the reference data stream.

A mapping between the trading phases and the functional timeline of the Enhanced Broadcast Solution Interface is shown below:

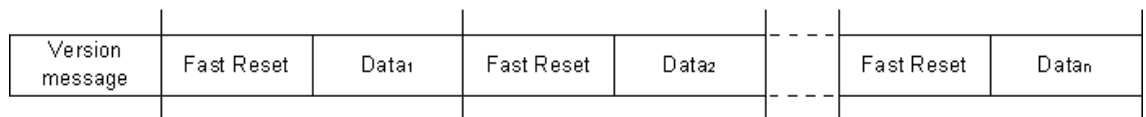
Xetra Exchange Period	Xetra Enhanced Broadcast Solution	Member System
Start of business day	Start Xetra Enhanced Broadcast Solution	
	Issue start of service message. Start the instrument reference data and maintenance reference data stream.	Start of Trading day. Join the reference data stream and store the received instrument and maintenance reference data.
	Start snapshot stream	Join the snapshot stream and build the base order book for all instruments.
	Start Delta/Incremental stream	Join the Delta/Incremental channel <sup>1</sup> . Leave the snapshot stream and apply incoming deltas to the order book. Capture the non-incremental messages.
	Start All Trade Price stream	Join the All Trade price stream to receive the trades
	Start State Changes stream	Join the State Changes stream to receive the system and exchange state changes
Trading Phases: Pretrade Pre-open .....	Continue broadcasting the information...	Receive the disseminated information.
Post end-of-day-processing	Issue end of service message. Service is brought down	Leave the streams. End of trading day

<sup>1</sup> If any gaps are detected in the sequence numbers on the delta stream, member applications can either connect to the snapshot stream and receive the latest order book snapshot or wait for the next delta message for this instrument to rebuild their copy of the order book. Member applications must be prepared for the fact that multicast does not guarantee the correct sequence of messages.

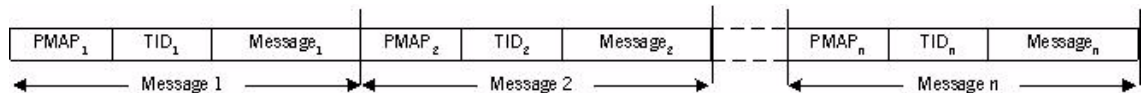


## 5 Structure of Messages

The Xetra Enhanced Broadcast Solution disseminates data in UDP datagrams. Every UDP datagram sent through the Enhanced Broadcast Solution will have the structure described in the diagram below:



Data consists of following blocks:



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.

All messages, all field data types and encoding formats disseminated by the Enhanced Broadcast Solution interface conform to the FAST standard 1.1.

Regarding FAST:

- Enhanced Broadcast Solution uses global dictionary scope for FAST operators. 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.

### Presence Map (PMAP)

The presence map is a bit combination indicating the presence or absence of a field in the message body. The allocation of a bit for a field in the presence map is governed by the FAST field encoding rules<sup>2</sup>.

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." (non-applicable).

In templates except for Instruments Reference Data and Maintenance Reference Data delivered via the Reference data stream, the following should be taken into account:

- Repeating groups will have no presence map, except in case of the instrument and maintenance reference data messages. The number of entries will always be sent, either with the actual number of entries or set to zero if there is no data in the repeating group to be sent.
- The FAST operator will be "none" for fields inside repeating groups that are not "delta", since any other value will imply the need of a presence map.

### Template Identifier (TID)

The template identifier is an integer field to inform which template should be used for decoding the message.

<sup>2</sup> see section 10.5 of the FAST specification document at <http://www.fixprotocol.org/documents/3066/FAST%20Specification%201%20x%201.pdf>

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 in Eurex and to FAST rules as to template numbering<sup>3</sup>.

Message	Template ID
Packet Header Message	34
Beacon Message	2
Instrument Reference Data	3
Maintenance Reference Data	4
Snapshot Message	6
Delta/Incremental Message	7
All Trade Price Message	9
State Changes Message	10
Cross Request Message	11
Market Reset Message	12
Instrument suspend/unsuspend, knock out/revoke	13
Start of Service Message	128
End of Service Message	129
Start of Instrument Reference Data Message	130
End of Instrument Reference Data Message	131
Start of Maintenance Reference Data Message	132
End of Maintenance Reference Data Message	133

The template with TID= 120 is not included in the FAST Message Template 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 and it is not necessary to mention it in the template file. Nevertheless, the Reset Message is sent out, and the dictionary must be initialized<sup>4</sup>.

### Message (Body)

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

Some special comments that should not be forgotten:

- The SendingTime in the Packet Header message contains the time since midnight in Central European Time (CET/CEST) and will be formatted in microseconds.

---

<sup>3</sup> To learn more about the template identifiers rules, read appendix 2.2 of the following FAST document:  
<http://www.fixprotocol.org/documents/3795/FAST%20Session%20Control%20Protocol%201.1.doc>

<sup>4</sup> Standard FAST template:

<template name="Reset" scp:reset="yes" id="120"><typeRef name="Reset"/></template>

---

- All other Timestamps contain the time since midnight in Central European Time (CET/CEST) and will be formatted in microseconds, unless otherwise specified. Member systems can use these timestamps and source identifiers along with the sequence numbers (wherever sent) to order messages.
- The Xetra Enhanced Broadcast Solution maintains the sequence numbers per instrument per source.
- Price and quantity fields are sent as scaled numbers with a mantissa of 8 bytes and an exponent of 4 bytes.
- The sequence of the fields within a message is defined by functional reasons, the sequence of the fields in the message, sent over the wire, is defined by technical reasons in the delivered xml file. It is strongly recommended to use the xml-files for developing decoders.

## 5.1 Structure of 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.

### 5.1.1 Packet Header Message

The Xetra Enhanced Broadcast Solution packet header (“version information message”) will be sent in every UDP datagram.

The previously named “version information message “will be named “packet header” to be in sync with other multicast based interfaces of Deutsche Börse Group.

Presence Map	Template ID	SenderCompID	PacketSeqNum	Sending Time	Performance Indicator
1 byte	1 byte	1 byte	5 bytes	6 bytes	4 bytes

Additionally field names are harmonized the following way:

PACKET HEADER MESSAGE STRUCTURE		
Field Name	Descriptive Name	Field Format
TID	Template ID	Template ID type (7.83); fixed value to indicate that this is the message header.
senderCompID	Source identifier	Source ID type (7.73); Identifies the Xetra broadcast instance from which this packet is originated.

PACKET HEADER MESSAGE STRUCTURE		
Field Name	Descriptive Name	Field Format
packetSeqNum	Sequence number of the datagram	Sequence number type (7.68); Byte vector encoded packet sequence number; does not fall back intraday; applications can use this field combined with the senderCompld to determine whether the packet in question has already been received and hence discard duplicate packets quickly without having to first decode them. Byte vector encoded.
sendingTime	Time stamp	SendingTime type (7.67); Byte vector encoded time at which this packet left the Xetra host, specified in microseconds since midnight
performanceIndicator	Performance Indicator	PerformanceIndicator type (7.57) Xetra host performance indicator, byte vector encoded.

Please note that this datagram sequence number cannot be used to detect data loss on the application level. The receiver application has to use the sequence numbers inside the messages for this purpose.

The packet header ID will be transported with template ID 34.

### 5.1.2 Functional Beacon Message

Functional beacon messages indicate the availability of exchange services. They contain the exchange timestamp and the last sequence number sent for the channel and source. These messages are sent for the delta stream.

BEACON MESSAGE STRUCTURE		
Field Name	Descriptive Name	Field Format
TID	Template ID	Template ID type (7.83)
timestamp	Timestamp	Timestamp type (7.85); generation time of message
srcId	Source identifier	Source ID type (7.73)
SeqNum	Sequence number	Sequence number type (7.68); sequence number of last delta for this isix
isix	Instrument identifier	Isix type (7.34)

### 5.1.3 Technical Beacon Message

Technical beacon message is sent out periodically on every multicast address. The last two digits of the port number are always 96. One cycle duration does not exceed 120 seconds. The technical beacon message consists of the Packet Header Message (TID=34).

The presence of the technical beacon message does not assure that the prices are being sent using the defined host; the intention is to keep the PIM sparse routing trees alive.

The technical beacon message should only be used as an indication that the respective Xetra broadcast host is available.

#### 5.1.4 Start/End of Services

The start and end of service messages are sent by the reference data stream and can be used by member systems to synchronize their services with that of the interface.

START/END OF SERVICE MESSAGE STRUCTURE		
Field Name	Descriptive Name	Field Format
TID	Template ID	Template ID type (7.83)
timestamp	Timestamp type	Timestamp type (7.85)
busDate	Business date	Business date type (7.11)

Start of service messages are generally only sent once during a business day. For technical reasons, they may occasionally occur more than once per day, and client applications must be able to handle this by simply ignoring such messages.

#### 5.1.5 Start/End of Instrument/Maintenance Reference Data Information

The start/end of instrument and maintenance reference information will be disseminated by the instrument and maintenance reference data stream to indicate the start and end of the reference information.

START/END OF ALL REFERENCE MESSAGES STRUCTURE		
Field Name	Descriptive Name	Field Format
TID	Template ID	Template ID type (7.83)
timestamp	Timestamp type	Timestamp type (7.85)
busDate	Business date	Business date type (7.11)
noOfMsgs	Number of messages	Number of values type (7.55)

## 5.2 Structure of Data Messages

### 5.2.1 Timestamp Information

This section provides a list of each timestamp field in the Enhanced Broadcast Solution; it describes the measure point and explains what is being measured. All timestamps contain the elapsed time since midnight in Central European Time (CET)/Central European Summer Time (CEST) and are formatted in **microseconds**.

The following picture shows all timestamps on Xetra for requests/responses for orders and quotes and execution messages:

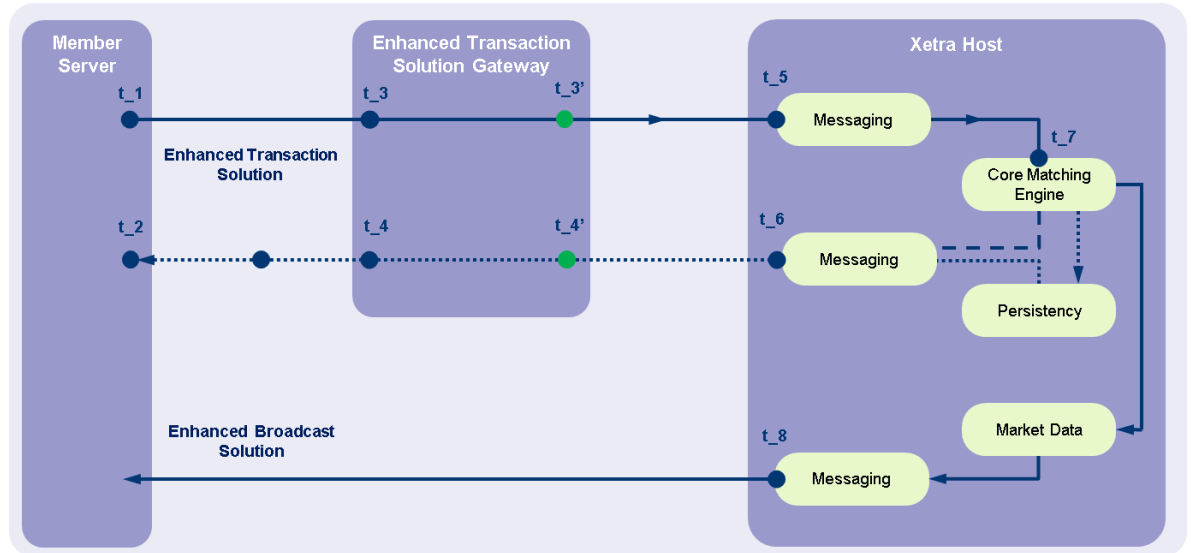


Figure 1: Time stamps and their measure points

$t_1, t_2$ : can be taken by the application when the request/response is read from/written to the socket.

$t_3, t_4$ : taken by the Enhanced Transaction Solution gateway when request/ response is read from/written to the socket on the participant ´s side of the gateway; contained in (private) Enhanced Transaction Solution response.

$t_5, t_6$ : taken by the matching engine when request/response is read from/written to the socket; contained in (private) Enhanced Transaction Solution response (note that consecutive  $t_5$  timestamps will differ by at least 18usec).

$t_7$ : taken when the order is functionally processed (e.g. put in book or matched); available in the (private) Enhanced Transaction Solution response and Enhanced Broadcast Solution order book delta and Enhanced Broadcast Solution trade message in case the order matches.

$t_8$ : taken just before a market data UDP datagram is written to the socket.

Message	Field name	Timestamp	Description
Packet Header	Performance Indicator	$T_8-T_5$	Time between the arrival of an incoming order/quote transaction on the Xetra matching engine and send time of the corresponding outgoing market data. The Performance Indicator is filled for order book incremental messages only, otherwise a 0 (zero) is sent.

Message	Field name	Timestamp	Description
		$T_8 - T_7$	Time between the outgoing market data and the time of the Xetra matching engine. This calculation is taken place for matching events resulting from auctions.
	Sending Time	$T_8$	Time the Xetra Market Data Component writes packet onto the socket.
Delta	entryTime	$T_7$	Two possibilities: - In case of an order: Time of last order book update of all updates in the message. - In case of a trade: Match time.
	aggressorTime	$T_5$	Entry time of the incoming order that triggered the trade. This time stamp is only available in case of a trade. The aggressorTime is empty if - the trade resulted from an auction.
Snapshot	entryTime	$T_7$	Time of the last order book update.
All Trade Price	entryTime	$T_7$	Match time.
Cross Request Instrument state change Technical market reset System or exchange state change	entryTime	$T_7$	Time when the event was processed by the core system.

Note: The transaction timestamp of an order/quote transaction in the Enhanced Transaction Solution (lastEventTrnId) has a different format than the timestamp in the Enhanced Broadcast Solution (entryTime); nevertheless the customer will be able to map both after conversion at microsecond granularity.

The message creation time (always denoted by “timestamp”) is provided in following messages

- Start of Xetra service message
- End of Xetra service message
- Start of instrument reference data message
- End of instrument reference data message
- Start of maintenance reference data message
- End of maintenance reference data message
- Beacon message
- All trade price message
- System/exchange state message (snapshot)
- Snapshot message
- Delta/Incremental message
- Cross request message
- Technical market reset message
- Instrument state message
- System/exchange state message (event driven)

## 5.2.2 Reference Information

The reference data stream disseminates instrument and maintenance data reference information. Which fields are actually sent depends e.g. from the instrument type. Market Maker fields are provided if Market Maker roles are defined for an instrument. Included is the whole tick table.

### 5.2.2.1 Instrument Reference Data

Instrument reference data will be disseminated over the static reference data stream.

Field Name	Descriptive Name	Field Format
TID	Template ID	Template ID type (7.83)
SeqNum	Sequence number	Sequence number type (7.68)
srclId	Source identifier	Source ID type (7.73)
exchId	Market Identifier Code of the trading market on Xetra according to ISO 10383.	Exchange ID type (7.20)
instGrp	Instrument group	Instrument group type (7.27)
noRelatedSym	Number of instruments	Number of Values type (7.55)
> isix	Instrument identifier	Isix type (7.34)

---



Field Name	Descriptive Name	Field Format
> isin	Instrument ISIN code	ISIN code type (7.33)
> instMnem	Instrument mnemonic	Mnemonic type (7.53)
> instShtNam	Instrument Long Name	instShtNam Type (7.30)
> wknNo	Wertpapierkennnummer	wknNo type (7.95)
> setld	Set identifier	Set Id type (7.69)
> currCode	Price currency	Currency code type (7.15)
> instTypCod	Instrument type	Instrument Type (7.29)
> instSubTypCod	Instrument Subtype Code	instSubTypCod type (7.31)
> lglMktSeg	Legal Markt Segment Code	lglMktSegCod type (7.39)
> mktSegCod	Market Segment Code	mktSegCod type (7.52)
> noTickRules	Number of tick rules.	Number of Values type (7.55)
>> tickIncrement	Tick increment.	tickIncrement type (7.84)
>> endTickPriceRange	End of tick price range.	endTickPriceRange type (7.17)
> trdmdl	Trading Model.	TrdMdlTypCod Type (7.88)
> optGwLoc	Optimal Gateway Location.	optGatewayLocation type (7.56)
> referencePrice	Reference Price.	referencePrice type (7.63)
> frstTrdDat	First Trading Date.	frstTrdDat type (7.22)
> lstTrdDat	Last Trading Date.	lstTrdDat type (7.41)
> bonMrtyDat	Maturity Date.	bonMrtyDat type (7.7)
> minOrdrSiz	Minimum Order Size.	minOrdrSiz type (7.48)
> minTrdbUnt	Minimum Tradable Unit.	minTrdbUnt type (7.50)
> rondLotQty	Round Lot Quantity.	rondLotQty type (7.66)
> tradCal	Trading Calendar.	tradCal type (7.86)
> untOfQotn	Unit of Quotation.	untOfQotn type (7.90)
> indicators	Array of supported order types and other flags.	Indicator collection type (7.26)
> minHiddOrdrVal	Minimum Hidden Order Value.	minHiddOrdrVal type (7.45)
> minMidPntOrdrVal	Minimum Midpoint Order Value.	minMidPntOrdrVal type (7.47)
> minIceQty	Minimum Iceberg Order Quantity.	minIceQty type (7.46)

Field Name	Descriptive Name	Field Format
> minPeakQty	Minimum Iceberg Order Peak Quantity.	minPeakQty type (7.49)
> vdoMinExec	Volume Discovery Minimum Execution Quantity	Volume Discovery Type (7.50)
> maxordrvalbest	Maximum Order Value BEST.	Maxordrvalbest type (7.43)
> clgloc	Clearing Location.	Clgloc type (7.12)
> postTrdAty	Post Trade Anonymity indicator.	postTrdAty type (7.59)
> stlPeriodFlag	Settlement Period Flag.	setlPeriodFlg Type (7.72)
> stlCurrCod	Settlement Currency Code.	setlCurrCod Type (7.71)
> stlCal	Settlement Calendar.	setlCal type (7.70)
> reportingMarket	Market Identification Code (ISO 10383) required for reporting to supervisory authority	reportingMarket type (7.65)
> homeMkt	Market Identifier Code (ISO 10383) of the market where the IPO took place.	homeMkt type (7.24)
> refMkt	Reference Market.	refMkt type (7.64)
> cmexind	Cum-Ex-Indicator.	Cmex Indicator Type (7.13)
> bonCpnRatWss	The coupon rate of a bond.	Coupon rate type (7.14)
> bonCurPoolFact	Bond Current Pool Factor.	bonCurPoolFact type (7.5)
> bonYldTrdInd	Bond Yield Trading Indicator.	bonYldTrdInd type (7.9)
> bonCrtCpnDat	Beginning of current coupon period.	bonCrtCpnDat type (7.4)
> bonNxtCpnDat	End of current coupon period.	bonNxtCpnDat type (7.8)
> dnomCurrCod	Denomination Currency.	dnomCurrCod type (7.16)
> bonFlatInd	Bond Flat Indicator.	bonFlatInd type (7.6)
> bonAcrlntCalcMD	Accrued interest calculation method	bonAcrlntCalcMD type (7.3)
> xetralssrMnem	Bond Issuer	xetralssrMnem type (7.96)
> warSeg	Warrant Segment.	warSeg type (7.92)
> warStrPrc	Warrant Strike Price.	warStrPrc type (7.93)
> warTyp	Warrant Type.	warTyp type (7.94)
> warUnd	Warrant Underlying.	UnderlyingSecurityID type (7.89)

Field Name	Descriptive Name	Field Format
> exchangeSegment	Exchange Segment Identifier	Exchange Segment Id type (7.21)
> issrMemblId	Issuer Member Id.	issrMemblId type (7.35)
> issrSubGrp	Issuer Subgroup.	issrSubGrp type (7.36)
> specMemblId	Specialist Member Id.	specMemblId type (7.74)
> specSubGrp	Specialist Subgroup.	specSubGrp type (7.75)
> knockOutInd	Knock-out indicator.	knockOutInd type (7.38)
> singleAuctionIndicator	Single auction indicator.	Indicator type (7.25)
> specialAuctionIndicator	Special auction indicator.	Indicator type (7.25)
> issueDate	The date on which a security is issued.	issueDat type (7.37)
> inSubscription	Indicator for subscription trading (primary market).	Y = instrument in subscription trading. Only provided for instruments that are currently in subscription trading.. inSubscription type (7.32)
> noOfLicenses	Number of Licenses.	Number of Values type (7.55)
>> licensetype	License type.	Licensetype type (7.40)
>> sprdtypecode	Spread Type Code.	Sprdtypecode type (7.78)
>> sprdFact	Spread Factor.	sprdFact type (7.76)
>> sprdminqty	Spread Minimum Quantity.	Sprdminqty type (7.77)
> maxsrpqty	Maximum Surplus Quantity.	Maxsrpqty type (7.44)
> basund	BASIS instrument underlying.	UnderlyingSecurityID type (7.89)
noOfStreams	Number of streams	Number of values type (7.55)
> streamType	Type of stream	Stream type (7.82)
> streamService	Service class of the stream	Stream service type (7.81)
> inetAddr	Multicast address of the stream	Address type (7.2)
> port	Port number	Port type (7.58)
> mktDepth	Stream depth	Stream depth type (7.80)
> mdBookType	Book type	Book type (7.10)

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

### 5.2.2.2 Maintenance Reference Data

Maintenance reference data messages provide the multicast address of the state changes stream. It will be disseminated over the static reference interface.

Field Name	Descriptive Name	Field Format
TID	Template ID	Template ID type (7.83)
srclId	Source identifier	Source ID type (7.73)
SeqNum	Sequence number	Sequence number type (7.68)
noOfExchanges	Number of Trading Markets	Number of values type (7.55)
>exchId	Market Identifier Code of the trading market on Xetra according to ISO 10383.	Exchange ID type (7.20)
> noOfExchangeSegments	Number of Exchange Segments	Number of values type (7.55)
>>exchangeSegment	Exchange Segment	Exchange Segment Id type (7.21)
noOfStreams	Number of streams	Number of values type (7.55)
>streamService	Service class of the stream	Stream service type (7.81)
>streamType	Stream Type	Stream Type type (7.82)
>inetAddr	Multicast address of the stream	Address type (7.2)
>port	Port number	Port type (7.58)
>mktDepth	Stream depth	Stream depth type (7.80)
>mdBookType	Book type	Book type (7.10)

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

## 5.2.3 Order Book Information

### 5.2.3.1 Order Book Snapshot

The snapshot messages will be disseminated over the snapshot stream.

The following principles are valid for the design of the snapshot messages.

- Snapshot messages include top-of-book and complete price-level information (depth  $\geq 1$ ).
- 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.
- Instrument trading phase for instruments traded in continuous auction model

At start time each tradable instrument starts with phase START. This is delivered with the full incremental broadcast with sequence number 1 and also delivered with the snapshot. Further incremental shows always the current phase. If no incremental must be created for an instrument also no state change information is sent. In the case that phase of instrument has been START, the state is implicitly changed with the exchange segment state change but no individual messages are distributed on instrument-level. The snapshot stream will display the instrument state, but the sequence number is not increased.

		<i>Exchange state</i>					
		START	PRETR	TRADE	POSTR	ENDTR	HALT
<i>Instrument state</i>	XPREC	START	PRETR	XPREC	POSTR	ENDTR	HALT
	XCALL	n/a	PRETR	XCALL	XCALL	n/a	HALT
	XFRZ	n/a	PRETR	XFRZ	XFRZ	n/a	HALT
	HALT	HALT	HALT	HALT	HALT	HALT	HALT
	SUSP	SUSP	SUSP	SUSP	SUSP	SUSP	SUSP
	DEL	DEL	DEL	DEL	DEL	DEL	DEL
	ADD	ADD	ADD	ADD	ADD	ADD	ADD
	HOL	HOL	HOL	HOL	HOL	HOL	HOL

The snapshot data has four different logical groups (three repeating groups and one optional group). The number of entries of the repeating group will always be filled, if there is no information for that group, the number will be set to 0.

- EntriesAtp (former Trade information): Repeating “All Trade Price” group. This group will be filled when there is any information on last trade price, in subscription price, last auction price, last midpoint trade, last best price or last trade price with “Bundesbank” participation.
- EntriesDepth (former Order book data): Repeating “Depth” group. When orderbook entries are sent, the field noEntriesDepth will be set to the number of entries being sent, entryType will be equal to “*BID\_PRC*” or “*ASK\_PRC*” and all the fields in the repeating group will be filled. Market Orders have entryType = *ASK\_PRC\_MARKET* / *BID\_PRC\_MARKET* and a price = 0.
- auctionGroup (former potential auction price and qty group): Optional group for values occurring in auctions. The auction group is only active in an auction, when the order book is crossed or the current book situation will go to a volatility or market order interruption. For instruments, which are traded in the specialist model of continuous auction, the specialist quote is delivered here.
- EntriesPrc (former statistics): Repeating “Price” group. Identical in structure to the delta message “EntriesPrc” group. This repeating group will be filled whenever there is a new daily high, daily low, opening price, closing price, valuation price.

Every snapshot contains the sequence number for the last “consolidated” delta message for this instrument in order to be able to position it among the delta messages. Members are expected to remain connected to the delta channels while receiving the snapshot for an instrument and buffer all the deltas until the complete snapshot is received. After the snapshot is received, these deltas have to be applied to the snapshot to obtain the most recent picture of the market.

There is no reference to the ATP stream, all values delivered with the ATP stream are part of the EntriesAtp group in the snapshot.

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

SNAPSHOT DATA MESSAGE STRUCTURE		
Field Name	Descriptive Name	Field Format
TID	Template ID	Template ID type (7.83)
entryTime	Time of entry	Timestamp type (7.85); timestamp T <sub>7</sub> (Figure 1)
srclId	Source identifier	Source ID type (7.73)
isix	Instrument identifier	Isix type (7.34)
consolSeqNum	Sequence number	Sequence number type (7.68)
instrStatus	Instrument status	Instrument status type (7.28)
noEntriesAtp	Number of entries	Number of values type (7.55)
>entryType	Type of entry	Entry type. Allowed values are: LAST AUCTION, LAST TRADE, SUBSCR_PRC, LAST MIDPOINT, LAST BEST, BUBA_PRC. (7.19)
>entryPrc	Traded price	General price type (7.23)
>entryQty	Traded quantity	Quantity type (7.62)
>totTrdQty	Total traded quantity	Quantity type (7.62)
>entryTime	Time of entry	Timestamp type (7.85); T <sub>7</sub> in Figure 1.
>numTrades	Number of trades	Number of trades type (7.54)
>tranMtchIdNo	Unique trade price ID	Transaction Id type (7.87)
noEntriesDepth	Number of entries	Number of values type (7.55)
>entryType	Type of entry	Entry type. Allowed values are: BID, ASK, ASK PRC MARKET, BID PRC MARKET, EMPTY BOOK (7.19)
>entryPrc	Price for the level below	General price type (7.23)
>entryQty	Quantity offered at above price	Quantity type (7.62)
>numOrders	Number of orders	Number of values type (7.55)
>entryPrcLvl	Level of price in ODB	Entry level type (7.18)
auctionGroup		Optional group of auction values
> moiInd	Moi indicator	Market Order Interruption indicator type (7.42)
> volInd	Vola indicator	Vola indicator type (7.91)
> noEntriesAuction	Number of entries	Number of values type (7.55)

SNAPSHOT DATA MESSAGE STRUCTURE		
Field Name	Descriptive Name	Field Format
>> entryType	Type of entry	Entry type. Allowed values are: POTENTIAL AUCTION PRICE, SURPLUS BID/ASK and MATCHING RANGE BID/ASK (7.19)
>>entryPrc	Potential auction price or matching range bid/ask price	General price type (7.23)
>>entryQty	Potential auction quantity, matching range BID/ASK quantity or surplus BID/ASK quantity	Quantity type (7.62)
noEntriesPrc	Number of entries	Number of values type (7.55)
>entryType	Type of entry	Entry Type. Allowed values are: OPENING, CLOSING, VALUATION, HIGH, LOW (7.19)
>entryPrc	Price	General price type (7.23)

### 5.2.3.2 Order Book Delta/Incremental

The order book delta messages will be disseminated over the delta stream.

The following principles are valid for the design of the delta messages.

- Delta/incremental messages include partial price-level information (depth  $\geq 1$ ).
- 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, order book and trade information.
- Price levels are provided explicitly (by a number) and do not need to be inferred through the price itself.
- The order book has a valid state, after all instructions of the repeating group *EntriesDepth* has been processed.
- Every delta message has a sequence number for synchronization purposes.
- If the number of trade prices exceeds the limit of 20, the trade gap indicator is set
- During the pre-trading and post-trading phases when no price levels exist for an instrument, an empty book will be disseminated.

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

The delta data has four different logical groups (three repeating groups and one optional group). The number of entries of the repeating group will always be filled, if there is no new information for that group, the number will be set to 0.

- “EntriesTrade” is a repeating group, which is sent if a trade occurs. All parameters delivered with the ATP stream are now also part of the trade sequence. An auction price is always the earliest price and therefore the last entry in the “entriesTrade” group. In this case the maximum number of prices provided is restricted to 19. If the field priceTypeCod describes an auction price, this price becomes the “last auction price” and is not delivered separately.

Note: The gapIndicator is set always if the matcher overwrites not yet delivered data. It is always guaranteed that the values within the message are the current market values. If the trade sequence is empty in this message, but the last traded price has changed in between, the last traded price with all its values is sent in the trade sequence.

For other prices, which could be sent via the price group, apply: If not sent, it is unchanged during the gap, otherwise the current value is sent.

- The cancelled quantity of a resting order due to a prevented in house crossing (aka self match prevention) is delivered in the sequence “EntriesTradePrices” member of the sequence “EntriesTrade”.

For SMP cancelled bid limit orders the entryType is BID\_RESTING\_CLX. For SMP cancelled ask limit orders the entryType is ASK\_RESTING\_CLX. For SMP canceled bid market orders the entryType is BID\_RESTING\_CLX\_MARKET. For SMP canceled ask market orders the entryType is ASK\_RESTING\_CLX\_MARKET. The "entryPrc" shows the limit of the cancelled order. In case of a cancelled market order the “entryPrc” is set to 0. The actnCod is always “Add” and the TranMchldNo is always 0.

- “EntriesDepth” (Order book Data. “Depth” group) is a repeating group similar in structure to the snapshot message order book group, only difference being the updateAction field needed to know in what way to update the order book. When order book entries are sent, the field noEntriesDepth will be set to the number of entries being sent, entryType will be equal to “BID\_PRC” or “ASK\_PRC” and all the fields in the repeating group will be filled. Market Orders have entryType = ASK\_PRC\_MARKET / BID\_PRC\_MARKET and a price = 0.
- “auctionGroup” (former potential auction price and ty group): Optional group for values occurring in auctions. The auction group is only active in an auction, when the order book is crossed or the current book situation will go to a volatility or market order interruption. The first time, when this group becomes active in an auction all available values will be sent. The values become invalid, when the auction is finished. For instruments, which are traded in the specialist model of continuous auction, the specialist quote is delivered here. A cleared surplus or potential auction price will be reported with a delete entryType (former version quantity zero).
- EntriesPrc (former statistics): Repeating “Price” group. Identical in structure to the delta message “EntriesPrc” group. This repeating group will be filled whenever there is a new daily high, daily low, opening price, closing price, valuation price or new reference price.

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

- New: Creating a price level, adds the new price at the specified entryPrcLvl say x. All price levels y where  $y \geq x$  are shifted to  $y + 1$ .
  - Change: Changing a price level, replaces the quantity of the price level specified by the entryPrcLvl with the information sent in the message.
  - Delete: Deleting a price level, removes the price at the level specified by entryPrcLvl say x. All price levels y where  $y > x$  are shifted to  $y - 1$ .
  - Delete From: Deletes all price levels from entryPrcLvl  $\geq x$  to maximum price levels maintained in the order book for the instrument.
-



- Delete Thru: Deletes all price levels from price level = '1' to entryPrclvl = 'x'. All price levels y where  $y > x$  are shifted to  $y - x$ .
- For all delete actions (delete, delete from, delete thru) the fields entryPrc, entryQty and numOrders have to be ignored in the repeating group noEntriesDepth.

DELTA/INCREMENTAL DATA MESSAGE STRUCTURE		
Field Name	Descriptive Name	Field Format
TID	Template ID	Template ID type (7.83)
entryTime	Time of entry	Timestamp type (7.85)
srclcd	Source identifier	Source ID type (7.73)
lsix	Instrument Identifier	lsix type (7.34)
seqNum	Sequence Number	Sequence number type (7.68)
instrStatus	Instrument Status	Instrument status type (7.28)
gapIndicator	flag to indicate a gap	Gap indicator type (7.25); there are more than one update collected in this message
trdgapIndicator	flag to indicate a gap	Gap indicator type (7.25); there are more prices (> 20) than delivered in the Entries Trade sequence
noEntriesTrade	Number of entries	Number of values type (7.55)
>entryType	Type of entry	Entry type. Allowed values are: TRADE_PRC_LIST, LAST_TRD_PRC, MPO_PRC, BEST_PRC (7.19)
>aggressorTime	Time of entry	Timestamp type (7.85); T <sub>5</sub> in Figure 1.
>totTrdQty	Total traded quantity	Quantity type (7.62)
>numTrades	Number of trades	Number of trades type (7.54)
>noEntriesTradePrices	Number of entries	Number of values type (7.55)
>>entryType	Type of entry	Entry type. Allowed values are: LAST_TRD_PRC, SUBSCR_PRC, SPECIAL_AUCTION, BUBA_PRC, BID_RESTING_CLX, BID_RESTING_CLX_MARKET, ASK_RESTING_CLX, ASK_RESTING_CLX_MARKET, MPO_PRC, BEST_PRC (7.19)
>>entryPrc	Traded price	General price type (7.23)
>>entryQty	Traded quantity	Quantity type (7.62)
>>entryTime	Time of entry	Time type (7.85); T <sub>7</sub> in Figure 1.

DELTA/INCREMENTAL DATA MESSAGE STRUCTURE		
Field Name	Descriptive Name	Field Format
>>priceTypeCod	Price Type Code	priceTypeCod (7.61)
>>actnCod	Action code	Action code type (7.1)
>>tranMtchldNo	Unique trade price ID	Transaction Id type (7.87)
noEntriesDepth	Number of entries	Number of values type (7.55)
>entryType	Type of entry	Entry type. Allowed values are: BID, ASK, ASK PRC MARKET, BID PRC MARKET, EMPTY BOOK (7.19)
>entryPrc	Price for the level below	General price type (7.23)
>entryQty	Quantity offered at above price	Quantity type (7.62)
>numOrders	Number of orders	Number of values type (7.55)
>entryPrcLvl	Level of price in ODB	Entry level type (7.18)
>updateAction	What operation will be performed with this entry	Price action type (7.60)
auctionGroup		Optional group of auction values
> moiInd	Moi indicator	Market Order Interruption indicator type (7.42)
> volInd	Vola indicator	Vola indicator type (7.91)
> noEntriesAuction	Number of entries	Number of values type (7.55)
>> entryType	Type of entry	Entry type. Allowed values are: POTENTIAL AUCTION PRICE, SURPLUS BID/ASK and MATCHING RANGE BID/ASK (7.19)
>>entryPrc	Potential auction price or matching range bid/ask price	General price type (7.23)
>>entryQty	Potential auction quantity, matching range BID/ASK quantity or surplus BID/ASK quantity	Quantity type (7.62)
noEntriesPrc	Number of entries	Number of values type (7.55)

DELTA/INCREMENTAL DATA MESSAGE STRUCTURE		
Field Name	Descriptive Name	Field Format
>entryType	Type of entry	Entry Type. Allowed values are: LAST_TRD_PRC, OPENING, CLOSING, VALUATION, HIGH, LOW (7.19)
>entryPrc	Price	General price type (7.23)

Xetra works on the basis of “units of work”. A single unit of work may consist of multiple order book updates and multiple trades. Each of the trades generates a discrete trade message (All Trade Price). Each unit of work always delivers only a single delta message, which includes the final order book after the processing of all orders plus the individual trade prices and accumulated trade quantities per price level for each processed order.

In addition the delta message contains the field total traded quantity. Total traded quantity is the cumulated quantity of the instrument for the current business day.

Each unit of work may consist of up to 20 different trade prices and their related quantities, listed in reverse time order (latest trade price first). In the unlikely event that more than 20 trade prices occur in one single unit of work only the last 20 trade price and quantities are disseminated via the delta message. Whether any trade is not included in the delta message can be obtained by evaluating the `trdgapIndicator`.

In case any trade price is not included in the delta message the full view of all trades may be obtained from the All Trade Price stream.

The delta message sequence numbers, field `seqNum`, increment sequentially only within the context of a single source identifier. Therefore, this field can only be used when comparing messages coming from the same source.

A single unit of work on the host may result in many trades (which are reported individually) but only a single order book delta, delivered at the end of the unit of work.

This field should therefore not be expected to increment sequentially.

### Order book “Trimming”

Enhanced Broadcast solution does not send explicit delete messages beyond `mktDepth`. Client applications must, however, allow for temporary growth beyond `mktDepth` until all actions of a message are processed.

## 5.2.4 Trade Information

### 5.2.4.1 All Trade Price Message

The Xetra All Trade Price subscription is the Xetra public broadcast data stream for dissemination of information about every single trade.

All Trade Price messages will be disseminated over the All Trade Price stream.

Following points should be taken into account:

- The reversal of a trade generates a **message with delete trade indicator** with the appropriate trade quantity.
  - Trade information is sequenced per instrument.
-

- For more information on the FAST implementation, refer to the XML template.

ALL TRADE PRICE MESSAGE STRUCTURE		
Field Name	Descriptive Name	Field Format
TID	Template ID	Template ID type (7.83)
timestamp	Timestamp	Timestamp type (7.85)
srclId	Source identifier	Source ID type (7.73)
isix	Instrument Identifier	Isix type (7.34)
seqNum	Sequence Number	Sequence number type (7.68)
priceTypeCod	Price Type Code	priceTypeCod (7.61)
entryType	Type of Entry	Valid Values: LAST_TRD_PRC, BEST_PRC, MPO_PRC, SUBSCR_PRC, SPECIAL_AUCTION (7.19)
entryPrc	Price @ the above level	General Price type (7.23)
entryQty	Quantity offered @ above price	Quantity type (7.62)
entryTime	Time of entry	Timestamp type (7.85)
tranMtchIdNo	Internal trade identifier	Transaction Id type (7.87)
actnCod	Action code	Action code type (7.1)

The field “tranMtchIdNo” corresponds to the field in the delta and snapshot message.

#### 5.2.4.2 Trade Reversals

The Xetra Enhanced Broadcast Solution features trade reversal broadcasts. Trade reversals are disseminated over All Trade Price stream with actnCod = 6.

Trade reversals do not affect the Snapshot and the Delta stream. Notably the following fields are not updated:

- Last trade price
- Daily high price of the instrument
- Daily low price of the instrument

#### 5.2.5 State Changes

This section covers status information that applies to exchange, system and special instrument changes.

The Xetra state change information subscription is used by the application to subscribe to the Xetra public broadcast data stream for dissemination of Xetra state change information.

The state change messages are sent over the state changes stream both cyclically and event-driven. The current system and exchange states are sent cyclically. An event occurs as soon as an exchange or system change happens (see 7.79), and is marked with the `changeIndicator = 'Y'`;

The messages will not be sequenced.

### 5.2.5.1 State Change Message

State changes on system or exchange or exchange segment level.

STATE CHANGES MESSAGE STRUCTURE		
Field Name	Descriptive Name	Field Format
TID	Template ID	Template ID type (7.83)
timestamp	Timestamp	Timestamp type (7.85)
srcId	Source identifier	Source ID type (7.73)
changeIndicator	flag to indicate a change	Indicator type (7.25); if set, an event driven state change has occurred.
state	Exchange or System State	State Type (7.79)
exchId	Exchange or ExchangeSegment where the state belongs to.	Exchange ID type (7.20) or Exchange Segment Identifier Type (7.21)

### 5.2.5.2 Instrument State Change Message

State change on instrument level.

INSTRUMENT STATE CHANGE MESSAGE STRUCTURE		
Field Name	Descriptive Name	Field Format
TID	Template ID	Template ID type (7.83)
entryTime	Time of entry	Timestamp type (7.85)
srcId	Source identifier	Source ID type (7.73)
isix	Instrument Identifier	Isix type (7.34)
state	Instrument Status State	State Type (7.28)

### 5.2.5.3 Technical Market Reset Message

Sends a technical market reset.

TECHNICAL MARKET RESET MESSAGE STRUCTURE		
Field Name	Descriptive Name	Field Format
TID	Template ID	Template ID type (7.83)
entryTime	Time of entry	Timestamp type (7.85)
srcId	Source identifier	Source ID type (7.73)
setId	Set Identifier	Set type (7.69)

#### 5.2.5.4 Cross Request Message

Sends a cross request.

CROSS REQUEST MESSAGE STRUCTURE		
Field Name	Descriptive Name	Field Format
TID	Template ID	Template ID type (7.83)
entryTime	Time of entry	Timestamp type (7.85)
srcId	Source identifier	Source ID type (7.73)
isix	Instrument Identifier	Isix type (7.34)
entryQty	Quantity offered at above price	Quantity type (7.62)

## 6 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.
Empty Book	An order book which does not contain any information (price levels).
ETS	Enhanced Transaction Solution
FAST	FIX Adapted for Streaming <sup>SM</sup> (FAST Protocol <sup>SM</sup> ).
FIX	Financial Information eXchange.
Gap	Missing information related to trades or changes to the order book.
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 streams.
Multicast	Many-to-many network architecture as against one-to-one (unicast).
Opening Price	This is the price of the first trade for the instrument. This price is recoverable through the snapshot stream.
Order	A contractually-binding request to other market participants to buy or sell a specific quantity of a financial instrument at a defined price.
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. These single prices represent 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.

---

Term	Explanation
Stream	A stream is a collection of one or more instruments and is the basic unit of subscription.
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.

---



## 7 Appendix – Data Field Dictionary

The following section provides specific information on the Xetra Enhanced Broadcast Solution data fields, listing valid values, ranges or giving an example value for the field.

### 7.1 Action Code Type

Description: This field indicates what type of action should be performed in case of last trade price information.

Type: Unsigned long

Value	Meaning
4	Add
5	Update
6	Delete

Where sent:

- All Trade Price Message
- Delta / Incremental Message

### 7.2 Address Type

Description: The multicast address of the channel.

Type: Unsigned long uint32

Value Range	Meaning
224.000.000.000 ... 239.255.255.255	Ipv4 address
E0 00 00 00 ... EF FF FF FF	Hexadecimal

Where sent:

- Instrument Reference Data Message
- Maintenance Reference Data Message

### 7.3 BonAcrIntCalcMd

Description: The identifier used to indicate the accrued interest calculation method.

Type: Unsigned long uint32

Number	Method
0	No interest calculation
1	German (30/360)
2	English (ACT./365/6)

---

- 3 French (ACT./360)
- 4 US TRSY 2
- 5 ACT/365 (no leap year)
- 6 ISMA: 30/360
- 7 ACT/365
- 8 US TRSY 4
- 9 ISMA: ACT/ACT
- 10 ISDA: ACT/365(6)
- 11 France: COMPND METH
- 12 Italy: TRSY BONDS
- 13 Poland
- 14 Hungary
- 15 ACT/365(6)
- 16 Bond Basis

Where sent:

- Instrument Reference Data Message

#### **7.4 BonCrtCpnDat Type**

Description: Beginning of current coupon period.  
Type: Unsigned Long

Value Range	Meaning
e.g. 20110605	05.06.2011

Where sent:

- Instrument Reference Data Message

#### **7.5 BonCurPoolFact Type**

Description: Bond Current Pool Factor  
Type: Signed decimal

---

Value Range	Meaning
0.0 ... 100.0	Pool Factor

Where sent:

- Instrument Reference Data Message

## 7.6 bonFlatInd Type

Description: Bond Flat indicator  
Type: ASCII Character String

Value	Meaning
0	Accrued interest calculation and pool factoring enabled.
1	flat, no accrued interest calculation.
2	x-flat, no accrued interest calculation and no pool factoring.

Where sent:

- Instrument Reference Data Message

## 7.7 bonMrtyDat Type

Description: Maturity Date.  
Type: Unsigned long., 8 digits

Value Range	Meaning
e.g. 20110605	05.06.2011

Where sent:

- Instrument Reference Data Message

## 7.8 bonNxtCpnDat Type

Description: End of current coupon period.  
Type: Unsigned long, 8 digits

Value Range	Meaning
e.g. 20110605	05.06.2011

Where sent:

- Instrument Reference Data Message

## 7.9 bonYldTrdInd Type

Description: Indicates, whether the corresponding price contains a yield.  
Type: ASCII Character String., 1 character

---

### Value Range

“Y” or “N”

Where sent:

- Instrument Reference Data Message

## 7.10 Book Type

Do not mix up the type “Book Type” with the field name “mdBookType” of type “Book Type”.

Description: Type of Book

Type: Unsigned long with NULL support

### Value Range    Meaning

2                    Price depth

Where sent:

- Instrument Reference Data Message
- Maintenance Reference Data Message

## 7.11 Business Date Type

Description: Current business date.

Type: ASCII Character String

### Value Example

e.g. “20070101”

Where sent:

- Start / End Service Message
- Start / End Instrument Reference Data Message
- Start / End Maintenance Reference Data Message

## 7.12 clgLoc Type

Description: Clearing location of the respective instrument..

Type: ASCII Character String.

### Value Range

e.g. “BOFRA”, “BECCP”

Where sent:

- Instrument Reference Data Message
-

### 7.13 Cmx Indicator Type

Description: Indicator for CUM/EX events or capital adjustment  
Type: ASCII Character String

Value	Meaning	FIX Corporate Action (292)
(blank) – CMEX_IND_NO_CHG, CMEX_IND_REST	No change / Reset	Not sent
“C” – CMEX_IND_CUM	Cum dividend	<no standard value>
“A” – CMEX_IND_ADJ	Capital adjustment	<no standard value>
“E” – CMEX_IND_EX	Ex dividend	“A”

Where sent:

- Instrument reference data

### 7.14 coupon rate Type

Description: The coupon rate of a bond.  
Type: Decimal

**Value Range**  
0.0000001 ... 99.9999999

Where sent:

- Instrument Reference Data Message

### 7.15 Currency Code Type

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:

- Instrument Reference Data Message

### 7.16 dnomCurrCod Type

Description: Denomination Currency code of the instrument.

---

Type: This field identifies the currency in ISO 4217 representation.  
ASCII Character String

Value Examples	Meaning
“EUR”	Euro
“GBP”	British Pound
“USD”	US Dollar

Where sent:

- Instrument Reference Data Message

### 7.17 endTickPriceRange Type

Description: End of Tick range for reach Tick increment.  
Type: Decimal

Value Range
Like prices

Where sent:

- Instrument Reference Data Message

### 7.18 Entry Level Type

Description: The type of entry level  
Type: Unsigned long

Value	Meaning
1	Top of book
1...20	Depth level

Where sent:

- Snapshot Message
- Delta / Incremental Message

### 7.19 entryType

Description: Type of entry; generally not all possible values are valid in all cases of application.  
Type: Unsigned long

Value	Meaning
0 EMPTY_ENTRY	No entry
1 ASK_PRC	Ask price

---

<b>Value</b>	<b>Meaning</b>
2 BID_PRC	Bid price
3 EMPTY_BOOK	Empty Order book
4 LAST_TRD_PRC	Last trade price
5 OPENING_PRC	Opening price
6 LAST_AUC_PRC	Last auction price
7 VAL_PRC	Valuation price
8 CLOSE_PRC	Closing price
9 SUBSCR_PRC	Price from subscription period
10 BEST_PRC	BEST trade price
11 MPO_PRC	Midpoint trade price
12 PTNL_AUCT_PRC	Potential auction price
13 MTCH_RNG_ASK	Matching range ask
14 MTCH_RNG_BID	Matching range bid
15 SRP_BID	Surplus bid
16 SRP_ASK	Surplus ask
20 HIGH_PRC_ENTRY	Daily high price
21 LOW_PRC_ENTRY	Daily low price
23 ASK_PRC_MARKET	Ask market order
24 BID_PRC_MARKET	Bid market order
25 BUBA_PRC	Currently not in use
26 TRADE_PRC_LIST	List of trade prices
27 DEL_PTNL_PRC	Delete potential auction price
28 DEL_SRP	Delete surplus entry
29 SPECIAL_AUCT	Special auction; for Federal Bonds the Special Auction indicator is used for prices determined with Bundesbank participation.
30 BID_RESTING_CLX	Cancelled quantity and limit of resting bid limit order due to SMP reason.
31 ASK_RESTING_CLX	Cancelled quantity and limit of resting ask limit order due to SMP reason.
32 BID_RESTING_CLX_MARKET	Cancelled quantity of resting bid market order due to SMP reason. "entryPrc" is 0.
33 ASK_RESTING_CLX_MARKET	Cancelled quantity of resting ask market

---

**Value**

**Meaning**

order due to SMP reason. "entryPrc" is 0

Where sent:

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

## 7.20 Exchange Id Type

Description: Market Identifier Code of the trading market on Xetra according to ISO 10383.

Type: ASCII Character String

Value Examples	Meaning
XEUB	Eurex Bonds
XETR	Xetra Deutsche Börse
XFRA	Frankfurt Stock Exchange
XDUB	Dublin / Irish Stock Exchange
XVIE	Vienna / Wiener Börse
XCEG	Central European Gas Hub
XPRA	Prague Stock Exchange
XLJU	Ljubljana Stock Exchange
XBUD	Budapest Stock Exchange
XMAL	Malta Stock Exchange
XCAY	Cayman Islands Stock Exchange
XAFX	African Stock Exchange
XZAG	Zagreb Stock Exchange

Where sent:

- Instrument Reference Data Message
- Maintenance Reference Data Message
- State Change Message

## 7.21 Exchange Segment Identifier Type

Description: The exchange segment code in which the instrument is traded.

Type: ASCII Character String

---



Value Examples
XEXC
XEXD

Where sent:

- Instrument Reference Data Message
- Maintenance Reference Data Message
- State Change message

## 7.22 frstTrdDat Type

Description: The earliest date for which an instrument becomes tradable.  
Type: Unsigned Long.

### Value Range

17990101 ... 55551231

Where sent:

- Instrument Reference Data Message

## 7.23 General Price Type

Description: Price field. Float field expressed as exponent (4 bytes) and mantissa (8 bytes) fields.  
Type: Scaled Number

### Value Example

exponent = -2  
mantissa = 2555  
resulting value = 25.55

Where sent:

- Instrument Reference Data Message
- Snapshot Message
- Delta/Incremental Message
- All Trade Price Message

## 7.24 homeMkt Type

Description: Market Identifier Code (ISO 10383) of the market where the IPO took place.  
Type: ASCII Character String

### Value Range

Like exchange identifier

---

Where sent:

- Instrument Reference Data Message

## 7.25 Indicator Type

Description: A flag to indicate whether the information is there or not. Used always for (Y)es/(N)o decisions..

Type: ASCII Character String

Value	Meaning
-------	---------

Y	Information is there (e.g. gap, trade gap exists)
---	---

Where sent:

- Instrument Reference Data Message
- Delta/Incremental Message
- All Trade Price Message
- State Change message

## 7.26 Indicator Collection Type

Description: Collection of indicators, e.g. order types.

Type: Unsigned long uint32

Power of 2	Meaning
------------	---------

0	Closed Book Indicator
1	Discretionary Order Indicator
2	Hidden Order Indicator
3	Iceberg Order Indicator
4	Limit Order Indicator
5	Midpoint Order Indicator
6	Market Imbalance Indicator
7	Market Order Indicator
8	Market-to-Limit Order Indicator
9	Quote Book Indicator
10	Market maker protection flag
11	Liquidity Interruption Indicator
12	Mini Auction Indicator
13	Matching Instruction Cross Id

---

Where sent:

- Instrument Reference Data Message

## 7.27 Instrument Group Type

Description: Instrument group will usually be disseminated via the same stream.  
Type: ASCII Character String

### Value Example

DAX1

Where sent:

- Instrument Reference Data Message

## 7.28 Instrument Status Type

Description: Status of the instrument. Stream.  
Type: ASCII Character String

Mapping of VALUES prcsStsValCod to EnBS:

Name	EnBS
START	0
PRETR	1
OCALL	5
ICALL	6
CCALL	7
ECALL	8
OIPO	10
OFRZ	11
IIPO	12
IFRZ	13
OPOBB	16
IPOBB	17
CPOBB	18
EPOBB	19
OOBB	21
IOBB	22

---

Name	EnBS
COBB	23
EOBB	24
TRADE	26
BETW	27
POSTR	28
ENDTR	29
HALT	30
SUSP	31
VOLA	32
ADD	35
DEL	36
XPREC	39
XCALL	40
XFRZ	41
HOL	43

**SecurityStatus:**

Value	Status
1	Active
3	Knocked out
4	Knocked out / revoked

**Instrument States:**

Enhanced Broadcast Solution Instrument State	Description	SecurityStatus (965)	SecurityTradingStatus (326)
31 SUSP	Suspend	'1'	107
50 -	Unsuspend	'1'	21
51 -	Knocked out	'3'	17
52 -	Knocked out / revoked	'4'	17
53 MIDFR	Midpoint book freeze	'1'	126
54 MIDUF	Midpoint book unfreeze	'1'	17

Where sent:

- Snapshot Message
- Delta / Incremental Message
- Instrument Status Message

## 7.29 Instrument Type

Description: Type of instrument  
Type: ASCII Character String

Value Example	Meaning
BAS	Basis Instrument
BON	Bond
EQU	Equity
WAR	Warrant

Where sent:

- Instrument Reference Data Message

## 7.30 instShtNam Type

Description: Instrument Long Name (although “instShtNam” indicates a “short” name).  
Type: ASCII Character String

**Value Range**  
String with max. 30 characters

Where sent:

- Instrument Reference Data Message

## 7.31 instSubTypCod Type

Description: Instrument Subtype.  
Type: ASCII Character String, 3 characters

**Value Range**  
e.g. “ANL”, “VAR”

Where sent:

- Instrument Reference Data Message
-

### 7.32 inSubscription Type

Description: Indicator for subscription trading (primary market). Y = instrument in subscription trading. Only provided for instruments that are currently in subscription trading..

Type: ASCII Character String

Value	Meaning
“Y”	Instrument is in subscription period
“N”	Instrument is not in subscription period

Where sent:

- Instrument Reference Data Message

### 7.33 ISIN Code Type

Description: The ISIN code of the instrument. The ISIN code follows standard in ISO 6166 (<http://www.anna-nna.com>).

Type: ASCII Character String

Value Example	Meaning
DE0005810055	Deutsche Börse AG

Where sent:

- Instrument Reference Data Message

### 7.34 isix Type

Description: Internal identifier assigned to each instrument.

Type: Unsigned long

Value Range
1..4294967295

Where sent:

- Beacon Message
- Instrument Reference Data Message
- Snapshot Message
- Delta / Incremental Message
- All Trade Price Message
- Instrument State Change Message
- Cross Request Message

### 7.35 issrMemblD Type

Description: Member ID of the issuer (Quote Provider) comprising institution (3 characters) and

---

Type: branch (2 characters).  
ASCII Character String

**Value Range**

“AAAAA” ... “ZZZZZ”

Where sent:

- Instrument Reference Data Message

### 7.36 issrSubGrp Type

Description: Trader subgroup to which a user of an issuer (Quote Provider) has been assigned to.  
Type: ASCII Character String

**Value Range**

e.g. “A10”, “22Y”

Where sent:

- Instrument Reference Data Message

### 7.37 issueDat Type

Description: The date on which a security is issued.  
Type: Long, 8 digits

Value Range	Meaning
-------------	---------

e.g. 20110605	05.06.2011
---------------	------------

Where sent:

- Instrument Reference Data Message

### 7.38 knockOutInd Type

Description: Indicator for knocked-out instruments.  
Type: ASCII Character String.

Value	Meaning
(blank)	Default
R	Reversed
K	Knocked out
L	Locked knock out

---

Where sent:

- Instrument Reference Data Message

### 7.39 IgIMktSegCod Type

Description: Type of market admission: The code for a legal market segment. These include Amtlicher Markt, Geregelter Markt and Freiverkehr.

Type: Unsigned long, 3 digits

Value	Examples	Meaning
3		Open Market
5		European Energy Exchange
6		Eurex Bonds
14		Open Market – Prädikatsmarkt
15		Open Market – Newex, NX.one
16		Open Market – Newex, NX.plus
17		Open Market – Newex, NX. Other
38		Open Market – Deutsche Boerse SMART TRADING
39		Open Market – Entry Standard
42		Open Market – Electronic Trading
43		Regulated Market – Electronic Trading
44		Regulated Market
45		Regulated Market – Prime Standard
46		Regulated Market – General Standard
47		Official Market
48		Unofficial Market
60		Budapest Stock Exchange MTF
80		Structured Products Regulated Market
81		Structured Products Open Market
82		Xetra Frankfurt 2 – Regulated Market, Prime Standard
83		Xetra Frankfurt 2 – Regulated Market, General Standard
84		Xetra Frankfurt 2 – Open Market, Entry Standard
85		Xetra Frankfurt 2 – Open Market
86		Xetra Frankfurt 2 – Regulated Market General Quoted
87		Official List Main Security Market Xetra
88		Irish Enterprise Security Market Xetra
97		Eurex
98		Eurex Repo
99		Global Exchange Market Xetra
318		Budapest Stock Exchange Regulated Market
437		WBAG Dritter Markt (MTF)
568		LJSE Official Market
735		Prague Stock Exchange Official Market
736		Prague Stock Exchange MTF
738		Prague Stock Exchange Free Market
930		WBAG – Wiener Wertpapierbörse
931		WBAG – Amtlicher Handel
934		WBAG – Geregelter Freiverkehr

Where sent:

- Instrument Reference Data Message
-



## 7.40 licensetype Type

Description: Type of License.  
Type: ASCII Character String

Value Range	Meaning
“M”	Designated Sponsor License
“E”	BEST Executor License
“Q”	Liquidity Manager License

Where sent:

- Instrument Reference Data Message

## 7.41 IstTrdDat Type

Description: The last date on which an instrument may be traded.  
Type: Unsigned long

Value Range
17990101 ... 55551231

Where sent:

- Instrument Reference Data Message

## 7.42 Market Order Interruption Indicator Type

Description: A flag to indicate a market order interruption.  
Type: ASCII Character String

Value	Meaning
“(space)”	No market order interruption
“P”	Potential market order interruption
“M”	Market order interruption
“X”	Market order interruption already performed

Where sent:

- Snapshot Message
- Delta/Incremental Message

## 7.43 maxOrdrValBest Type

Description: Maximum value that may apply to a Xetra BEST order, specified as a cash amount.  
Type: Signed Decimal

---

**Value Range**

-999999999999.999 ... +999999999999.999

Where sent:

- Instrument Reference Data Message

**7.44 maxSrpQty Type**

Description: Maximum surplus quantity which can be taken by a member in the Betreuer OBB phase.

Type: Signed Decimal

**Value Range**

-9999999999.999 ... +9999999999.999

Where sent:

- Instrument Reference Data Message

**7.45 minHiddOrdVal Type**

Description: Minimum Hidden Order Value: This field indicates the minimum order value that may apply to a hidden order, specified as cash amount.

Type: Signed Decimal

**Value Range**

-999999999999.999 ... +999999999999.999

Where sent:

- Instrument Reference Data Message

**7.46 minIceQty Type**

Description: Minimum Iceberg Order quantity.

Type: Signed Decimal

**Value Range**

-9999999999.999 ... +9999999999.999

Where sent:

- Instrument Reference Data Message

**7.47 minMidPntOrdVal Type**

Description: Minimum Midpoint Order value.

Type: Signed Decimal

---

**Value Range**

-999999999999.999 ... +999999999999.999

Where sent:

- Instrument Reference Data Message

**7.48 minOrdrSiz Type**

Description: Minimum Order Size for this instrument.

Type: Signed Decimal

**Value Range**

-9999999999.999 ... +9999999999.999

Where sent:

- Instrument Reference Data Message

**7.49 minPeakQty Type**

Description: Minimum Iceberg Order Peak quantity.

Type: Signed Decimal

**Value Range**

-9999999999.999 ... +9999999999.999

Where sent:

- Instrument Reference Data Message

**7.50 vdoMinExecType**

Description: Volume Discovery Minimum Execution Quantity. 0 means not available.

Type: Signed Decimal

**Value Range**

0 ... +9999999999.999

Where sent:

- Instrument Reference Data Message

**7.51 minTrdbUnitType**

Description: Minimum Tradable Unit of this instrument.

Type: Signed Decimal

---

### Value Range

-999999999.999 ... +999999999.999

Where sent:

- Instrument Reference Data Message

## 7.52 mktSegCod Type

Description: Market Segment Supplement.

Type: ASCII Character String, 3 characters

Value Example	Meaning
BGA	Market A
BGB	Market B
DED	US Stars
DEE	European Stars
DEL	XTF Exchange Traded Funds
DES	Deutschland – Fonds
DEX	Boerse Frankfurt (Fonds)
DEZ	Exchange Traded Commodities, ETC
DE2	Select Bonds
DE3	Prime Bonds
DE4	Deutsche Boerse REITs
DE5	Deutsche Boerse First Quotation Board
DX7	Xetra Bonds
DX8	Exchange Traded Notes (ETN)
DX9	Open Market Plus
LJA	LJSE Prime Market
LJB	LJSE Standard Market
LJC	LJSE Entry Market
LJD	LJSE Bonds
LJE	LJSE Fund Market
LJG	LJSE Closed-end Fund Shares
LJL	LJSE T-Bills
PRX	PSE Qualified investors funds
PRY	PSE Collective investment funds
VIB	WBAG Bonds Financial Sector
VIC	WBAG Equities Standard Market Auction
VIE	WBAG ETFs
VIG	WBAG Bonds Public Sector
VIK	WBAG Bonds Corporate Sector
VIL	WBAG Performance Linked Bonds
VIM	WBAG Mid Market
VIO	WBAG Other Securities
VIP	WBAG Equities Prime Market
VIS	WBAG Equities Standard Market Continuous
VIW	WBAG Warrants
VIZ	WBAG Certificates

Where sent:

- Instrument Reference Data Message
-

### 7.53 Mnemonic Type

Description: The abbreviated form of the instrument name. The field instMnem is only filled, if an instrument has an Instrument Mnemonic. In other cases the field will contain a single space.

Type: ASCII Character String

Value Example	Meaning
"DB1"	Deutsche Börse AG.

Where sent:

- Instrument Reference Data Message

### 7.54 Number of Trades Type

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

Type: Unsigned long

Value Range
0..4294967295

Where sent:

- Snapshot Message
- Delta / Incremental Message

### 7.55 Number of Values Type

Description: A general field carrying a number

Type: Unsigned long

Value Range
0..4294967295

Where sent:

- Start / End of Instrument / Maintenance Reference Data Message
- Instrument Reference Data Message
- Maintenance Reference Data Message
- Snapshot Message
- Delta / Incremental Message

### 7.56 OptiGatewayLocId Type

Description: Enhanced Transaction Solution: The optimal performance gateway location for trading

---

Type: a respective instrument..  
Unsigned long, 4 digits

**Value Range**

e.g. 4711

Where sent:

- Instrument Reference Data message.

### 7.57 PerformanceIndicator

Description: Xetra host performance indicator.  
Type: uint64(byte vector)

**Value Range**

0..16777215 (0xfffff)

Where sent:

- Packet Header Message

### 7.58 Port Type

Description: The port where the data is disseminated.  
Type: Unsigned long

**Value Range**

0..65535

Where sent:

- Instrument Reference Data Message
- Maintenance Reference Data Message

### 7.59 postTrdAty Type

Description: Post Trade Anonymity indicator. Field indicating, whether post trade anonymity is enabled ("P"), the central counterpart is used ("C") or not (blank).  
Type: ASCII Character String.

**Value Range**

"P", "C", " " (i.e. blank)

Where sent:

- Instrument Reference Data Message
-

## 7.60 Price Action Type

Description: Describes the update action for the price entry in orderbook  
Type: Unsigned long

Value	Meaning
1 - CRE	New
2 - CHG	Change
3 - DEL	Delete
4 - DEL_FRM	Delete From. Delete from level 'X' to maximum levels.
5 - DEL_THR	Delete Through. Delete from level 1 to level 'X'.

Where sent:

- Delta / Incremental Message

## 7.61 priceTypeCod Type

Description: Price Type Code.  
Type: ASCII Character String, 1 character

Value	Meaning
C	Continuous Trading
O	Opening Auction
A	Auction
F	Closing Auction
E	End-of-day Auction
V	Volatility Interruption in Continuous Trading
M	Mini Auction
L	Liquidity Interruption

Where sent:

- Delta / Incremental Message
- All Trade Price Message

## 7.62 Quantity Type

Description: The quantity or volume field. Float field expressed as exponent (4 bytes) and mantissa (8 bytes) fields.  
Type: Scaled Number

**Value Example**  
exponent = -1

---

#### Value Example

mantissa = 2500  
resulting value = 250.0

Where sent:

- Snapshot Message
- Delta/Incremental Message
- All Trade Price Message
- Cross Request Message

### 7.63 referencePrice Type

Description: Reference Price.  
Type: Decimal

#### Value Range

Like prices

Where sent:

- Instrument Reference Data Message

### 7.64 refMkt Type

The market from which the reference price is derived.

Description: Reference Market.  
Type: ASCII Character String

#### Value Range

Like exchange identifier

Where sent:

- Instrument Reference Data Message

### 7.65 reportingMarket Type

Description: Market Identifier Code (ISO 10383) required for reporting to supervisory authority  
Type: ASCII Character String

#### Value Example

“XETA” – Xetra regulated market

Where sent:

- Instrument Reference Data Message

### 7.66 rondLotQty Type

Description: Round Lot quantity, in number of units of the instrument.

---



Type: Signed Decimal

**Value Range**

-999999999.999 ... +999999999.999

Where sent:

- Instrument Reference Data Message

### 7.67 SendingTime Type

Description: The timestamp when the message was created by the Enhanced Broadcast Solution.

Type: Unsigned long64 (byteVector)

**Value Range**

**Meaning**

0..863999999999 Microseconds since midnight CET/CEST

Where sent:

- Packet Header Message

### 7.68 Sequence Number Type

Description: Sequence number to assist in identifying data loss, if not in Packet Header or Beacon message.

Note: Not all Xetra Enhanced Broadcast Solution messages are sequenced.

Type: Unsigned long

**Value Range**

0..4294967295

Where sent:

- Packet Header Message
- Beacon Message
- Instrument Reference Data Message
- Maintenance Reference Data Message
- Snapshot Message
- Delta / Incremental Message
- All Trade Price Message

### 7.69 Set Id Type

Description: Technical grouping of instruments.

Type: Unsigned long

---

#### Value Range

1..1007

Where sent:

- Instrument Reference Data Message
- Technical Market Reset Message

### 7.70 setlCal Type

Description: Settlement Calendar.  
Type: ASCII Character String

#### Value Range    Meaning

e.g. "XCAL"    Standard calendar

Where sent:

- Instrument Reference Data Message

### 7.71 setlCurrCod Type

Description: Settlement Currency Code.  
Type: ASCII Character String, 3 characters

#### Value Range

e.g. "EUR"

Where sent:

- Instrument Reference Data Message

### 7.72 setlPeriodFlg Type

Description: Number of business days from trade execution after which settlement is to be effected.  
Type: ASCII Character String.

#### Value Range

"0" ... "9"

Where sent:

- Instrument Reference Data Message

### 7.73 Source Id Type

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

---

#### Value Range

0..4294967295

Where sent:

- Packet Header Message
- Beacon Message
- Instrument Reference Data Message
- Maintenance Reference Data Message
- Snapshot Message
- Delta / Incremental Message
- All Trade Price Message
- State Changes Message
- Instrument State Change Message
- Technical Market Reset Message
- Cross Request Message

#### 7.74 specMembId Type

Description: Member ID of the Specialist.  
Type: ASCII Character String

#### Value Range

e.g. "ABCFR", "DBKFR"

Where sent:

- Instrument Reference Data Message

#### 7.75 specSubGrp Type

Description: The Specialist's subgroup; its users are authorized to act as Specialists for the instrument.  
Type: ASCII Character String

#### Value Range

e.g. "ISS"

Where sent:

- Instrument Reference Data Message

#### 7.76 sprdFact Type

Description: Maximum Spread permissible for the license.  
Type: Decimal

---

#### Value Range

-999999999.999 ... +999999999.999

Where sent:

- Instrument Reference Data Message

### 7.77 sprdMinQty Type

Description: Minimum allowable size of a Quote by license.

Type: Decimal

#### Value Range

-999999999.999 ... +999999999.999

Where sent:

- Instrument Reference Data Message

### 7.78 sprdTypCod Type

Description: Quote Spread Type for license.

Type: ASCII Character String

#### Value Meaning

“A” Absolute value

“P” Percentage value

Where sent:

- Instrument Reference Data Message

### 7.79 State Type

Description: Identifier for Exchange and System states depending on whether the change applies to the system or to trade model 8 (Trading Model Continuous Auction)

Type: Unsigned long

Value	Meaning	Refers to
1	System Online	System
2	System in Batch	System
3	System Halted	System
4	Fast Market Off for Equities	System
5	Fast Market On for Equities	System
6	Fast Market Off for Bonds	System

---

Value	Meaning	Refers to
7	Fast Market On for Bonds	System
8	Fast Market Off for Warrants	System
9	Fast Market On for Warrants	System
10	Fast Market Off for Basis Instruments	System
11	Fast Market On for Basis Instruments	System
12	Exchange Start	Trading Model Continuous Auction
13	Exchange Pre-trade	Trading Model Continuous Auction
14	Exchange Trade	Trading Model Continuous Auction
15	Exchange End-trade	Trading Model Continuous Auction
16	Exchange Post-trade	Trading Model Continuous Auction
17	Exchange Halt	Trading Model Continuous Auction

Where sent:

- State Changes Message

### 7.80 Stream Depth Type

Also known as mktDepth.

Description: The stream depth type indicates the defined order book depth for the channel  
Type: Unsigned long with NULL support

#### Value Range

0...20

Where sent:

- Maintenance Reference Data Message
- Instrument Reference Data Message

### 7.81 Stream Service Type

Description: The stream service type indicates whether the stream is part of service A or service B of the Xetra Enhanced Broadcast Solution.

Type: ASCII Character String

#### Value Meaning

“A” Service A

“B” Service B

---

Where sent:

- Instrument Reference Data Message
- Maintenance Reference Data Message

## 7.82 Stream Type

Description: The type of the stream  
Type: ASCII Character String

Value	Name	Service	Meaning
1	Snapshot	EnBS	Snapshot
2	Delta	EnBS	Delta / Incremental
3	ATP	EnBS	All Trade Price
4	Maintenance	EnBS	Maintenance Stream
5	Ticker	MDI	External Ticker
6	Market Data	MDI	Snapshots / Deltas
7	Maintenance	MDI	Maintenance Stream
8	ATP/PWT	MDI	All Trade Price/Price Without Turnover

Where sent:

- Instrument Reference Data Message
- Maintenance Reference Data Message

## 7.83 Template Id Type

Description: The identifier used to indicate the FAST template to be used for decoding.  
Type: Unsigned long

Value	Meaning
34	Packet Header Message
2	Beacon Message
3	Instrument Reference Data
4	Maintenance Reference Data
6	Snapshot Message
7	Delta/Incremental Message
9	All Trade Price Message
10	State Changes Message

---

<b>Value</b>	<b>Meaning</b>
11	Cross Request Message
12	Market Reset Message
13	Instrument suspend/unsuspend, knocked out/revoke, midpoint book freeze/unfreeze
120	FAST Reset Message
128	Start of Service Message
129	End of Service Message
130	Start of Instrument Reference Data Message
131	End of Instrument Reference Data Message
132	Start of Maintenance Reference Data Message
133	End of Maintenance Reference Data Message

Where sent:

- Packet Header Message
- Beacon Message
- Instrument Reference Data Message
- Maintenance Reference Data Message
- Snapshot Message
- Delta / Incremental Message
- All Trade Price Message
- State Changes Message
- Instrument State Change Message
- Technical Market Reset Message
- Cross Request Message
- Start / End Service Message
- Start / End Instrument Reference Data Message
- Start / End Maintenance Reference Data Message

#### **7.84 tickIncrement Type**

Description: Tick Increment.  
Type: Decimal

Where sent:

- Instrument Reference Data Message
-

## 7.85 Timestamp Type

Description: The timestamp when the message was created by the Enhanced Broadcast Solution or the official timestamp of the exchange, in the format microseconds since midnight CET/CEST.

Type: Unsigned long

Value Range	Meaning
0..86399999	Microseconds since midnight

Where sent:

- Beacon Message
- Snapshot Message
- Delta / Incremental Message
- All Trade Price Message
- State Changes Message
- Instrument State Change Message
- Technical Market Reset Message
- Cross Request Message
- Start / End Service Message
- Start / End Instrument Reference Data Message
- Start / End Maintenance Reference Data Message

## 7.86 tradCal Type

Description: Trading Calendar.

Type: ASCII Character String

Value Range	Meaning
e.g. "XCAL"	Example

Where sent:

- Instrument Reference Data Message

## 7.87 Transaction ID Type

Description: Internal transaction matching identifier. 0 for SMP

Type: Unsigned long

Value Range
0..4294967295

Where sent:

- All Trade Price Message
-



- Snapshot Message
- Delta / Incremental Message

### 7.88 trdMdITypCod Type

Description: Trading Model identifier.  
Type: ASCII Character String,

Value	Meaning
1	Multiple Auctions, M
2	Continuous Trading; C
3	One Auction Only, O
8	Continuous Auction, X

Where sent:

- Instrument Reference Data Message

### 7.89 UnderlyingSecurityID Type

Description: Underlying ISIN of a warrant or basis instrument.  
Type: ASCII Character String, 12 characters

**Value Range**  
e.g. "DE0008232125"

Where sent:

- Instrument Reference Data Message

### 7.90 untOfQuotn Type

Description: The unit, in which an instrument is quoted / stated.  
Type: Unsigned long, 3 digits

Value	Meaning
001	Pieces (EQU/WAR)
002	Percent (BON/BAS)
004	Points (Indices)

Where sent:

- Instrument Reference Data Message

### 7.91 Vola Indicator Type

Description: Indicates whether an instrument has a potentially volatility interruption, an actual

---

volatility interruption, a Liquidity Interruption, a Liquidity Interruption with a potential volatility interruption, a Mini Auction, a Mini Auction with a potential volatility interruption, or neither.  
Type: ASCII Character String

Value	Meaning
"E"	Extended volatility interruption
"F"	Freeze
"P"	Potential volatility interruption
"V"	Volatility interruption
"X"	Expired
"L"	Liquidity Interruption
"A"	Mini Auction
"I"	Liquidity Interruption with potential volatility interruption
"B"	Mini Auction with potential volatility interruption
" "(space)	No volatility interruption

Where sent:

- Snapshot Message
- Delta/Incremental Message

## 7.92 warSeg Type

Description: Segment of a Warrant instrument.  
Type: ASCII Character String, 2 characters

Value	Meaning
WA	Plain Vanilla Warrants
KO	Knock Out without Stop Loss
KL	Knock Out with Stop Loss
EL	Exotic Leverage Products
RV	Reverse Convertibles
BA	Basket Certificates
BO	Bonus Certificates
DS	Discount Certificates
GC	Guarantee Certificates
IX	Index Certificates
OP	Out Performance Certificates
MS	Misc. Investment Products.
(blank)	Other

---

Where sent:

- Instrument Reference Data Message

### 7.93 warStrPrc Type

Description: Strike Price for a warrant.  
Type: ASCII Character String, 12 characters

#### Value Range

Free form text.

Where sent:

- Instrument Reference Data Message

### 7.94 warTyp Type

Description: Type of warrant.  
Type: ASCII Character String, 1 character

Value	Meaning
C	Call
P	Put
R	Range
F	Certificate
O	Others

Where sent:

- Instrument Reference Data Message

### 7.95 wknNo Type

Description: Wertpapierkennnummer.  
Type: ASCII Character String.

#### Value Range

“000000000” ... “999999999”

Where sent:

- Instrument Reference Data Message

### 7.96 xetraIssrMnem Type

Description: Bond issuer  
Type: ASCII Character String, 4 character.

---

---

**Value Range**

e.g. "BUND"

Where sent:

- Instrument Reference Data Message

---

## 8 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 Enhanced Broadcast Solution.

Limit	Descriptive	Name / Message	Value
MAX LIVE LIVE SERVICES	The Enhanced Broadcast Solution interface disseminates data simultaneously in duplicate over two services, 'A' and 'B'	noStreams / Maintenance Reference Data	2
MAX STRING SIZE	Maximum size of any string disseminated through the interface. Enhanced Broadcast Solution disseminates mainly binary data.	N.A.	30
MAX TICK SIZE TABLE	Maximum size of tick size table.	Instrument Reference Data	20
TRADE_PRICE_LIST_MAX	Maximum number of trade prices in EntriesTradePrice group of delta messages	Delta	20
TRADE_ENTRY_MAX	Maximum number of entries in the EntriesTradePrices group	Delta	3
ATP_ENTRY_MAX	Maximum number of entries in the EntriesAtp group	Snapshot	4

---

---

## 9 Appendix - How to Use

This section explains how member applications should use the Xetra Enhanced Broadcast Solution 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 version message has to be considered to determine the market source.

### 9.1 Start of Day

#### 9.1.1 Connect to the Reference Data Stream

Member applications must connect to the reference data stream for each market source and receive the reference data messages as explained in chapter 4 (Broadcast Streams) before they start receiving and processing any data (snapshot, delta, All Trade Price broadcasts and system state changes).

The address for this first stream will be published by Xetra. Through this stream, instrument reference data and maintenance reference data is distributed.

Steps involved:

- Join the reference data stream multicast address.
- Wait until the start of instrument or maintenance reference data message is received.
- Once this message is received, start processing all subsequent UDP datagrams from this multicast address until the end of instrument (or maintenance) reference data message is received.
- Wait until the start of maintenance reference data message is received, if the instrument reference data has already been received or wait until the start of instrument reference data message is received, vice versa. Both types of reference data have to be received. It does not matter which one is received and processed first.
- Start processing all subsequent UDP datagrams until the corresponding end of reference data message is received.
- Leave this stream.

This information has to be stored for future use. The instrument reference data contains information on sets, instruments, streams, channels and the corresponding multicast addresses. The maintenance reference data contains information on the corresponding multicast addresses of state changes.

Members must use the information obtained to identify the sets and instruments in which they are interested. Members are expected to join only the required streams.

#### 9.1.2 Receive Snapshot

Members have to first join the snapshot streams for the instruments which they require and after receiving the order book snapshots, members should leave this stream.

---

Steps involved to receive orderbook snapshots:

- Connect to the delta channels (first) and buffer the orderbook deltas while they receive the snapshots, so that no delta message is lost.
- Join the snapshot streams for the instruments.
- Wait until the datagram carrying the orderbook snapshot for the instrument is received and discard all unwanted packets (see next section on synchronization).
- Apply the snapshot to the members' copy of the orderbook.
- Leave all the snapshot streams after receiving the initial snapshots.

Members may have to join more than one multicast address if the instruments of their interest are not broadcast on the same stream, multicast address.

### 9.1.3 Process Delta

Members are expected to stay connected to this stream throughout the day.

The following steps need to be performed for the delta streams:

- Join the delta streams of interest. This could involve joining multiple channels (multiple multicast addresses).
- The top of book channel disseminates trade information for the instrument.
- Process datagrams containing data related to the instruments of interest. Discard all unwanted datagrams.
- 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 snapshot and delta/incremental streams. Members will have to filter the information for the sets and instruments of their interest.

## 9.2 Messages Sequence Numbers Synchronization

Due to the unreliable nature of the UDP protocol, the order in which the broadcasts arrive, whether snapshots, deltas and/or All Trade Prices 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 these three streams (All Trade Price, deltas and snapshots), sequence numbers have been added to the orderbook and trade messages. These sequence numbers along with the source identifier (where is this message coming from) can be used to detect messages lost.

### 9.2.1 Snapshot - Delta Synchronization

Order book snapshots carry consolidated sequence numbers, field *consolSeqNum*, for each of the configured delta streams, providing the receiver with sufficient information to position the snapshots exactly amongst the delta messages in each stream.

---

Each delta message for a specific instrument has a unique sequence number. This sequence number is assigned by the source and is useless unless the source is also taken into account. The sequence number (field seqNum) increments sequentially, always within the context of a single source identifier. Therefore, these sequence numbers can be used for synchronization purposes whenever they are coming from the same source.

If any gaps are detected in the sequence numbers on the delta stream and they are not filled after a period of time (some time must be allowed for the unreliability of the UDP interface), member applications can either join the snapshot stream again and receive the latest order book snapshot or try rebuilding their copy of the order book with the next delta messages for the instrument.

Each snapshot message has a reference to the latest delta sequence number in each stream whose information it comprises.

With this information, the member knows as soon as he receives a snapshot, which delta messages are already included in the snapshot information and therefore can be thrown away since they will not add any new information. And, on the contrary, which messages have information more recent than the snapshot and must be taken into account to have an up-to-date orderbook.

For example, assuming a scenario like the one in the figure below, sequence numbers coming all from the same source and for the same instrument:

	Delta	Snapshot
t1	42	...
t2	43	...
t3	44	...
t4	45	44
t5	...	...
t6	46	...
t7	47	...
t8	...	48
t9	48	...
t10	49	...
t11	50	...
t12	...	50

During the period t1-t12, deltas message sequence numbers from 42 to 50 and three snapshots are received.

The member subscribes to the snapshot and delta streams for a specific instrument at time t1 and starts receiving messages, first come deltas. When a snapshot arrives, it states that, for that stream, the snapshot has consolidated the information up to delta message 44, so the first three received deltas can be ignored (delta messages numbers 42, 43 and 44) whereas delta message number 45 has to be incorporated into the global picture.

Then delta message 46 arrives at t6 and delta message 47 arrives at t7 and both changes are applied to the orderbook.

When snapshot message arrives at t8 with consolidated delta sequence number 48<sup>5</sup>, we know that it has some information that we are missing, so we take it.

<sup>5</sup> This case is highly improbable since delta messages are created and sent before snapshot, but it could happen due to problems in the transmission.



Then at t9, delta message with sequence number 48 arrives and we ignore it.

Then delta messages with sequence numbers 49 and 50 arrive and their information is saved.

Then snapshot message with delta sequence number 50 arrives and we ignore it because we have already received everything it has, via the delta stream.

## **9.2.2 All Trade Price Synchronization**

### **9.2.2.1 All Trade Price Stream**

The All Trade Price Stream has a sequence number for each instrument. This number can be used to monitor the trade prices for one instrument. The trade price message includes the field `tranMtchIdNo`, assigned by the Xetra matching engine. Two different number pools are used, one for the mid point trade prices, and one for BEST and other trade prices.

### **9.2.2.2 Delta – All Trade Price Synchronization**

Xetra works on the basis of “units of work”. A single unit of work may consist of multiple order book updates and multiple trades. Each of the trades generates a discrete trade message (All Trade Price). Each unit of work always delivers only a single delta message, which includes the final order book after the processing of all orders plus the individual trade prices and accumulated trade quantities per price level for each processed order.

The individual trade prices and associated values, including the total traded quantity, are mapped to the repeating group `EntriesTrade`.

Each unit of work may consist of up to 20 different trade prices, listed in reverse time order (latest trade price first). In the unlikely event that more than 20 trade prices occur in one single unit of work only the last 20 trade price and quantities are disseminated via the delta message. Whether a trade is not included in the delta message may be obtained from the `trdgapIndicator`. In case any trade price is not included in the delta message the full view of all trades may be obtained from the All Trade Price stream.

Members requiring a guaranteed full view of all reported trades must therefore include information from the All Trade Price messages, not only the delta messages

### **9.2.2.3 Snapshot – All Trade Price Synchronization**

In the snapshot stream the `tranMtchIdNo` is delivered for the last trade price for each type (BEST, midpoint, etc.)

The synchronization between the Snapshot and All Trade Price stream works the same way as the synchronization between Delta stream and All Trade Price stream.

## **9.3 Order Book Recovery**

Due to the inherently unreliable nature of the UDP protocol, occasional loss of UDP packets in the network is expected. Loss of data packets can be detected from non-sequential numbers of messages for an instrument.

---

The order book will be available for recovery purposes on the snapshot streams throughout the day.

### **9.3.1 Loss of Packets**

If the member joins both the interface services A and B, packets may be received from either of the two services. A permanent loss of packet(s) is established when a member detects a sequence number gap.

If a copy of the order book cannot be generated, a recovery is required and the following steps need to be performed.

Please note that as a property of multicast networks, packets may not be delivered in the order they are sent by the information source.

#### **9.3.1.1 Receive Snapshot for an Instrument**

Join the snapshot stream of the instrument to receive the current ODB snapshot:

- Join the snapshot stream for the instrument. Stay connected to the delta channel and buffer all incoming deltas.
- Wait for the packets carrying snapshot information message on the instrument and discard all the other packets.
- Apply the snapshot to the local copy of the ODB. Every snapshot carries the consolidated sequence numbers for all the delta channels.
- Apply the buffered deltas that were issued after the snapshot.

#### **9.3.1.2 Leave the Snapshot Stream**

Leave the snapshot channel and continue working with the deltas.

#### **9.3.1.3 Continue Operation with the Delta Stream**

Maintain the orderbook by processing all incoming deltas from the delta stream.

## **9.4 Events Demanding Special Response**

Due to the transparent operation of the Xetra Enhanced Broadcast Solution, 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.

### **9.4.1 Gap Indicator**

Type	Normal
Where sent	The Gap Indicator field is disseminated in ODB delta/incremental message for an instrument.
Value	Y

---

Description	<p><u>ODB delta/incremental message</u> If sent and the value is “Y”, it indicates that the Xetra Enhanced Broadcast Solution interface may have missed some of the latest orderbook changes. This situation can occur if the Xetra Enhanced Broadcast Solution data generators were busy with other instruments when there was more than one ODB update for an instrument. The next ODB delta message for such an instrument is sent out with the gap indicator set to indicate this event. This message will contain the latest orderbook update, so that the members’ orderbook view will be consistent. When the gap indicator is not sent, it should be assumed that there is no gap, that Xetra has sent all the order book updates for the instrument.</p> <p><u>Trade information message</u> When this field carries value “Y”, it indicates that some intermediate trades had occurred for the instrument which may not be reported due to heavy processing load. When the gap indicator is not sent it should be assumed that Xetra has sent information on all trades for the instrument.</p>
-------------	---

#### 9.4.2 Trade Gap Indicator

Type	Normal
Where sent	The Trade Gap Indicator field is disseminated in ODB delta/incremental message for an instrument.
Value	Y
Description	<p><u>ODB delta/incremental message</u> If sent and the value is “Y”, it indicates that the Xetra Enhanced Broadcast Solution can’t deliver all trade prices within this message. This situation can occur if the Xetra Enhanced Broadcast Solution detects more than 20 trade prices within on unit of work. In this case the latest 20 trade prices were sent. Some older prices are missing. These trade prices can be found in the ATP stream.. When the trade gap indicator is not sent, it should be assumed that there is no gap, that Xetra has sent all the trade prices for the instrument.</p>

#### 9.4.3 Source Identifier

Type	Exceptional
Where sent	Every message sent out of the Xetra Enhanced Broadcast Solution 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 Enhanced Broadcast Solution changes, namely the Xetra trade matching system, this value is changed accordingly. This value is unique per business day. Members should maintain the market information for each instrument per source identifier.

---

#### 9.4.4 Sequence Number

Type	Normal
Where sent	The Xetra Enhanced Broadcast Solution sequences Delta and All Trade Price messages per instrument per source.
Value	Non-consecutive sequence numbers.
Description	<p>This can occur due to the unreliable nature of the UDP protocol. As the interface sequences all messages per instrument per source, any message loss will be known to the member immediately. If members wish to recover from this situation, they can join the snapshot channel for the instrument to receive the recent orderbook snapshot and then continue with the delta channel.</p> <p>Note that while connecting to the snapshot channel the client application has to remain connected to the delta channels and buffer all the incoming order book deltas. The order book snapshot message carries the last consolidated sequence number for the delta channel. After the snapshot is received the client application has to discard all the deltas already included in the snapshot and then apply the remaining deltas to the order book view carried by the snapshot to create the recent market picture once again.</p>

Type	Exceptional
Where sent	The Xetra Enhanced Broadcast Solution sequences Delta and All Trade Price messages per instrument per source.
Value	Sequence numbers abruptly restart.
Description	<p>This can occur due to two reasons:</p> <p>(1) If the source of market data, namely the order matching system, changes. Every time the Xetra Enhanced Broadcast Solution interface starts receiving data from a new source, it restarts sequencing the messages.</p> <p>(2) Technical malfunction. If there is some technical malfunction and if the interface is unable to recover the old market information, the Xetra Enhanced Broadcast Solution interface can choose to restart sequencing the messages.</p>

#### 9.4.5 Start/End of Service Message

Type	Exceptional
Where sent	On start and end of the interface services.
Value	Messages sent within the trading day.
Description	<p>Due to technical malfunctions Xetra could restart some of the interface services. This will result in an additional start and end of service messages.</p> <p>Members should treat this as an exceptional situation and respond to this event in an appropriate manner.</p> <p>The only purpose of the start and end of service messages is to alert members about the window in which data will be disseminated by the interface.</p>

#### 9.4.6 Host Fail-over

Internally, the Xetra Enhanced Broadcast Solution platform is distributed across multiple host machines. During normal processing, each set and its associated instruments are processed by a single host, and it is this host that generates the Xetra Enhanced Broadcast Solution messages for these instruments.

---

If a host fails, responsibility for the instruments managed by that host is assigned to one or more of the remaining hosts, which also takes over the broadcasting of Xetra Enhanced Broadcast Solution information for the affected instruments. This is referred to as a fail-over.

When the failed host is returned to service, it continues work under a new sender/source ID. It does not continue to work under its former sender/source ID. With the failed host returning back to service a fail-back occurs which sees responsibility for the fail-over instrument returning to the original host, but with a new sender/source ID. A fail-back will not always occur as soon as a failed host returns to service. It may be delayed until a transaction for a fail-over instrument occurs.

During such a fail-over or fail-back, depending on the mode of failure, it is possible for there to be a certain amount of “flutter” as responsibility is passed forth and back to clear queued transactions.

For technical reasons, it is also possible – though very rare – for this to occur during normal processing, without a node failure having occurred. This is an extremely rare occurrence, typically caused by a transient network-related problem within Xetra.

This information is important for client programmers because all Xetra Enhanced Broadcast Solution sequence numbers are maintained on a per-host basis. This applies to all fields identified as "Sequence Number Type" in the message descriptions (see Data Field Dictionary Appendix in this document). So on the fail-back with a new sender/source ID for the returning host, the sequence numbers start again. The sequence numbers do not continue at the last sequence number before the fail-over.

Hosts are opaquely but uniquely identified to members through the field *srclid*, present in all Xetra Enhanced Broadcast Solution messages. It is important that clients maintain sequence numbers only in the context of a single source identifier. It is never appropriate to perform any operation (such as a comparison) on sequence numbers from different sources.

During a change of host (either as a result of a fail-over or a fail-back), a number of events occur that require special handling by the client code:

- Messages for the affected instruments start arriving with a different source identifier.
- The sequence numbers received from the new host no longer follows the sequence established by the previous host.
- They will typically start from 1, but no assumption must be made about this behaviour as there are certain circumstances under which it is not true.
- If a sequence number of an instrument is 1, a full delta/incremental is delivered. In this case all data must be set to invalid and the book is empty, before processing the message. No further processing (synchronization, etc.) is necessary for this instrument. This also occurs in relationship to a technical market reset. In this case no *srclid* change happens, but the instrument sequence number starts with 1. In such a case, with handling the message as described above, it is not necessary to synchronize with the snapshot stream.

The only correct response to these events is for the client to invalidate its view of the order book until an explicit message has been received containing new information. This is a recovery from snapshots.

To summarize the basic guidelines:

1. Messages generated by one sender/source id must never be mixed with messages from another sender/source id.
  2. By inference, an order book must only be constructed of messages from the one and the same sender/source id.
-

- 
3. Any change in the sender/source identifier must be treated in the same way as an unexpected backward jump in delta sequence numbers, the client's view of the order book must be considered invalid, and must not be relied upon until a valid value has been received from the new sender/source id.
  4. Sequence numbers of all types only have meaning within the context of a single sender/source id. Failure to adhere to these guidelines will result in an unpredictable order book under fail-over respectively fail-back conditions.

#### **9.4.7 Market Reset**

In case of a technical market reset (see also Market Reset message) the sequence number of an instrument restarts with 1 and a snapshot (full incremental) is provided in the delta/Incremental stream. No further synchronization with the snapshot stream is required. In this case the order book must be emptied before processing this message.

## 10 Appendix - Message Encoding

This chapter explains the FAST concepts important for understanding the Xetra Enhanced Broadcast Solution data encoding logic. The Xetra Enhanced Broadcast Solution encoding adheres to the FAST 1.1 specification.

Please note that FPL's Market Data Optimization Working Group may in future modify the FAST concepts. Therefore it is important to describe/understand the FAST concepts which the Xetra Enhanced Broadcast Solution interface is based on.

### 10.1 FIX Adapted for STreaming<sup>SM</sup> (FAST Protocol<sup>SM</sup>)

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 the link <http://www.fixprotocol.org/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 Enhanced Broadcast Solution.

#### 10.1.1 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: <code>&lt;uInt32 name="timestamp" /&gt;</code>
Group	A group is a named type comprising an unordered set of fields. A group appearing at the top most level of a stream is also referred to as a message.

Term	Explanation
Instructions	<p>There are two categories of instructions:</p> <ul style="list-style-type: none"><li>• Field instructions specify how to encode fields of the instance to the stream.</li><li>• Template reference instructions provide the means for defining parts of a template by reference to other templates.</li></ul> <p>Encoding and decoding are performed with in the context of an instruction.</p> <p>An instruction context consists of:</p> <ul style="list-style-type: none"><li>• set of templates</li><li>• current template</li><li>• set of application types</li><li>• current application type</li><li>• set of dictionaries</li><li>• optional initial value</li></ul> <p>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.</p>
Presence map	<p>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).</p> <p>When present, bit n indicates whether the corresponding field in the template is sent ('1') or not ('0').</p>
Primitive field	<p>A field that is not a group or sequence and can have a field operator. This operator specifies the optimization operation for the field.</p>
Primitive type	<p>Are ASCII strings, Unicode string, ulnt32, int32, ulnt64, int64, decimal (8 bytes mantissa plus 4 bytes exponent) and byte vector.</p>
Segments	<p>Consist of the following:</p> <ul style="list-style-type: none"><li>• segment identifier</li><li>• presence map</li><li>• set of field instructions</li></ul> <p>In general, a segment is a complete set of information. A FAST message is the highest level segment. A segment continuation 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.</p>
Sequence	<p>Comprises a length and an ordered set of elements. Each element is of group type and must not have identical group types.</p>



---

Term	Explanation
Template	Specifies how to encode an instance of an application type or a part thereof as a stream of bytes.

## 10.2 Field Instructions

Field instructions refer to how the fields will be sent in the information stream. For further information, please refer to the FAST Specification Version 1.1:

<http://www.fixtradingcommunity.org/pg/file/fplpo/read/43062>

## 10.3 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 earlier. 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.

## 10.4 Transfer Encoding

Transfer encoding refers to how the fields will be encoded in an information stream.

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

---

---

## 11 Appendix – The Xetra Member Homepage

There are various ways in which the Xetra Enhanced Broadcast Solution message layouts could be described. FAST Market Data Optimization Working Committee provides message templates in many layouts such as Compact Notation, XML notation, etc.

We describe the Xetra Enhanced Broadcast Solution message templates in XML format. Please note that:

- These are only additional formats of message representation and the tabular representation of messages described in section 5 (Structure of Messages) should be considered as master layouts in case of discrepancies.
- Fields not explicitly mentioned as optional are mandatory, unless they are suppressed by FAST compression.

### Xetra Member homepage

The current and latest version of the templates in XML format is provided on the Xetra Member homepage; find the Xetra Member Homepage here:

- Xetra Webpage  
[www.xetra.com](http://www.xetra.com)

Please visit there:

Xetra > Documentation > Xetra 16.0 > Interfaces > Enhanced Broadcast Solution

---