Eurex Exchange's T7 Release 3.0

Eurex Enhanced Order Book Interface

Manual

Version	v3.0.100
Date	03. August 2015

Eurex Exchange's T7 Release 3.0

Eurex Frankfurt AG

Eurex Enhanced Order Book Interface

v3.0.100

Contents

1 List of Abbreviations

The following are the abbreviations and definitions used in this manual:

Eurex EOBI	Eurex **E**nhanced **O**rder **B**ook **I**nterface
Eurex EMDI	Eurex **E**nhanced **M**arket **D**ata **I**nterface
Eurex ETI	Eurex **E**nhanced **T**rading **I**nterface
Eurex MDI	Eurex **M**arket **D**ata **I**nterface
Eurex RDI	Eurex **R**eference **D**ata **I**nterface
Eurex RDF	Eurex **R**eference **D**ata **F**ile
FIX	**F**inancial **I**nformation e**X**change. The Financial Information eXchange ("FIX") Protocol is a series of messaging specifications for the electronic communication of trade-related messages.
Out-of-Band	Incremental-messages and Snapshot-messages are delivered on different multicast channels.
Live - Live	Concept whereby data is disseminated simultaneously via two separate channels called "Service A" and "Service B".
BBO	**B**est **B**id and **O**ffer (can refer to price and size).
CRE	**C**ommon **R**eport **E**ngine
Match Step	Product-wide day-unique identifier for each price level of the match event.
Potential Auction Price	If the order book becomes crossed during an auction, then a potential auction price is formed and communicated to all participants.
T7	Eurex Exchange's trading system developed by Deutsche Börse Group.
IPS	**I**nter **P**roduct **S**preads, realised as Complex Instruments with multiple legs, in general belonging to a set of future products.
OCO	One-Cancels-the-Other order.
Eurex EOBI Channel	Eurex EOBI snapshot and incremental feeds consist of several channels, each of which delivers the information for a group of products. All channels are sent on two different multicast addresses via different physical connections (Service A and Service B). Both services are identical in terms of the information provided.

This manual uses conventions to highlight certain words and phrases and draw attention to specific pieces of information.

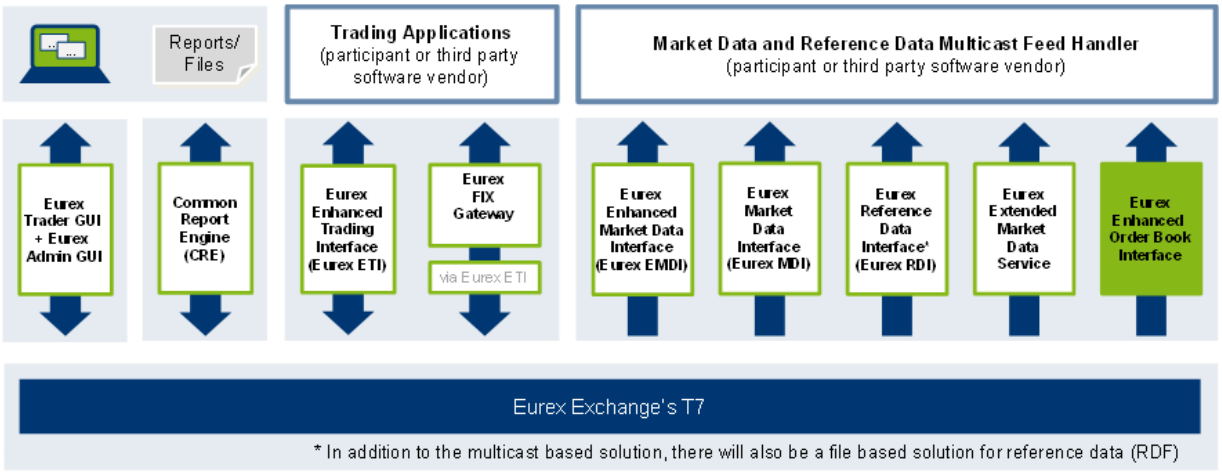
Therefore, all message names related to Eurex Enhanced Order Book Interface feeds are in `fixed width font like this` and all field names are in *italic* to separate them from ordinary text. **Bold** highlighting will be used when a new term is introduced, or to emphasize the importance of a word.

2 Introduction

The **Eurex** **E**nhanced **O**rder **B**ook **I**nterface (Eurex EOBI) provides the entire visible order book, by publishing information on each individual order and quote side, along with executions and state information in real-time and in an un-netted manner. The interface is available for a selected group of Eurex benchmark Futures products and all IPS products (see Appendix 9.1 - "Product Scope"), and provides an alternative to recipients of the Eurex Enhanced Market Data Interface (Eurex EMDI).

Though most of the functional concepts used are similar to those of Eurex EMDI, the interface provides greater transparency and efficiency, together with a high throughput at minimal latency. The Eurex EOBI disseminates public market data with the following features:

- A full order depth feed; there is no depth restriction.
- Information is sent in form of fixed-length binary messages.
- Intelligent packing of messages into a datagram by including repetitive entities only once in a message.
- Utilization of the widely adopted FIX standard to decrease integration efforts and on-going support costs.
- Corresponding reference data information is available via the existing Eurex Reference Data Interface (Eurex RDI) and the Eurex Reference Data Files (Eurex RDF).
- Dissemination of incremental messages (following state changes) and all Snapshot messages follow a publishing in sequence based on:
    1. Side (bid first, offer second),
    2. Price (best price first),
    3. Time (highest time-priority first).



Picture 1: Interface Landscape of Eurex's T7

As depicted above, the interface provides an additional market data interface alongside the existing Eurex EMDI and Eurex MDI interfaces.

The Eurex EOBI is designed for participants that rely on **low-latency** at a high throughput with a **high band-width network.** The interface disseminates all visible orders and quotes without any depth restriction, when the order books are open, along with order executions and state information via incremental messages in un-netted manner. Furthermore, snapshot messages always carry existing visible orders and quotes without any depth restriction at the time of sending.

Multicast address and port combinations of Eurex EOBI are different from Eurex EMDI and Eurex MDI.

2.1 Purpose of this document

The purpose of this document is to provide guidance for programmers developing applications that receive public market data from the Eurex EOBI feeds.

It covers a complete reference, describes the general business behavior and provides concepts for the implementation.

The most recent version is available at:

www.eurexchange.com > Technology > T7 > System Documentation > Release 3.0 > Market and Reference Data Interfaces

2.2 Document Outline

The following chapter, Chapter 3 - "Characteristics" gives an overview of the functional and technical features of the Eurex Enhanced Order Book Interface.

Chapter 4 - "Order Book Management" outlines the availability of messages, the initial build-up of the order book, and the processing of order book updates.

Chapter 5 - "Eurex's T7 Timestamps" provides an overview of timestamps referred throughout the document.

Chapter 6 – "Availability of Enhanced Order Book Service" presents the availability of the Enhanced Order Book Interface according to the state of trading during the day.

Chapter 7 – "Message Formats" outlines the general structure of messages sent out over the Eurex Enhanced Order Book Interface, followed by the specific individual message layouts in Chapter 8.7 – "Message Layout".

Chapter 8 – "Appendix" describes:

- the product scope of this interface, (see Appendix 9.1 - "Product Scope"),
- where synthetic pricing information can be found, (see Appendix 9.3 - "Synthetic prices"),
- how public market data and private data can be synchronized between Eurex EOBI and Eurex Enhanced Transaction Interface (Eurex ETI). See Appendix 9.4 - "Connecting Eurex EOBI and Eurex ETI data",
- how the reference data can be extracted from Eurex RDI and/or Eurex RDF (see Appendix 9.6 - "Reference data for Eurex EOBI").

2.3 Further Reading Material

Eurex recommends participants to be familiar with the concepts described in the following documents:

- Eurex Functional Reference
- Eurex Functional and Interface Overview
- Eurex Market and Reference Data Interfaces Manual
- Eurex Extended Market Data Service Manual
- Eurex Trading Interface Manual
- Fixed-length binary messaging related documentation

Eurex related documents are available at:

www.eurexchange.com > Technology > T7 > System Documentation > T7 > Release 3.0

FIX-messages and FIX-tag related information is available at:

www.fixtradingcommunity.org > FIX Application Layer

www.fixtradingcommunity.org > FIXimate

Also FIX wiki (same as FIXimate but with additional annotations) is available at:

www.fixtradingcommunity.org > FIXwiki

3 Characteristics

The Eurex EOBI is based on the same concepts as the Eurex EMDI. However, there are some functional and technical characteristics that distinguish the Eurex EOBI from Eurex EMDI.

3.1 Functional Characteristics

The Eurex EOBI disseminates:

- The instrument identifier, side, price, priority timestamp and quantity of each visible order and quote side.
- Trade price and traded quantity for each executed on-exchange trade.
- Order book information disseminated without any depth limitation.
- The trading status of each product and corresponding instruments.
- Intra-day changes regarding complex instruments.
- Request for Quotes and Cross Trade Announcements.
- Manually entered trades and trade reversals by Eurex Market Supervision.
- Recovery via Eurex EOBI snapshots.

Each order and quote can be **uniquely** identified by the combination of instrument identifier, side and priority timestamp.

In order to send public market data as fast as possible, the Eurex EOBI publishes only very specific market information. However, participants can derive certain information themselves based on the messages sent out by the Eurex EOBI. The following information is not explicitly provided, however can be derived, if needed (from here onwards the term "order" is used to refer both to orders and quotes):

- Price levels; can be derived from individual orders.
- Aggregation at price levels; can be derived from individual orders.
- Information about synthetic prices; can be derived from visible orders received on the Eurex EOBI feed.
- Fully matched incoming visible orders; can be derived from execution messages.
- Trade statistics are not provided via the incremental channel to keep the size of messages as small as possible. They can be derived from the order execution messages sent out on the Eurex EOBI incremental channel. But, on the other hand, trade statistics are sent out on the Eurex EOBI snapshot channel for recovery purposes.

3.2 Technical Characteristics

The Eurex EOBI contains similar technical characteristics as the Eurex EMDI, such as "Live - Live" multicast, distribution mode and sequence numbering schemes. Anticipating a high load, the size of messages is kept as small as possible.

The following are highlights of the technical characteristics of the Eurex EOBI :

- Low-latency multicast for data dissemination with "Live - Live" concept.
- Fixed length optimized message layouts without any compression.
- Uses push-based publishing model in Out-Of-Band distribution mode.
- Packet and message sequence number schemes (same as on the Eurex EMDI feed). However, as opposed to Eurex EMDI/MDI/RDI, the field *MarketSegmentID,* i.e., the product identifier will be used instead of *SenderCompID* in packet header. Therefore, the *SenderCompID* will be absent in both the Packet Header and the Message Header.
- Matching Engine-In timestamp, *TrdRegTSTimeIn*, as part of orders.
- Little Endian and basic data types are used.

- Message padding for better byte alignment.
- Recovery via Eurex EOBI snapshot channel as similar to Eurex EMDI.

All messages are designed to be as small as possible and are following FIX 5.0 SP2 semantics. The maximum number of bytes per transmission unit (MTU) is limited to 1372 bytes.

The rule for the **distribution sequence** across partitions is as follows:
Even partitions : Publish on Service A first, then on Service B.
Odd partitions : Publish on Service B first, then on Service A.

The above rule is applied by using the field *PartitionID*. It is available in the `Product Snapshot` message via the Eurex RDI and Eurex RDF and in the packet header of Eurex EOBI.

All functional and technical reference data information needed for the Eurex EOBI is provided by the Eurex RDI and/or Eurex RDF, similar to the current procedure for the Eurex EMDI. The multicast addresses and ports for both services are disseminated in the product reference information. Multicast addresses and port information don't change during trading hours. See 9.6 - "Reference data for Eurex EOBI" .

4 Order Book Management

The Eurex EOBI provides an **explicit** message for each order book update by publishing the instrument identifier, side, price, displayed quantity, priority timestamp and "matching engine in" timestamp of each visible order in the entire order book, along with the order execution and state information. The order book information will be published for all IPS products and Future products which are enabled on Eurex EOBI. As described earlier, each order is uniquely identified by the combination instrument identifier, side and priority timestamp.

An outline of the **visibility** of orders on the Eurex EOBI is shown below:

Order Type	Visible in Order book
Regular Limit Order	yes
Quote	yes
Triggered Order – Closing Auction Only	no
Triggered Order – Stop Limit Order	yes
OCO	yes
Regular Order – GFD / GTC / GTD	yes
All types of Market Orders	no
Stop Market Order (un-triggered)	no
Stop Limit Order (un-triggered)	no
Regular Limit Order – IOC	no
All types of Rejected Orders	no

Table 1 - Visibility of orders on the Eurex EOBI

For each instrument within a product, snapshot messages can be received via the Eurex EOBI snapshot channel to build the initial order book. Once the initial order book is built, the order book must be maintained using the corresponding order book updates received on the Eurex EOBI incremental channel. On the Eurex EOBI incremental channel, order messages are used by participants to maintain the order book, while explicit state change messages are provided to communicate current product and instrument state. Intra-day complex instrument changes will also be communicated via the Eurex EOBI incremental channel.

To assist fine filtering and error discovery on the participant side, the Eurex EOBI keeps messages in line using a multi-sequencing paradigm. It uses the following two sequencing methods: **packet sequence number** and **message sequence number**.

Packet Sequencing

Each packet on the Eurex EOBI feeds is sequenced using contiguous packet sequence numbers. The packet sequence number is incremented for each packet across products on the same feed.

Message Sequencing

In addition to packet sequencing, each product on the Eurex EOBI feeds is sequenced contiguously by using message sequencing. This should allow participants to filter products of interest only. The message sequence number is incremented per product across the different message types.

The following sections describe the order book management with respect to the messages sent over the Eurex EOBI.

Message layouts can be identified by the *templateID* field which is the (exchange wide) unique identifier for the message layout, and is included in each Message Header. The *templateID* also determines the fixed size of the message.

Message	Template ID
Order Add	13100
Order Modify	13101
Order Modify Same Priority	13106
Order Delete	13102
Order Mass Delete	13103
Partial Order Execution	13105
Full Order Execution	13104
Execution Summary	13202
Auction Best Bid Offer	13500
Auction Clearing Price	13501
Top Of Book	13504
Product State Change	13300
Instrument State Change	13301
Cross Request	13502
Quote Request	13503
Add Complex Instrument	13400
Trade Report	13201
Trade Reversal	13200
Product Summary	13600
Instrument Summary	13601
Snapshot Order	13602
Heartbeat	13001

Table 2 - Eurex EOBI messages with assigned template IDs

4.1 Building the Order Book

Product and instrument reference data information required to process the Eurex EOBI market data is provided by the Eurex RDI and/or Eurex RDF, similar to the current procedure for Eurex EMDI, also see Appendix See 9.6 - "Reference data for Eurex EOBI".

Messages in the Eurex EOBI snapshot channels are grouped by product. In order to build an initial order book, participants subscribe to the Eurex EOBI snapshot channel. The content of one **snapshot cycle** for one product is denoted in Picture 6 (see 7.3 Snapshot Messages). The individual orders in the order book are represented in the snapshot message using the `Snapshot Order` messages. The snapshot messages contain the field *LastMsgSeqNumProcessed* to enable participant synchronization between the Eurex EOBI snapshot channel and the Eurex EOBI incremental channel.

While subscribed to the Eurex EOBI snapshot channel, participants should keep processing incoming data from the Eurex EOBI incremental channel. Any incoming incremental messages with a sequence number higher than the *LastMsgSeqNumProcessed* received in the snapshot message should be applied to the order book after the full snapshot message is processed.

The following data is provided via the Eurex EOBI snapshot channel:

- Product State information,
- Instrument State information,
- Trade Statistics per instrument,
- All visible orders in the order book.

During the Continuous Trading instrument state, all visible orders in the order book will be published on the Eurex EOBI incremental channel.

During the Auction instrument state, the Eurex EOBI snapshot channel will broadcast auction information, Best Bid and Offer (BBO) or the auction clearing price (indicative auction price),

only. After the auction phase, trades that took place during the auction will be published using the `Trade Report` messages, before the corresponding state changes.

As soon as trading is in the state Continuous, all visible orders in the order book will be immediately published on the Eurex EOBI incremental channel.

The sequencing of the data in a snapshot cycle is based on the product identifier, the instrument identifier and on the price level. For the product and instrument identifier, the **sending order sequence** is ascending and the orders are sorted from best to worst prices (buy orders are sorted from highest to lowest, and sell orders from lowest to highest).

The visible orders are sent alternating between buy and sell sides, where orders at the same price level are sorted by order time priority from the oldest to the newest order. The visible order book is disseminated per price level in a zig-zag manner, meaning both the sides (Bid and Offer) at each price level are disseminated before moving on to the next price level. If one side providing more orders on the same price level as the opposite side, all orders of the same price level are processed before switching to the next price level.

Assuming the following arbitrary order book is sorted according to imaginary order priority timestamps and order prices where in the orders with the same order prices are sorted according to imaginary order priority timestamps.

Picture 2: Order book in a zig-zag manner

As it can be seen from table above, the orders denoted by B1, B2 and S1 are on the first price level. The orders denoted by B3, S2, S3 and S4 are on the second price level. The orders B4 and S5 are on the third price level. In price level fourth and fifth buy orders exists only.

The resulting sending order sequence in zig-zag fashion is: B1, S1 and B2, B3, S2, S3, S4, B4, S5, B5 and B6.

The following data is not provided via the Eurex EOBI snapshot channel:

- Cross Trade Announcements / Cross Requests,
- Requests for Quotes,
- Intra-day created complex instruments,
- Manually entered trades by Eurex Market Supervision,

These messages will be published on the Eurex EOBI incremental channel only. The snapshot messages will contain all order book information about the intra-day created complex instruments. Please note that the intra-day created complex instruments are published on the Eurex RDI snapshot and incremental channel as well.

4.2 Adding an Order

An `Order Add` message will be sent each time a visible order is added to the order book of the corresponding instrument. The message includes the instrument identifier, priority timestamp, side, price, displayed quantity of the order and its matching engine in timestamp.

The `Order Add` message includes among other the priority timestamp and side, which are to be used as the instrument-wide **unique identifier** of this order, as long as the order is not modified. See 4.4 - "Modifying an Order". The instrument identifier, priority timestamp and side will be the reference key for all future updates for the order. See 4.3 - "Identifying an Order".

Information about an incoming order, that matched fully against to one or more orders in the order book, can be derived from the associated execution messages or execution summary only.

The remaining part of an incoming order that matches partially will be reported with an `Order Add` message after all associated executions.

The `Order Add` messages also include the "matching engine in" timestamp (conveyed by *TrdRegTSTimeIn*) of the order, which conveys when the corresponding order transaction has been received by the matching engine. Please note that *TrdRegTSTimeIn* will not be set in case of a self triggered transaction (without any external actor). For ex. a state change resulting in opening of the book.

4.3 Identifying an Order

Participants are able to identify their own orders on the Eurex EOBI by using the unique identifier, the priority timestamp and side, as stated earlier.

In order to provide participants with the priority timestamp of the orders, the field *TrdRegTSTimePriority*, will be provided in the Order messages of the Eurex EOBI and in the Eurex ETI responses.

In order to identify matching of own orders, the priority timestamp of the order and a unique match step identifier for each price level of the match event will be provided, i.e., the fields *TrdRegTSTimePriority* and *TrdMatchID* in the execution messages in the Eurex EOBI will correspond to the field *TrdRegTSTimePriority* and *FillMatchID* and/or *QuoteEventMatchID* in the Eurex ETI (See Appendix 9.4 - "Connecting Eurex EOBI and Eurex ETI data").

4.4 Modifying an Order

If the time-priority, price and/or displayed quantity of an existing order changes, then an `Order Modify` or `Order Modify Same Priority` message will be sent.

A modification might result in the order being assigned a new priority timestamp (for example, in the case of a price modification). If it is the case, then an `Order Modify` message will be sent. The old priority timestamp will be given by *TrdRegTSPrevTimePriority*, whereas the new priority timestamp will be given by *TrdRegTSTimePriority*. Henceforth the new *TrdRegTSTimePriority* along with the side will be the new unique key for the order in the future.

Please note that, change in the priority timestamp might also occur due to change in non-visible order attribute e. g. modification of stop price of an OCO order. In such a case, there will be no change in price and quantity hence *PrevPrice* will contain the original price and *PrevQuantity* will contain the original quantity. Please refer to Eurex Functional Reference documentation for further details.

However, if there is no priority loss with the modification (which may occur for example when quantity is reduced) then the `Order Modify Same Priority` message will be sent and *TrdRegTSTimePriority* field will contain the original order priority-timestamp.

"Matching engine in" timestamp, *TrdRegTSTimeIn*, of the order will be amended accordingly.

4.5 Deleting an Order

When an order is deleted, the Eurex EOBI will publish the instrument identifier, the priority timestamp of the order, side, price and transaction time , i.e., the fields *SecurityID, TrdRegTSTimePriority, Side, Price* and *TransactTime,* which will enable participants to quickly identify and delete the corresponding order from the order book. The "matching engine in" timestamp of delete request, *TrdRegTSTimeIn*, will be provided as well. Please note that *TrdRegTSTimeIn* will not be set in case of a self triggered transaction (without any external actor). For ex. an instrument expiry causing an order book clean up.

4.6 Order Executions

In order to ease the processing of matches along with the other order book updates by participants the following information is disseminated for each match corresponding to an incoming order:

- first, an execution summary message will be sent when an incoming order has been matched against orders that were already in the order book,
- second, messages that convey the individual executions of visible orders are published[1].

The `Execution Summary` message contains the instrument identifier, side, aggressor time-stamp, gateway in time-stamp indicated by *RequestTime*, worst price, total executed quantity, resting hidden quantity (if any) and match-time information of the incoming order. Please note that, aggressor time-stamp and gateway in time-stamp are provided only for executions triggered by an incoming order.

For conveying the individual executions of the visible orders two template messages will be used for fully and partially executed orders.

In a sense, the `Execution Summary` messages can be used by participants for fast trading decisions. The individual order execution messages should be used by participants for order book maintainance to ensure the correctness of the order book.

However, it should also be noted that, the `Execution Summary` message will **not** be published in the case a match is not triggered by an incoming order. It is illustrated by the following use case.

After an opening auction is committed, all simple instrument order books are published and then the spread order books are uncrossed. A spread order book may cross against a simple order book leading to synthetic matches with full or partial order executions reported on simple instrument order books. The `Execution Summary` message will not be sent in this case, because executions are not triggered by an incoming order. Additionally, the orders on simple instrument order books could be matched at a price which is different from the displayed price. This information is conveyed by fields `Price` and `LastPx` in full or partial order executions messages. `Price` informs the price at which the order was entered into the book, whereas `LastPx` indicates the price at which it was matched.

[1] That implies individual executions are not sent if an incoming order matches against non-visible orders, i.e. if an incoming order matches against a market order or if a synthetic match happens.

The order execution messages will be sent whenever a visible order is **fully** or **partially** executed at its displayed price (except for uncrossing scenario described above) . Each **match step** will include a **product-wide day-unique identifier** of the trade, represented by the field *TrdMatchID*. This field will always have a value in the execution messages for a full or partial execution. The same unique identifier of the trade is made available to participants by the Eurex ETI.

If the incoming order has been partially executed, then the remaining quantity will be reported with an `Order Add` message after all associated individual executions have been provided.

Triggered Stop Market orders or Stop Limit orders are reported like incoming Market or Limit orders, respectively.

4.7 Trade Statistics

Instrument trade statistics such as opening, closing, daily low and high prices are available via the Eurex EOBI snapshot messages only. They are provided to participants for recovery purposes and are published included in the `Instrument Summary` message on the Eurex EOBI snapshot channel. By design, they are provided as a repeating group as part of the `Instrument Summary` message and are not cut off.

When subscribed to the Eurex EOBI incremental channel, participants can derive order book and trade statistics by combining the information received via the order and execution messages.

4.8 Auctions

The visibility of the order books is limited during an auction. When an instrument goes to an auction state, an `Instrument State Change` message is immediately published via the incremental channel.

Auction information will be published via `Auction Best Bid Offer` and `Auction Clearing Price` messages, which will carry either the Best Bid Offer prices for uncrossed order books or the potential auction price for crossed order books respectively. No other order book information is available during Auctions. Quantities and depth information are not published during auction state.

An **uncrossed** order book is identified by means of Top-Of-Book prices that are published by `Auction Best-Bid Offer` messages. A **crossed** order book is identified by means of `Auction Clearing Price` messages. So, the change from a crossed to an uncrossed book situation and vice versa is implicitly identified by sending Top-Of-Book information instead of an auction clearing price and vice versa. Both messages, `Auction Best-Bid Offer` and `Auction Clearing Price`, are mutually exclusive.

On the snapshot channels, the auction instrument state is reflected in the `Instrument Summary` message along with the trade statistics. In order to provide an empty book situation in the snapshots, an `Auction Best Bid Offer` message will be present even when the corresponding order book does not contain any Best Bid Offer prices.

When an auction closes, i.e., an instrument leaving an auction, an `Instrument State Change` message is immediately published via Eurex EOBI incremental channel. There could also be an optional `Trade Report` message published before `Instrument State Change`, for the trades those have occured during the auction phase.

As soon as trading is in the state Continuous Trading, all visible orders in the order book will be published on the Eurex Order Book incremental channel in a zig-zag manner. There will be no explicit message clearing any previous sent Top-Of-Book prices or Auction Clearing Price during the auction phase. `Product State Change` messages and `Instrument State`

`Change` messages will be used to publish status changes of tradable products and corresponding instruments.

4.9 Product and Instrument States

In a `Product State Change` message, the product state can normally be found in the field *TradingSessionSubID*. Only for quiescent product states, the field *TradingSessionID* must be evaluated additionally to determine the actual product state.
A Halt state is additionally indicated by the field *TradSesStatus* containing the value "*1 = Halted"*.
A Fast Market is reported with the same message type using the field *FastMarketIndicator* which can take the values "*0 = No"* or "*1 = Yes"*.
The instrument state is published with an `InstrumentStateChange` message and can be found directly in the field *SecurityTradingStatus*.
Please note that, `ProductStateChange` message will not be published for IPS products. However, the instrument state for an IPS instrument will be published by `InstrumentStateChange` message . That means, the fields *TradingSessionID* and *TradingSessionSubID* will not be set in `ProductSummary` message on Eurex EOBI snapshot channel.

The status of the instrument (as opposed to the instrument state) distinguishes active, suspended and inactive instruments and is contained in the field *SecurityStatus*.

4.10 Intra-day Created Complex Instruments

`AddComplexInstrument` message will be used to publish complex instruments created intra-day.

A full description of intra-day created complex instruments is available via the Eurex RDI and/or the Eurex RDF only.

Empty book information for the intra-day created complex instruments is sent for any order maintenance activity.

If a participant's application has a late start and the application uses the "Start-Of-Day" public reference data without applying the intra-day created complex instruments, then order book data may be received for unknown instruments.

4.11 Manual Trade Entry and Trade Reversal

The Eurex EOBI reports all on-exchange trades. In addition to order book trades, participants receive trade messages for trades or trade reversals that are manually entered by Eurex Market Supervision.

A manually entered trade will not affect the price statistics. Even if the manually entered trade price is higher than the daily high price, it does not change the daily high price.

In case of a manually entered on-exchange trade by Eurex Market Supervision, participants will be informed via a `Trade Report` message.
A trade can only be reversed by Eurex Market Supervision for its full quantity. For such a trade reversal, participants will be informed by a `Trade Reversal` message.
The `Trade Reversal` message consists of quantity, price, match event identifier and timestamp when the trade reversal request was processed:

- *LastQty (32)* and *LastPx (31)* carry quantity and price of the reversed trade,
- *TradeMatchID (880)* carries the match event identifier of reversed trade,
- *TransactTime (60)* when the trade reversal request was processed.

Please note, that a `Trade Reversal` message will be sent only for trades which have been previously reported on the Eurex EOBI incremental channel. Additionally, the message may include other trade statistics such as the new last price, opening price, closing price, low price and high price, in case they are affected. The new trade volume will not be reported with the `Trade Reversal` message however, the same can be calculated by subtracting the quantity of the reversed trade from the existing trade volume.

4.12 Heartbeats

Functional heartbeat messages, `Heartbeat`, are sent at a regular interval for less active products on the Eurex EOBI incremental channels. A functional heartbeat message provides the message sequence number last sent in the field *LastMsgSeqNumProcessed* to allow participants to identify potential gaps. Heartbeats will be sent out as of the product state "Start-Of-Day.

Technical heartbeats will be provided on the specific ports assigned to technical heartbeat messages.

4.13 Recovery

Due to the unreliable nature of UDP multicast, UDP packets may be duplicated, delayed, missing, or arrive in an incorrect sequence. Therefore, Eurex EOBI uses a "Live - Live" concept, as in Eurex EMDI, for recovery purposes. Both live services (A and B) are sequenced identically and participants should ideally process both services to detect data losses at an early stage.

If a packet is lost on both (Live - Live) services of the Eurex EOBI incremental channel, then participants can take advantage of the **out-of-band** nature of Eurex EOBI. Participants can utilize the Eurex EOBI snapshot channel to obtain the corresponding lost information, i.e., rebuild the initial order book, determine trade statistics and instrument states. For recovery, participants should recover on a product level (i.e., for all instruments of one product), for two reasons:

- The field *LastMsgSeqNumProcessed* in the snapshot cycle is given on product level, so in order to synchronize the Eurex EOBI snapshot channel and the Eurex EOBI incremental channel, participants should recover for all instruments in the product.
- Given the fact that there is no explicit information on synthetic price and quantity in the Eurex EOBI, participants will have to re-determine the order books of all instruments to derive this information.

Detecting duplicates and gaps by means of the packet header

The packet header allows receiving applications to identify identical packets between service A and service B. This could be achieved by a simple memory comparison on the first 12 bytes of a Eurex EOBI datagram containing the *ApplSeqNum* as shown in 8.2 `Packet Header`. *BodyLen* and *TemplateID* will be constant for `Packet Header` and *MsgSeqNum* is always filled with 0xFFFFFFFF.

Please note, that packets have contiguous sequence numbers per EOBI channel (service A and service B). This means, that field *ApplSeqNum* can be used not only to detect duplicates but also to detect missing packets.
Please note, that EOBI channels are not shared between different partitions.

Participant Fail-Over

In the event of a packet loss on both (Live - Live) services of an Eurex EOBI channel, recovery on the participant side can be achieved by recovering the order book information via the Eurex EOBI snapshot channel.

The Eurex EOBI snapshot channel is synchronized with the Eurex EOBI incremental channel through the use of message sequence numbering. Participants should subscribe to the Eurex Order Book Snapshot channel while buffering incoming messages from the Eurex EOBI incremental channel. Any incoming message from the Eurex EOBI incremental channel with a *MsgSeqNum* higher than the value of the *LastMsgSeqNumProcessed* field received in the `Product Summary` snapshot message should be applied to the order books after the full product snapshot is processed.

Exchange Fail-Over

A fail-over of the Eurex EOBI incremental channel will be recognizable by the following features:

- The *ApplSeqNum* in the `Packet Header` is reset to 1.
- The *MsgSeqNum* in the `Message Header` will continue to be incremented contiguously (Ideally without any gap).

When a participant receives packets on a specific multicast address with an unexpected (lesser or equal) sequence number (either on service A or service B), it is advised, that the participant subscribes to the Eurex EOBI snapshot channel again to rebuild the order book. Note that, because of the unreliable nature of the UDP protocol, packets may arrive out of sequence.

For a full restart of the Eurex EOBI service, the Eurex EOBI incremental channel will have the same features as the Eurex EOBI snapshot channel:

- The *ApplSeqNum* in the `Packet Header` is reset to 1.
- The *MsgSeqNum* in the `Message Header` is reset to 1.

In case of a full restart of the Eurex EOBI service, participants must wait for the first message after the restart to be certain that a restart was executed. It is expected, that a full restart of Eurex EOBI feed will take much longer than the configured heartbeat interval.

The field *ApplSeqResetIndicator* is always set in the `Packet Header` of the first few incremental messages after a (re-)start.

5 Timestamps

The various Eurex Exchange's T7 timestamps mentioned throughout the document, are taken at high-frequency gateways, matching engines and market data servers both in production and simulation. They are also provided through messages sent on Eurex EMDI, Eurex MDI and Eurex EOBI feeds. These can be used to analyze one way transport times. To reiterate, all timestamps are in UTC, and represented as nanoseconds past the UNIX epoch (00:00:00 UTC on 1 January 1970).
An incoming transaction is timestamped at,

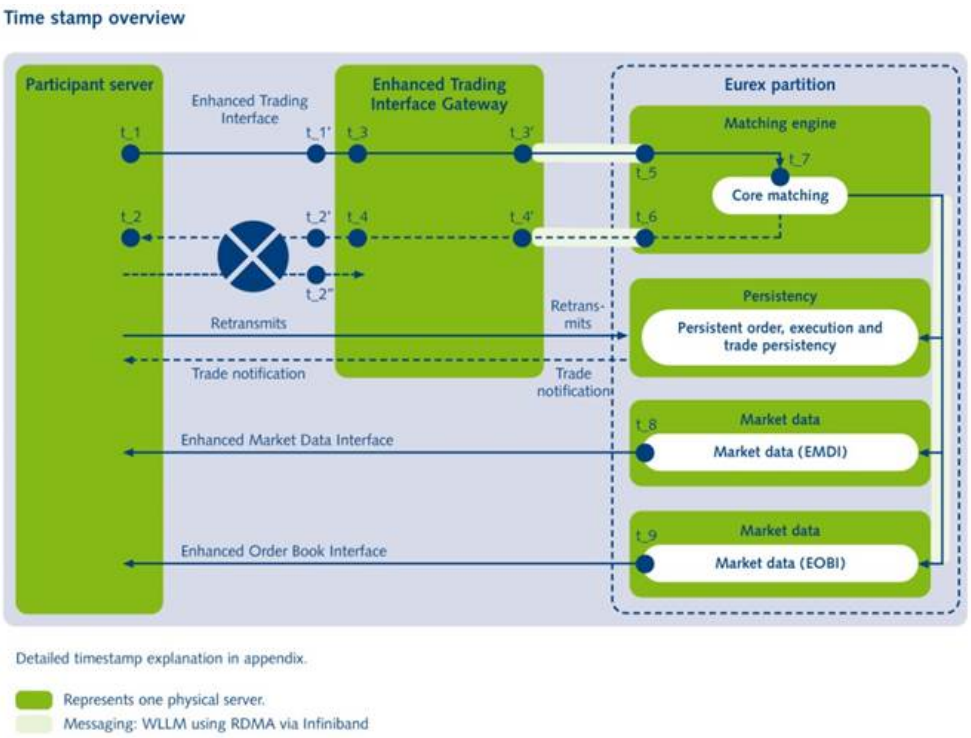
Matching Engine:

- order book maintenance and execution,
- creation of direct responses as well as execution messages all for passive orders and quotes,
- creation of listener broadcast for standard orders (see Eurex ETI Manual).

Market Data (Eurex EMDI, Eurex MDI and Eurex EOBI):

- *SendingTime* for order book delta and snapshot messages,
- addtionally timestamps from Matching Engine such as *PriorityTimestamp*, *TransactTime* etc. are provided on market data messages.

The following picture provides an overview of Eurex Exchange's T7 timestamps:



Picture 3: An overview of Eurex Exchange's T7 Timestamps

The following table lists the mapping of Eurex Exchange's T7 timestamps:

Timestamp	Semantic	FIX fields	Description
t_3	Gateway request in	RequestTime (5979)	Provides the time the Eurex application has read an inbound message on a gateway from the TCP socket.
t_3‘	Gateway request out	RequestOut (7764)	Provides the time the Eurex application has sent an outbound message from a gateway to the matching engine.
t_4‘	Gateway response in	ResponseIn(7765)	Provides the time the Eurex application has received an inbound message on a gateway from a matching engine.
t_4	Gateway response out	SendingTime (52)	Provides the time the Eurex application has written an outbound message on a gateway to the TCP socket.
t_7	Priority timestamp, Creation timestamp, Transaction timestamp, etc.	TrdRegTSTimePriority (21008), ExecID (17), TransactTime (60), etc.	Taken when a transaction is functionally processed and is unique per product. It could be seen in either of the FIX fields depending on if it corresponds to fresh order or quote transaction, strategy creation, execution or as transaction timestamp for others.
t_5	Matching engine in	TrdRegTSTimeIn (21002)	Provides the time the Eurex application has received an inbound message on a matching engine.
t_6	Matching engine out	TrdRegTSTimeOut (21003)	Provides the time the Eurex application has sent an outbound message from a matching engine.
t_8	Eurex EMDI out	SendingTime (byte vector)	Provides the sending time when Eurex EMDI has put the datagram on the wire.
t_9	Eurex EOBI out	TransactTime (60)	Provides the sending time when Eurex EOBI has put the datagram on the wire.

Table 3 - Mapping of Eurex Exchange's T7 timestamps

6 Availability of Enhanced Order Book Service

The Eurex EOBI is available during the entire business day between product states "Start-Of-Day" and "End-Of-Day".

Table 4 below shows the information typically sent on the Eurex EOBI during each product state. The messages listed in the table should serve as a super-set of messages and inform participants on "what-to expect" during each product state. However, it does not state any deterministic behaviour and should only be used as a guideline. The actual message set could be a sub-set of the listed messages depending on market conditions.

Product State	Messages
Start-Of-Day	Product State Change, Instrument State Change, Add Complex Instrument, Product Summary, Instrument Summary (incl. Trade Statistics), Heartbeat
Pre-Trading	Product State Change, Instrument State Change, Order Mass Delete, Add Complex Instrument, Product Summary, Instrument Summary (incl. Trade Statistics), Trade Report (manual entered Trades), Trade Reversal, Heartbeat
Trading	Product State Change, Instrument State Change, Add Complex Instrument, Add Order, Modify Order, Modify Order Same Priority, Delete Order, Partial Order Execution , Full Order Execution, Execution Summary, Auction Best Bid Offer (during Auction), Auction Clearing Price (during Auction), Cross Request, Quote Request, Heartbeat, Product Summary, Instrument Summary (incl. Trade Statistics), Snapshot Order, Trade Report (Manual entered Trades), Trade Reversal
Closing	Product State Change, Instrument State Change, Auction Best Bid Offer (during Auction), Auction Clearing Price (during Auction), Product Summary, Instrument Summary, (incl. Trade Statistics), Trade Report (Manual entered Trades), Heartbeat, Trade Reversal
Post-Trading	Product State Change, Instrument State Change, Order Mass Delete, Product Summary, Instrument Summary (incl. Trade Statistics), Top Of Book, Trade Report (Manual entered Trades), Trade Reversal, Heartbeat

End-Of-Day	Product State Change, Instrument State Change, Product Summary, Instrument Summary (incl. Trade Statistics), Top Of Book, Heartbeat
Post-End-Of-Day	-
Halt	Product State Change, Instrument State Change, Order Mass Delete, Product Summary, Instrument Summary (incl. Trade Statistics)
Holiday	Product State Change, Instrument State Change, Product Summary, Instrument Summary (incl. Trade Statistics), Heartbeat

Table 4 - Availability of Order Book Messages within Different Product States.

Please note that the Eurex EOBI snapshot channels stop after migration of all products to "Post-End-Of-Day".

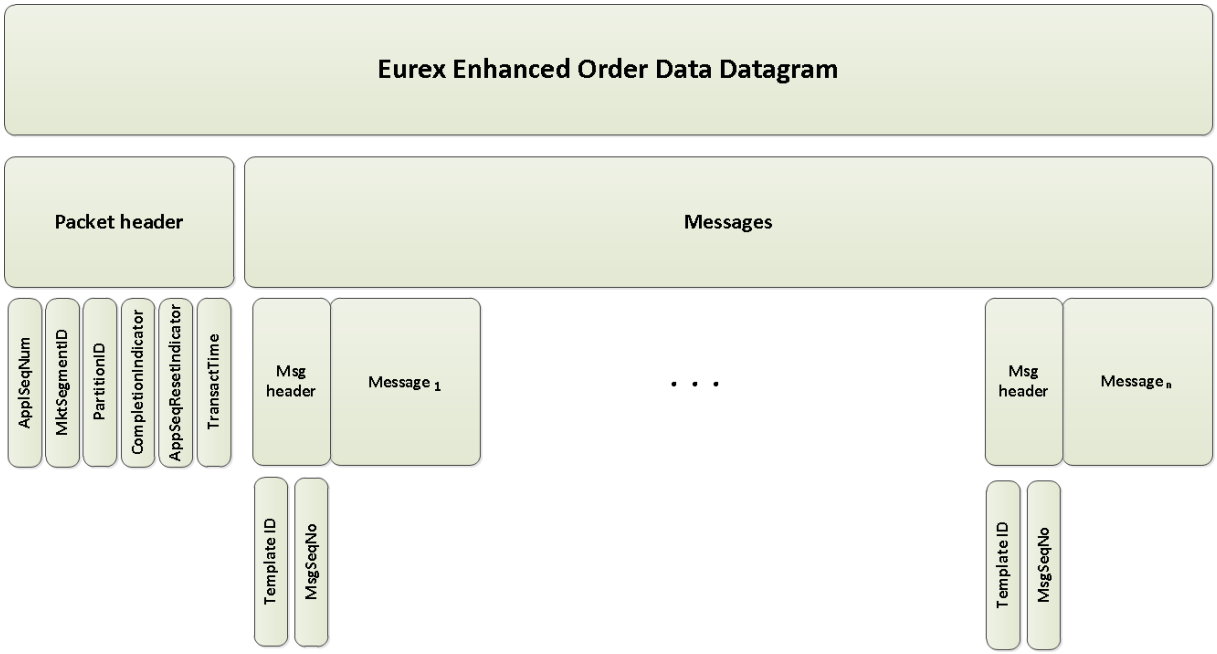
7 Message Formats

This chapter provides a global overview of the structure of datagram and message layouts and the data types used in these messages.

7.1 Datagram Structure

Each UDP datagram[1] starts with a `Packet Header` followed by one or more public market data messages and is terminated on the product level boundary, meaning that a datagram contains not more than order book updates for one product.

The Eurex EOBI follows the following structure for the datagrams sent on the network:



Picture 4: Generic Datagram structure of Eurex EOBI

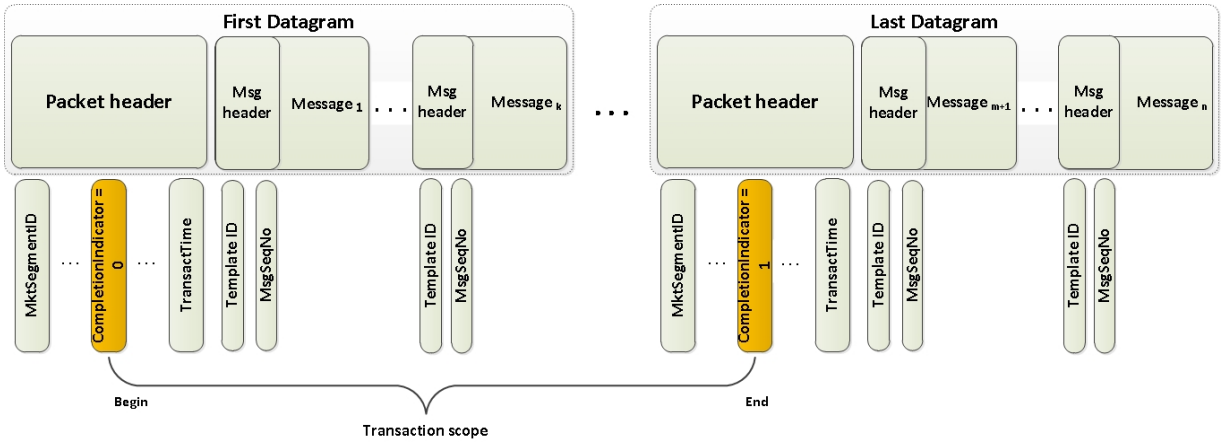
The `Packet Header` in each datagram contains information about

- The product and the partition ID of corresponding product,
- A contiguous packet sequence number,
- An indicator whether the **atomic unit of work** fits into one datagram,
- An indicator whether a fail-over has occurred, and
- When the packet has been sent out.

The product, *MarketSegmentID*, information can be used by participants for product filtering purposes.
The packet sequence numbers, *ApplSeqNum*, are contiguous and are incremented per Eurex EOBI channel (service A and service B). They can be used by participants to detect gaps, duplicate and missing packets (see 4.13 Recovery).
Please note, that EOBI channels are not shared between different partitions.
Furthermore, the `Packet Header` provides information whether the atomic unit of work that was processed by the corresponding matching engine fits into one datagram or is spread over

[1] Shortly called a datagram to ease the readability.

several datagrams. By design, a datagram will contain one atomic unit of work that was processed by the corresponding matching engine. However, if the resulting public market data of one atomic unit of work doesn't fit into one datagram due to datagram size restriction, then the resulting market data information is spread over several datagrams. In this case, as it is shown in the picture below, the completion flag, i.e., *CompletionIndicator*, in the first packet header of the first datagram is set to *Incomplete (=0)* and in the packet header of the last datagram is set to *Complete (=1)*. As a result, participants are able to gather all market data information belonging together.



Picture 5: Transaction scope spread over several datagrams

When the public market data fits into one datagram, the completion indicator in the packet header will be set to *Complete (=1)*.

The time when the datagram is sent out is provided by, *TransactTime.*

The functional structure of each Eurex EOBI datagram will always be the same; a message header will specify the fixed layout of the message content by a *templateID*, followed by a message sequence number of the corresponding product. Message sequence numbers, *MsgSeqNum*, contained in the Eurex EOBI incremental messages are incremented per product. Message sequence numbers for the Eurex EOBI snapshot messages are incremented per snapshot cycle.

The repeating groups in incremental and snapshot messages are not cut off.

7.2 Incremental Messages

Incremental messages are sent according to the Eurex EOBI datagram structure as described above.

A message header will indicate the fixed layout of the message content, followed by the actual messages.

There is **no well-defined sending order** for the incremental messages. However, the *template ID* in the message header identifies each incremental message uniquely.

Eurex EOBI incremental messages will be sent as long as the Eurex EOBI service is available. The `Heartbeat` messages are repeated in the configured heartbeat interval in a single datagram by setting the message sequence number last sent to the *LastMsgSeqNumProcessed* field of the corresponding product. If the *LastMsgSeqNumProcessed* is not available, i.e., until the product state "Start-Of-Day", then it is set to *"0"*.

As noted, if one atomic unit of work doesn't fit in one datagram, then the resulting market data information is spread over several datagrams. The completion flag will be used for this scenario.

Message	Template ID
Order Add	13100
Order Modify	13101
Order Modify Same Priority	13106
Order Delete	13102
Order Mass Delete	13103
Partial Order Execution	13105
Full Order Execution	13104
Execution Summary	13202
Auction Best Bid Offer	13500
Auction Clearing Price	13501
Top Of Book	13504
Product State Change	13300
Instrument State Change	13301
Cross Request	13502
Quote Request	13503
Add Complex Instrument	13400
Trade Report	13201
Trade Reversal	13200
Heartbeat	13001

Table 5 - Eurex Enhanced Order Book incremental messages

For order book maintenance, the order messages `Order Add`, `Order Modify`, `Order Delete` and `Order Mass Delete` will be provided along with the product and instrument state messages. Execution for orders will be published via `Partial Order Execution` and `Full Order Execution` messages for partially and fully matched orders. Additionally, an execution summary, `Execution Summary`, message will be provided for the mass execution scenarios.

Any update to the complex instruments will be provided via complex instrument messages. Auction information will be published as described in 4.8 - "Auctions" in detail.
Manually entered trades and reversed trades by Eurex Market Supervision will be published by using Trade Report and Trade Reversal messages.

Cross Trade Announcements and Request for Quotes are disseminated by via the `Cross Request` and the `Quote Request` messages. Request for Quotes and Cross Trade Announcements will be published via incremental messages only.

Functional Heartbeats will be published if there is no activity on a specific product.

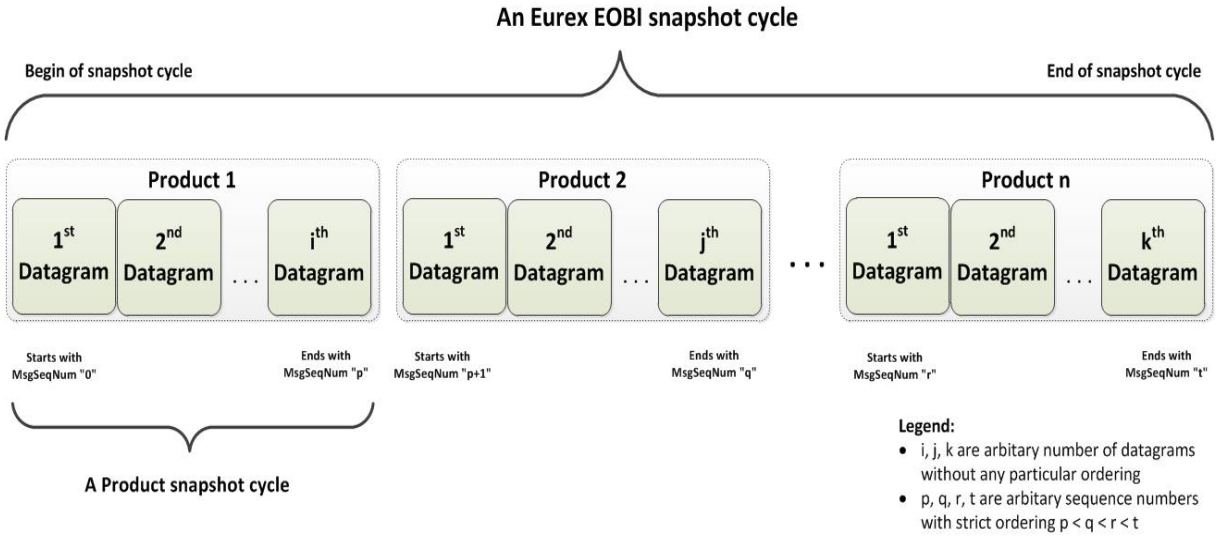
7.3 Snapshot Messages

By design, the snapshot messages are sent periodically and can be used by participants for recovery purposes, i.e. start-up processing or closing gaps in incremental messages. In contrast to Eurex EOBI incremental messages, Eurex EOBI snapshot messages will provide the trade statistics information at the time of sending. Furthermore, they contain the last message sequence number sent on the incremental feed, to provide a synchronization mechanism to participants for incremental and snapshots messages.

Like incremental messages, the snapshot messages will follow the Eurex EOBI datagram structure as described in section 7.1 - "Datagram Structure".

Eurex EOBI snapshot messages will be sent in product states between "Start-Of-Day" and "End-Of-Day".

The picture below provides an overview of a typical **snapshot cycle**.



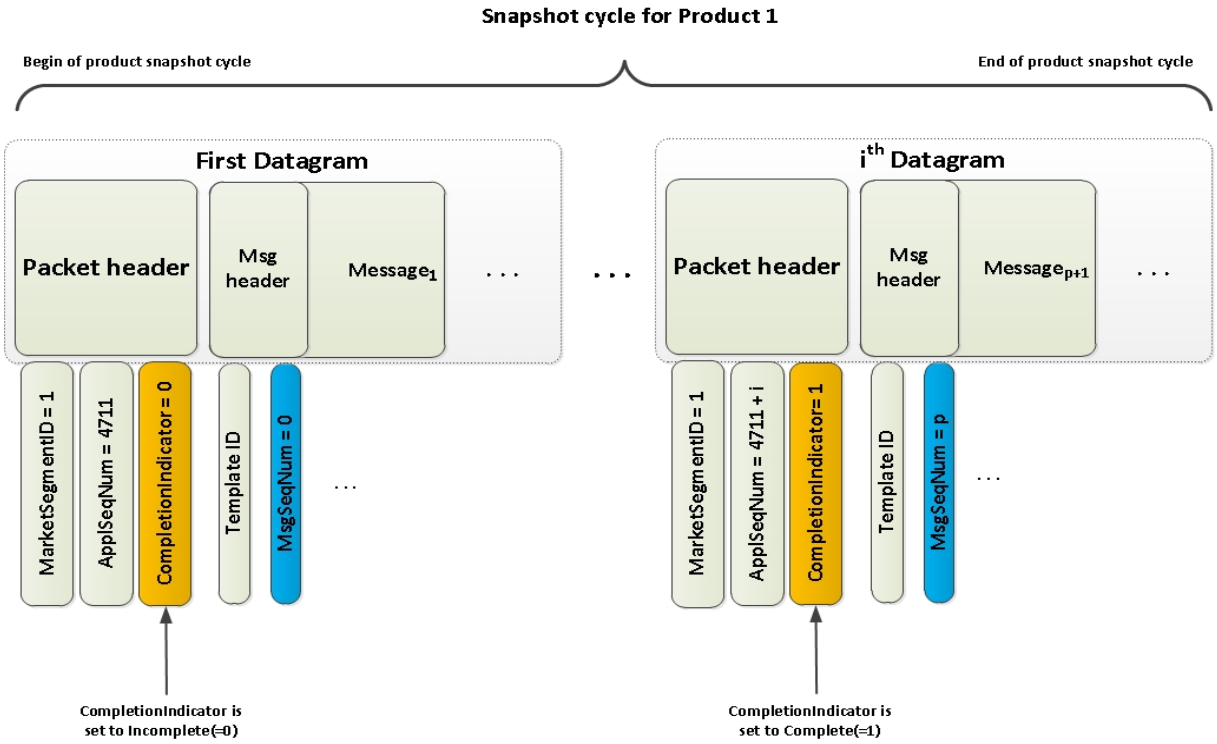
Picture 6: An overview of a snapshot cycle

It is characterized by,

- The packet sequence numbers, *ApplSeqNum,* are contiguous and are incremented across products,
- The message sequence number, *MsgSeqNum*, of the first message in the first datagram of a new snapshot cycle is set to zero(=0),
- The message sequence number, *MsgSeqNum,* within the same snapshot cycle is incremented for each message across all messages and all products,
- The *CompletionIndicator* in the last datagram of a product snapshot cycle is set to *Complete(=1)* to inform about the end of a product snapshot cycle.

That implies, a full snapshot cycle on Eurex EOBI snapshot feed comprises of multiple product snapshot cycles. In order to assist an easy identification of a product snapshot boundary, the *CompletionIndicator* is set to *Complete(=1)* in the last datagram of a product. Each snapshot cycle starts by re-setting the message sequence number, *MsgSeqNum*, to zero(=*0*) for the first message in the first datagram.

The following picture further outlines **product snapshot cycle** for the *Product1* from the picture above.



Picture 7: The Snapshot cycle for Product1

Each message header containing the *templateID* of a message within a snapshot cycle will specify the message content.

Two summary messages are introduced to reduce the total size of snapshot messages in a snapshot cycle by avoiding redundant information:

- A `Product Summary` containing the last message sequence number of the last message sent on the incremental feed and trading state information, and
- An `Instrument Summary` for each instrument of the product including instrument state information and trade statistics such as last trade price and volume, daily low and high prices, opening prices etc. Additionally, the number of visible orders in the current product's snapshot cycle is provided to participants in advance.

The last message sequence number, *LastMsgSeqNumProcessed*, in the product summary message denotes the last message sent on the incremental feed, i.e., it provides a link between incremental and snapshot feed.

A snapshot cycle might contain order book information for multiple products. The following describes the snapshot cycle for one product.

A product has multiple instruments. The `Product Summary` will be given once, as it includes attributes that are identical for all instruments. However, it can include multiple `Instrument Summary` messages, each followed by the individual orders for that instrument.

As it shown in picture below, a **snapshot cycle of a product** will always start with a product summary followed by an instrument summary followed by all visible orders of the corresponding instrument and so on. Logically, the whole process is repeated for all instruments of a product.

Picture 8: A snapshot cycle of a product

Finally, as snapshot cycle of product is terminated on the product level boundary, i.e., *CompletionIndicator* is set to *Complete(=1)*, the next `Product Summary` message implicitly defines the start of a snapshot cycle for the next product, inhernently defining the product level boundary. All messages within a product level boundary are self-contained.

Order messages within a snapshot cycle will be sent in a zig-zag manner as described in 4.1 - "Building the Order Book". All subsequent products follow a similar pattern, forming a snapshot cycle.
Eurex EOBI snapshot messages will contain order book information about the intra-day created complex instruments as well, even if there is no trading activity in that complex instrument.

Please note that, during Auctions the snapshot messages contain either `Auction Best Bid – Offer` or `Auction Clearing Price` messages instead of the order messages, i.e., visible orders aren't published during Auctions via snapshot messages.
Additionally, the `Top Of Book` messages will be published starting from post trading state until end of day trading state to provide participants with last available instrument's BBO information.

7.4 Data Types

The following table provides an overview of the data types used in the fixed-length binary encoded messages sent out by the Eurex EOBI. These data types will be indicated for each field in the Chapter 8.7 - "Message Layout", which covers the message layouts.

Data Type	Description	No Value
signed int.	little endian byte order supported are 1, 2, 4 and 8-byte, signed integers the most significant bit contains the sign.	1 byte signed int: 0x80 2 byte signed int: 0x8000 4 byte signed int: 0x80000000 8 byte signed int: 0x8000000000000000
unsigned int.	little endian byte order supported are 1, 2, 4 and 8-byte unsigned integer.	1 byte unsigned int: 0xFF 2 byte unsigned int: 0xFFFF 4 byte unsigned int: 0xFFFFFFFF 8 byte unsigned int: 0xFFFFFFFFFFFFFFFF
PriceType	Price in integer format including 8 decimals. For certain asset classes, prices may have negative values.	see 8 byte signed int.
Counter	Contains a record or message counter.	see 4 byte signed int.
UTCTimestamp	Date and time, in UTC, represented as nanoseconds past the UNIX epoch (00:00:00 UTC on January 1st, 1970).	see 8 byte unsigned int.

Table 6 - Data types on the Eurex EOBI

8 Message Layout

8.1 Overview of Supported Message Types

The following message formats are based on:

- Interface Version: 3.0
- Build Number: 3.0.71.ga-3000100-483

General

EOBI Message	TemplateID (28500)	FIX Message	MsgType (35)
Packet Header	13004	MarketDataReport	U20
Heartbeat	13001	Heartbeat	0

Order Data

EOBI Message	TemplateID (28500)	FIX Message	MsgType (35)
Order Add	13100	MarketDataOrder	U21
Top of Book	13504	MarketDataInstrument	U23
Order Modify	13101	MarketDataOrder	U21
Order Modify Same Priority	13106	MarketDataOrder	U21
Order Delete	13102	MarketDataOrder	U21
Order Mass Delete	13103	MarketDataOrder	U21
Partial Order Execution	13105	MarketDataOrder	U21
Full Order Execution	13104	MarketDataOrder	U21
Auction Best Bid/Offer	13500	MarketDataInstrument	U23
Auction Clearing Price	13501	MarketDataInstrument	U23

Trade Data

EOBI Message	TemplateID (28500)	FIX Message	MsgType (35)
Execution Summary	13202	MarketDataTrade	U22
Quote Request	13503	MarketDataInstrument	U23
Cross Request	13502	MarketDataInstrument	U23
Trade Report	13201	MarketDataTrade	U22
Trade Reversal	13200	MarketDataTrade	U22

State Change

EOBI Message	TemplateID (28500)	FIX Message	MsgType (35)
Product State Change	13300	TradingSessionStatus	h
Instrument State Change	13301	SecurityStatus	f

Reference Data

EOBI Message	TemplateID (28500)	FIX Message	MsgType (35)
Add Complex Instrument	13400	SecurityDefinitionUpdateReport	BP

Snapshot

EOBI Message	TemplateID (28500)	FIX Message	MsgType (35)
Product Summary	13600	MarketDataInstrument	U23
Snapshot Order	13602	MarketDataOrder	U21
Instrument Summary	13601	MarketDataInstrument	U23

8.2 General

Packet Header

The Packet Header is a technical header which is delivered in every UDP-datagram, and is used for identification of datagrams. The Packet Header will be published on a multicast-channel basis, with each packet containing information for one product only, recognizable by the field MarketSegmentID. Whenever there is an amount of information that doesn't fit in one datagram, the field CompletionIndicator will be set to 'Incomplete'. A CompletionIndicator field set to 'Incomplete' implies that another (new) datagram will follow, containing the remaining data. This will be applied to the incremental messages only. Every partition stamps the outgoing datagrams with a sequence number: ApplSeqNum and a sending time: TransactTime. It also includes the ApplSeqResetIndicator field that can be set in case of market data fail-over and/or a market data restart.

Tag	Field Name	Req'd	Len	Ofs	Data Type	Description
<MessageHeader>						
9	BodyLen	Y	2	0	unsigned int	Number of bytes for the message, including this field.
28500	TemplateID	Y	2	2	unsigned int	Unique identifier for a Eurex EOBI message layout. Value: 13004 (MarketDataReport, MsgType = U20)
34	MsgSeqNum	U	4	4	unsigned int	not used
<Message Body>						
1181	ApplSeqNum	Y	4	8	unsigned int	Message sequence number is contiguous and is incremented across products.
1300	MarketSegmentID	Y	4	12	signed int	Product identifier.
5948	PartitionID	Y	1	16	unsigned int	Grouping of Eurex products. Belongs to the scope of Service Availability.
6228	CompletionIndicator	Y	1	17	unsigned int	Indicated whether an unit of works fits into a single datagram for incremental messages. Value: 0 – Description: Incomplete Value: 1 – Description: Complete
28841	ApplSeqResetIndicator	Y	1	18	unsigned int	Value: 0 – Description: No Reset Value: 1 – Description: Reset
25020	Pad5	U	5	19	Fixed String	not used
60	TransactTime	Y	8	24	UTCTimestamp	Time when market data feed handler writes packet on the wire.

Implied Message Constants

These constant values are to be considered as part of the above message, although they are not transmitted.

Tag	Field Name	Field Value	Length	Data Type	Description
35	MsgType	U20	3	Fixed String	U20 = Market Data Report
28827	MDReportEvent	0	1	unsigned int	0 = Scope Definition.

Heartbeat

A functional Heartbeat message will be published regularly when there is no activity on the Eurex Enhanced Order Book Interface incremental channel. The functional Heartbeat message will be available on the Eurex Enhanced Order Book Interface incremental channel as of the trading state 'Start-Of-Day' and will contain the last processed message sequence number, enabling participants to check for missed or lost packets.

Tag	Field Name	Req'd	Len	Ofs	Data Type	Description
<MessageHeader>						
9	BodyLen	Y	2	0	unsigned int	Number of bytes for the message, including this field.
28500	TemplateID	Y	2	2	unsigned int	Unique identifier for a Eurex EOBI message layout. Value: 13001 (Heartbeat, MsgType = 0)
34	MsgSeqNum	U	4	4	unsigned int	not used
<Message Body>						
369	LastMsgSeqNum-Processed	Y	4	8	unsigned int	Last Message Sequence number that was processed, regardless of message type.
25019	Pad4	U	4	12	Fixed String	not used

Implied Message Constants

These constant values are to be considered as part of the above message, although they are not transmitted.

Tag	Field Name	Field Value	Length	Data Type	Description
35	MsgType	0	3	Fixed String	0 = Hearbeat

8.3 Order Data

Order Add

An Order Add message will be published for each new order that was entered in the order book. The unique key for each order will be based on the instrument identifier, the priority timestamp and the order side, represented by the fields SecurityID, TrdRegTSTimePriority and Side respectively.

Tag	Field Name	Req'd	Len	Ofs	Data Type	Description
<MessageHeader>						
9	BodyLen	Y	2	0	unsigned int	Number of bytes for the message, including this field.
28500	TemplateID	Y	2	2	unsigned int	Unique identifier for a Eurex EOBI message layout. Value: 13100 (MarketDataOrder, MsgType = U21)
34	MsgSeqNum	Y	4	4	unsigned int	Message sequence number, incremented per product across all message types.
<Message Body>						
21002	TrdRegTSTimeIn	N	8	8	UTCTimestamp	Matching engine In timestamp.
48	SecurityID	Y	8	16	signed int	Unique instrument identifier.
<OrderDetails>						
21008	TrdRegTSTimePriority	Y	8	24	UTCTimestamp	Priority timestamp.
1138	DisplayQty	Y	4	32	signed int	Quantity.
54	Side	Y	1	36	unsigned int	Side of the order. Value: 1 = Buy; 2 = Sell
25018	Pad3	U	3	37	Fixed String	not used
44	Price	Y	8	40	PriceType	Price.

Implied Message Constants

These constant values are to be considered as part of the above message, although they are not transmitted.

Tag	Field Name	Field Value	Length	Data Type	Description
35	MsgType	U21	3	Fixed String	U21 = Market Data Order
28842	MarketDataType	1	1	unsigned int	1 = Order Book Maintenance
279	MDUpdateAction	0	1	unsigned int	0 = New Type of Market Data update action.
22	SecurityIDSource	M	1	Fixed String	M = Marketplace Marketplace assigned identifier.

Top of Book

Top of Book messages will be published via incremental and snapshot messages starting from post trading state until end of day trading state to provide the BBO instrument's information.

Tag	Field Name	Req'd	Len	Ofs	Data Type	Description
<MessageHeader>						
9	BodyLen	Y	2	0	unsigned int	Number of bytes for the message, including this field.
28500	TemplateID	Y	2	2	unsigned int	Unique identifier for a Eurex EOBI message layout. Value: 13504 (MarketDataInstrument, MsgType = U23)
34	MsgSeqNum	Y	4	4	unsigned int	Message sequence number, incremented per product across all message types.
<Message Body>						
60	TransactTime	Y	8	8	UTCTimestamp	Transaction timestamp.
48	SecurityID	Y	8	16	signed int	Unique instrument identifier.
132	BidPx	N	8	24	PriceType	Bid price/rate.
133	OfferPx	N	8	32	PriceType	Offer price/rate.

Implied Message Constants

These constant values are to be considered as part of the above message, although they are not transmitted.

Tag	Field Name	Field Value	Length	Data Type	Description
35	MsgType	U23	3	Fixed String	U23 = Market Data Instrument
28842	MarketDataType	13	1	unsigned int	13 = Top Of Book See also Eurex EOBI Schema (XSD) file.
22	SecurityIDSource	M	1	Fixed String	M = Marketplace Marketplace assigned identifier.

Order Modify

An Order Modify message will be published, if an existing order in the book is modified, whereby the new parameters of the order might cause a change in time priority. If an order is modified to another price, or if the quantity of this order is increased, the time priority of the order will change. The order that was modified is recognizable by the field TrdRegTSPrevTimePriority and a new priority key will be set by using the TrdRegTSTimePriority field. Note that, the time priority might also change in case any no-visible attribute of an order is changed e.g. stop price of an OCO order.

Tag	Field Name	Req'd	Len	Ofs	Data Type	Description
<MessageHeader>						
9	BodyLen	Y	2	0	unsigned int	Number of bytes for the message, including this field.
28500	TemplateID	Y	2	2	unsigned int	Unique identifier for a Eurex EOBI message layout. Value: 13101 (MarketDataOrder, MsgType = U21)
34	MsgSeqNum	Y	4	4	unsigned int	Message sequence number, incremented per product across all message types.
<Message Body>						
21002	TrdRegTSTimeIn	Y	8	8	UTCTimestamp	Matching engine In timestamp.
21100	TrdRegTSPrevTimePriority	Y	8	16	UTCTimestamp	Previous order priority timestamp.
28855	PrevPrice	Y	8	24	PriceType	Previous order price.
28867	PrevDisplayQty	Y	4	32	signed int	Previous display quantity
25019	Pad4	N	4	36	Fixed String	not used Valid characters: \x01-\x7E
48	SecurityID	Y	8	40	signed int	Unique instrument identifier.
<OrderDetails>						
21008	TrdRegTSTimePriority	Y	8	48	UTCTimestamp	Priority timestamp (new)
1138	DisplayQty	Y	4	56	signed int	Quantity.
54	Side	Y	1	60	unsigned int	Side of the order. Value: 1 = Buy; 2 = Sell
25018	Pad3	U	3	61	Fixed String	not used
44	Price	Y	8	64	PriceType	Price.

Implied Message Constants

These constant values are to be considered as part of the above message, although they are not transmitted.

Tag	Field Name	Field Value	Length	Data Type	Description
35	MsgType	U21	3	Fixed String	U21 = Market Data Order

Tag	Field Name	Field Value	Length	Data Type	Description
28842	MarketDataType	1	1	unsigned int	1 = Order Book Maintenance See also Eurex EOBI Schema (XSD) file.
279	MDUpdateAction	1	1	unsigned int	1 = Change
22	SecurityIDSource	M	1	Fixed String	M = Marketplace

Order Modify Same Priority

An Order Modify Same Priority message will be published, if the time priority of an existing order is not changed. Please note that, if an order is modified to another price, or if the quantity of this order is increased, then the time priority of the order will change. Otherwise, the time priority remains as it is. The time priority of the order is available in the TrdRegTSTimePriority field.

Tag	Field Name	Req'd	Len	Ofs	Data Type	Description
<MessageHeader>						
9	BodyLen	Y	2	0	unsigned int	Number of bytes for the message, including this field.
28500	TemplateID	Y	2	2	unsigned int	Unique identifier for a Eurex EOBI message layout. Value: 13106 (MarketDataOrder, MsgType = U21)
34	MsgSeqNum	Y	4	4	unsigned int	Message sequence number, incremented per product across all message types.
<Message Body>						
21002	TrdRegTSTimeIn	Y	8	8	UTCTimestamp	Matching engine In timestamp.
60	TransactTime	Y	8	16	UTCTimestamp	Transaction timestamp.
28867	PrevDisplayQty	Y	4	24	signed int	Previous display quantity
25019	Pad4	N	4	28	Fixed String	Data structure padding (4 bytes). Valid characters: \x01-\x7E
48	SecurityID	Y	8	32	signed int	Unique instrument identifier.
<OrderDetails>						
21008	TrdRegTSTimePriority	Y	8	40	UTCTimestamp	Priority timestamp. Identical to the original time priority.
1138	DisplayQty	Y	4	48	signed int	Quantity.
54	Side	Y	1	52	unsigned int	Side of the order. Value: 1 = Buy; 2 = Sell
25018	Pad3	U	3	53	Fixed String	not used
44	Price	Y	8	56	PriceType	Order Price.

Implied Message Constants

These constant values are to be considered as part of the above message, although they are not transmitted.

Tag	Field Name	Field Value	Length	Data Type	Description
35	MsgType	U21	3	Fixed String	U21 = Market Data Order
28842	MarketDataType	1	1	unsigned int	1 = Order Book Maintenance
279	MDUpdateAction	1	1	unsigned int	1 = Change

Tag	Field Name	Field Value	Length	Data Type	Description
22	SecurityIDSource	M	1	Fixed String	M = Marketplace Marketplace assigned identifier.

Order Delete

Whenever an existing order is deleted from the order book, an Order Delete message will be published. The Order Delete message will contain all necessary fields needed to delete the correct order; SecurityID, TrdRegTSTimePriority, Side. For convenience, the order delete message will also contain the former displayed quantity and the former price.

Tag	Field Name	Req'd	Len	Ofs	Data Type	Description
<MessageHeader>						
9	BodyLen	Y	2	0	unsigned int	Number of bytes for the message, including this field.
28500	TemplateID	Y	2	2	unsigned int	Unique identifier for a Eurex EOBI message layout. Value: 13102 (MarketDataOrder, MsgType = U21)
34	MsgSeqNum	Y	4	4	unsigned int	Message sequence number, incremented per product across all message types.
<Message Body>						
21002	TrdRegTSTimeIn	N	8	8	UTCTimestamp	Matching engine In timestamp.
60	TransactTime	Y	8	16	UTCTimestamp	Transaction timestamp.
48	SecurityID	Y	8	24	signed int	Unique instrument identifier.
<OrderDetails>						
21008	TrdRegTSTimePriority	Y	8	32	UTCTimestamp	Priority timestamp.
1138	DisplayQty	Y	4	40	signed int	Quantity.
54	Side	Y	1	44	unsigned int	Side of the order. Value: 1 = Buy; 2 = Sell
25018	Pad3	U	3	45	Fixed String	not used
44	Price	Y	8	48	PriceType	Limit price. Required if OrdType (40) is Limit (2) or Stop Limit (4).

Implied Message Constants

These constant values are to be considered as part of the above message, although they are not transmitted.

Tag	Field Name	Field Value	Length	Data Type	Description
35	MsgType	U21	3	Fixed String	U21 = Market Data Order
28842	MarketDataType	1	1	unsigned int	1 = Order Book Maintenance
279	MDUpdateAction	2	1	unsigned int	2 = Delete Type of Market Data update action.
22	SecurityIDSource	M	1	Fixed String	M = Marketplace Marketplace assigned identifier.

Order Mass Delete

An Order Mass Delete message will be published when the order book is expected to be emptied. The message contains the instrument identifier indicating which order book has to be fully deleted. Please note, that the Order Mass Delete Message is mutually exclusive to messages like TopOfBook, Auction Best Bid Offer and Auction Clearing Price which are published during non-continuous state. However it will be sent explicitly, if the potential auction price or BBO is not to be determined during auction call phase, i.e., neither Auction Best Bid Offer nor Auction Clearing Price messages are to be sent.

Tag	Field Name	Req'd	Len	Ofs	Data Type	Description
<MessageHeader>						
9	BodyLen	Y	2	0	unsigned int	Number of bytes for the message, including this field.
28500	TemplateID	Y	2	2	unsigned int	Unique identifier for a Eurex EOBI message layout. Value: 13103 (MarketDataOrder, MsgType = U21)
34	MsgSeqNum	Y	4	4	unsigned int	Message sequence number, incremented per product across all message types.
<Message Body>						
48	SecurityID	Y	8	8	signed int	Unique instrument identifier.
60	TransactTime	Y	8	16	UTCTimestamp	Transaction timestamp.

Implied Message Constants

These constant values are to be considered as part of the above message, although they are not transmitted.

Tag	Field Name	Field Value	Length	Data Type	Description
35	MsgType	U21	3	Fixed String	U21 = Market Data Order
28842	MarketDataType	1	1	unsigned int	1 = Order Book Maintenance See also Eurex EOBI Schema (XSD) file.
279	MDUpdateAction	2	1	unsigned int	2 = Delete
22	SecurityIDSource	M	1	Fixed String	M = Marketplace

Partial Order Execution

Whenever a visible order is partially executed at its displayed price, a Partial Order Execution message will be published, containing the execution information; instrument identifier, priority timestamp, price and executed quantity of the executed passive order as well as the match identifier. The remaining quantity in the order book for this order must be calculated by subtracting the executed quantity in the Partial Order Execution message from the initial quantity in the order book.

Tag	Field Name	Req'd	Len	Ofs	Data Type	Description
<MessageHeader>						
9	BodyLen	Y	2	0	unsigned int	Number of bytes for the message, including this field.
28500	TemplateID	Y	2	2	unsigned int	Unique identifier for a Eurex EOBI message layout. Value: 13105 (MarketDataOrder, MsgType = U21)
34	MsgSeqNum	Y	4	4	unsigned int	Message sequence number, incremented per product across all message types.
<Message Body>						
54	Side	Y	1	8	unsigned int	Side of the order. Value: 1 = Buy; 2 = Sell
25022	Pad7	U	7	9	Fixed String	not used
44	Price	Y	8	16	PriceType	The price at which the order entered the book. Typically it is equal to LastPx except during auction uncrossing.
21008	TrdRegTSTimePriority	Y	8	24	UTCTimestamp	Priority timestamp.
48	SecurityID	Y	8	32	signed int	Unique instrument identifier.
880	TrdMatchID	Y	4	40	unsigned int	Unique identifier for each price level (match step) of a match event; it is used for public trade reporting.
32	LastQty	Y	4	44	signed int	Quantity executed in this fill.
31	LastPx	Y	8	48	PriceType	The price at which the order was matched.

Implied Message Constants

These constant values are to be considered as part of the above message, although they are not transmitted.

Tag	Field Name	Field Value	Length	Data Type	Description
35	MsgType	U21	3	Fixed String	U21 = Market Data Order
28842	MarketDataType	2	1	unsigned int	2 = Order Book Execution
279	MDUpdateAction	1	1	unsigned int	1 = Change

Tag	Field Name	Field Value	Length	Data Type	Description
22	SecurityIDSource	M	1	Fixed String	M = Marketplace

Full Order Execution

Whenever a visible order is fully executed at its displayed price, a Full Order Execution message will be published, containing the execution information; instrument identifier, priority timestamp, price and executed quantity of the executed passive order and the match identifier. As this order is executed in full, it has to be deleted from the order book.

Tag	Field Name	Req'd	Len	Ofs	Data Type	Description
<MessageHeader>						
9	BodyLen	Y	2	0	unsigned int	Number of bytes for the message, including this field.
28500	TemplateID	Y	2	2	unsigned int	Unique identifier for a Eurex EOBI message layout. Value: 13104 (MarketDataOrder, MsgType = U21)
34	MsgSeqNum	Y	4	4	unsigned int	Message sequence number, incremented per product across all message types.
<Message Body>						
54	Side	Y	1	8	unsigned int	Side of the order. Value: 1 = Buy; 2 = Sell
25022	Pad7	U	7	9	Fixed String	not used
44	Price	Y	8	16	PriceType	The price at which the order entered the book. Typically it is equal to LastPx except during auction uncrossing.
21008	TrdRegTSTimePriority	Y	8	24	UTCTimestamp	Priority timestamp.
48	SecurityID	Y	8	32	signed int	Unique instrument identifier.
880	TrdMatchID	Y	4	40	unsigned int	Unique identifier for each price level (match step) of a match event; it is used for public trade reporting.
32	LastQty	Y	4	44	signed int	Quantity executed in this fill.
31	LastPx	Y	8	48	PriceType	The price at which the order was matched.

Implied Message Constants

These constant values are to be considered as part of the above message, although they are not transmitted.

Tag	Field Name	Field Value	Length	Data Type	Description
35	MsgType	U21	3	Fixed String	U21 = Market Data Order
28842	MarketDataType	2	1	unsigned int	2 = Order Book Execution
279	MDUpdateAction	1	1	unsigned int	1 = Change
22	SecurityIDSource	M	1	Fixed String	M = Marketplace

Auction Best Bid/Offer

During auctions, no order book depth information is published. For an uncrossed order book, only the BBO information for an instrument is published using Auction Best Bid/Offer messages.

Tag	Field Name	Req'd	Len	Ofs	Data Type	Description
<MessageHeader>						
9	BodyLen	Y	2	0	unsigned int	Number of bytes for the message, including this field.
28500	TemplateID	Y	2	2	unsigned int	Unique identifier for a Eurex EOBI message layout. Value: 13500 (MarketDataInstrument, MsgType = U23)
34	MsgSeqNum	Y	4	4	unsigned int	Message sequence number, incremented per product across all message types.
<Message Body>						
60	TransactTime	Y	8	8	UTCTimestamp	Official timestamp of order book entry.
48	SecurityID	Y	8	16	signed int	Unique instrument identifier.
132	BidPx	N	8	24	PriceType	Bid price/rate.
133	OfferPx	N	8	32	PriceType	Offer price/rate.

Implied Message Constants

These constant values are to be considered as part of the above message, although they are not transmitted.

Tag	Field Name	Field Value	Length	Data Type	Description
35	MsgType	U23	3	Fixed String	U23 = Market Data Instrument
28842	MarketDataType	5	1	unsigned int	5 = Auction BBO See also Eurex EOBI Schema (XSD) file.
22	SecurityIDSource	M	1	Fixed String	M = Marketplace Marketplace assigned identifier.

Auction Clearing Price

During auctions, no order book depth information is published. For a crossed order book in an auction, an Auction Clearing Price message will be published, indicating the potential auction price.

Tag	Field Name	Req'd	Len	Ofs	Data Type	Description
<MessageHeader>						
9	BodyLen	Y	2	0	unsigned int	Number of bytes for the message, including this field.
28500	TemplateID	Y	2	2	unsigned int	Unique identifier for a Eurex EOBI message layout. Value: 13501 (MarketDataInstrument, MsgType = U23)
34	MsgSeqNum	Y	4	4	unsigned int	Message sequence number, incremented per product across all message types.
<Message Body>						
60	TransactTime	Y	8	8	UTCTimestamp	Transaction timestamp.
48	SecurityID	Y	8	16	signed int	Unique instrument identifier.
31	LastPx	Y	8	24	PriceType	Indicating the potential Auction price for a crossed order book.

Implied Message Constants

These constant values are to be considered as part of the above message, although they are not transmitted.

Tag	Field Name	Field Value	Length	Data Type	Description
35	MsgType	U23	3	Fixed String	U23 = Market Data Instrument
28842	MarketDataType	6	1	unsigned int	6 = Auction Clearing Price See also Eurex EOBI Schema (XSD) file.
22	SecurityIDSource	M	1	Fixed String	M = Marketplace Marketplace assigned identifier.

8.4 Trade Data

Execution Summary

Whenever an incoming order is executed, an Execution Summary message will be published, containing information on the execution of the incoming order. The Execution Summary message only contains information for the initial instrument (security), that was specified by the incoming order, i.e. any synthetic matches/changes can not be derived from the summary message. The Execution Summary message is meant for fast trading decisions only. In fact, to be absolutely sure the order book is correct, participants should always process the execution messages following the Execution Summary message. The fields in the Execution Summary message provide information on the instrument specified in the incoming order, the time the incoming order entered the gatway and the matching engine, match time, the side of the incoming order, an indicator for a synthetic match, the quantity that was executed (of the specified instrument) in the fill, and the worst price of the fill, represented by the fields SecurityID, RequestTime, AggressorTimestamp, ExecID, AggressorSide, TradeCondition, LastQty, RestingHiddenQty and LastPx respectively. The RestingHiddenQty in the context of an execution would refer to the resting hidden quantity, included in the LastQty (of the specified instrument). It is set to zero if no such quantity is involved and is empty if the TradeCondition is flagged as ImpliedTrade.

Tag	Field Name	Req'd	Len	Ofs	Data Type	Description
<MessageHeader>						
9	BodyLen	Y	2	0	unsigned int	Number of bytes for the message, including this field.
28500	TemplateID	Y	2	2	unsigned int	Unique identifier for a Eurex EOBI message layout. Value: 13202 (MarketDataTrade, MsgType = U22)
34	MsgSeqNum	Y	4	4	unsigned int	Message sequence number, incremented per product across all message types.
<Message Body>						
48	SecurityID	Y	8	8	signed int	Unique instrument identifier.
28820	AggressorTimestamp	N	8	16	UTCTimestamp	
5979	RequestTime	N	8	24	UTCTimestamp	Gateway request in timestamp.
17	ExecID	Y	8	32	UTCTimestamp	Matching timestamp.
32	LastQty	Y	4	40	signed int	Total quantity of this match.
28731	AggressorSide	Y	1	44	unsigned int	Value: 1 = Triggered by the buy side; 2 = Triggered by the sell side
277	TradeCondition	N	1	45	unsigned int	Indicates whether a synthetic match is occured. Value: 1 = Implied Trade
25017	Pad2	U	2	46	Fixed String	not used
31	LastPx	Y	8	48	PriceType	Worst price of this match.

Tag	Field Name	Req'd	Len	Ofs	Data Type	Description
28868	RestingHiddenQty	N	4	56	signed int	Quantity of matched passive orders that is not displayed to the market. Set to zero if no such quantity is involved. Is empty if the TradeCondition is flagged as ImpliedTrade.
28869	RestingCxlQty	Y	4	60	signed int	The deleted quantity due to Self Match Prevention within a Match Event. This quantity is not part of LastQty which could even be 0 in certain cases.

Implied Message Constants

These constant values are to be considered as part of the above message, although they are not transmitted.

Tag	Field Name	Field Value	Length	Data Type	Description
35	MsgType	U22	3	Fixed String	U22 = Market Data Trade
28842	MarketDataType	12	1	unsigned int	12 = Match Event See also Eurex EOBI Schema (XSD) file.
279	MDUpdateAction	0	1	unsigned int	0 = New Type of Market Data update action.
22	SecurityIDSource	M	1	Fixed String	M = Marketplace Marketplace assigned identifier.

Quote Request

Market participants can enter a quote request (trading interest), asking market makers to enter quotes into a specific instrument. The Quote Request message shows these quote requests.

Tag	Field Name	Req'd	Len	Ofs	Data Type	Description
<MessageHeader>						
9	BodyLen	Y	2	0	unsigned int	Number of bytes for the message, including this field.
28500	TemplateID	Y	2	2	unsigned int	Unique identifier for a Eurex EOBI message layout. Value: 13503 (MarketDataInstrument, MsgType = U23)
34	MsgSeqNum	Y	4	4	unsigned int	Message sequence number, incremented per product across all message types.
<Message Body>						
48	SecurityID	Y	8	8	signed int	Unique instrument identifier.
32	LastQty	Y	4	16	signed int	Defines the requested quantity in a Quote Request.
54	Side	Y	1	20	unsigned int	Side of the order. Value: 1 = Buy; 2 = Sell
25018	Pad3	U	3	21	Fixed String	not used
60	TransactTime	Y	8	24	UTCTimestamp	Transaction timestamp.

Implied Message Constants

These constant values are to be considered as part of the above message, although they are not transmitted.

Tag	Field Name	Field Value	Length	Data Type	Description
35	MsgType	U23	3	Fixed String	U23 = Market Data Instrument
28842	MarketDataType	8	1	unsigned int	8 = Quote Request See also Eurex EOBI Schema (XSD) file.
22	SecurityIDSource	M	1	Fixed String	M = Marketplace Marketplace assigned identifier.

Cross Request

A crossing is defined as an intentional or unintentional execution of orders and quotes against a preselected participant or in-house. Using the Cross Request message, all participants are informed of a crossing that shall be executed in the Eurex order book (on-exchange). Other market participants can see the order(s) and can also match against them. Eurex expects the orders to be entered within a certain time frame.

Tag	Field Name	Req'd	Len	Ofs	Data Type	Description
<MessageHeader>						
9	BodyLen	Y	2	0	unsigned int	Number of bytes for the message, including this field.
28500	TemplateID	Y	2	2	unsigned int	Unique identifier for a Eurex EOBI message layout. Value: 13502 (MarketDataInstrument, MsgType = U23)
34	MsgSeqNum	Y	4	4	unsigned int	Message sequence number, incremented per product across all message types.
<Message Body>						
48	SecurityID	Y	8	8	signed int	Unique instrument identifier.
32	LastQty	Y	4	16	signed int	Defines the requested quantity in a Cross Request.
25019	Pad4	U	4	20	Fixed String	not used
60	TransactTime	Y	8	24	UTCTimestamp	Transaction timestamp.

Implied Message Constants

These constant values are to be considered as part of the above message, although they are not transmitted.

Tag	Field Name	Field Value	Length	Data Type	Description
35	MsgType	U23	3	Fixed String	U23 = Market Data Instrument
28842	MarketDataType	7	1	unsigned int	7 = Cross Trade Announcement
22	SecurityIDSource	M	1	Fixed String	M = Marketplace Marketplace assigned identifier.

Trade Report

Whenever a trade results from an auction or a trade is entered manually by Eurex Market Supervision, participants will be informed by a Trade Report message. The reason for the trade will be indicated by the field MatchType.

Tag	Field Name	Req'd	Len	Ofs	Data Type	Description
<MessageHeader>						
9	BodyLen	Y	2	0	unsigned int	Number of bytes for the message, including this field.
28500	TemplateID	Y	2	2	unsigned int	Unique identifier for a Eurex EOBI message layout. Value: 13201 (MarketDataTrade, MsgType = U22)
34	MsgSeqNum	Y	4	4	unsigned int	Message sequence number, incremented per product across all message types.
<Message Body>						
48	SecurityID	Y	8	8	signed int	Unique instrument identifier.
60	TransactTime	Y	8	16	UTCTimestamp	Transaction timestamp.
880	TrdMatchID	Y	4	24	unsigned int	Unique identifier for each price level (match step) of a match event; it is used for public trade reporting.
32	LastQty	Y	4	28	signed int	Quantity executed in this fill.
31	LastPx	Y	8	32	PriceType	Price of this fill.
574	MatchType	Y	1	40	unsigned int	The point in the matching process at which this trade was matched. Value: 3 — Manual Trade Entry Value: 5 — Trade from Uncrossing Value: 7 — Trade from Auction
28610	MatchSubType	N	1	41	unsigned int	Indicates the auction type the trade originates from. Not filled for uncrossing, i.e. when a complex instrument switches to the instrument state "Continuous". Value: 1 — Opening Auction Value: 2 — Closing Auction Value: 3 — Intraday Auction Value: 4 — Circuit Breaker Auction
25021	Pad6	U	6	42	Fixed String	not used

Implied Message Constants

These constant values are to be considered as part of the above message, although they are not transmitted.

Tag	Field Name	Field Value	Length	Data Type	Description
35	MsgType	U22	3	Fixed String	U22 = Maraket Data Trade
28842	MarketDataType	4	1	unsigned int	4 = Trade Report See also Eurex EOBI Schema (XSD) file.
279	MDUpdateAction	0	1	unsigned int	0 = New
22	SecurityIDSource	M	1	Fixed String	M = Marketplace Marketplace assigned identifier.

Trade Reversal

Whenever a trade is reversed by Eurex Market Supervision, participants will be informed by a Trade Reversal message.

Tag	Field Name	Req'd	Len	Ofs	Data Type	Description
<MessageHeader>						
9	BodyLen	Y	2	0	unsigned int	Number of bytes for the message, including this field.
28500	TemplateID	Y	2	2	unsigned int	Unique identifier for a Eurex EOBI message layout. Value: 13200 (MarketDataTrade, MsgType = U22)
34	MsgSeqNum	Y	4	4	unsigned int	Message sequence number, incremented per product across all message types.
<Message Body>						
48	SecurityID	Y	8	8	signed int	Unique instrument identifier.
60	TransactTime	Y	8	16	UTCTimestamp	Transaction timestamp.
880	TrdMatchID	Y	4	24	unsigned int	Unique identifier for each price level (match step) of a match event; it is used for public trade reporting.
32	LastQty	Y	4	28	signed int	Quantity executed in this fill.
31	LastPx	Y	8	32	PriceType	Price of this fill.
21001	TrdRegTSExecutionTime	N	8	40	UTCTimestamp	Matching timestamp of new last trade.
268	NoMDEntries	Y	1	48	Counter	
25022	Pad7	U	7	49	Fixed String	not used
<MDTradeEntryGrp>						Variable size array, Record counter: NoMDEntries
270	>MDEntryPx	N	8	56	PriceType	Price.
271	>MDEntrySize	N	4	64	signed int	Quantity.
269	>MDEntryType	Y	1	68	unsigned int	Type of market data entry. Value: 2 – Trade 4 – Opening Price 5 – Closing Price 7 – High Price 8 – Low Price
25018	>Pad3	U	3	69	Fixed String	not used

Implied Message Constants

These constant values are to be considered as part of the above message, although they are not transmitted.

Tag	Field Name	Field Value	Length	Data Type	Description
35	MsgType	U22	3	Fixed String	U22 = Market Data Trade
28842	MarketDataType	3	1	unsigned int	3 = Trade Reversal See also Eurex EOBI Schema (XSD) file.
279	MDUpdateAction	2	1	unsigned int	2 = Delete
22	SecurityIDSource	M	1	Fixed String	M = Marketplace Marketplace assigned identifier.

8.5 State Change

Product State Change

The Product State Change message provides updates on the trading state for (all instruments in) a particular product.

Tag	Field Name	Req'd	Len	Ofs	Data Type	Description
<MessageHeader>						
9	BodyLen	Y	2	0	unsigned int	Number of bytes for the message, including this field.
28500	TemplateID	Y	2	2	unsigned int	Unique identifier for a Eurex EOBI message layout. Value: 13300 (TradingSessionStatus, MsgType = h)
34	MsgSeqNum	Y	4	4	unsigned int	Message sequence number, incremented per product across all message types.
<Message Body>						
336	TradingSessionID	Y	1	8	unsigned int	Miscellaneous state information. Value: 1 = Day; 3 = Morning; 5 = Evening; 7 = Holiday
625	TradingSessionSubID	Y	1	9	unsigned int	Product state information. Value: 1 = Pre Trading; 3 = Trading; 4 = Closing; 5 = Post Trading; 7 = Quiescent
340	TradSesStatus	Y	1	10	unsigned int	Miscellaneous state information. Value: 1 = Halted; 2 = Open; 3 = Closed
28828	FastMarketIndicator	Y	1	11	unsigned int	Indicates if product is in the state Fast Market. Value: 0 = No; 1 = Yes
25019	Pad4	U	4	12	Fixed String	not used

Tag	Field Name	Req'd	Len	Ofs	Data Type	Description
60	TransactTime	Y	8	16	UTCTimestamp	Transaction timestamp.

Implied Message Constants

These constant values are to be considered as part of the above message, although they are not transmitted.

Tag	Field Name	Field Value	Length	Data Type	Description
35	MsgType	h	3	Fixed String	h = Trading Session Status
1368	TradSesEvent	3	1	unsigned int	3 = Status Change

Instrument State Change

The Instrument State Change message provides state information for a single instrument. Furthermore, it informs participants about intra-day expiration of instruments.

Tag	Field Name	Req'd	Len	Ofs	Data Type	Description
<MessageHeader>						
9	BodyLen	Y	2	0	unsigned int	Number of bytes for the message, including this field.
28500	TemplateID	Y	2	2	unsigned int	Unique identifier for a Eurex EOBI message layout. Value: 13301 (SecurityStatus, MsgType = f)
34	MsgSeqNum	Y	4	4	unsigned int	Message sequence number, incremented per product across all message types.
<Message Body>						
48	SecurityID	Y	8	8	signed int	Unique instrument identifier.
965	SecurityStatus	Y	1	16	unsigned int	Instrument status. Value: 1 – Active; 2 – Inactive; 4 – Expired; 9 – Suspended
326	SecurityTradingStatus	Y	1	17	unsigned int	Instrument state status. Value: 200 – Closed; 201 – Restricted; 202 – Book; 203 – Continuous; 204 – Opening Auction; 205 – Opening Auction Freeze; 206 – Intraday Auction; 207 – Intraday Auction Freeze; 208 – Circuit Breaker Auction; 209 – Circuit Breaker Auction Freeze; 210 – Closing Auction; 211 – Closing Auction Freeze
28828	FastMarketIndicator	Y	1	18	unsigned int	Value: 0 – No; 1 – Yes
25020	Pad5	U	5	19	Fixed String	not used

Tag	Field Name	Req'd	Len	Ofs	Data Type	Description
60	TransactTime	Y	8	24	UTCTimestamp	Transaction timestamp.

Implied Message Constants

These constant values are to be considered as part of the above message, although they are not transmitted.

Tag	Field Name	Field Value	Length	Data Type	Description
35	MsgType	f	3	Fixed String	f = Security Status
22	SecurityIDSource	M	1	Fixed String	M = Marketplace Marketplace assigned identifier.

8.6 Reference Data

Add Complex Instrument

Whenever a new complex instrument is created or an existing complex instrument is modified, an Add Complex Instrument message will be published.

Tag	Field Name	Req'd	Len	Ofs	Data Type	Description
<MessageHeader>						
9	BodyLen	Y	2	0	unsigned int	Number of bytes for the message, including this field.
28500	TemplateID	Y	2	2	unsigned int	Unique identifier for a Eurex EOBI message layout. Value: 13400 (SecurityDefinitionUpdateReport, MsgType = BP)
34	MsgSeqNum	Y	4	4	unsigned int	Message sequence number, incremented per product across all message types.
<Message Body>						
48	SecurityID	Y	8	8	signed int	Unique instrument identifier.
60	TransactTime	Y	8	16	UTCTimestamp	Transaction timestamp.
762	SecuritySubType	N	4	24	signed int	Strategy Type.
1227	ProductComplex	Y	1	28	unsigned int	This field qualifies an instrument type on Eurex. Value: 5 – Futures Spread 6 – Inter Product Spread 7 – Standard Futures Strategy 8 – Pack And Bundle 9 – Strip
1144	ImpliedMarketIndicator	Y	1	29	unsigned int	Indicates that an implied market to be created for either the legs of a multi-leg instrument (Implied-in) or for the multi-leg instrument based on the existence of the legs (Implied-out). Determination as to whether implied markets should be created is generally done at the level of the multi-leg instrument. Commonly used in listed derivatives. Value: 0 – Not Implied 3 – Implied In Out
555	NoLegs	Y	1	30	Counter	Number of Legs repeating group instances.

Tag	Field Name	Req'd	Len	Ofs	Data Type	Description
25016	Pad1	U	1	31	Fixed String	not used
<InstrmtLegGrp>						Variable size array, Record counter: NoLegs
600	>LegSymbol	Y	4	32	signed int	Product identifier of the leg security (only applicable for underlying leg).
25019	>Pad4	U	4	36	Fixed String	not used
602	>LegSecurityID	Y	8	40	signed int	Instrument identifier of the leg security.
623	>LegRatioQty	N	4	48	signed int	The ratio of quantity for this individual leg relative to the entire multi-leg security.
624	>LegSide	Y	1	52	unsigned int	The side of the individual leg of a strategy as defined in signature. Value: 1 = Buy; 2 = Sell
25018	>Pad3	U	3	53	Fixed String	not used

Implied Message Constants

These constant values are to be considered as part of the above message, although they are not transmitted.

Tag	Field Name	Field Value	Length	Data Type	Description
35	MsgType	BP	3	Fixed String	BP = Security Definition Update Report
980	SecurityUpdateAction	A	1	Fixed String	A = Add
22	SecurityIDSource	M	1	Fixed String	M = Marketplace Marketplace assigned identifier.
603	LegSecurityIDSource	M	1	Fixed String	Marketplace assigned identifier.
1310	NoMarketSegments	1	1	unsigned int	1 = One
167	SecurityType	MLEG	4	Fixed String	MLEG = Multi Leg

8.7 Snapshot

Product Summary

A Product Summary message will be published once each snapshot cycle, and will contain attributes that are equal for all instruments that belong to that product.

Tag	Field Name	Req'd	Len	Ofs	Data Type	Description
<MessageHeader>						
9	BodyLen	Y	2	0	unsigned int	Number of bytes for the message, including this field.
28500	TemplateID	Y	2	2	unsigned int	Unique identifier for a Eurex EOBI message layout. Value: 13600 (MarketDataInstrument, MsgType = U23)
34	MsgSeqNum	Y	4	4	unsigned int	Message sequence number, incremented per product.
<Message Body>						
369	LastMsgSeqNum-Processed	Y	4	8	unsigned int	Last Message Sequence number that was processed, regardless of message type.
336	TradingSessionID	N	1	12	unsigned int	Miscellaneous state information. Value: 1 = Day; 3 = Morning; 5 = Evening; 7 = Holiday
625	TradingSessionSubID	N	1	13	unsigned int	Product state information. Value: 1 = Pre Trading; 3 = Trading; 4 = Closing; 5 = Post Trading; 7 = Quiescent
340	TradSesStatus	N	1	14	unsigned int	Miscellaneous state information. Value: 1 = Halted; 2 = Open; 3 = Closed

Tag	Field Name	Req'd	Len	Ofs	Data Type	Description
28828	FastMarketIndicator	N	1	15	unsigned int	Indicates if product is in the state Fast Market. Value: 0 = No; 1 = Yes

Implied Message Constants

These constant values are to be considered as part of the above message, although they are not transmitted.

Tag	Field Name	Field Value	Length	Data Type	Description
35	MsgType	U23	3	Fixed String	U23 = Market Data Instrument
28842	MarketDataType	9	1	unsigned int	9 = Market Segment Snapshot

Snapshot Order

Each individual order or quote is represented in a Snapshot Order in a snapshot cycle on the Eurex Enhanced Order Book Interface snapshot channel. The format of the snapshot order enables participants to build the order book according to price-time priority.

Tag	Field Name	Req'd	Len	Ofs	Data Type	Description
<MessageHeader>						
9	BodyLen	Y	2	0	unsigned int	Number of bytes for the message, including this field.
28500	TemplateID	Y	2	2	unsigned int	Unique identifier for a Eurex EOBI message layout. Value: 13602 (MarketDataOrder, MsgType = U21)
34	MsgSeqNum	Y	4	4	unsigned int	Message sequence number, incremented per product across all message types.
<Message Body> *<OrderDetails>*						
21008	TrdRegTSTimePriority	Y	8	8	UTCTimestamp	Priority timestamp.
1138	DisplayQty	Y	4	16	signed int	Quantity.
54	Side	Y	1	20	unsigned int	Side of the order. Value: 1 = Buy; 2 = Sell
25018	Pad3	U	3	21	Fixed String	not used
44	Price	Y	8	24	PriceType	Limit price. Required if OrdType (40) is Limit (2) or Stop Limit (4).

Implied Message Constants

These constant values are to be considered as part of the above message, although they are not transmitted.

Tag	Field Name	Field Value	Length	Data Type	Description
35	MsgType	U21	3	Fixed String	U21 = Market Data Order
28842	MarketDataType	11	1	unsigned int	11 = Order Book Snapshot See also Eurex EOBI Schema (XSD) file.
279	MDUpdateAction	5	1	unsigned int	5 = Overlay

Instrument Summary

An Instrument Summary message will be published for each instrument in one snapshot cycle on the Eurex Enhanced Order Book Interface snapshot channel, and will contain instrument state information and trade statistics for one instrument. Note that one product can have multiple instruments. The repeating group MDEntryGrp, instrument's trade statistics, are not cut of by design.

Tag	Field Name	Req'd	Len	Ofs	Data Type	Description
<MessageHeader>						
9	BodyLen	Y	2	0	unsigned int	Number of bytes for the message, including this field.
28500	TemplateID	Y	2	2	unsigned int	Unique identifier for a Eurex EOBI message layout. Value: 13601 (MarketDataInstrument, MsgType = U23)
34	MsgSeqNum	Y	4	4	unsigned int	Message sequence number, incremented per product.
<Message Body>						
48	SecurityID	Y	8	8	signed int	Unique instrument identifier.
779	LastUpdateTime	Y	8	16	UTCTimestamp	Last update time of the corresponding order book.
21001	TrdRegTSExecution-Time	N	8	24	UTCTimestamp	Last matching execution timestamp.
68	TotNoOrders	Y	2	32	Counter	Corresponding number of orders for this instrument.
965	SecurityStatus	Y	1	34	unsigned int	Instrument status. Value: Description 1: Active 2: Inactive 4: Expired 9: Suspended

Tag	Field Name	Req'd	Len	Ofs	Data Type	Description
326	SecurityTradingStatus	Y	1	35	unsigned int	Instrument state status. Value: 200 Closed; 201 Restricted; 202 Book; 203 Continuous; 204 Opening Auction; 205 Opening Auction Freeze; 206 Intraday Auction; 207 Intraday Auction Freeze; 208 Circuit Breaker Auction; 209 Circuit Breaker Auction Freeze; 210 Closing Auction; 211 Closing Auction Freeze
28828	FastMarketIndicator	Y	1	36	unsigned int	Value: 0 No; 1 Yes
268	NoMDEntries	Y	1	37	Counter	Number of entries in Market Data message for MDEntryGrp.
25017	Pad2	U	2	38	Fixed String	not used
<MDInstrumentEntryGrp>						Variable size array, Record counter: NoMDEntries
270	>MDEntryPx	N	8	40	PriceType	Price.
271	>MDEntrySize	N	4	48	signed int	Quantity.
269	>MDEntryType	Y	1	52	unsigned int	Type of market data entry. Value: 2 Trade; 4 Opening Price; 5 Closing Price; 7 High Price; 8 Low Price; 66 Trade Volume; 101 Previous Closing Price; 200 Opening Auction; 201 Intraday Auction; 202 Circuit Breaker Auction; 203 Closing Auction
25018	>Pad3	U	3	53	Fixed String	not used

Implied Message Constants

These constant values are to be considered as part of the above message, although they are not transmitted.

Tag	Field Name	Field Value	Length	Data Type	Description
35	MsgType	U23	3	Fixed String	U23 = Market Data Instrument
28842	MarketDataType	10	1	unsigned int	10 = Single Instrument Snapshot See also Eurex EOBI Schema (XSD) file.
22	SecurityIDSource	M	1	Fixed String	M = Marketplace Marketplace assigned identifier.

9 Appendix

9.1 Product Scope

Eurex EOBI interface is designed for selected Eurex benchmark Futures products and all IPS products. The daily public reference data provided by Eurex RDI and Eurex RDF contains the future benchmarks products and all IPS products which are enabled for the Eurex EOBI.

9.2 Multicast addresses for Eurex EOBI

The reference data information provided by Eurex RDI contains the multicast channel information, i.e., IP multicast addresses, for selected Eurex benchmark Futures products. The table below lists the foreseen the multicast IP address range.

Environment	Service A – Multicast Addresses	Service B – Multicast Addresses
Production	224.0.114.32 224.0.114.63	224.0.114.64 224.0.114.95
Simulation	224.0.114.96 224.0.114.111	224.0.114.112 224.0.114.127

Table 7 – Range for Multicast addresses

9.3 Synthetic prices

The order books for Futures products and IPS products, can be enabled for synthetic matching. Any incoming order can match synthetically against the order books enabled for synthetic matching. Information about whether a Futures product or IPS product is enabled (*ImpliedMarketIndicator*) for synthetic matching, is published by public reference data in both in the `Instrument Snapshot` message and in the `Complex Instrument Update` message. Eurex Functional Reference documentation describes the synthetic matching for futures spread and inter product spread instruments that are enabled for synthetic matching in detail. Furthermore in this document, the terms such as Match Path, Synthetic Book Path, Synthetic Pricing and available Quantity refering to Synthetic Book Path, are explained thoroughly. Additionally, the general rules for the calculation of a synthetic price are explained in-depth. For more details please see paragraph about Synthetic Matching in Continuous Trading.

9.4 Connecting Eurex EOBI and Eurex ETI data

Eurex EOBI and Eurex ETI provide information to synchronize private responses and public market data.

Order transactions

The Order time priority and matching engine in timestamp information is provided by both interfaces, i.e., *ExecID* field from Eurex ETI in Order Status and Execution Reports and *TrdRegTSTimePriority* field from Eurex EOBI in incremental and snapshot messages along with the *securityID* information as shown in the table below*:*

Field Description	Public Market Data via Eurex EOBI	Private Market Data via Eurex ETI
Security Identifier	*securityID*	*securityID*
Priority Timestamp of an Order	*TrdRegTSTimePriority*	*TrdRegTSTimePriority / ExecID for Standard orders*
		ExecID for Lean orders
Matching Engine In Timestamp	*TrdRegTSTimeIn*	*TrdRegTSTimeIn*
Match Step Identifier	*TrdMatchID*	*FillMatchID QuoteEventMatchID TrdMatchID*

Table 8 – Provided private and public data via interfaces

An order that is modified will lose its time priority, i.e., it will get a new priority time stamp, if its price or its quantity or order type is changed. So,
For Quotes the *QuoteResponseID*, which is the priority time stamp of the quote side, is delivered in `MassQuoteResponse` message by the Eurex ETI. For Quote activations the *MassActionReportID* in `QuoteActivationResponse` provides the transaction timestamp. Please note that, if a mass quote is modified, then the old time priority time stamp is provided by the Eurex ETI. Logically, if the time priority of an existing order doesn't change, then no order time priority information, *ExecID*, is provided by the Eurex ETI.

Order executions

When an order executes against the order book at multiple price levels, this is reflected by a matching event with multiple match steps. Each match step includes the trades at one price level and is represented by an unique *TrdMatchID (880)* and published in the public market data.
The field *TrdMatchID (880)* is a unique id on product level for each business day. A synthetic match can result in more than one trade volume record with the same *TrdMatchID (880).* Every match step occurring in the exchange has an identifier in Eurex ETI that is provided in the field *FillMatchID (28708)*in the `Execution Report (8)`, *QuoteEventMatchID (8714)* in the `Quote Execution Report (U8)` and *TrdMatchID (880)* in the `Trade Capture Report (AE)`. The match time of all involved orders is reported in the `Execution Summary`message by using the field *ExecID (17)*. These identifier allows participants to link trade capture reports and the corresponding execution report of the Eurex ETI with the market data incremental feed of the Eurex EOBI.
The aggressor timestamp, whenever set, is always identical to the matching engine in timestamp.

9.5 Multicast addresses

The reference information provided by Eurex RDI contains the respective multicast channel information (i.e., multicast addresses and port numbers) for all available products.

Reference data snapshot channels

Environment	Service A	Service B
Production	224.0.50.0:59098	224.0.50.128:59098
Simulation	224.0.50.96:59598	224.0.50.224:59598

Table 9 – Multicast address and ports for reference data snapshot feeds

Reference data incremental channels

Environment	Service A	Service B
Production	224.0.50.1:59098	224.0.50.129:59098
Simulation	224.0.50.97:59598	224.0.50.225:59598

Table 10 – Multicast address and ports for reference data incremental feeds

Please note that the reference data is provided in file form as compressed Reference Data Files **(RDF)** in FIXML-layout, updated approximately every 5 minutes via the Common Report Engine(CRE).

9.6 Reference data for Eurex EOBI

The reference data information such as order book type, multicast addresses and port numbers of corresponding products etc., which is needed to receive public market data via Eurex EOBI, is available via the existing Eurex RDI and/or the Eurex RDF.

The `Product Snapshot` message will contain the following information for the products configured for Eurex EOBI:

- Book Type, *MDBookType*(*1021*), field will carry the valid value (*Order Depth* = *3*),
- Feed Type, *MDFeedType*(*1022*), field will carry the valid values (*HI* = high bandwidth incrementals) for Eurex EOBI incremental messages and (*HS* = high bandwidth snapshots) for Eurex EOBIsnapshot messages in combination with the multicast addresses and port numbers,
- IP Multicast address, *MDPrimaryFeedLineID*(*28590*), field will carry the IP Multicast address of primary Eurex EOBI feed along with the primary port number, *MDPrimaryFeedLineSubID*(28591).
- IP Multicast address, *MDSecondaryFeedLineID*(*28590*), field will carry the IP Multicast address of secondary Eurex EOBI feed along with the secondary port number (*MDSecondaryFeedLineSubID*(*28591*)).

The same information is also available via Eurex Reference Data Files (Eurex RDF).

10 Change log

The updated Eurex EOBI interface version, Release 3.0 (as part of Eurex Exchange's T7 Release 3.0).

Chapter	Description
All	Top Of Book message is added to provide instrument's BBO information beyond post trading phases. Implied message constants made available along with the EOBI message layouts. Description of Heartbeat message is improved. Complex Instrument message layout changes: *SecurityDesc* is removed and *ProductComplex* is added.
4.1, 5, 4.13, 7.3, 7.4, 8.2	Description of the sending order sequence in zig-zag fashion is improved by an example. Transaction timestamp is added in the Auction messages. *TradeCondition* is added in the Execution Summary message. Desciption of Recovery handling is improved. Snapshot Order message is not sent during Pre-Trading phases. IP Multicast address range table is added. Description of implied message constants have been improved in the message layouts. Description of Order Modify is improved.
7.3	The previous fields in Order Modify message are mandatory.
4.11, 7.3	FIX Tag ID of *TransactTime* is changed from 273 to 60. Description of Partial Order Execution message is improved.
4, 4.8, 8.2	Closing-Auction-Only Orders aren't visible. Description of Order Mass Delete message is improved. It is sent explicitly, if the potential auction price is not determined during auction call phase. A new sub-chapter Synthetic prices with a reference to Functional Reference document is added. Description of Partial Order Execution message is improved.
4, 4.4, 4.5, 5, 6.2, 6.3, 7.1, 7.3	*CompletionIndicator* set to Complete(=1) on the last datagram of every Product Snapshot cycle on Eurex EOBI snapshot feed. Description of Order Mass Delete message is improved. It is sent explicitly, if the auction bbo or auction clearing price is not to be determined during auction call phase. Added new message description and layout for OrderModifySamePrio. Added *TransactTime* to OrderDelete, OrderMassDelete and OrderModifySamePrio messages. Updated packet header *TemplateID* to 13002.
3.2, 4.6, 7.4	The MTU size is set to 1372 bytes. Added field *RestingHiddenQty* to Execution Summary message.
4.2, 4.5	The field *TrdRegTSTimeIn* is not mandatory in OrderAdd and OrderDelete messages.
4.11, 8.1	Corrected trade volume is also reported together with the Trade Reversal message. Removed list of products enabled for EOBI interface.
1, 4, 4.9, 4.11, 9.1, 9.4	Added construct for IPS.
2.3, 4.6, 5, 7.5, 8.3	Changed *www.fixprotocol.org* to *www.fixtradingcommunity.org*. Described specific scenario with respect to matches resulting from spread uncrossing. Added new section *Timestamps*. Added concrete definition for PriceType data type. Field descriptions for *Price* and *LastPx* in message layouts for Full/Partial excecution.
4,6 8.4	Added new field *RequestTime* to *Execution Summary* message.

Chapter	Description
2, 4.4	Updated the Introduction and Order Modification section for the case wherein change in priority timestamp might occur due to a change in non-visible field.
4.11, 8.4	The new trade volume will not be reported with *TradeReversal* message. Updated message layout for *TradeReversal* message.
7.4	Removed redundant section 7.4 *Message Overview*. Section 6 *Availability of Enhanced Order Book Service* provides requisite details.
4.13	Added paragraph Detecting duplicates and gaps by means of the packet header
	Either fixed or removed a few broken references
8.1, 8.4	Updated packet header *TemplateID* to 13004 Added field *RestingCxlQty* to *Execution Summary* message